Segment Routing

Setting the scene...

Dhruv Dhody
(dhruv.ietf@gmail.com)

Why Segment Routing (SR)?

- Ability for a node to specify a path
  - So that apart from the usual shortest-path, another path can be taken up!
  - Basically Source Routing!
    - In contrast to the usual hop-by-hop destination-based routing!
    - *But what happened to source routing in the past?*
      - *Significant security concerns for IP in the open Internet and thus was deprecated!*
- No state in the network
  - Everything needed is in the packet header *(no signaling protocols)*
  - Highly Scalable
- Simplify Network Operations
  - Agile and flexible Traffic Engineering
  - Better resource utilization
  - Network Operations and Management

What is Segment Routing (SR)?

- A modern variant of source-routing!
- No state on the transit routers, instead it is part of the packet header, injected at the ingress.
- Some concepts
  - SR Domain: the set of nodes participating in the SR-based routing model
  - SR Path: an ordered list of **segments** that connects an SR ingress node to an SR egress node
  - SR Segment: an instruction a node executes on the incoming packet (e.g., forward packet through a specific interface, or forward via shortest path to the destination).
  - SID: a segment identifier
  - Two dataplanes are supported : SR-MPLS & SRv6

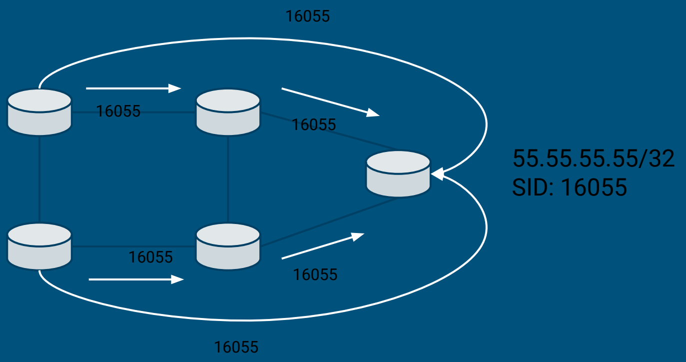
SR Forwarding & Control Planes

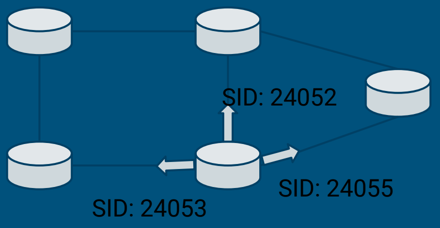
- SR Forwarding Plane
  - SR-MPLS
    - an ordered list of segments is represented as a stack of MPLS labels
    - No change in MPLS header and dataplane
    - Segment represented as MPLS label and SR Path as label stack
  - SRv6
    - an ordered list of segments is encoded in a routing extension header (SRH)
    - More later on…

- SR Control Plane
  - The “brains” of the whole operation!
  - IGP / BGP-LS
    - Distribute the SR information such as SIDs
  - PCEP
    - When a centralized Path Computation Element (PCE) is used to compute SR Paths
  - SR Path/Policy
    - a framework that enables the instantiation of an ordered list of segments on a node for the steering of traffic for a specific purpose
    - Distributed via BGP/PCEP/YANG

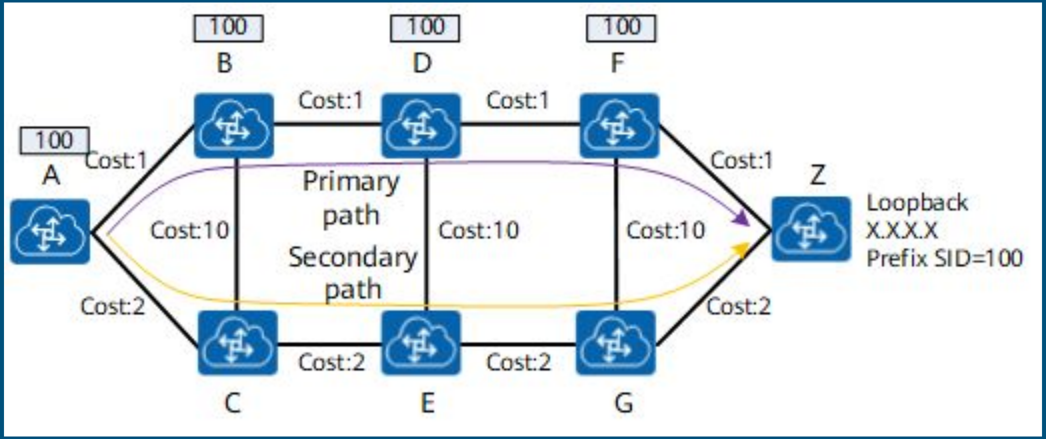
Segments / SIDs

- Prefix/Node SID - Globally unique segment identifier represent ECMP-aware shortest route to the prefix/node.
- Adjacency SID - Local segment identifier for an adjacency
- Others
  - Anycast SID
  - Binding SID
  - Egress Peer Engineering (EPE) Segments

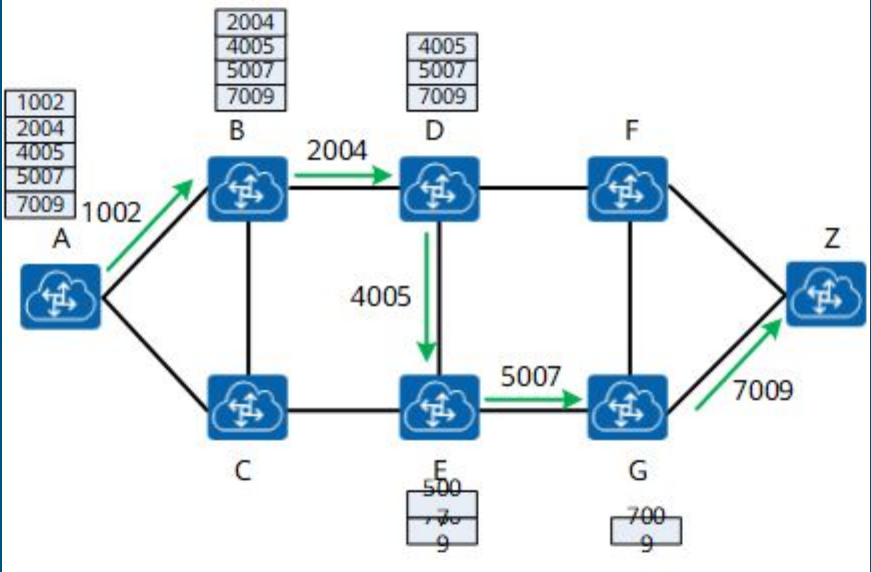




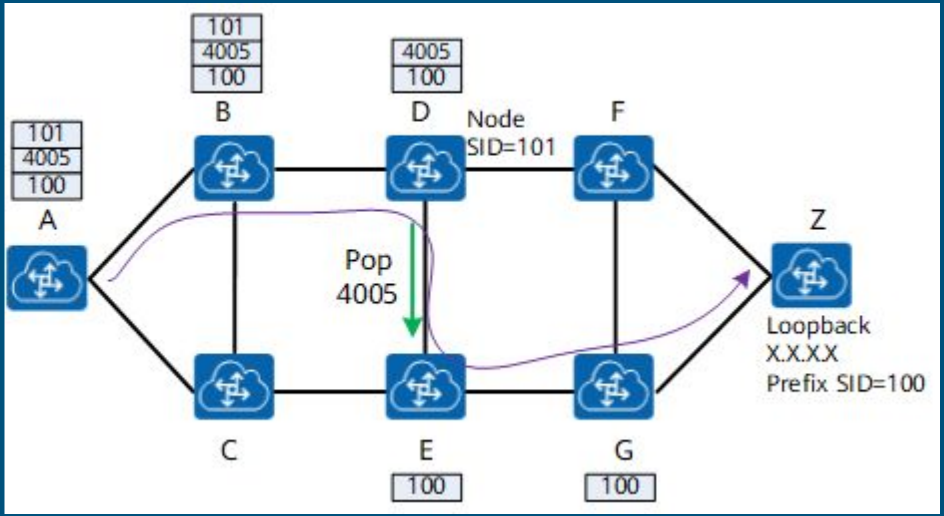
Prefix SID Example



Adjacency SID Example



Mixed Example



Our focus is on the SR on IPv6 (SRv6) dataplane today!

Introduction to SRv6

Darren Dukes ddukes@cisco.com

INTC/IIESoc Webinar – March 9, 2023

SRv6 is IETF Proposed Standard

Architecture

- Segment Routing Architecture RFC 8402
- Source Packet Routing in Networking (SPRING) Problem Statement and Requirements RFC 7855
- IPv6 Segment Routing Header (SRH) RFC 8754
- Segment Routing over IPv6 (SRv6) Network Programming RFC 8986

Use-cases

- SRv6 based overlay services RFC9259
- IGP Flexible Algorithm RFC9350
- Topology Independent Fast Reroute using Segment Routing (WG Draft)

Protocol Extensions

ISIS

- IS-IS Extensions for SRv6 RFC 9352

OSPF

- OSPFv3 Extensions for SRv6 (WG Draft)

BGP

- BGP Link State Extensions for SRv6 (WG Draft)

PCEP

- PCEP Extensions for SRv6 (WG Draft)

OAM

- OAM In SRv6 Networks 9259

Performance Measurement

- A Two-Way Active Measurement Protocol (TWAMP) RFC 5357
- Simple Two-Way Active Measurement Protocol RFC 8762
- Enhanced Performance Measurement Using Simple TWAMP in Segment Routing Networks (Draft)
- Performance Measurement Using Simple TWAMP (STAMP) for Segment Routing Networks (WG Draft)
- Simple TWAMP (STAMP) Extensions for Segment Routing Networks (WG Draft)
- Simple Two-Way Direct Loss Measurement Procedure (Draft)

Up-to-date list on https://segment-routing.net/ietf

SRv6
Network Programming
Introduction

Segment Routing

- Source Routing
  - the topological and service (NFV) path is encoded in packet header
- Scalability
  - the network fabric does not hold any per-flow state for TE or NFV
- Simplicity
  - automation: TILFA sub-50msec FRR
  - protocol elimination: LDP, RSVP-TE, VxLAN, NSH, GTP, ...
- End-to-End
  - DC, Metro, WAN

Two dataplane instantiations

Segment Routing

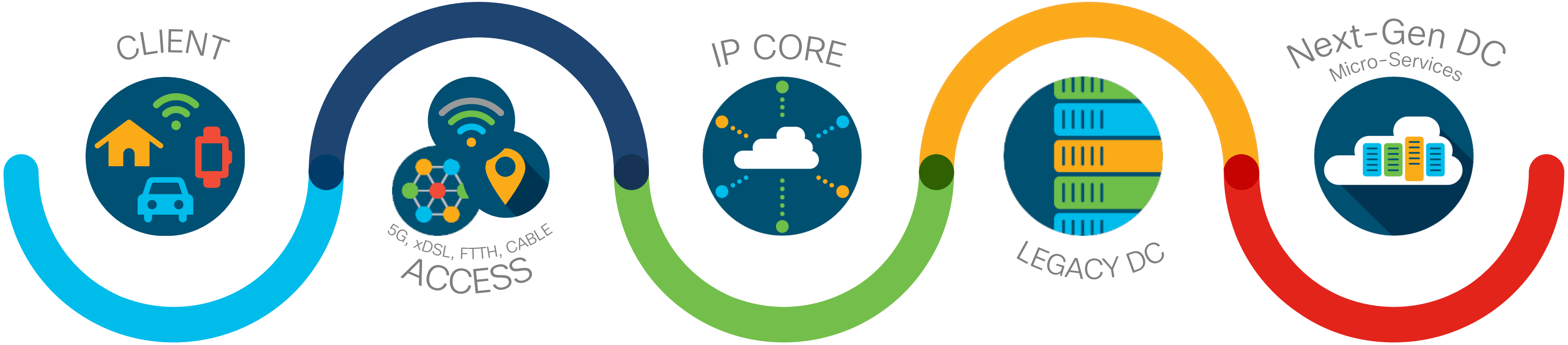
MPLS

- leverage the mature MPLS HW with only SW upgrade
- 1 segment = 1 label
- a segment list = a label stack

