Network Programming with SRv6

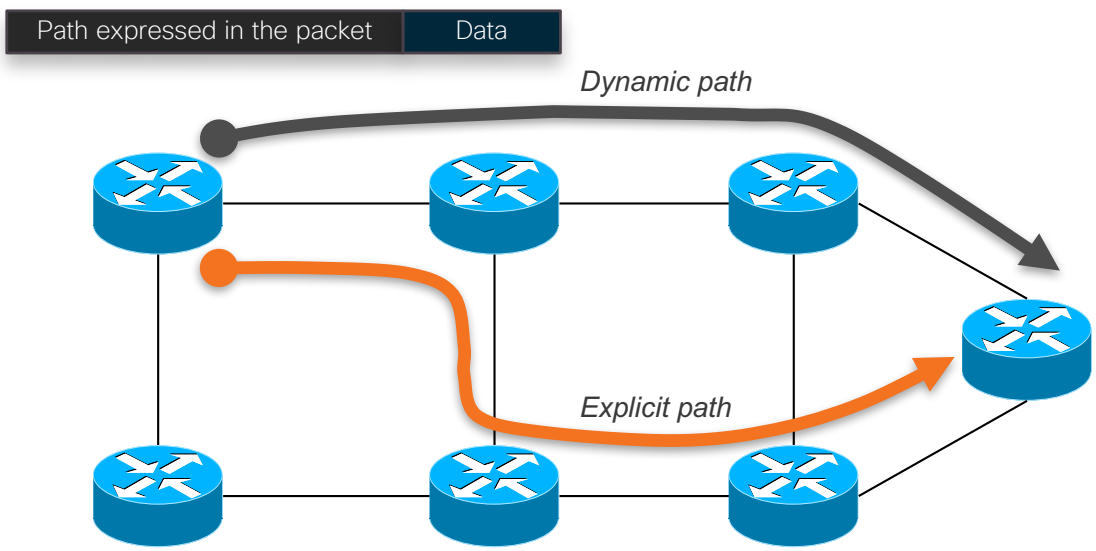
Hernán Contreras G.
Technical Solution Architect, Cisco Systems
LACNOG - September 2018

Agenda

- Introduction to Segment Routing
- Introduction to SRv6
- SRv6 Network Programming
- SRv6 NP Use Cases
- SRv6 Interoperability and Demos
- References

Segment Routing

Segment Routing – Technical view



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)

IETF

- Strong commitment for standardization and multi-vendor support
- SPRING Working-Group (started Nov 2013)
- All key documents are WG-status
- Over 25 drafts maintained by SR team
  - Over 50% are WG status
  - Over 75% have a Cisco implementation
- Several interop reports are available
- First RFC document - RFC 7855 (May 2016)
- RFC 8402 – Segment Routing Architecture

www.segment-routing.net
tools.ietf.org/wg/spring/

Technology and Problem Statement

- Architecture (draft)
- Problem Statement
  - Generic (draft)
  - Resiliency (draft)
  - IPv6 (draft)
  - OAM (draft)
- Applicability
  - SR illustration to problem statement (draft)
  - Centralized Egress Peer Engineering (draft)

Protocol Extension

- ISIS extension for SR (draft)
- OSPF extension for SR (draft)
- OSPFv3 extension for SR (draft)
- BGP-LS extension for SR (draft)
- BGP-LS extension for SR EPE use-case (draft)
- PCEP extension for SR (SR ext, setup method)

FRR

- Topology-Independent LFA FRR with SR (draft)

MPLS instantiation of Segment Routing

- MPLS support for SR (draft)
- SR/LDP interaction and interworking (draft)

IPv6 instantiation of Segment Routing

- IPv6 SR routing extension header (draft)
- IPv6 use-cases (draft)

OAM

- SR/LSP Ping (draft)
- OAM (draft)

A Standardized Technology

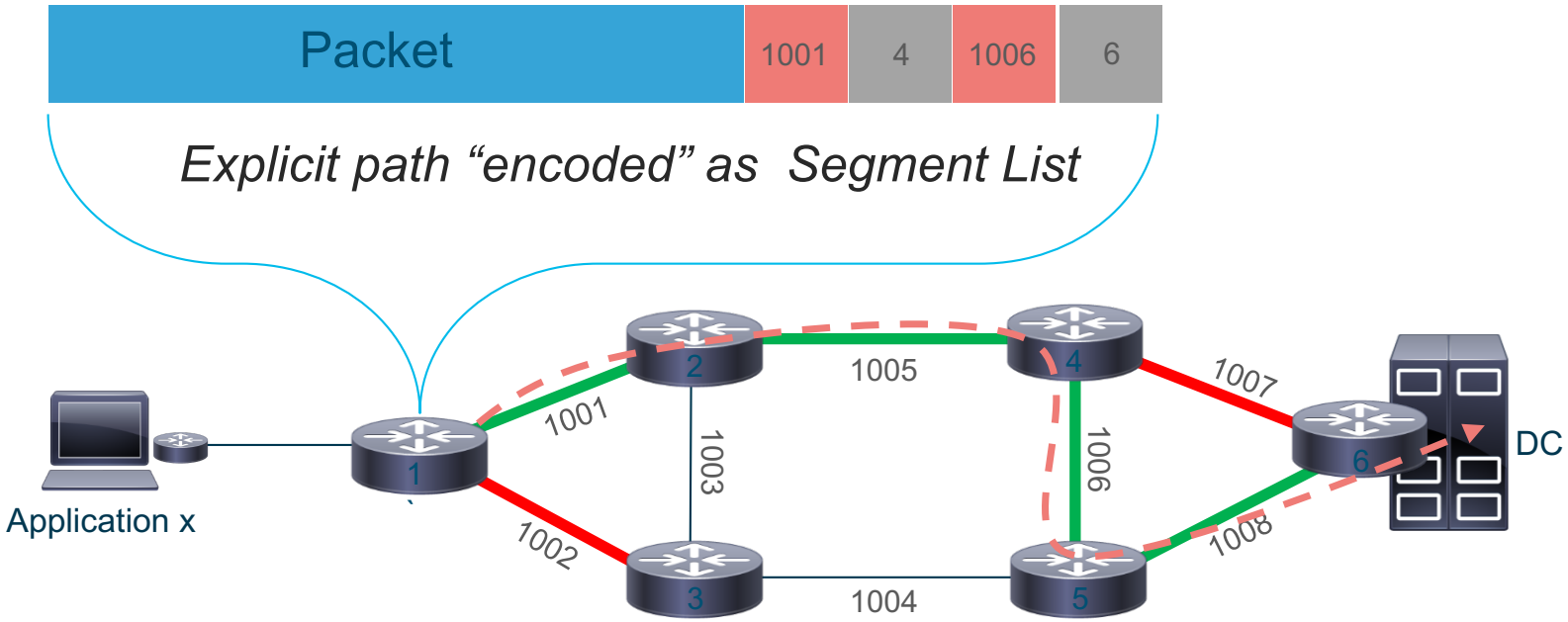
No Vendor Lock-In

- IETF Standardization in SPRING working group
- Protocol extensions progressing in multiple groups
  - IS-IS
  - OSPF
  - PCE
  - IDR
  - 6MAN
- Broad vendor and customer support
- 50% Draft-based working implementations
- 25% under development

Sample IETF Documents
Problem Statement and Requirements (RFC 7855)
Segment Routing Architecture (RFC 8402)
IPv6 SPRING Use Cases (RFC 8354)
Segment Routing with MPLS data plane (draft-ietf-spring-segment-routing-mpls)
Topology Independent Fast Reroute using Segment Routing (draft-francois-rtgwg-segment-routing-ti-lfa)
IS-IS Extensions for Segment Routing (draft-ietf-isis-segment-routing-extensions)
OSPF Extensions for Segment Routing (draft-ietf-ospf-segment-routing-extensions)
PCEP Extensions for Segment Routing (draft-ietf-pce-segment-routing)

Around 40 IETF drafts in progress

SR = near-stateless traffic engineering

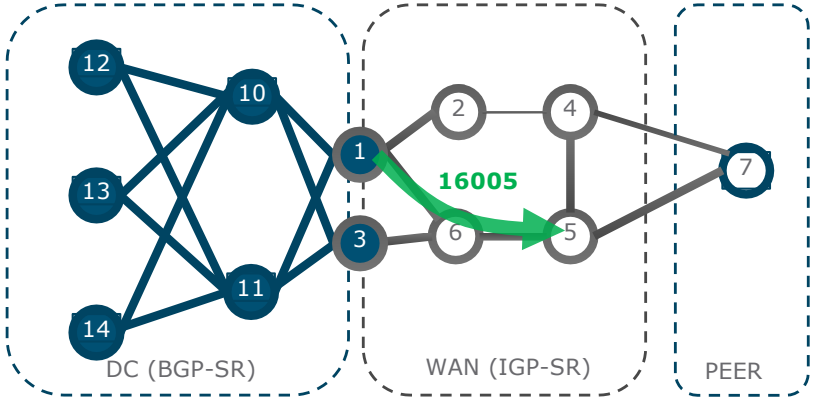


SR Segments

IGP Prefix Segment

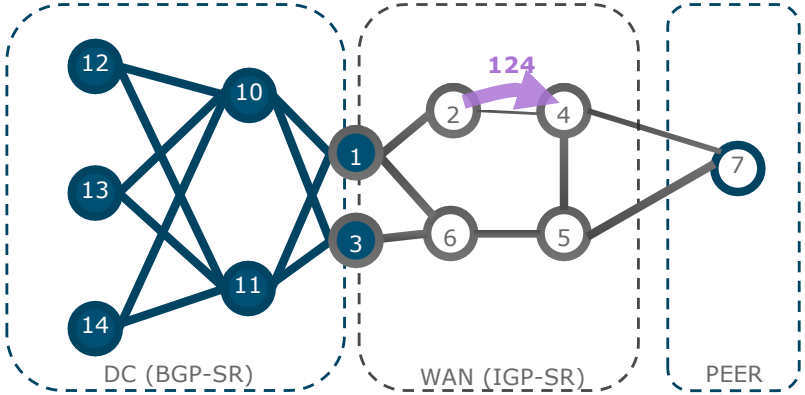
- Signaled by ISIS/OSPF
- Minor extensions to the existing link-state routing protocols (OSPF and IS-IS)
- Shortest-path to the IGP prefix
- Global in SR domain
- SRGB + Index => 16000+5 = 16005

SRGB: Segment Routing Global Block:
default [16000 – 23999]



IGP Adjacency Segment

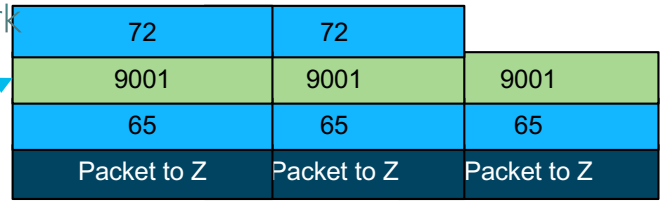
- Signaled by ISIS/OSPF
- Minor extensions to the existing link-state routing protocols (OSPF and IS-IS)
- Forward on the IGP adjacency
- Local
- Automatically allocated by the router