IPv6

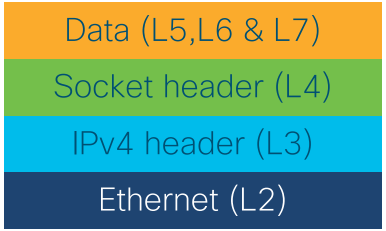
- leverages RFC8200 provision for source routing extension header
- 1 segment = 1 address
- a segment list = an address list in the SRH (RFC8754)

IPv6 provides reachability



IPv4 limitations & work-arounds

- × Limited address space
- × No engineered Load Balancing
- × No VPN
- × No Traffic Engineering
- × No Service Chaining



IPv4 limitations & work-arounds

- × Limited address space → NAT
- × No engineered Load Balancing → MPLS Entropy Label, VxLAN UDP
- × No VPN → MPLS VPN's, VxLAN
- × No Traffic Engineering → RSVP-TE, SR-TE MPLS
- × No Service Chaining → NSH

IPv4 limitations & work-arounds

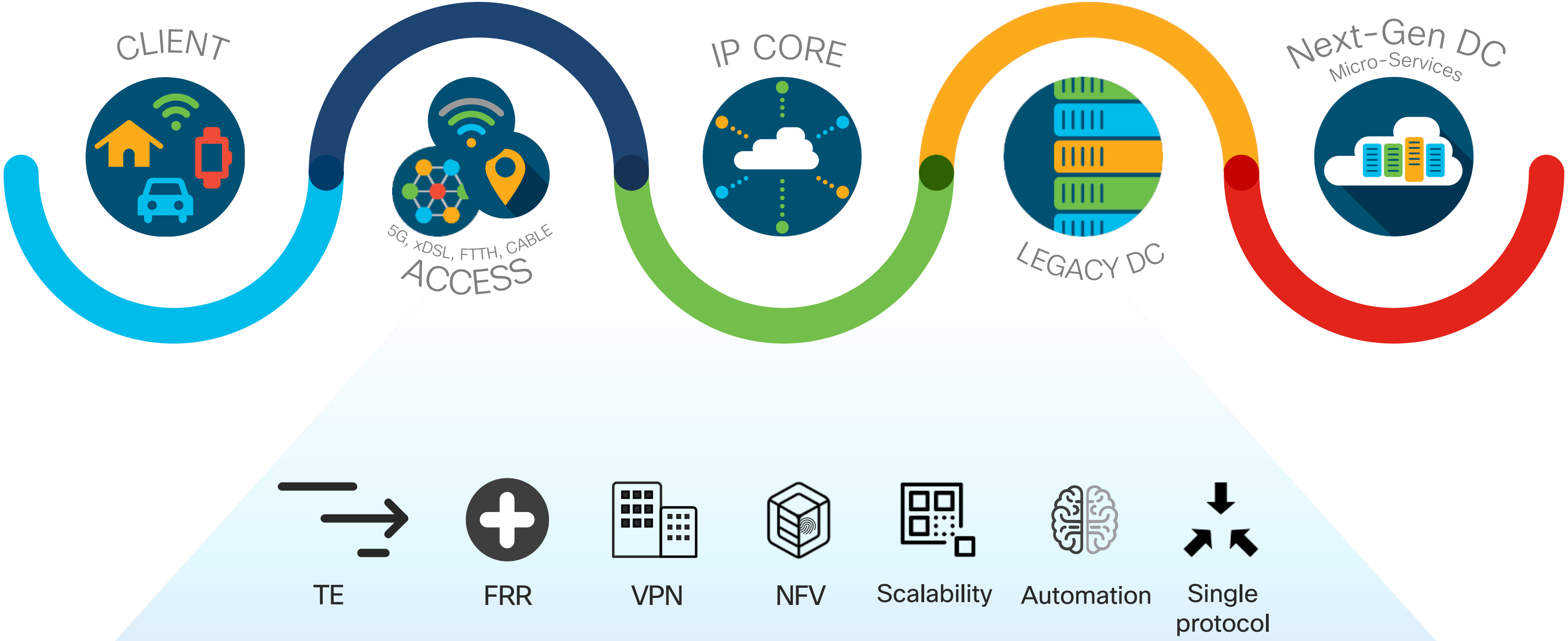
- × Limited address space → NAT
- × No engineered Load Balancing → MPLS Entropy Label, VxLAN UDP
- × No VPN → MPLS VPN's, VxLAN
- × No Traffic Engineering → RSVP-TE, SR-TE MPLS
- × No Service Chaining → NSH

Data (L5,L6 & L7)
Socket header (L4)
IPv4 header (L3)
Ethernet (L2)

Simplicity (back to the OSI model)

Data (L5,L6 & L7)
Socket header (L4)
IPv4 header (L3)
NSH
VxLAN (+ IP and UDP)
MPLS
Ethernet (L2)

SRv6 unleashes IPv6 potential



SR for anything:
Network as a Computer

Network instruction

- 128-bit SRv6 SID
  - Locator: routed to the node performing the function
  - Function: any possible function
    either local to NPU or app in VM/Container
  - Flexible bit-length selection

Network instruction

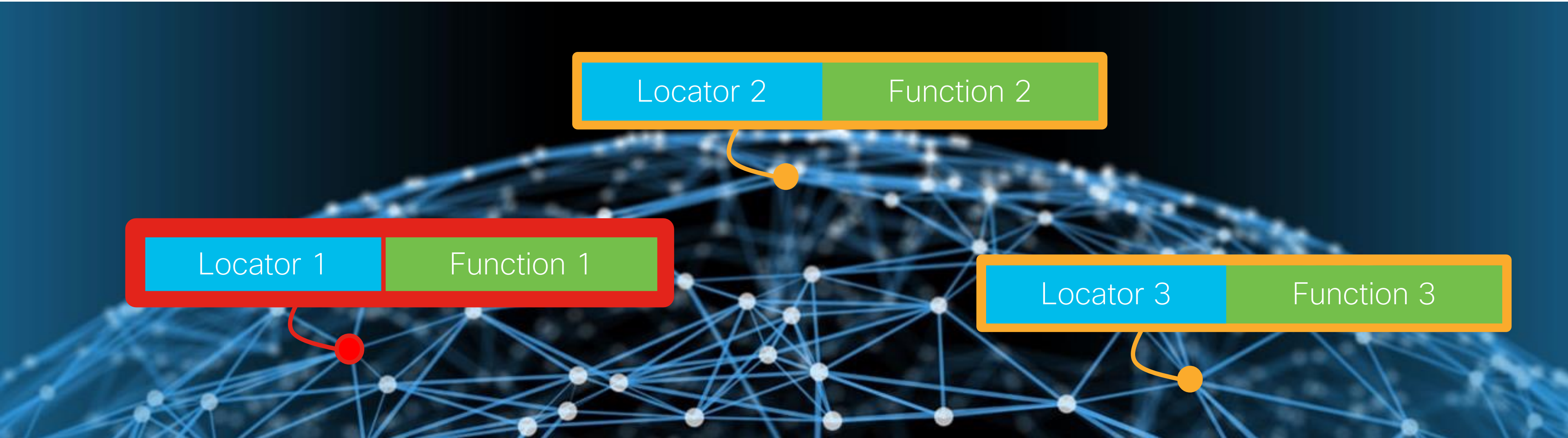
- 128-bit SRv6 SID
  - Locator: routed to the node performing the function
  - Function: any possible function

    either local to NPU or app in VM/Container
  - Arguments: optional argument bits to be used only by that SID
  - Flexible bit-length selection

Network Program

Active Segment →

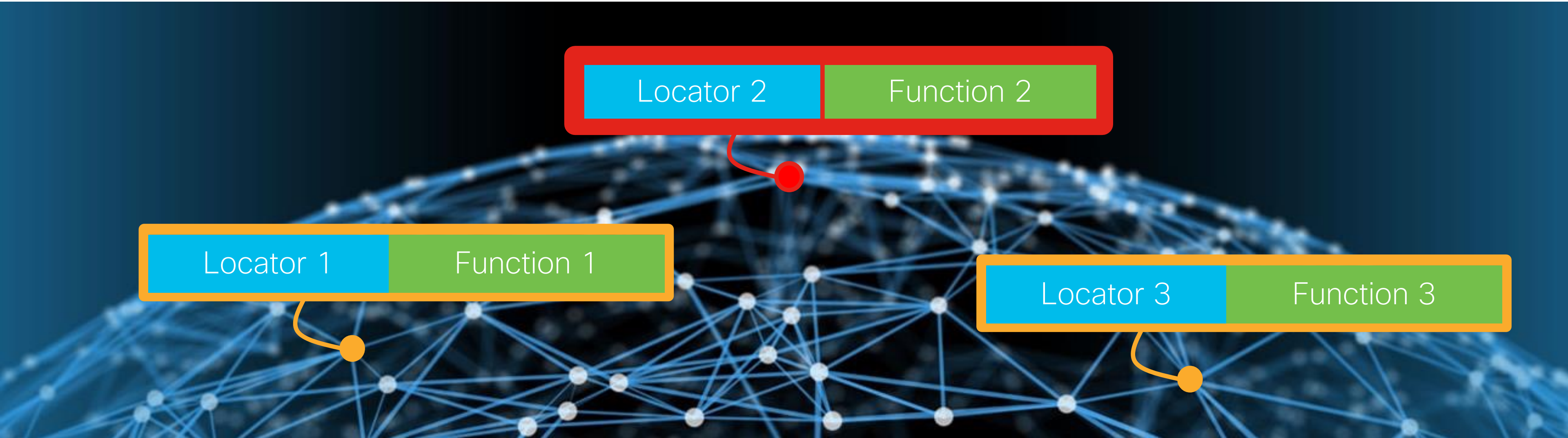
Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3