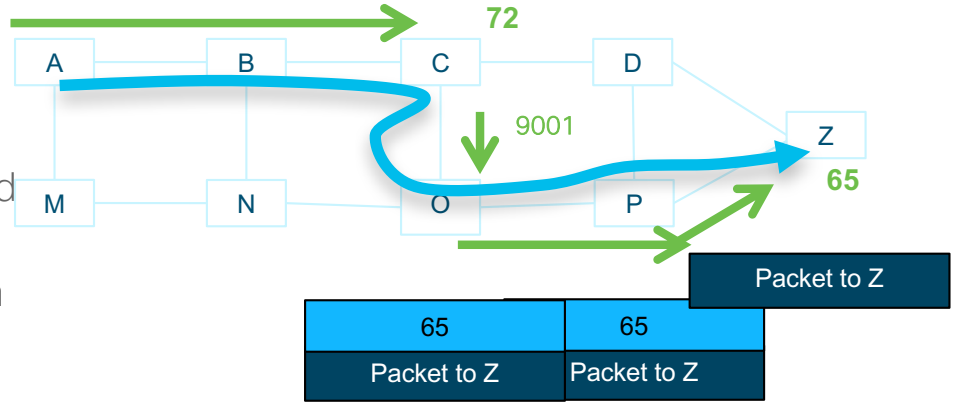
Segment Routing

- Source Routing
  - the source chooses a path and encodes it in the packet header as an ordered list of segments
  - the rest of the network executes the encoded instructions

The flow state is in the header, not in the network



Segment: an identifier for any type of instruction (forwarding or service)



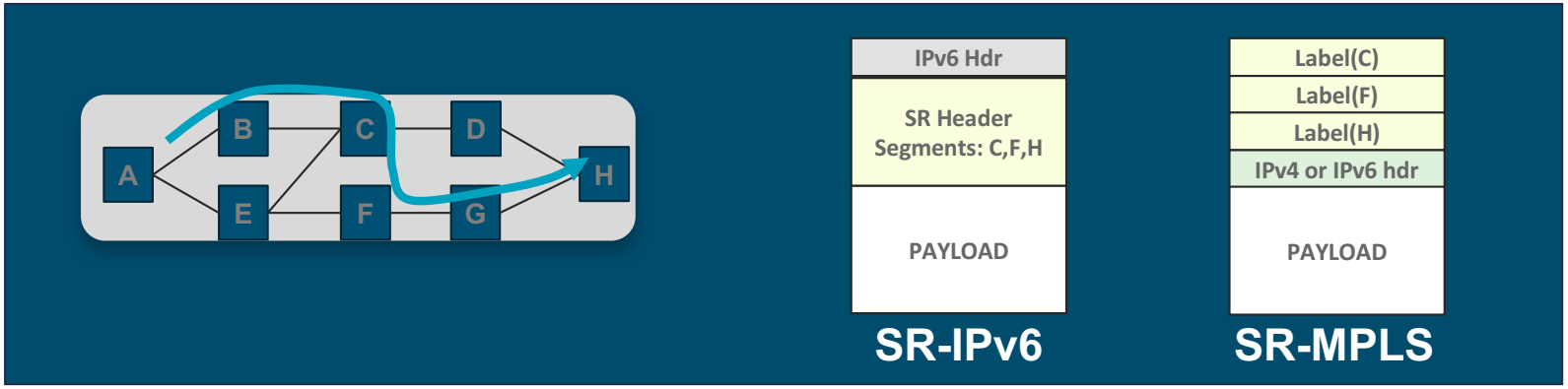
- Forwarding Plane:
  - MPLS: an ordered list of segments is represented as a stack of labels
  - IPv6: an ordered list of segments is encoded in a routing extension header

Segment Routing Dataplane

Segment Routing applies to both IPv6 and MPLS dataplanes
Difference is in the bits encoded in the packet not in the architecture

MPLS: an ordered list of segments is represented as a stack of labels
- Segment Routing re-uses MPLS data plane without any change
- Segment represented as MPLS label
- Applicable to IPv4 and IPv6 address families
- defined in ietf-spring-segment-routing-mpls

IPv6: an ordered list of segments is encoded in a routing extension header



SRv6 Header

draft-ietf-6man-segment-routing-header-14

This document describes the Segment Routing Extension Header and how it is used by Segment Routing capable nodes.

```
Network Working Group                                   S. Previdi, Ed.
Internet-Draft                                              C. Filsfils
Intended status: Standards Track                             K. Raza
Expires: September 14, 2017                                 D. Dukes
                                                  Cisco Systems, Inc.
                                                             J. Leddy
                                                             B. Field
                                                              Comcast
                                                             D. Voyer
                                                           D. Bernier
                                                          Bell Canada
                                                        S. Matsushima
                                                             Softbank
                                                             I. Leung
                                                  Rogers Communications
                                                           J. Linkova
                                                               Google
                                                             E. Aries
                                                             Facebook
                                                           T. Kosugi
                                                                 NTT
                                                           E. Vyncke
                                                  Cisco Systems, Inc.
                                                            D. Lebrun
                                       Universite Catholique de Louvain
                                                        D. Steinberg
                                                 Steinberg Consulting
                                                            R. Raszuk
                                                           Bloomberg
                                                       March 13, 2017


                 IPv6 Segment Routing Header (SRH)
           draft-ietf-6man-segment-routing-header-06
```

IPv6 SR Header

- IPv6 header
  - Next header field: **43 → Routing**
- IPv6 Routing extension header
  - Generic header format defined in RFC 8200
    - Next Header: IPv4, TCP, UDP, …
    - Hdr Ext Len: Any IPv6 device can skip this header
    - Segments Left: Ignore extension header if equal to 0
  - Specific data depends on Routing Type field:
    - ~~*0 Source Route*~~ (deprecated since 2007)
    - ~~*1 Nimrod*~~ (deprecated since 2009)
    - 2 Mobility (RFC 6275)
    - 3 RPL Source Route (RFC 6554)
    - **4 Segment Routing** (suggested value 4)

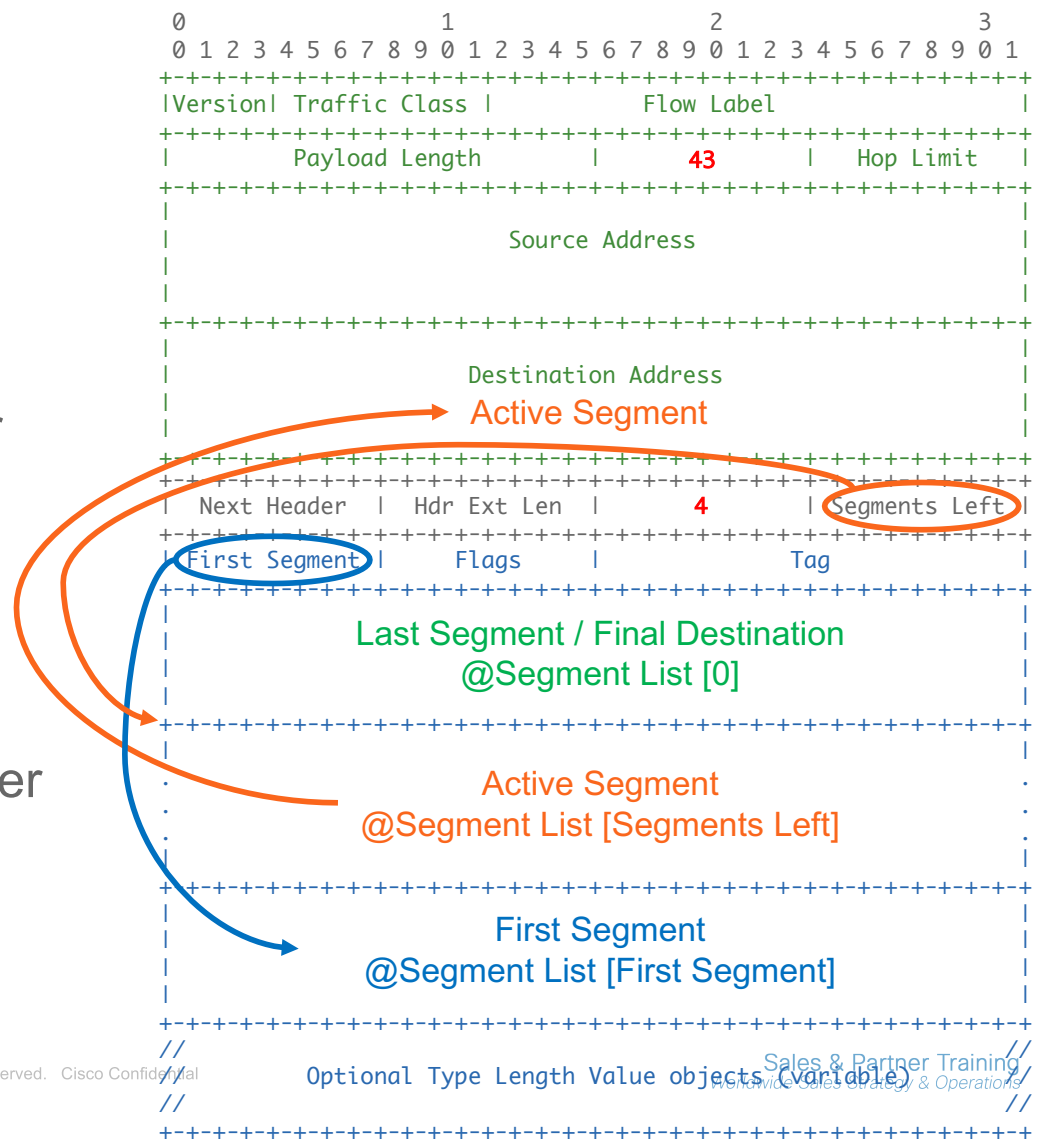
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |      43       |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Source Address                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Destination Address                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Header   | Hdr Ext Len   | Routing Type  | Segments Left |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                       type-specific data                      .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination.

https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml

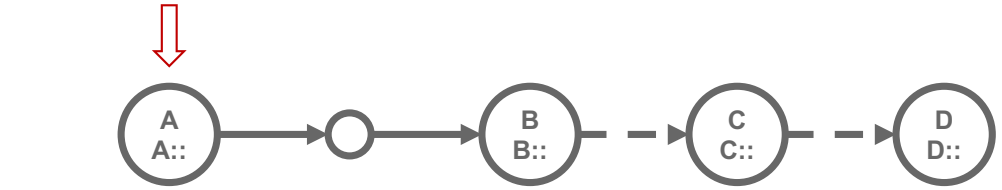
IPv6 SR Header

- Each segment is an IPv6 address
- Segments are encoded in reverse order
  - Last segment index is 0
  - First segment index is **First Segment**
  - Active segment index is **Segments Left**
- Active Segment is copied in the Destination Address field of the IP header
- Additional data can be stored in TLVs
  - Security (HMAC), NFV metadata, …



SRv6 Source Node

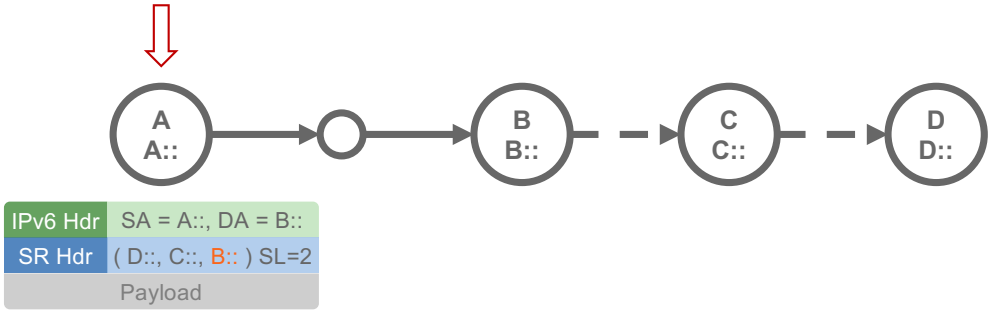
- Source node is SR-capable
- SR Header (SRH) is created with
  - Segment list in reversed order of the path
    - Segment List [0] is the LAST segment
    - Segment List [$n-1$] is the FIRST segment
  - Segments Left is set to $n-1$
  - First Segment is set to $n-1$
- IP DA is set to the first segment
- Packet is send according to the IP DA
  - Normal IPv6 forwarding



IPv6 Hdr	SA = A::, DA = B::
SR Hdr	(D::, C::, B::) SL=2
Payload	

IPv6 Hdr	Version	Traffic Class	Flow Label	
	Payload Length		Next = 43	Hop Limit
	Source Address = A::			
	Destination Address = B::			
SR Hdr	Next Header	Len= 6	Type = 4	SL = 2
	First = 2	Flags		RESERVED
	Segment List [0] = D::			
	Segment List [1] = C::			
	Segment List [2] = B::			
	Payload			

Non-SR Transit Node



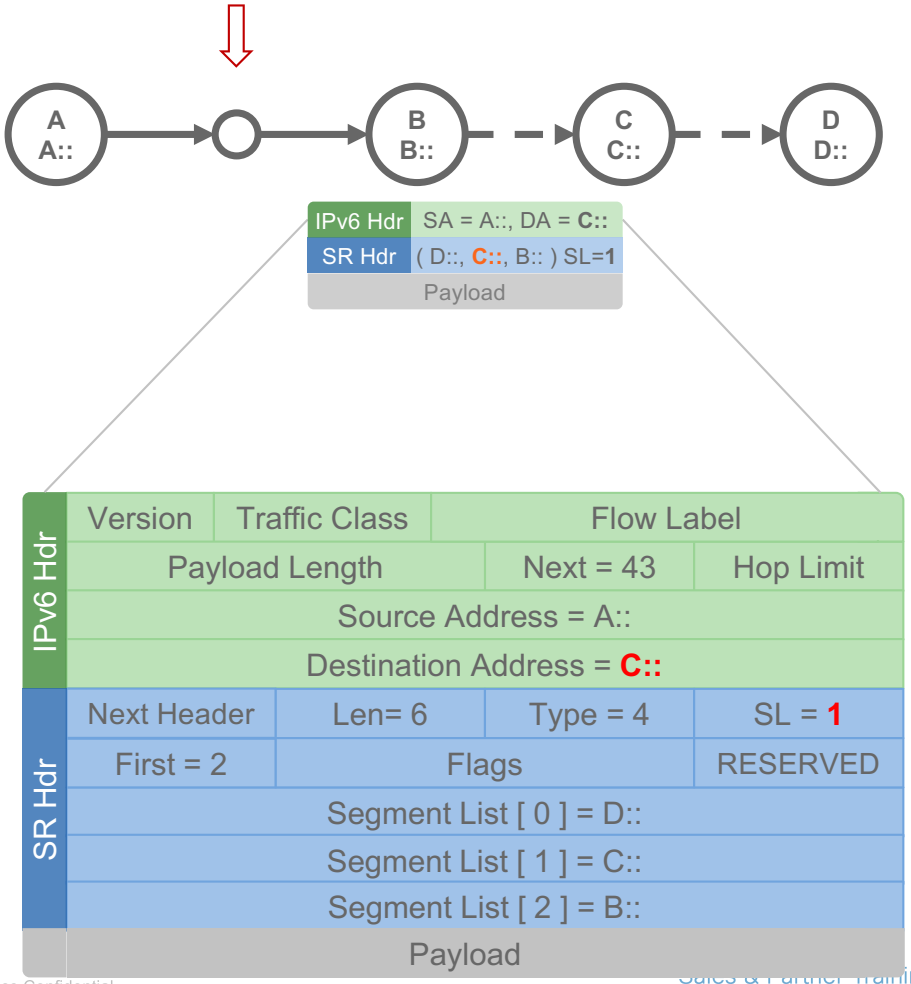
Enabling SR-IPv6, means that ONLY the nodes that have to process the packet header must have SR-IPv6 dataplane support

All other nodes in the infrastructure are just plain IPv6 nodes

- Plain IPv6 forwarding
- Solely based on IPv6 DA
- No SRH inspection or update

SR Segment Endpoints

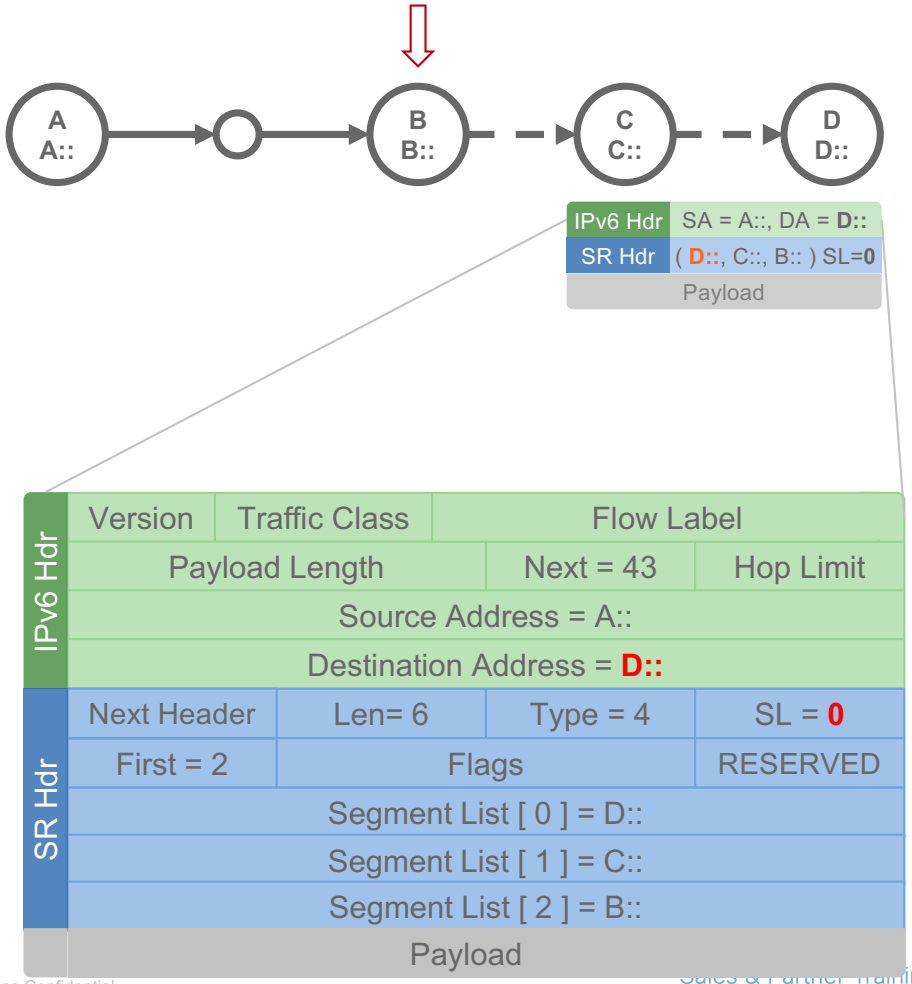
- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA



SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA
  - ELSE (Segments Left = 0)
    - Remove the IP and SR header
    - Process the payload:
      - Inner IP: Lookup DA and forward
      - TCP / UDP: Send to socket
      - …

Standard IPv6 processing
The final destination does not have to be SR-capable.



SRv6 Network Programming

- SRv6 looks similar to SR MPLS, as in both cases you have a list of Segment IDs used to direct traffic over a specific path through the network
  - the exception of the Segment ID size – 128-bits for SRv6 compared to 32-bits for SR MPLS,
- That increase in size on SID opens the door to an entirely new paradigm: “Network Programming.”
  - With 128 bits, 4 times 32 bits, you can pack more than mere IP addresses into a Segment ID and hence go beyond routing purposes.
- 128-bits Segment IDs can be used and allocated for different purposes, for example:
  - The first 64 bits can be used to direct traffic to a specific node in the network – the *“main body”* of the program
  - The next 32 bits can be used to enforce some actions on the traffic – the *“function”*part
  - The remaining 32 bits can be used to pass some additional information – the *“argument”* part

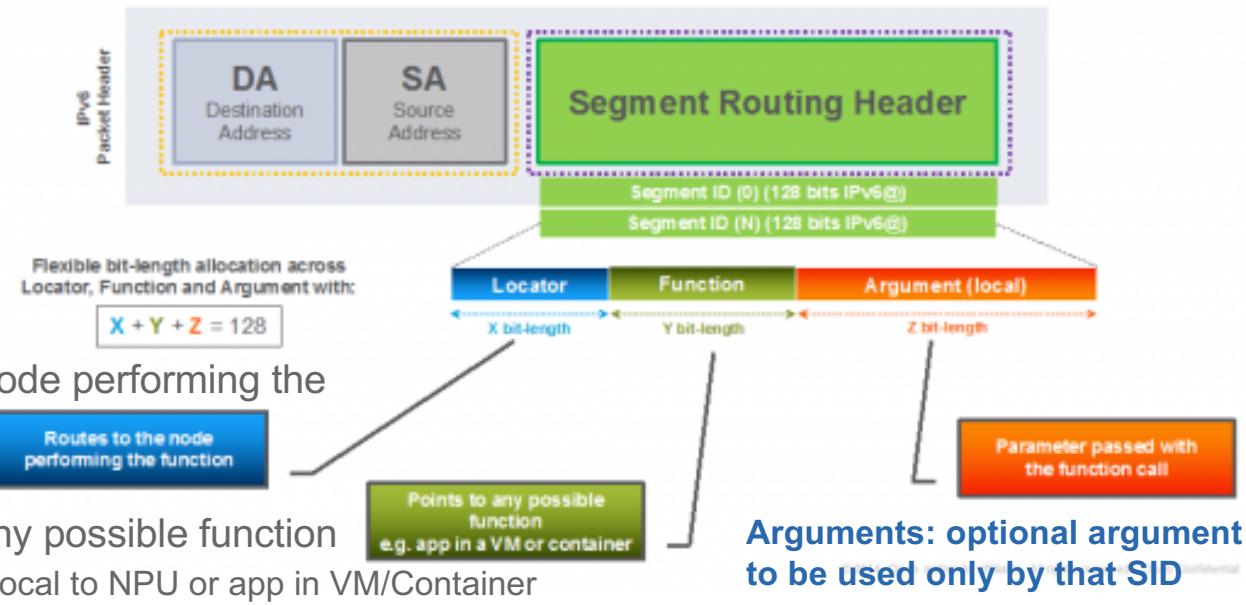
128-bit SRv6 SID

Locator: routed to the node performing the function
Function: any possible function
Flexible bit-length selection