Network Program

Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3

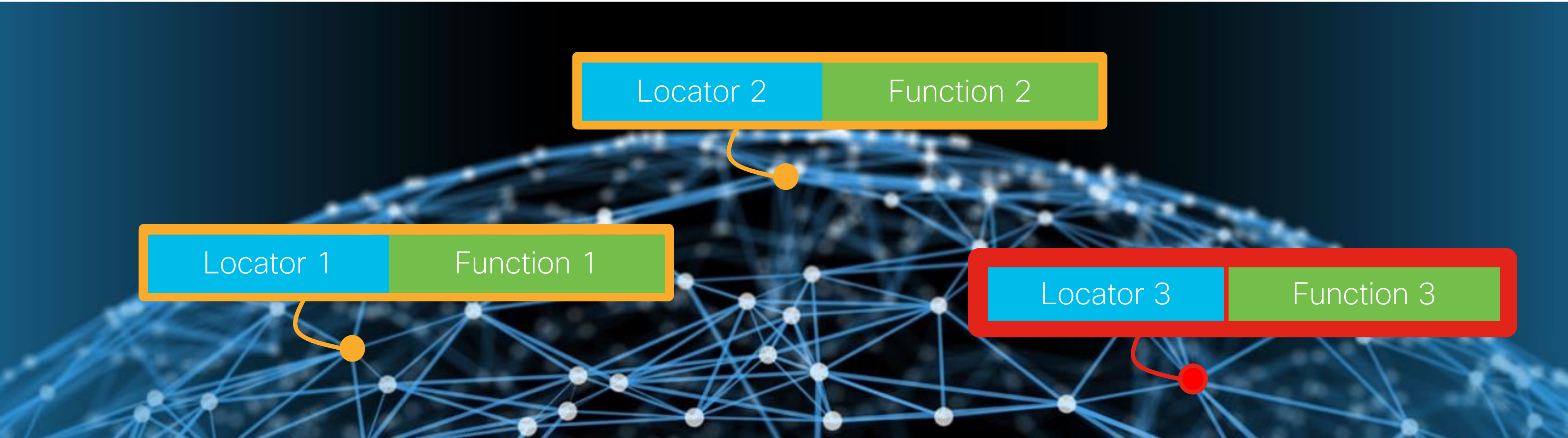
Active Segment → Locator 2 | Function 2



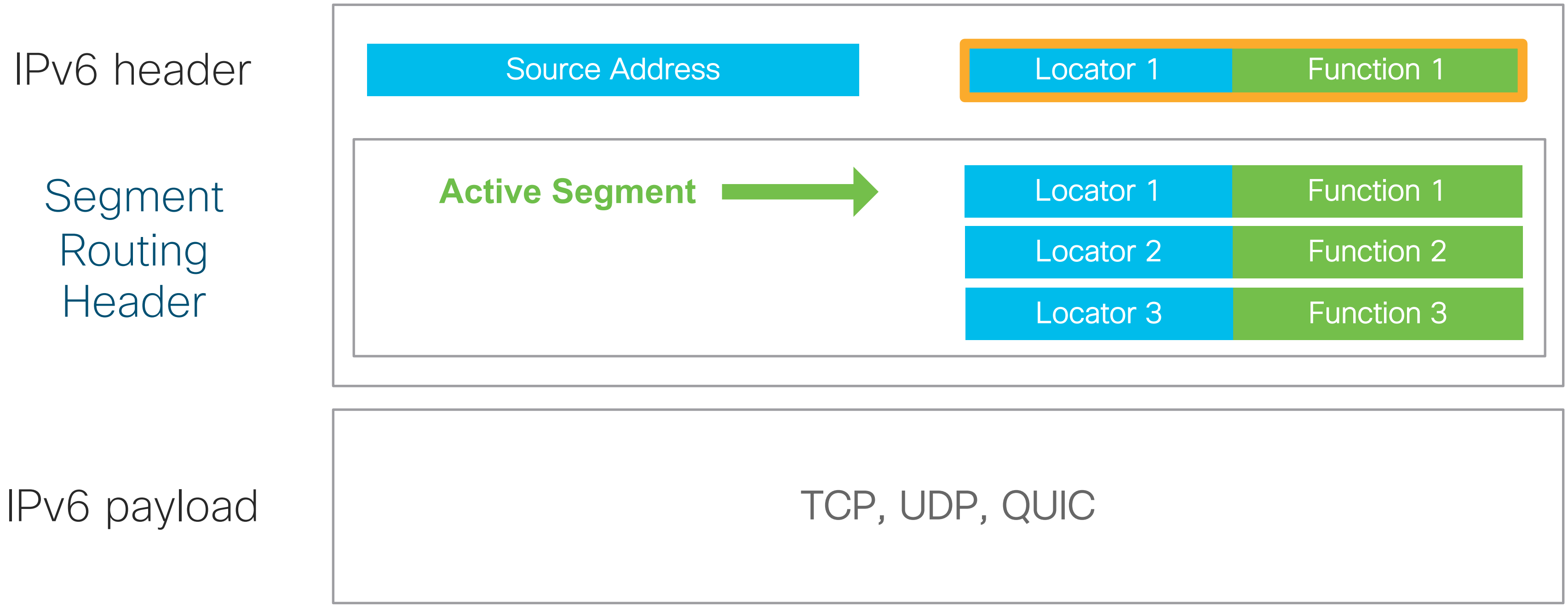
Network Program

Locator 1	Function 1
Locator 2	Function 2
Locator 3	Function 3

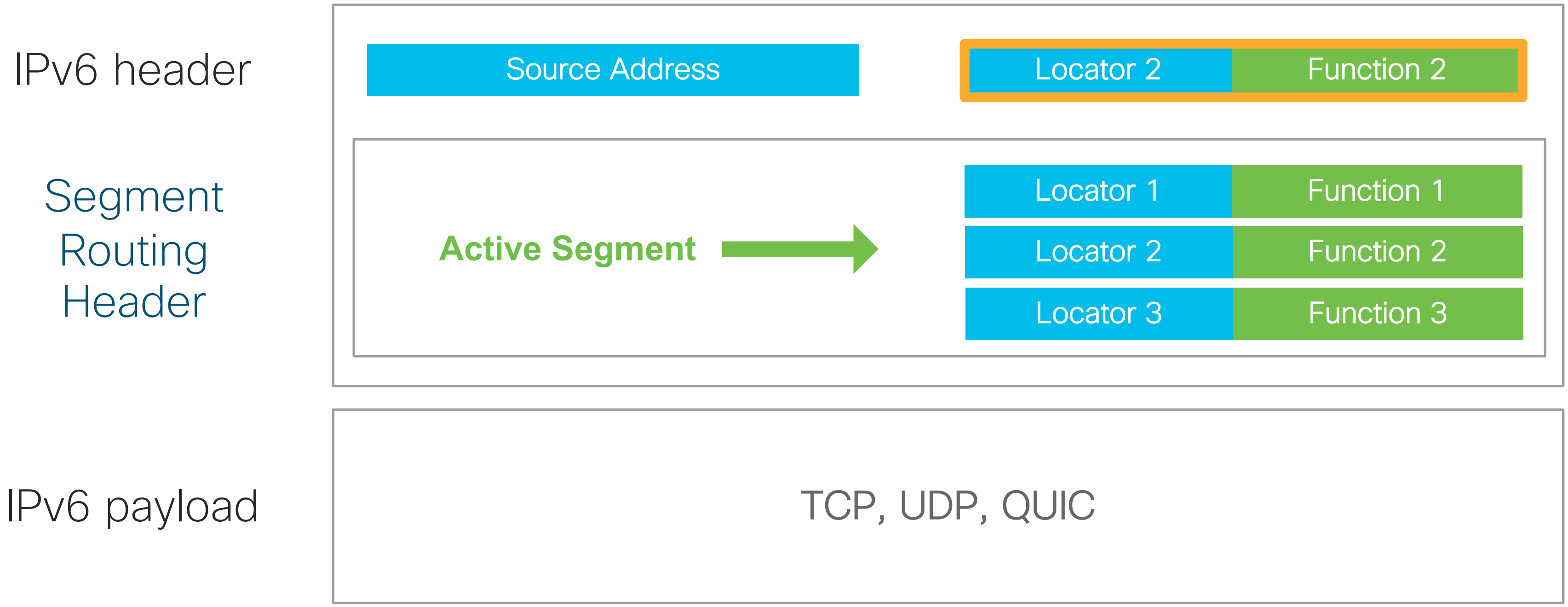
Active Segment → Locator 3 | Function 3



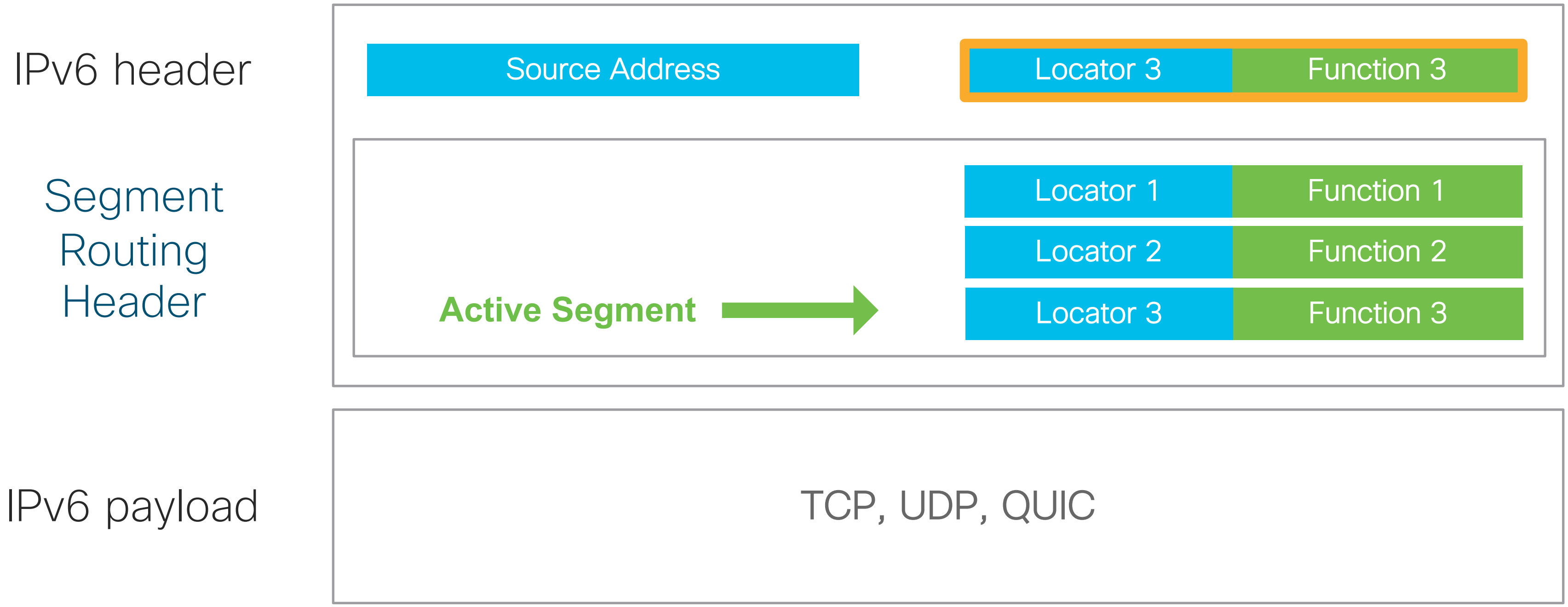
Network Program in the Packet Header



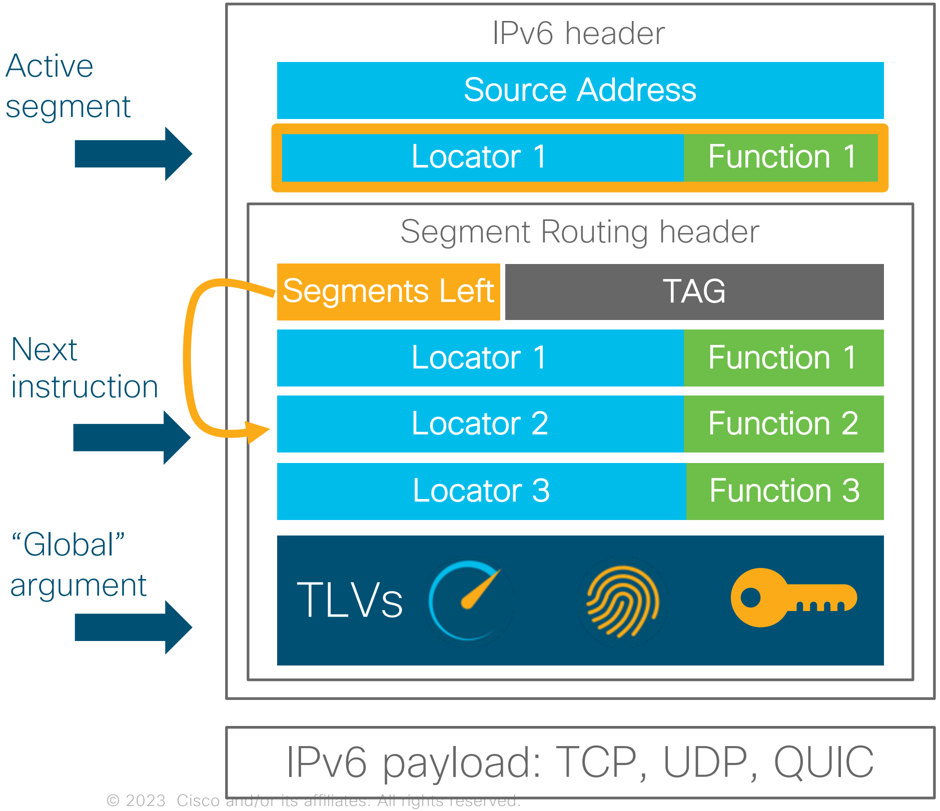
Network Program in the Packet Header



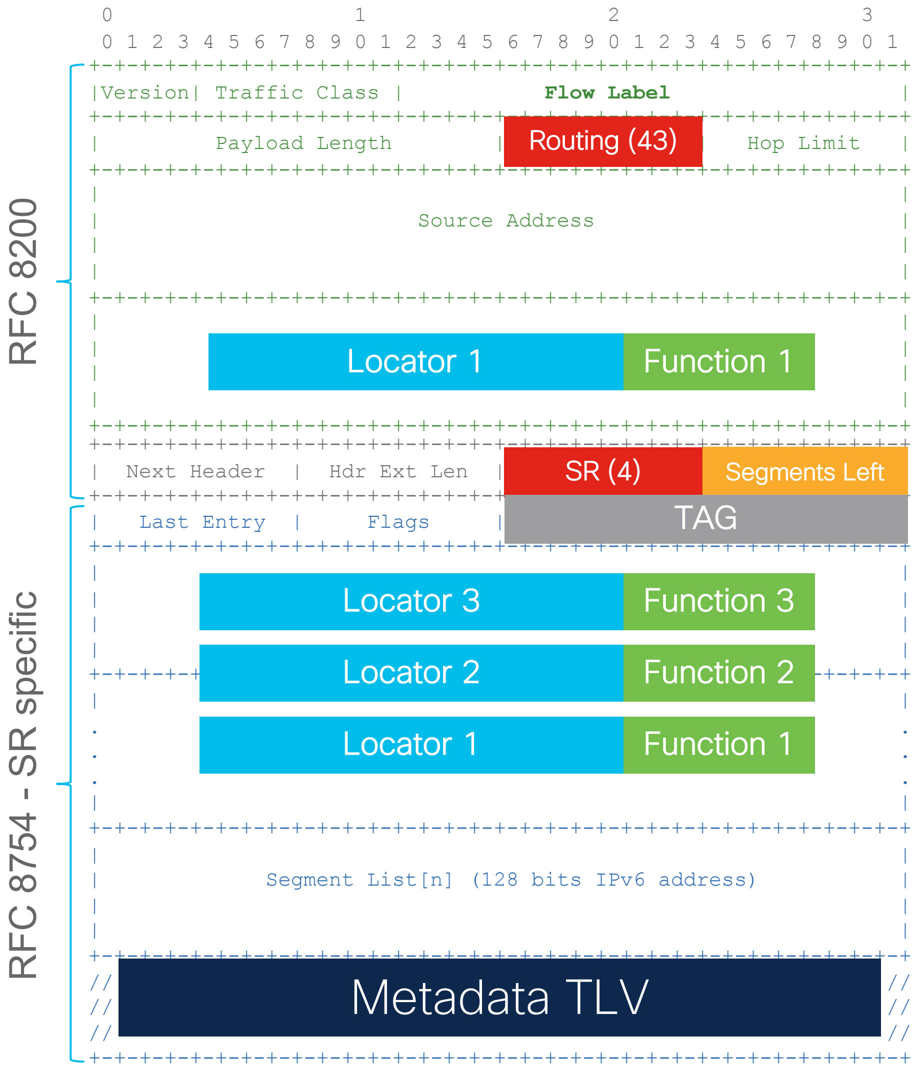
Network Program in the Packet Header



SRv6 Header

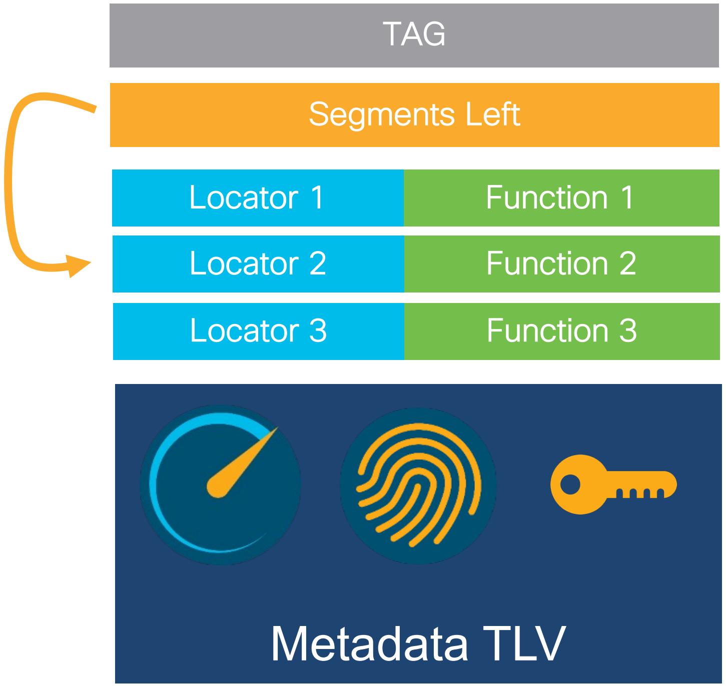


Active segment
Next instruction
"Global" argument
IPv6 header
Source Address
Locator 1
Function 1
Segment Routing header
Segments Left
TAG
Locator 1
Function 1
Locator 2
Function 2
Locator 3
Function 3
TLVs
IPv6 payload: TCP, UDP, QUIC



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
RFC 8200
|Version| Traffic Class |