SRv6 Network Programming

SRv6 goes one step further by enabling the infrastructure to perform some actions on the applications: you have now the ability to code directly into each packet header where the traffic should be sent and how the traffic should be treated

Network Programming



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

https://blogs.cisco.com/sp/segment-routing-ready-to-surf-the-v6-wave

SRv6 Network Programming

```
SPRING                                                        C. Fisfils
Internet-Draft                                     Cisco Systems, Inc.
Intended status: Standards Track                             J. Leddy
Expires: September 10, 2017                                   Comcast
                                                            D. Voyer
                                                          D. Bernier
                                                        Bell Canada
                                                       D. Steinberg
                                                 Steinberg Consulting
                                                           R. Raszuk
                                                       Bloomberg LP
                                                      S. Matsushima
                                                  SoftBank Telecom
                                                           D. Lebrun
                                   Universite catholique de Louvain
                                                       B. Decraene
                                                            Orange
                                                        B. Peirens
                                                          Proximus
                                                        S. Salsano
                                   Universita di Roma "Tor Vergata"
                                                           G. Naik
                                              Drexel University
                                                        H. Elmalky
                                                          Ericsson
                                                     P. Jonnalagadda
                                                          M. Sharif
                                              Barefoot Networks
                                                       A. Ayyangar
                                                            Arista
                                                          S. Mynam
                                           Dell Force10 Networks
                                                       A. Bashandy
                                                           K. Raza
                                                          D. Dukes
                                                           F. Clad
                                              P. Camarillo, Ed.
                                            Cisco Systems, Inc.
                                                  March 9, 2017


                        SRv6 Network Programming
              draft-filsfils-spring-srv6-network-programming-00
```

Segment format

Locator *Function*

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

- SRv6 SIDs are 128-bit addresses
  - **Locator**: most significant bits are used to **route** the segment to its **parent node**
  - **Function**: least significant bits identify the **action** to be performed on the **parent node**
    - **Argument** [optional]: Last bits can be used as a local function argument
- Flexible bit-length allocation
  - Segment format is **local knowledge** on the parent node
- SIDs have to be **specifically enabled** as such on their parent node
  - A local address **is not** by default a local SID
  - A local SID does not have to be associated with an interface

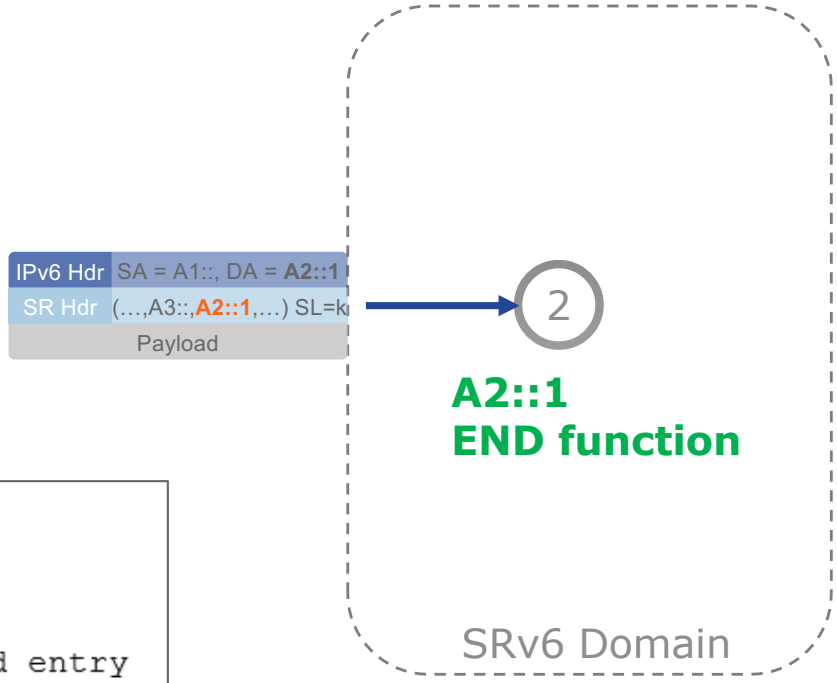
Endpoint behaviors specs summary

Codename	Behavior	
End	Endpoint	[PSP/USP flavors]
End.X	Endpoint with Layer-3 cross-connect	[PSP/USP flavors]
End.B6	Endpoint bound to an SRv6 policy	
End.B6.Encaps	Endpoint bound to an SRv6 Encapsulation policy	
End.DX6	Endpoint with decapsulation and IPv6 cross-connect	(per-CE VPN label)
End.DX4	Endpoint with decapsulation and IPv4 cross-connect	(per-CE VPN label)
End.DT6	Endpoint with decapsulation and specific IPv6 table lookup	(per-VRF VPN label)
End.DT4	Endpoint with decapsulation and specific IPv4 table lookup	(per-VRF VPN label)
End.DX2	Endpoint with decapsulation and Layer-2 cross-connect	

END SID

Node 2 advertises prefix A2::/64
(A2::/64 is the SID **locator**)

- END function
  - shortest-path to SID's endpoint
  - endpoint updates DA with next SID
  - endpoints forwards according to updated DA
  - 0:0:0:0:1 is commonly used for an END function




```
IF NH=SRH and SL > 0
   decrement SL
   update the IPv6 DA with SRH[SL]
   FIB lookup on updated DA
   forward accordingly to the matched entry
ELSE
   drop the packet
```

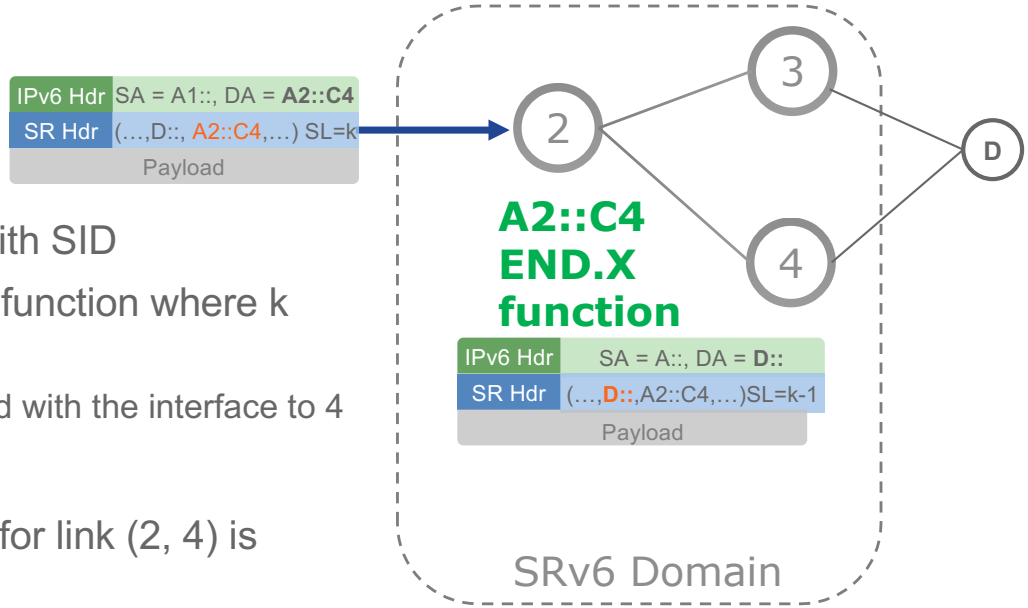
My LocalSID table: A2::1 -> End

END.X SID

Endpoint xconnect behavior (adjacency segment)

Forward on the interface associated with the Xconnect segment

- END.X function
  - shortest-path to SID's endpoint
  - endpoint updates DA with next SID
  - endpoint forwards to interface associated with SID
  - 0:0:0:0:Ck is commonly used for an END.X function where k identifies the interface
    - e.g. 0:0:0:0:C4 is the END.X of node 2 associated with the interface to 4

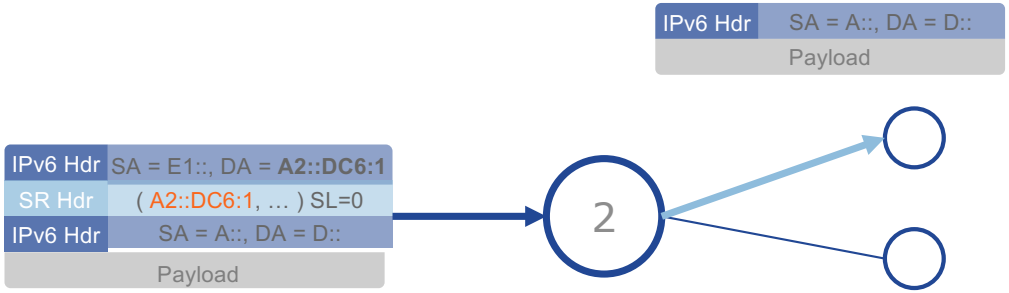


On Node 2, the *endpoint xconnect* behavior for link (2, 4) is associated with ID C4
The SID corresponding to *endpoint xconnect-(2,4)* behavior on node 2 is **A2::C4**