Flow Label
Payload Length
Routing (43)
Hop Limit
Source Address
Locator 1
Function 1
Next Header
Hdr Ext Len
SR (4)
Segments Left
Last Entry
Flags
TAG
RFC 8754 - SR specific
Locator 3
Function 3
Locator 2
Function 2
Locator 1
Function 1
Segment List[n] (128 bits IPv6 address)
Metadata TLV

SRv6 for anything



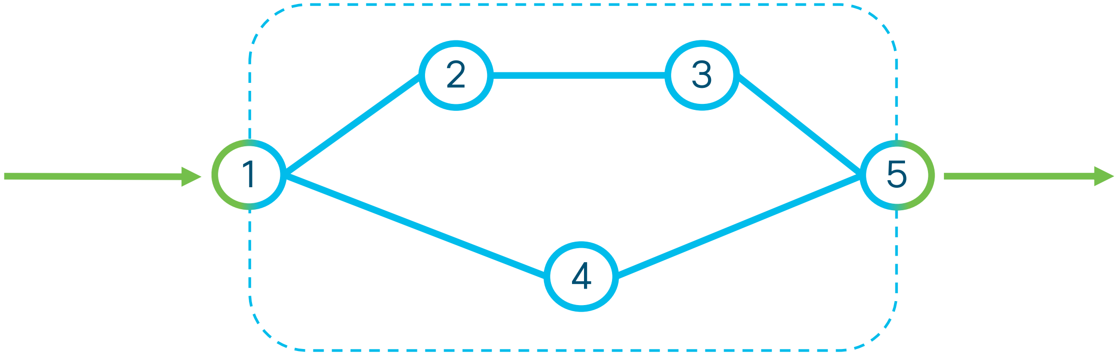
Optimized for HW processing
e.g. Underlay & Tenant use-cases

Optimized for SW processing
e.g. NFV, Container, Micro-Service

SRv6 Domain

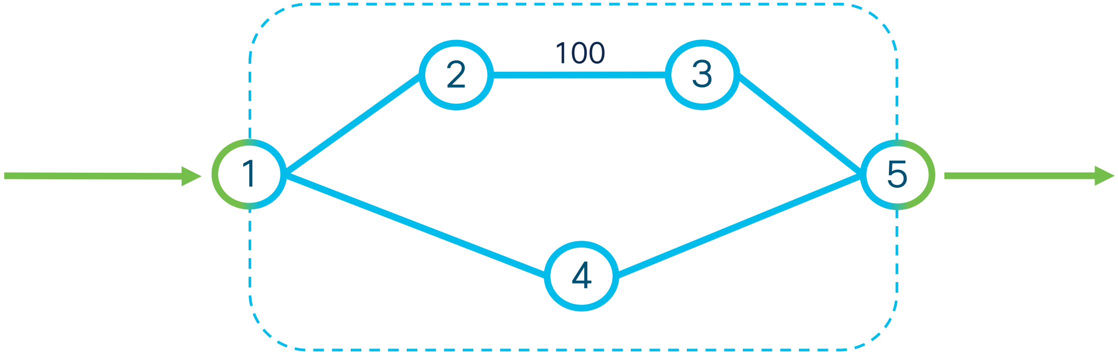
IPv6 enabled provider infrastructure

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

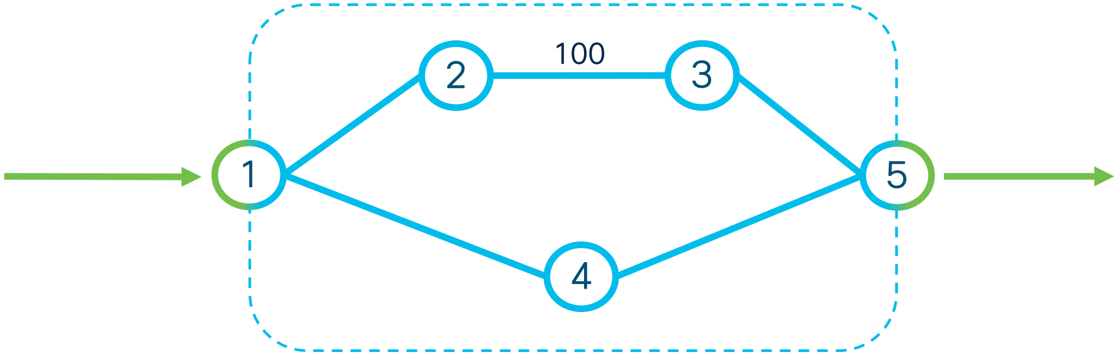


IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:2:C3::
SR	(B:0:0:5: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



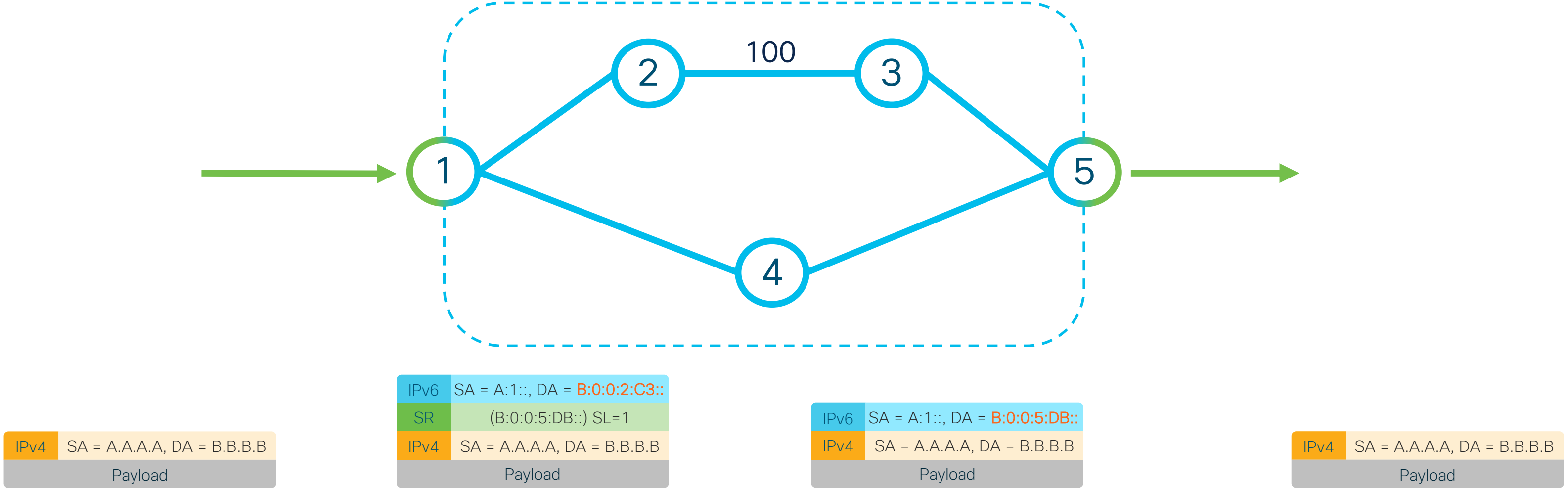
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:2:C3::
SR	(B:0:0:5:DB::) SL=1
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