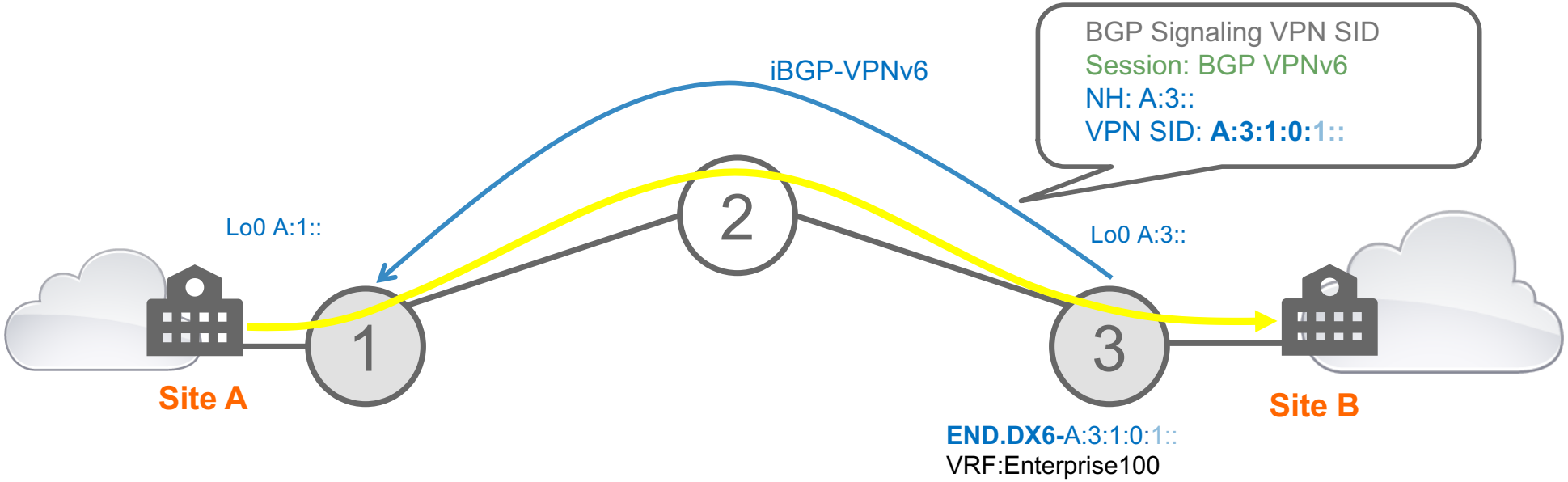
My LocalSID table: A2::C4 -> End.X {OIF=4; NH=2001::1}

End.DX6: Endpoint with decapsulation and IPv6 xconnect

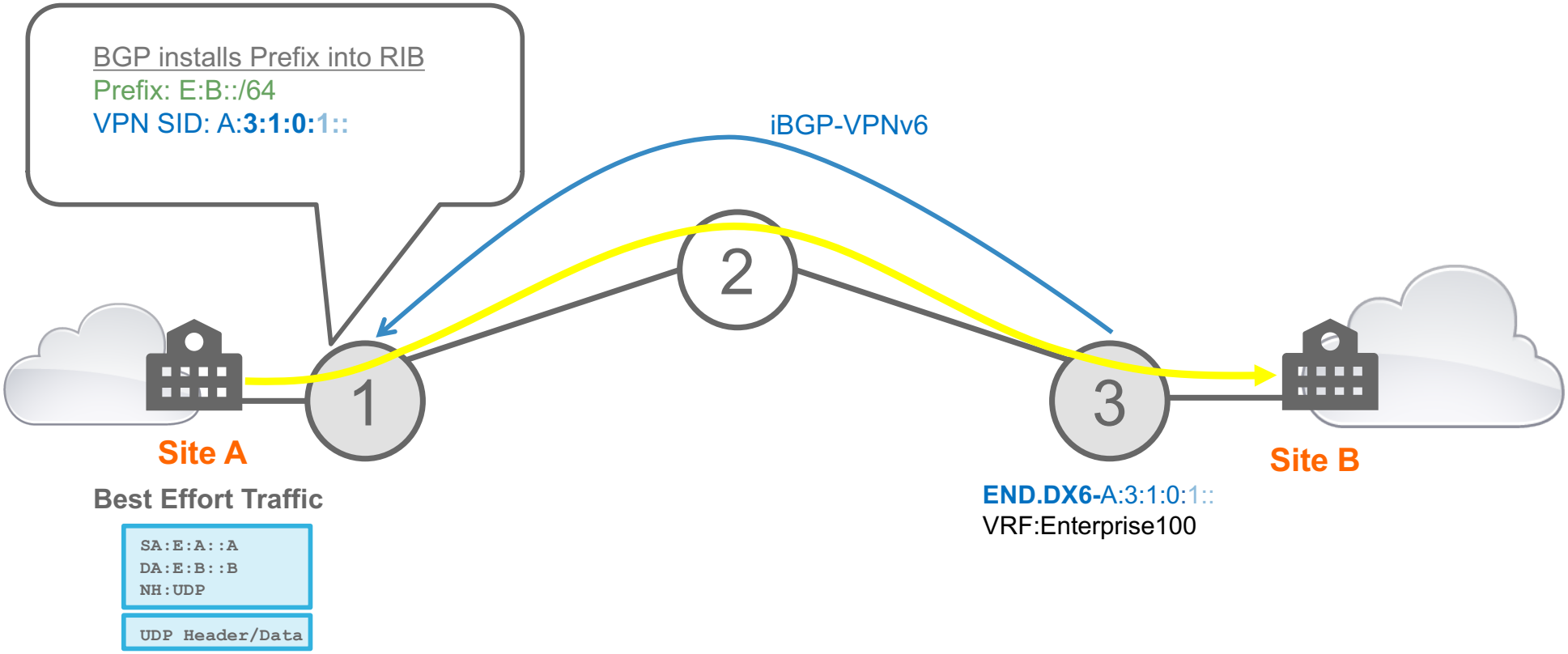
- Equivalent to per-CE VPN label in MPLS
- If (SL == 0 & NH == 41“IPv6”)
  - Pop the (outer) IPv6 header and its extension headers
  - **Forward on the outgoing interface associated with the localSID**
- My LocalSID table: A2::DC6:1 -> End.DX6 {OIF=1; NH=2001::1}



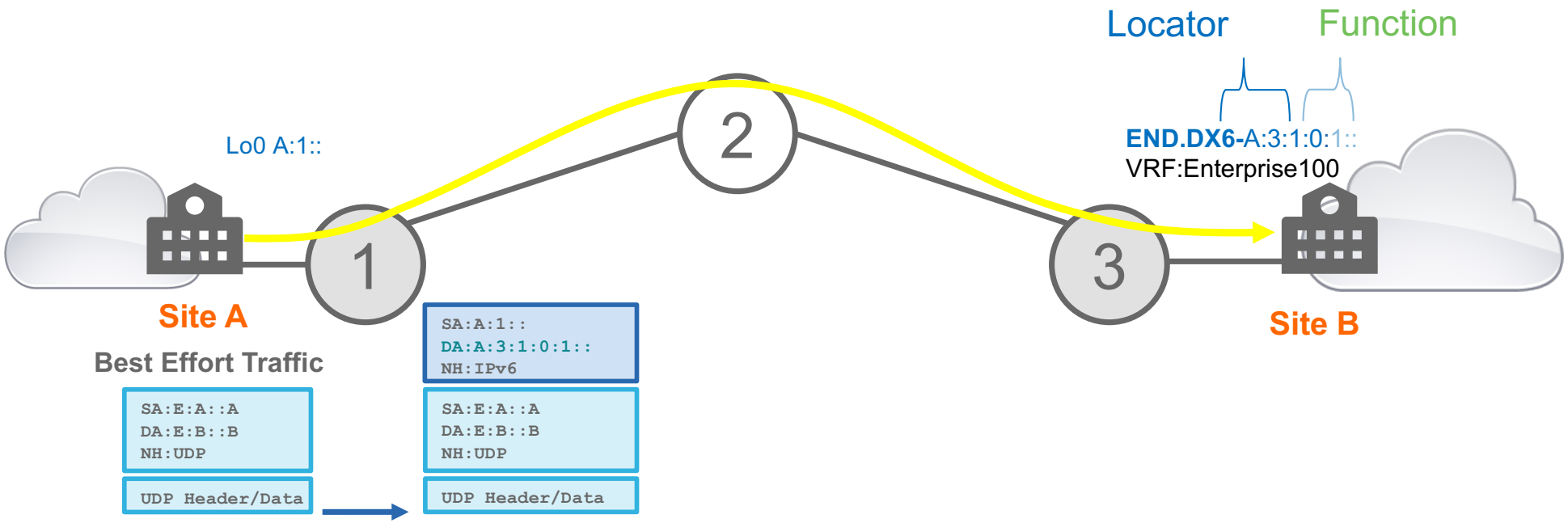
END.DX6 Use Case: BGP VPNv6



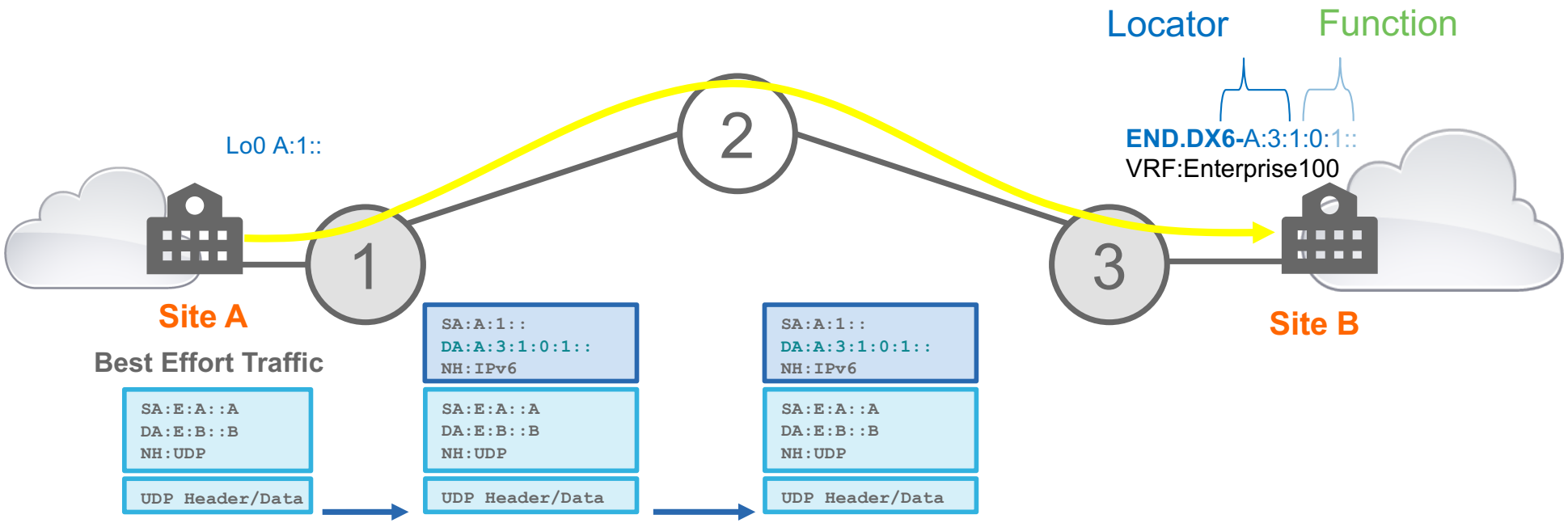
END.DX6 Use Case: BGP VPNv6



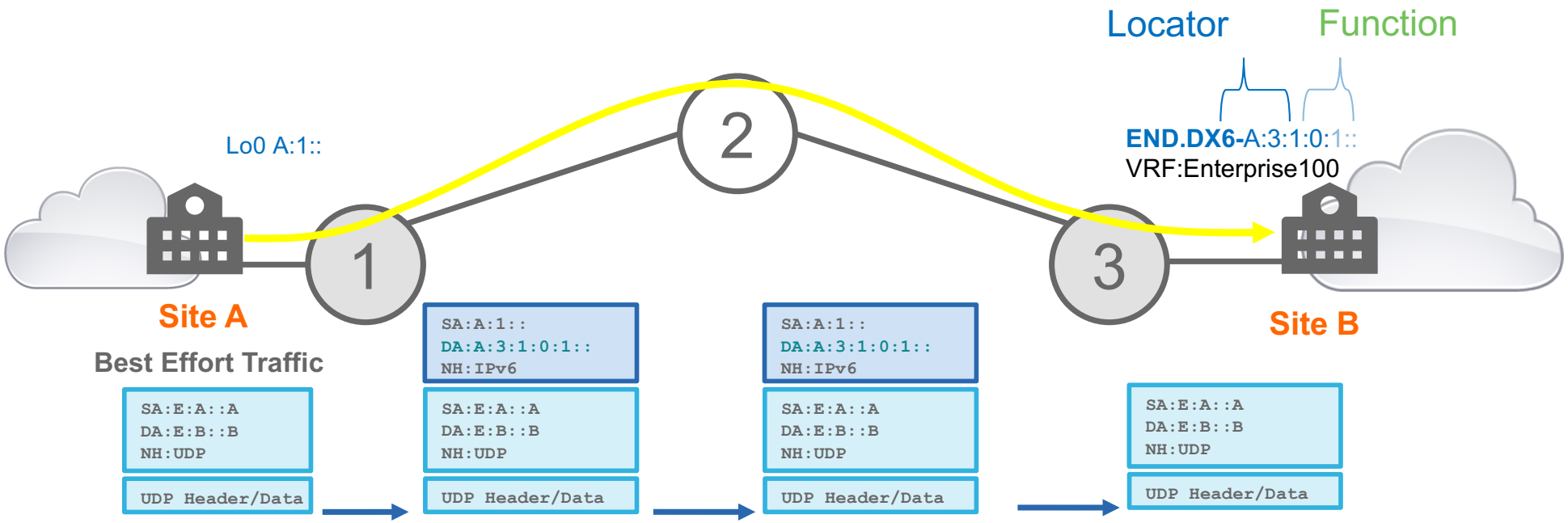
END.DX6 Use Case: BGP VPNv6



END.DX6 Use Case: BGP VPNv6



END.DX6 Use Case: BGP VPNv6



Versions: 00 draft-dawra-idr-srv6-vpn

Inter-Domain Routing
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2017

G. Dawra, Ed.
C. Filsfils
D. Dukes
P. Brissette
P. Camarilo
Cisco Systems
J. Leddy
Comcast
D. Voyer
D. Bernier
Bell Canada
D. Steinberg
Steinberg Consulting
R. Raszuk
Bloomberg LP
B. Decraene
Orange
S. Matsushima
SoftBank Telecom Japan
March 12, 2017

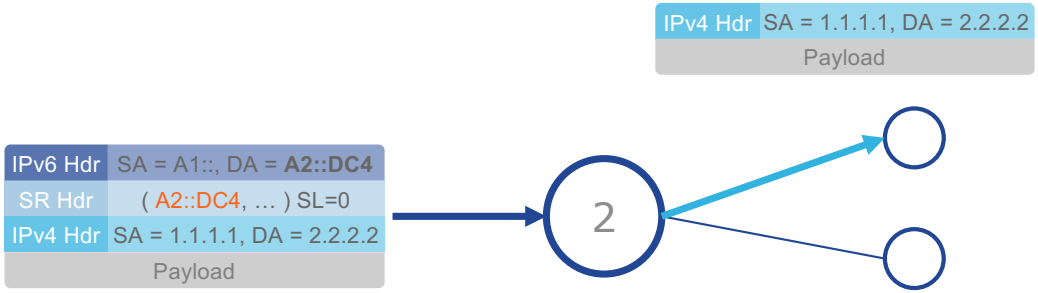
SRv6-based VPN (SRv6-VPN) refers to the creation of VPN between PE's leveraging the SRv6 dataplane and more specifically the END.DT* (crossconnect to a VRF) and END.DX* (crossconnect to a nexthop) functions defined in the SRv6 network programming document

Segment Routing v6 VPN

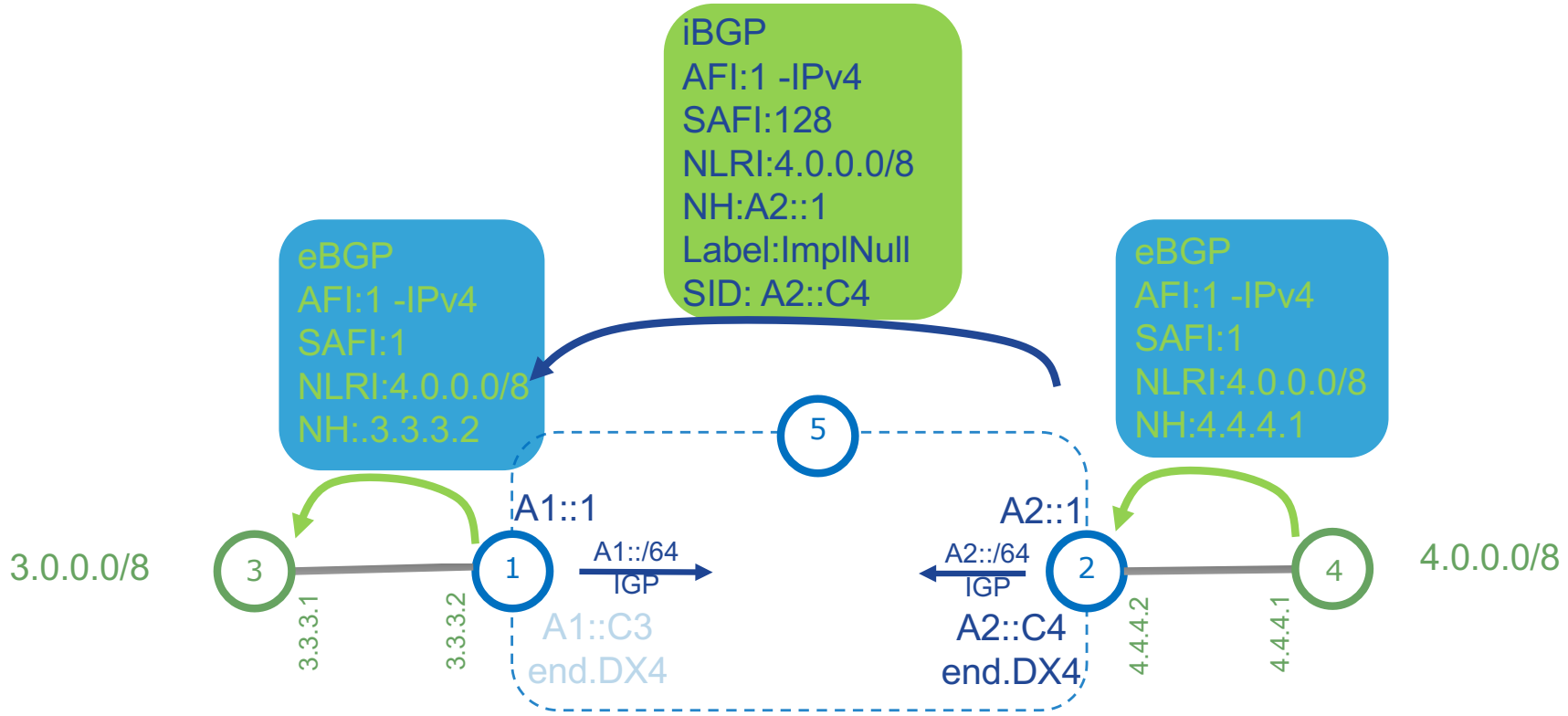
BGP Signaling of IPv6-Segment-Routing-based VPN Networks
draft-dawra-bgp-srv6-vpn-00.txt

End.DX4: Endpoint with decapsulation and IPv4 xconnect

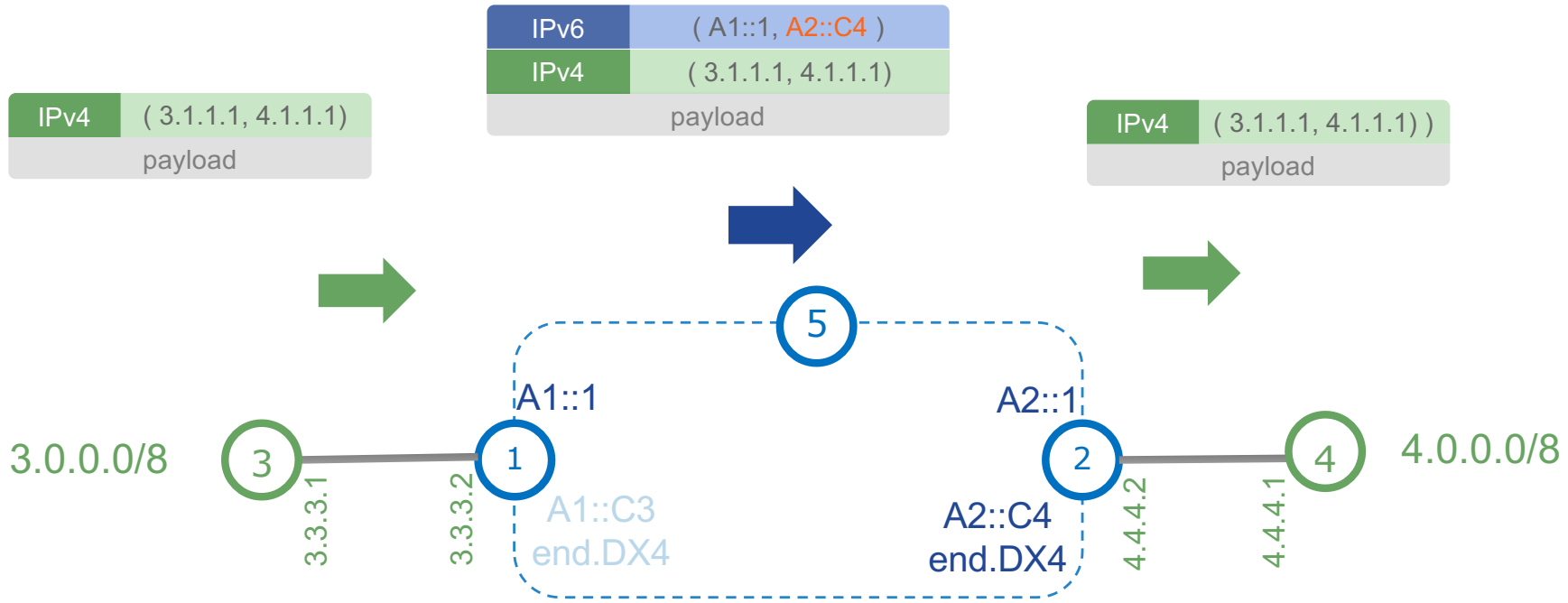
- Equivalent to per-CE VPN label in MPLS
- If (SL == 0 & NH == 4”IPv4”)
  - Pop the (outer) IPv6 header and its extension headers
  - **Forward on the outgoing interface associated with the localSID**
- My LocalSID table: A2::DC4 -> End.DX4 {OIF=1; NH=2.2.2.0}