IPv6	SA = A:1::, DA = B:0:0:5:DB::
IPv4	SA = A.A.A.A, DA = B.B.B.B
Payload	

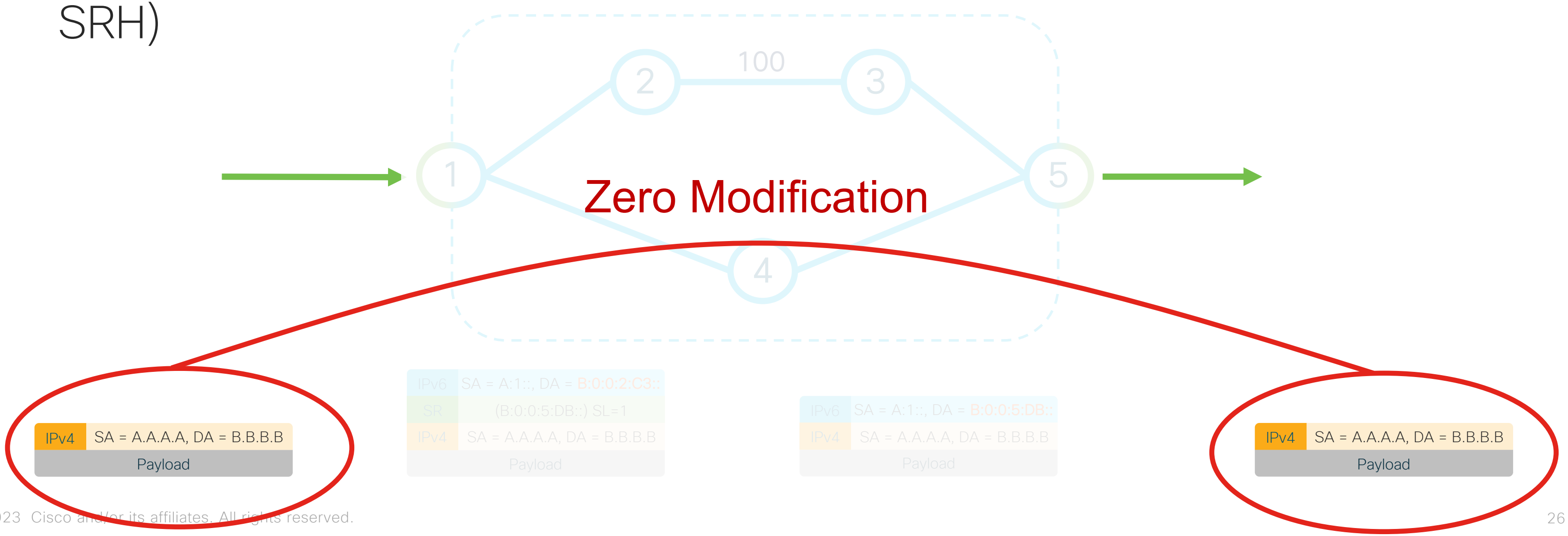
Decapsulation at Domain Egress

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



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)

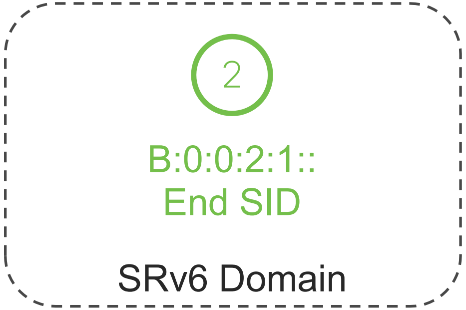


End and End.X SID behaviors

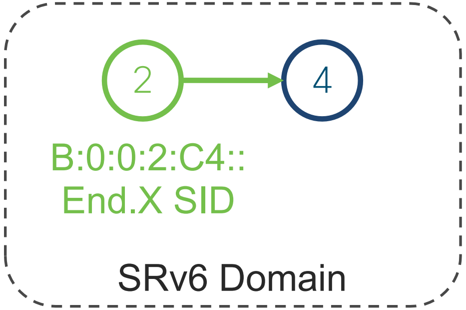
Illustration convention:

- IPv6 address of node k is **A:<k>::**
- SRv6 SID of node k is **B:0:0:<k>:<function>::**

- End – Default endpoint behavior
  - shortest-path to the SID's endpoint
  - endpoint updates DA with next SID
  - endpoint forwards according to updated DA

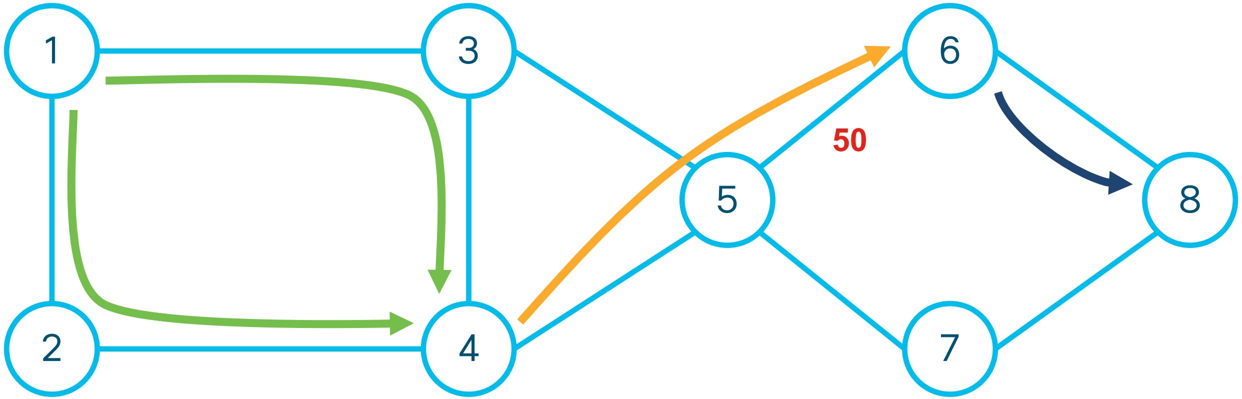


- End.X – Endpoint with cross-connect
  - shortest-path to SID's endpoint
  - endpoint updates DA with next SID
  - endpoint forwards to interface associated with SID



Endpoint behaviors illustration

SR: 〈 B:0:0:4:1::, B:0:0:5:C6::, A:8:: 〉



Default metric 10

- B:0:0:4:1:: shortest path to node 4
- B:0:0:5:C6:: shortest path to node 5, then cross-connect towards 6
- A:8:: regular IPv6 address of node 8

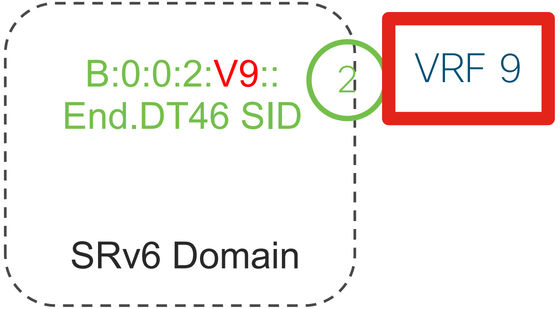
Usecase

End.DT46 Behavior

Illustration convention:

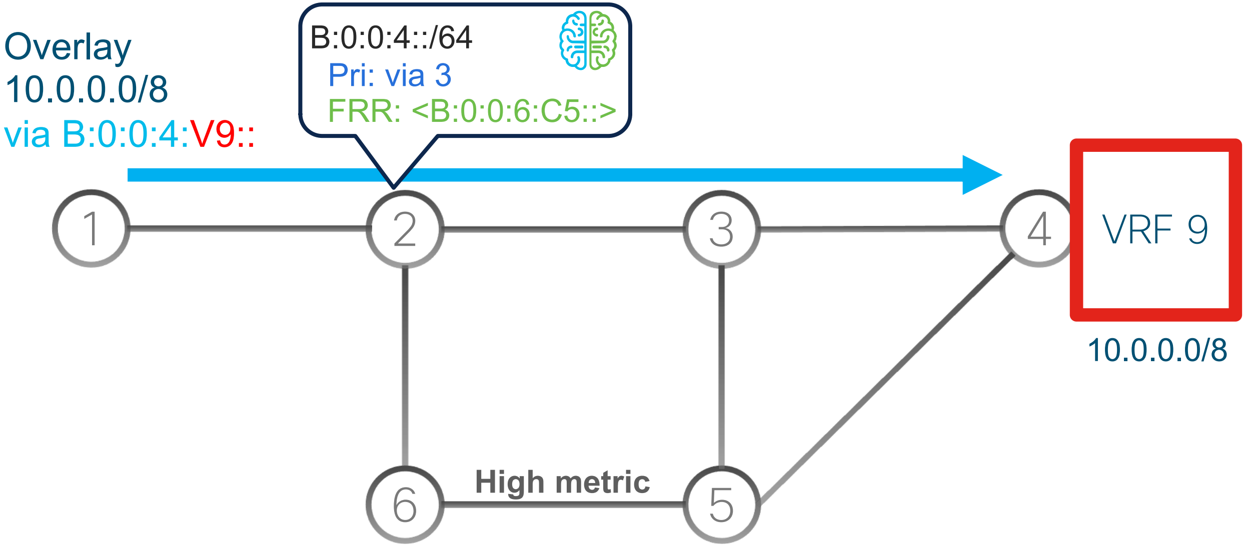
- IPv6 address of node k is **A:<k>::**
- SRv6 SID of node k is **B:0:0:<k>:<function>::**

- End.DT46 – Endpoint with decapsulation to a table
  - shortest-path to SID's endpoint
  - endpoint decapsulates outer IPv6 header
  - forwards to the new destination via a VRF
    - *IPv4 or IPv6 table associated with the VRF*



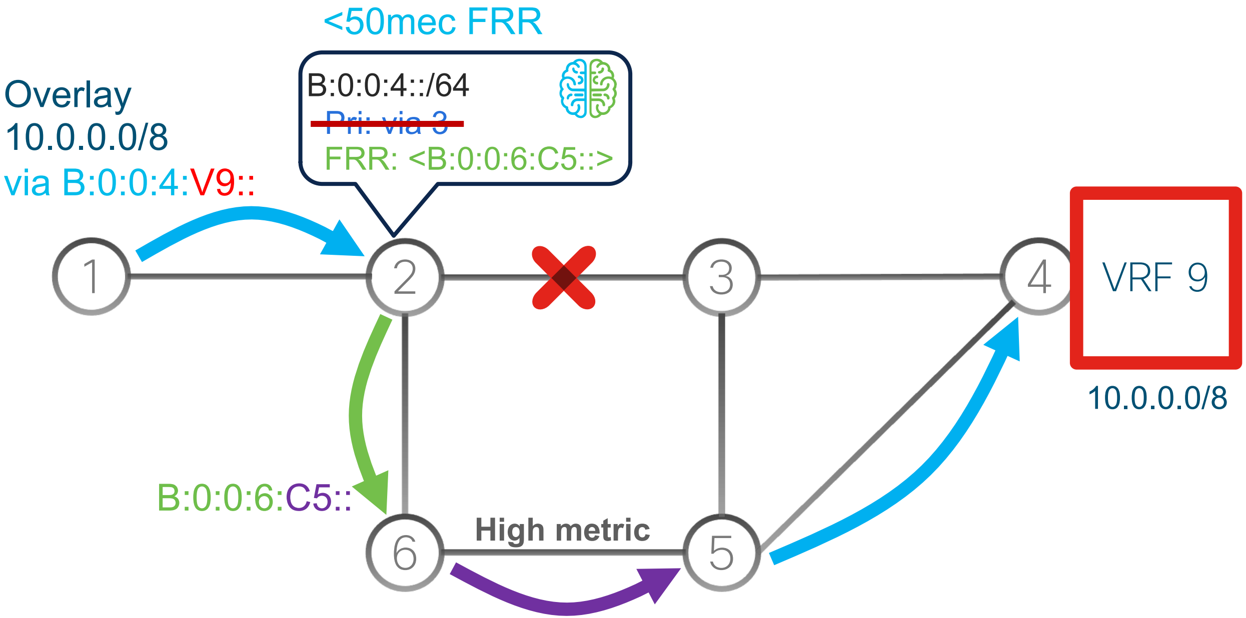
RFC8986 SRv6 SID Behaviors

TI-LFA



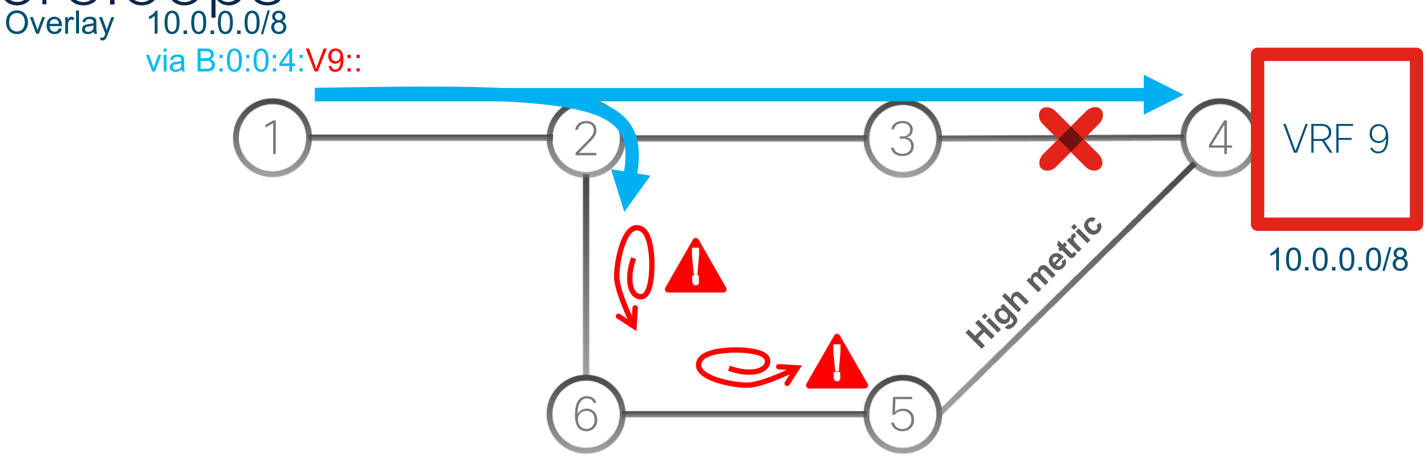
- 50msec Protection upon local link, node or SRLG failure
- Simple to operate and understand
  - automatically computed by the router's IGP process
  - 100% coverage across any topology
  - predictable (backup = post-convergence)
- Optimum backup path
  - leverages the post-convergence path
  - avoid any intermediate flap via alternate path
- Incremental deployment
- Distributed and Automated Intelligence