VPNv4 – Basic SRv6 VPN –Control Plane



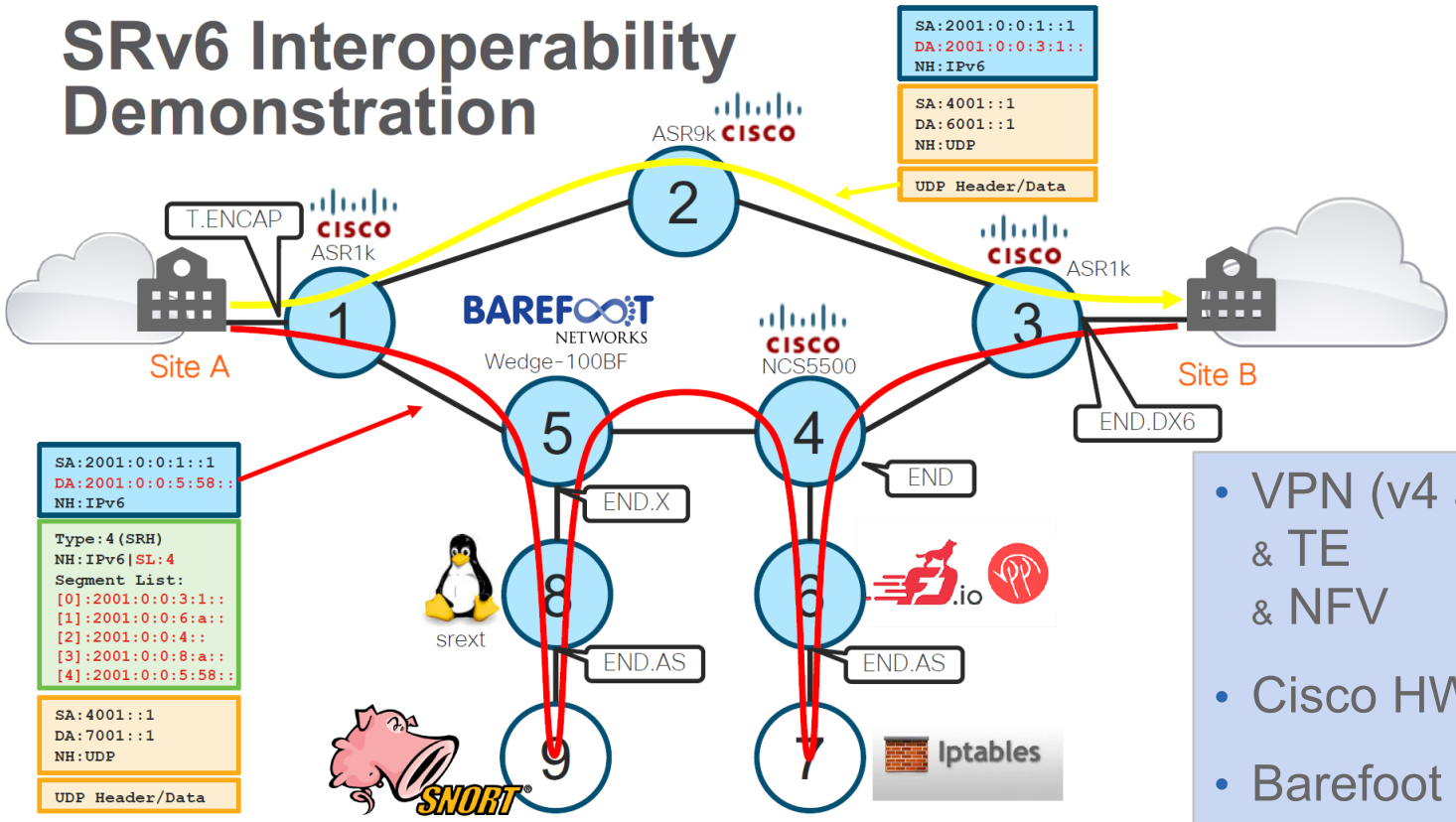
VPNv4 – Basic SRv6 VPN -Data Plane



Functions Defined in Net Programming

- **End** Endpoint function The SRv6 instantiation of a prefix SID
- **End.X** Endpoint function with Layer-3 cross-connect The SRv6 instantiation of a Adj SID
- **End.T** Endpoint function with specific IPv6 table lookup
- **End.DX2** Endpoint with decapsulation and Layer-2 cross-connect L2VPN use-case
- **End.DX2V** Endpoint with decapsulation and VLAN L2 table lookup EVPN Flexible cross-connect use-cases
- **End.DT2U** Endpoint with decaps and unicast MAC L2 table lookup EVPN Bridging unicast use-cases
- **End.DT2M** Endpoint with decapsulation and L2 table flooding EVPN Bridging BUM use-cases with ESI filtering
- **End.DX6** Endpoint with decapsulation and IPv6 cross-connect IPv6 L3VPN use (equivalent of a per-CE VPN label)
- **End.DX4** Endpoint with decapsulation and IPv4 cross-connect IPv4 L3VPN use (equivalent of a per-CE VPN label)
- **End.DT6** Endpoint with decapsulation and IPv6 table lookup IPv6 L3VPN use (equivalent of a per-VRF VPN label)
- **End.DT4** Endpoint with decapsulation and IPv4 table lookup IPv4 L3VPN use (equivalent of a per-VRF VPN label)
- **End.B6** Endpoint bound to an SRv6 policy SRv6 instantiation of a Binding SID
- **End.B6.Encaps** Endpoint bound to an SRv6 encapsulation Policy SRv6 instantiation of a Binding SID
- **End.BM** Endpoint bound to an SR-MPLS Policy SRv6/SR-MPLS instantiation of a Binding SID
- **End.S** Endpoint in search of a target in table T

SRv6 Interoperability Demonstration



- VPN (v4 and v6)
  & TE
  & NFV
- Cisco HW with XR and XE
- Barefoot HW with P4 code
- FD.IO
- Linux

https://tools.ietf.org/html/draft-filsfils-spring-srv6-interop-00

Linux

Implementing IPv6 Segment Routing in the Linux Kernel

David Lebrun, Olivier Bonaventure
ICTEAM, Université catholique de Louvain
Louvain-La-Neuve – Belgium
firstname.lastname@uclouvain.be

ABSTRACT

IPv6 Segment Routing is a major IPv6 extension that provides a modern version of source routing that is currently being developed within the Internet Engineering Task Force (IETF). We propose the first open-source implementation of IPv6 Segment Routing in the Linux kernel. We first describe it in details and explain how it can be used on both endhosts and routers. We then evaluate and compare its performance with plain IPv6 packet forwarding in a lab environment. Our measurements indicate that the performance penalty of inserting IPv6 Segment Routing Headers or encapsulating packets is limited to less than 15%. On the other hand, the optional HMAC security feature of IPv6 Segment Routing is costly in a pure software implementation. Since our implementation has been included in the official Linux 4.10 kernel, we expect that it will be extended by other researchers for new use cases.

1. INTRODUCTION

Segment Routing (SR) is a modern source routing architecture that is being developed within the Internet Engineering Task Force [1, 2]. While traditional IP routing uses destination-based hop-by-hop forwarding, Segment Routing can forward packets along non-shortest paths towards their destination by specifying a list of detours or waypoints called *segments*. Packets are forwarded along the shortest path from the source to the first segment, then through the shortest path from the first segment to the second segment, and so on. Segment Routing defines two main types of segments representing topological instructions. A *node* segment is used to steer packets through a specific network node. An *adjacency* segment allows to steer packets through a particular link.

The Segment Routing architecture has been instantiated in two different dataplanes: MPLS [3] and IPv6 [4]. The MPLS variant of SR is already implemented by network vendors and deployed by operators. Several researchers have proposed new traffic engineering solutions that leverage the unique characteristics of SR [5, 6, 7, 8, 9]. Other researchers have proposed improved monitoring solutions [10]

The IPv6 SR dataplane is younger and its specification became stable in early 2017 [4]. We argue that the inherent features of the IPv6 flavor of Segment Routing (SRv6) open the doors to a large, yet mostly unexplored, research area enabling the endhosts to actively participate in the selection of the paths followed by their packets.

SRv6 is realised through an IPv6 extension header, the routing header (RH). This header was already defined in the IPv6 protocol specification [11]. The IPv6 Segment Routing Header (SRH) is defined as an extension of the routing header [4]. The SRH contains a list of segments, encoded as IPv6 addresses (usually globally routable and announced by an underlying IGP such as OSPF or IS-IS). A segment can represent a topological instruction (node or link traversal) or any operator-defined instruction (*e.g.*, virtual function). An SRH can be used to steer packets through paths with given properties (*e.g.*, bandwidth or latency) and through various network functions (*e.g.*, firewalling). The list of segments present in the SRH thus specifies the network policy that applies to the packet. Each SRH contains at least a list of segments [4] and a `Segments Left` pointer that references the current active segment (a value of 0 refers to the last segment). In addition, an SRH can optionally include extended information encoded as Type-Length-Value (TLV) fields. One example is the optional HMAC TLV, used for authenticity and integrity checks.

When a router must impose an SRH on a forwarded packet, the packet is encapsulated in an outer IPv6 header containing the SRH. The original packet is left unmodified as the payload. The router is called the *SR ingress node*. The destination address of the outer IPv6 header is set to the first segment of the list and the packet is forwarded to the corresponding node following the shortest path. This node (called a *segment endpoint*) then processes the packet by updating the destination address to the next segment. The last segment of the list is the *SR egress node*. It decapsulates the inner packet and forwards it to its original destination. Figure 1 shows an illustration of path steering with Segment Routing.

Another SRH insertion technique is direct (or *inline*) insertion. In this case, the original packet is not encapsulated. Rather, the SRH is directly inserted immediately after the IPv6 header. This method yields less overhead than encapsulation, but is more susceptible to disruptions in case of network errors. For example, an ICMP generated for such a packet will reach the original source of the packet, which does not know about the inserted SRH. However, such a technique might be used within an SRv6 domain to, *e.g.*, decouple the IPv6 encapsulation and the SRH insertion.