TI-LFA



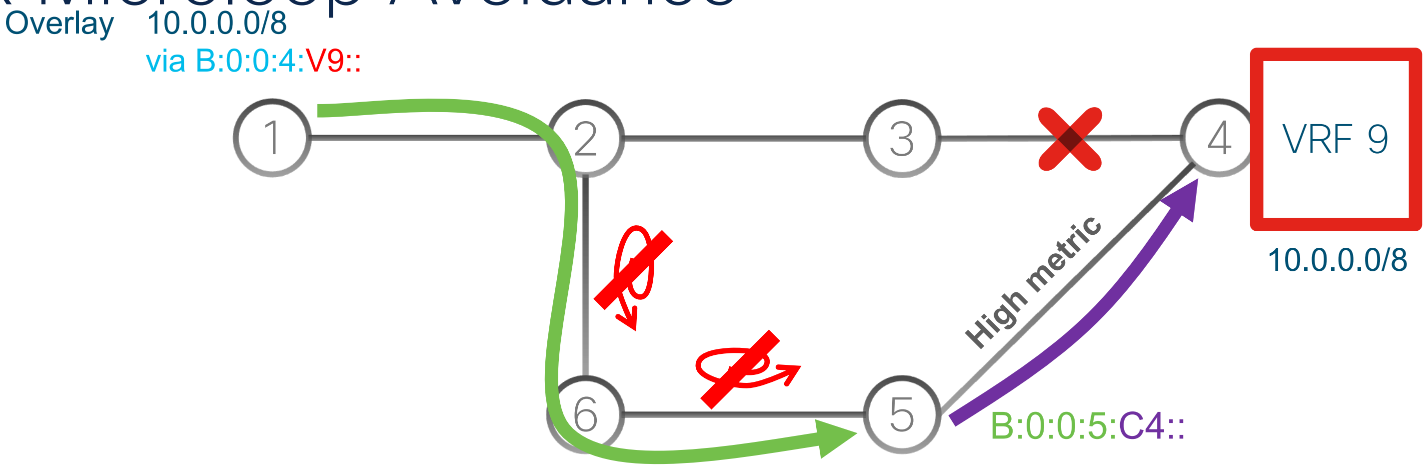
- 50msec Protection upon local link, node or SRLG failure
- Simple to operate and understand
  - automatically computed by the router's IGP process
  - 100% coverage across any topology
  - predictable (backup = post-convergence)
- Optimum backup path
  - leverages the post-convergence path
  - avoid any intermediate flap via alternate path
- Incremental deployment
- Distributed and Automated Intelligence

Microloops



- Microloops are a day-one IP drawback
  - Unsynchronized distributed convergence and IP hop-by-hop routing can cause transient packet loops after a topology change
- Microloops cause packet loss and out-of-order packets

SR Microloop Avoidance

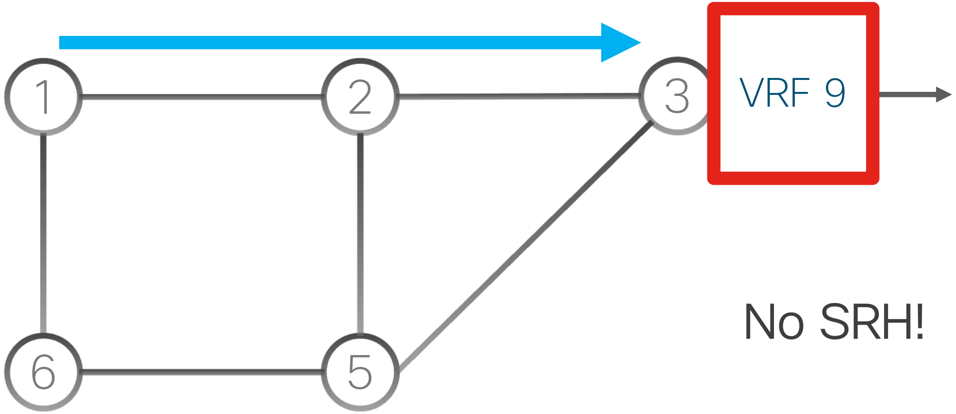


- SR Microloop Avoidance temporarily steers traffic on the loop-free post-convergence paths using SR Policies
- After the network has converged the SR Policies are deactivated

VPN over Best-Effort Slice

Network Program: B:0:0:3:V9::

B::/40 locator block is associated with ISIS base algo (Low Cost, Best Effort)

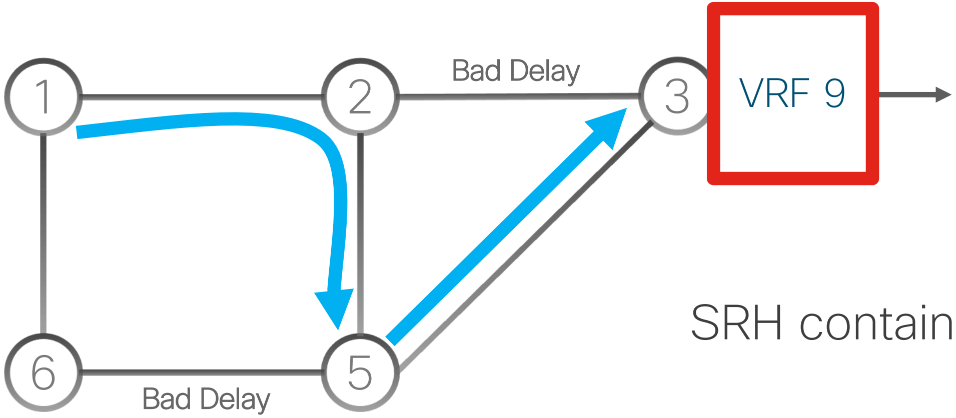


No SRH!

VPN with Low-Delay Slice – SR-TE Option

Network Program: B:0:0:2:C5:: then B:0:0:3:V9::

B::/40 locator block is associated with ISIS base algo (Low Cost)

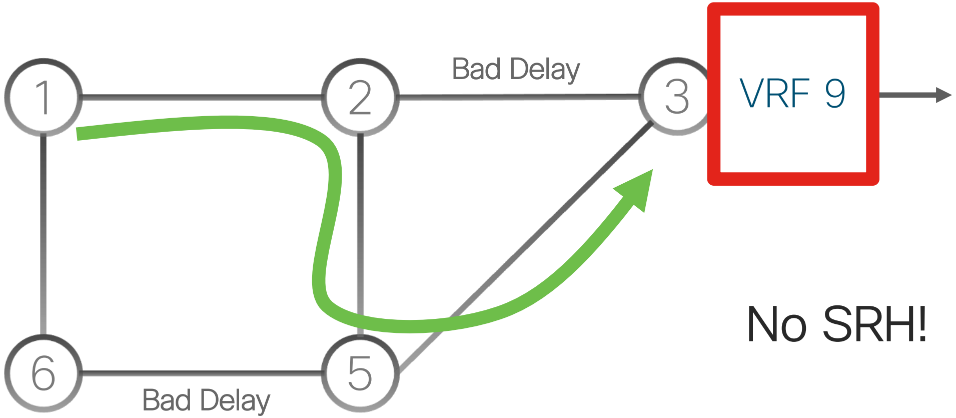


SRH contains 1 single SID

VPN with Low-Delay Slice – Flex-Algo Option

Network Program: D:0:0:3:V9::

D::/40 locator block is associated with Low Delay Flex-Algo

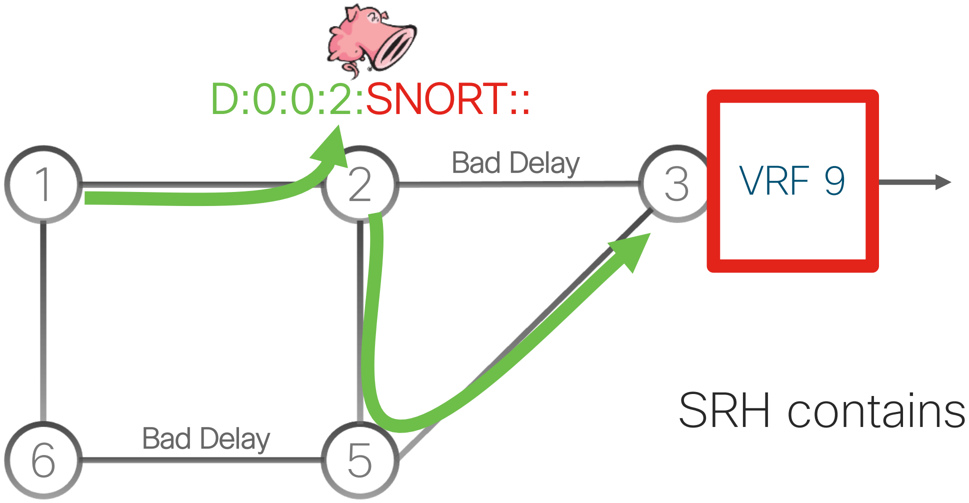


No SRH!

Snort Firewall, VPN & Low-Delay Slice

Network Program: D:0:0:2:SNORT:: then D:0:0:3:V9::

D::/40 locator block is associated with Low Delay Flex-Algo

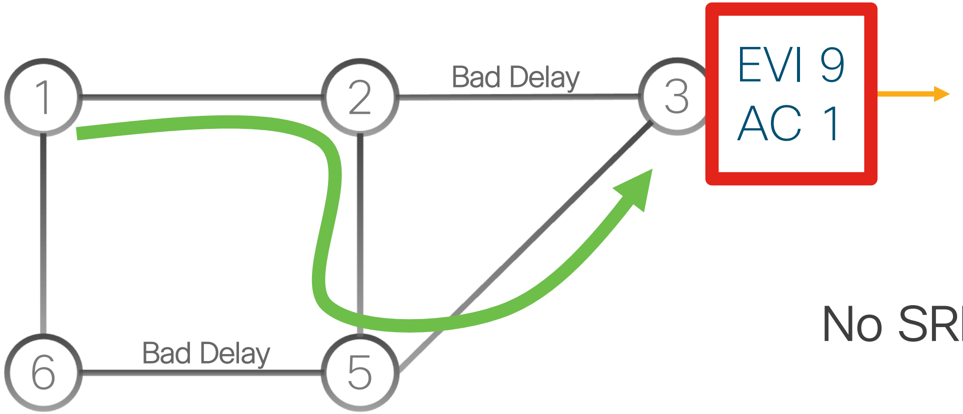


SRH contains 1 Single SID

EVPN VPWS Single-Home & Low-Delay Slice

Network Program: D:0:0:3:X1::

D::/40 locator block is associated with Low Delay Flex-Algo

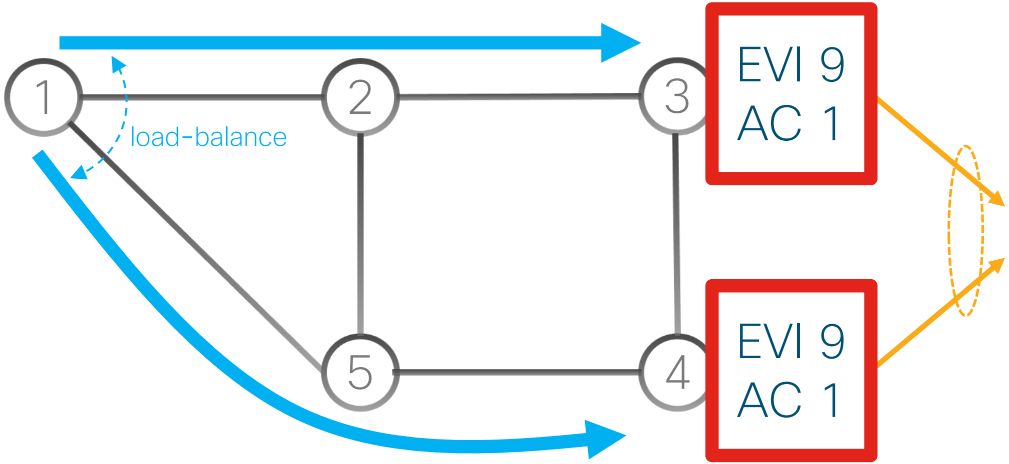


No SRH!

EVPN VPWS MH All-Active & Best-Effort Slice

Network Program: B:0:0:3:X1:: or B:0:0:4:X1::

B::/40 locator block is associated with ISIS base algo (Low Cost)



No SRH!

Load-balancing

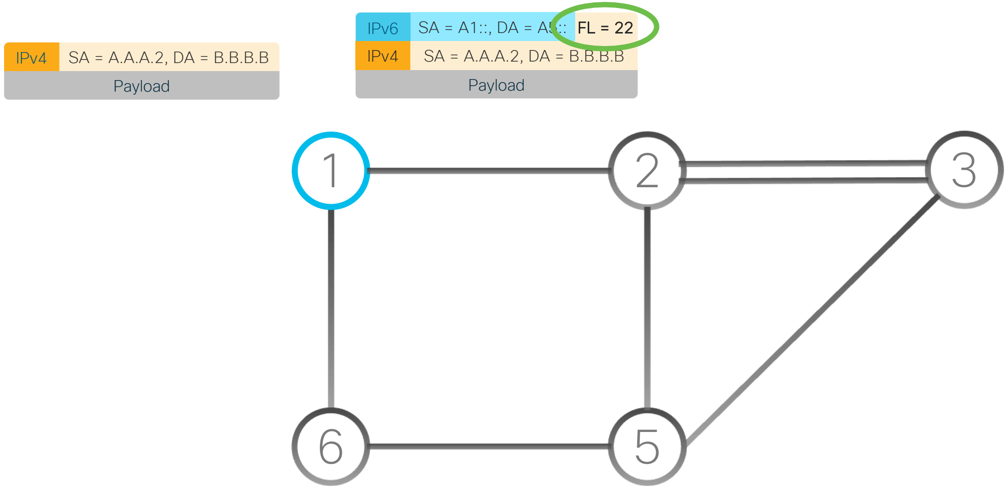
- 20-bit entropy
- No additional protocol
  - infamous mpls entropy label

```
 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                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Load-balancing

- Action at the ingress of SRv6 domain

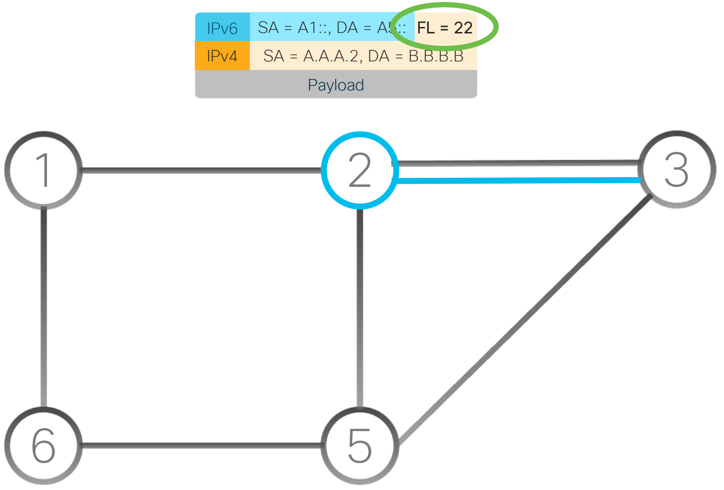
Flow Label is the result of the hash of the inner packet



Load-balancing

- Action at a transit node

Outer Flow Label used for hashing

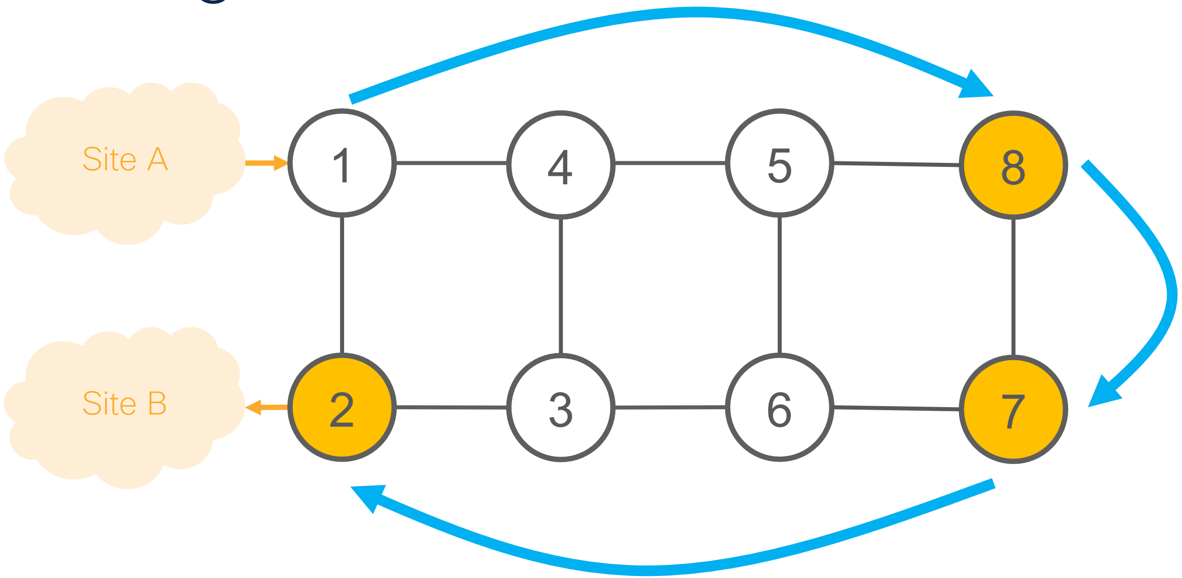


SRv6 SRH compression

draft-ietf-spring-srv6-srh-compression
NEXT and REPLACE C-SID flavors

- Program
  - list of instructions contained in DA/SRH
- Instruction
  - SRv6 SID
- C-SID Container
  - SRv6 SID that contains a list of C-SIDs
- C-SID
  - The locator-node and function bits of a SID that supports compressed encoding of SIDs

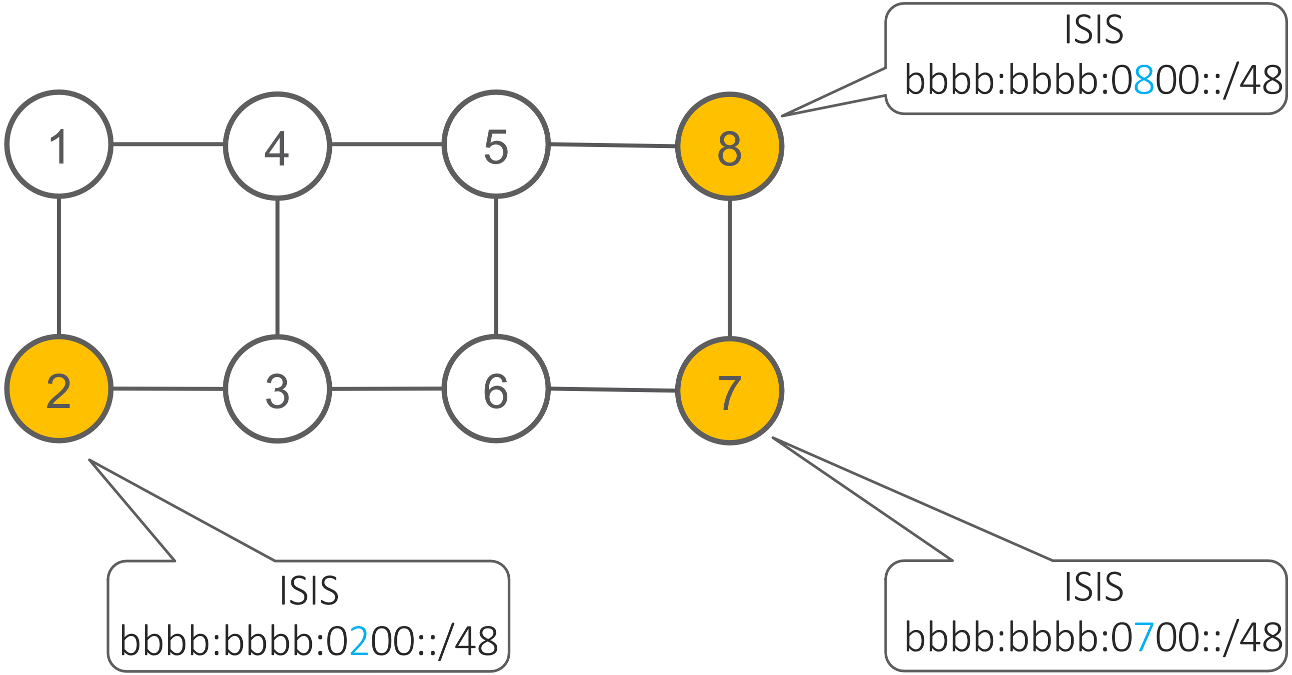
Illustration: go to 8 then 7 then 2 and decaps



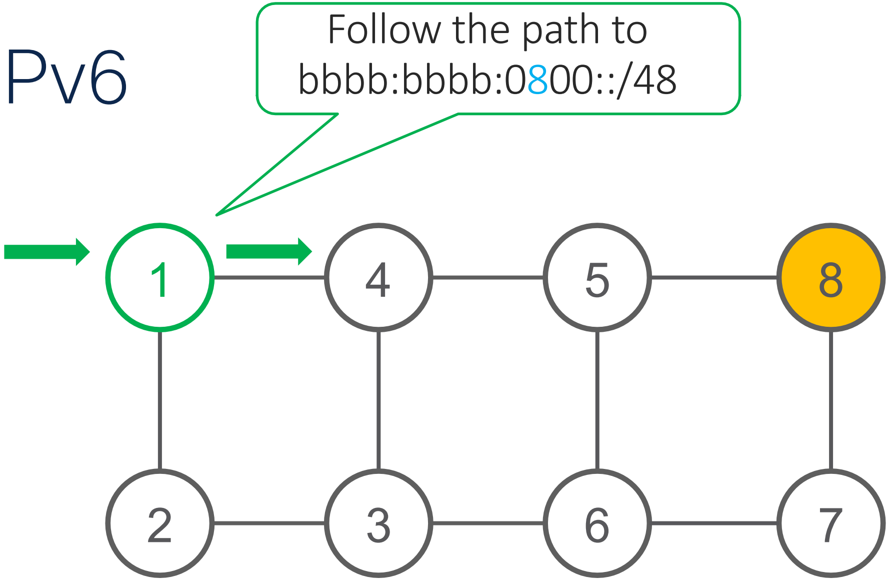
One single NEXT C-SID container in the DA is enough

DA = bbbb:bbbb:0800:0700:0200:FDT4:0000:0000

Basic IP Routing: no new extension

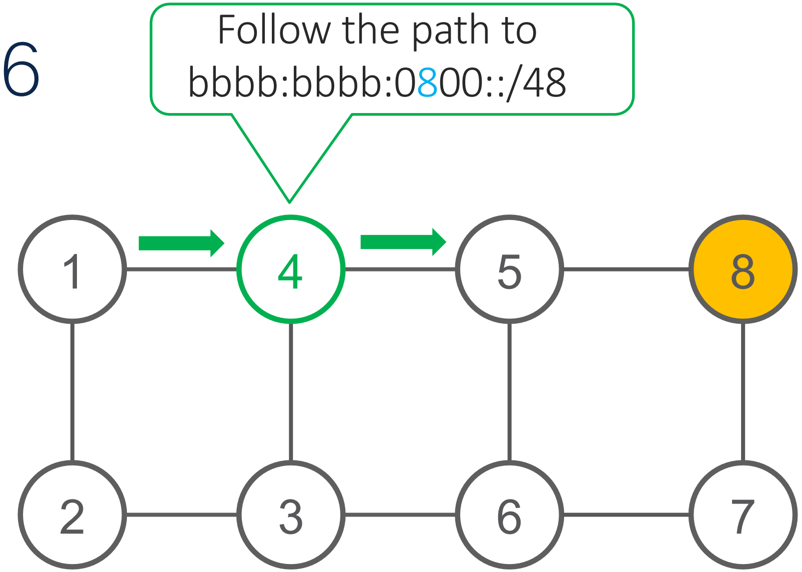


@1: basic IPv6



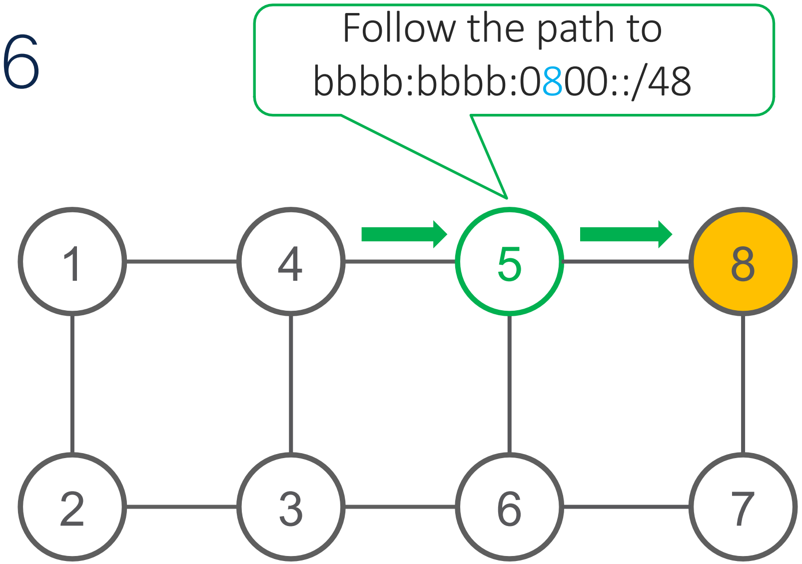
DA = bbbb:bbbb:0800:0700:0200:FDT4:0000:0000

@4: basic IPv6



DA = bbbb:bbbb:0800:0700:0200:FDT4:0000:0000

@5: basic IPv6



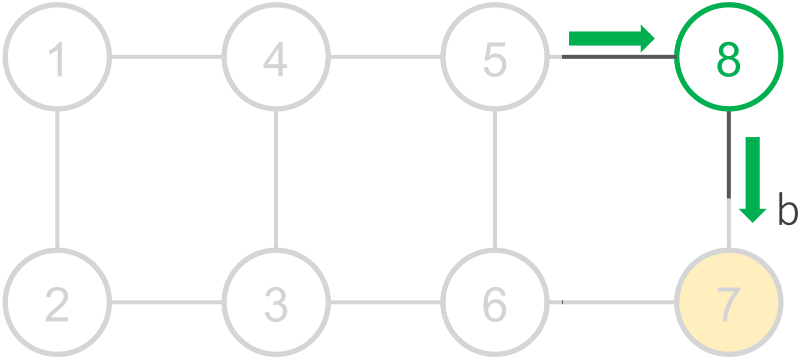
DA = bbbb:bbbb:0800:0700:0200:FDT4:0000:0000

@8: Copy Args and Forward

Rx'd DA: bbbb:bbbb:0800:0700:0200:FDT4:0000:0000

SHIFT << 16

Tx'd DA: bbbb:bbbb:0700:0200:FDT4:0000:0000:0000

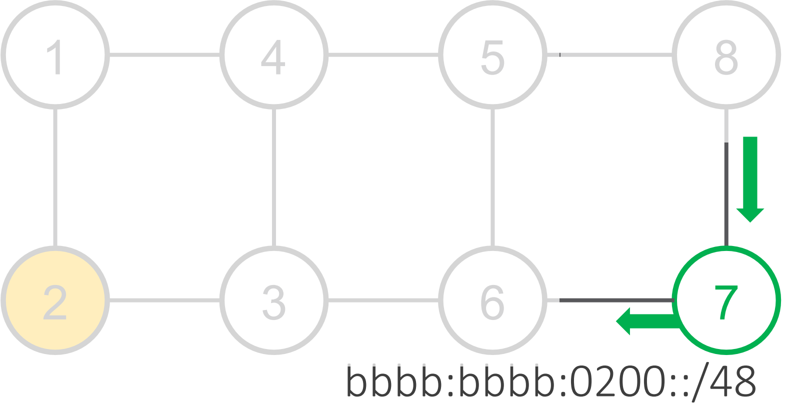


bbbb:bbbb:0700:

:/48

```
FIB Longest-Match bbbb:bbbb:0800::/48 → SRv6 function:
        Copy arguments into current function
        Set last remaining bits to 0
        Lookup the updated DA and forward
```

@7: Copy Args and Forward



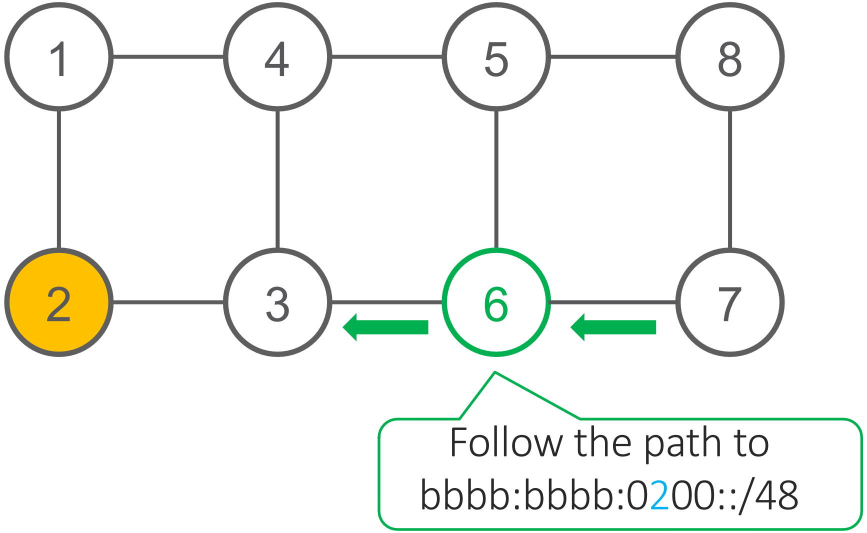
Rx'd DA: bbbb:bbbb:0700:0200:FDT4:0000:0000:0000

SHIFT << 16

Tx'd DA: bbbb:bbbb:0200:FDT4:0000:0000:0000:0000

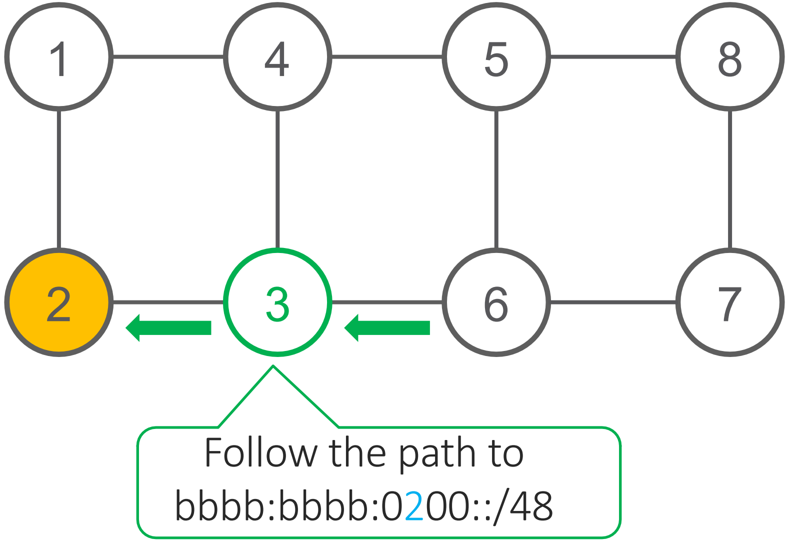
```
FIB Longest-Match bbbb:bbbb:0700::/48 → SRv6 Function:
        Copy arguments into current function
        Set last remaining bits to 0
        Lookup the updated DA and forward
```

@6: basic IPv6



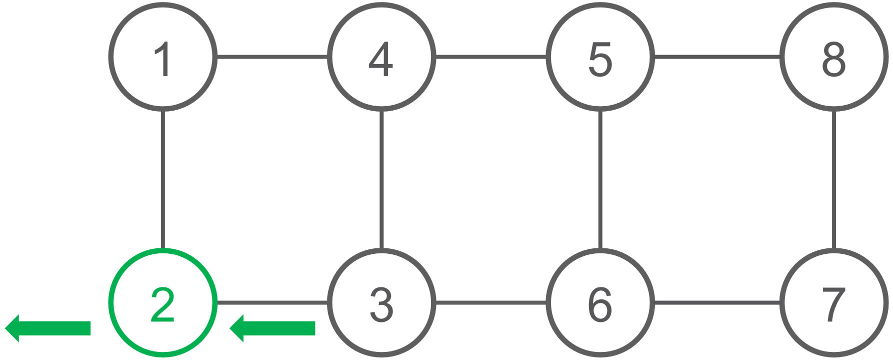
DA = bbbb:bbbb:0200:FDT4:0000:0000:0000:0000

@3: basic IPv6



DA = bbbb:bbbb:0200:FDT4:0000:0000:0000:0000

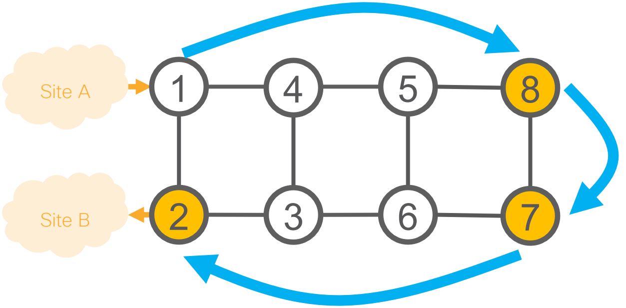
@2: SRv6 End.DT4 behavior



Rx'd DA: bbbb:bbbb:0200:FDT4:0000:0000:0000:0000

```
FIB Longest-Match bbbb:bbbb:0200:FDT4::/64 → SRv6 function:
  Decapsulate and forward inner IPv4 packet to Site B
```

Recap



- @1: inner packet P encapsulated with outer DA
  bbbb:bbbb:0800:0700:0200:FDT4:0000:0000
- @4 & @5: classic IP forwarding, outer DA unchanged
- @8: SRv6 NEXT behavior: copy args and forward, outer DA becomes
  bbbb:bbbb:0700:0200:FDT4:0000:0000:0000
- @7: SRv6 NEXT behavior: copy args and forward, outer DA becomes
  bbbb:bbbb:0200:FDT4:0000:0000:0000:0000
- @6 & @3: classic IP forwarding, outer DA unchanged
- @2: SRv6 End.DT4: Decapsulate and forward inner packet

SRv6

- IETF proposed standard
- Network Programming: RFC8754, RFC8986
- Use cases: TILFA, any VPN
- Compression: Working group draft

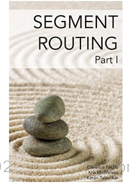
Resources / Stay Up-To-Date

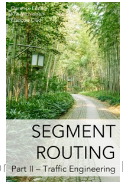
http://www.segment-routing.net/

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

https://twitter.com/SegmentRouting

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





Segment Routing, Part I / II - Textbooks