A particular feature of SRv6 is that it can be used in the network up to the endhosts, as they already support (or are meant to support) IPv6, as opposed to MPLS which is only used in core networks. Such feature enables the endhosts to actively participate in

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Name	Description	Release
End	Endpoint function	**4.10** (February 2017), **srext**
End.X	Endpoint function with Layer-3 cross-connect	**4.10** (February 2017), **srext**
End.T	Endpoint function with specific IPv6 table lookup	**4.14** (November 2017)
End.DX2	Endpoint with decapsulation and Layer-2 cross-connect	**4.14** (November 2017), **srext**
End.DX6	Endpoint with decapsulation and IPv6 cross-connect	**4.14** (November 2017), **srext**
End.DX4	Endpoint with decapsulation and IPv4 cross-connect	**4.14** (November 2017), **srext**
End.DT6	Endpoint with decapsulation and IPv6 table lookup	**4.14** (November 2017)
End.DT4	Endpoint with decapsulation and IPv4 table lookup	In development
End.B6	Endpoint bound to an SRv6 policy	**4.14** (November 2017)
End.B6.Encaps	Endpoint bound to an SRv6 encapsulation Policy	**4.14** (November 2017)
End.BM	Endpoint bound to an SR-MPLS Policy	In development
End.S	Endpoint in search of a target in table T	In development
End.AD	Endpoint to SR-unaware APP via dynamic proxy	**srext**
End.AM	Endpoint to SR-unaware APP via masquerading	**srext**

Code Review / wireshark.git / commit

summary | shortlog | log | commit | commitdiff | review | tree
(parent: cee3e82) | patch

IPv6: Add support for Segment Routing (Type 4) Extension Header 93/15593/10

the-tcpdump-group / **tcpdump**

Code | Issues 48 | Pull requests 26 | Insights

Add support for IPv6 routing header type 4

```
IPv6 routing header type 4 is known as Segment Routing Header (SRH).
It is used by Segment Routing (SR) to steer a packet through a
controlled set of instructions, called segments. It is described in
https://tools.ietf.org/html/draft-ietf-6man-segment-routing-header-07

An extension is added to allow tcpdump to print SRH information.
```

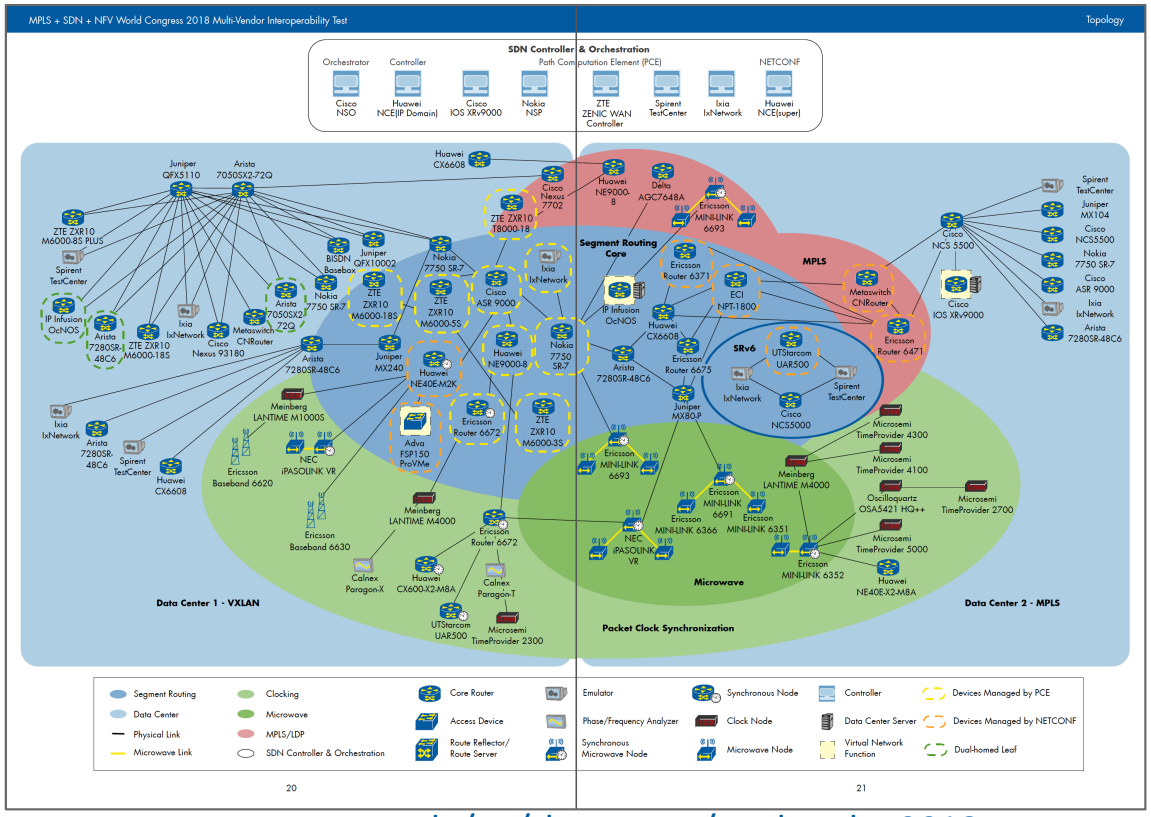
Name	Description	Release
T.Insert	Transit behavior with insertion of an SRv6 Policy	**4.10** (February 2017)
T.Encaps	Transit behavior with encapsulation in an SRv6 policy	**4.10** (February 2017)
T.Encaps.L2	T.Encaps behavior of the received L2 frame	**4.14** (November 2017)

FD.IO

Name	Description	Release
End	Endpoint function	**17.04** (April 2017)
End.X	Endpoint function with Layer-3 cross-connect	**17.04** (April 2017)
End.T	Endpoint function with specific IPv6 table lookup	In development
End.DX2	Endpoint with decapsulation and Layer-2 cross-connect	**17.04** (April 2017)
End.DX6	Endpoint with decapsulation and IPv6 cross-connect	**17.04** (April 2017)
End.DX4	Endpoint with decapsulation and IPv4 cross-connect	**17.04** (April 2017)
End.DT6	Endpoint with decapsulation and IPv6 table lookup	**17.04** (April 2017)
End.DT4	Endpoint with decapsulation and IPv4 table lookup	**17.04** (April 2017)
End.B6	Endpoint bound to an SRv6 policy	**17.04** (April 2017)
End.B6.Encaps	Endpoint bound to an SRv6 encapsulation Policy	**17.04** (April 2017)
End.BM	Endpoint bound to an SR-MPLS Policy	In development
End.S	Endpoint in search of a target in table T	In development
End.AS	Endpoint to SR-unaware APP via static proxy	In development
End.AM	Endpoint to SR-unaware APP via masquerading	In development

Name	Description	Release
T.Insert	Transit behavior with insertion of an SRv6 Policy	**17.04** (April 2017)
T.Encaps	Transit behavior with encapsulation in an SRv6 policy	**17.04** (April 2017)
T.Encaps.L2	T.Encaps behavior of the received L2 frame	**17.04** (April 2017)

EANTC Interop Showcase



www.eantc.de/en/showcases/ mpls_sdn_2018

Over 20 companies, including:

- Cisco
- Juniper
- Arista
- Huawei
- Ericsson
- Nokia
- ZTE
- ECI
- NEC

Technologies covered

- SR/SRv6
- EVPN
- PCE
- NC/YANG
- Microwave (IP/MPLS)
- Synchronization

EANTC SRv6 Interoperability tests

- In this scenario we tested both the END and END.X functions.
- During our test, we verified the expected END and END.X data plane forwarding behavior and IPv6 SR Header (SRH) handling by Cisco and UTStarcom devices for SRv6 encapsulated traffic generated by Spirent TestCenter (STC) and Ixia IxNetwork.
- In our first scenario, UTStarcom's UAR500 performed the END function and Cisco's NCS 5500 performed the END.X function. In the second scenario, the roles were inverted accomplishing the same results.

http://www.eantc.de/en/showcases/mpls_sdn_2018

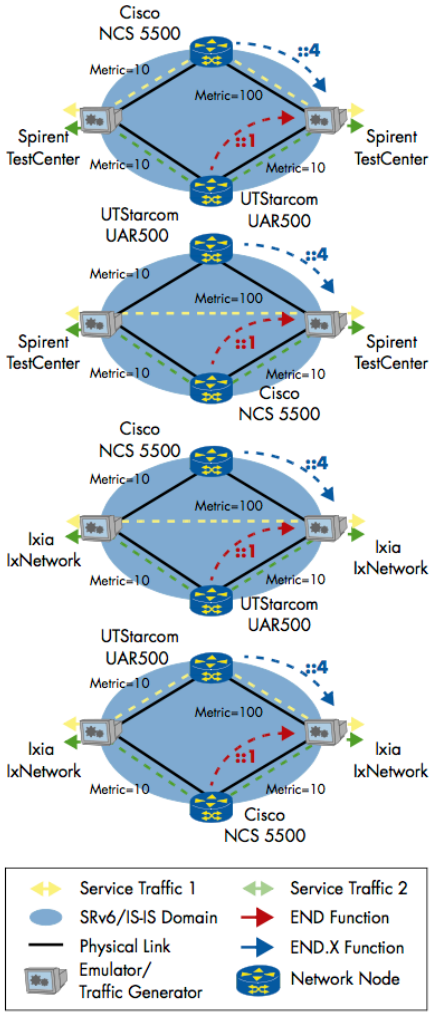
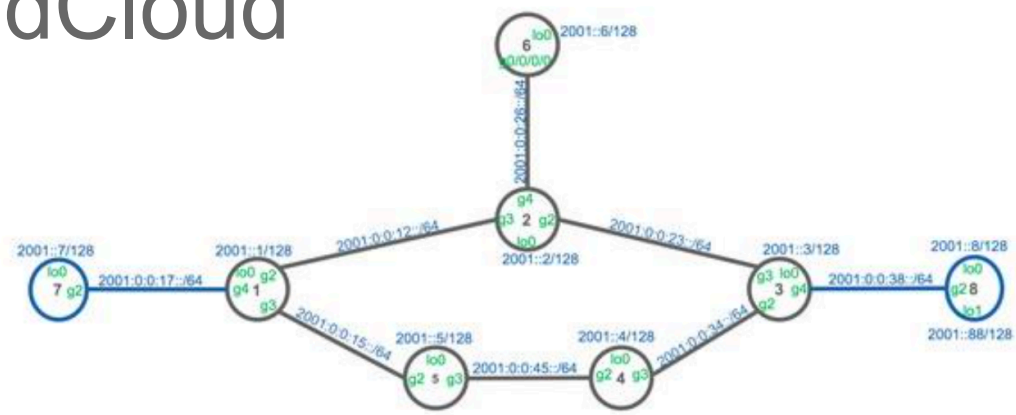
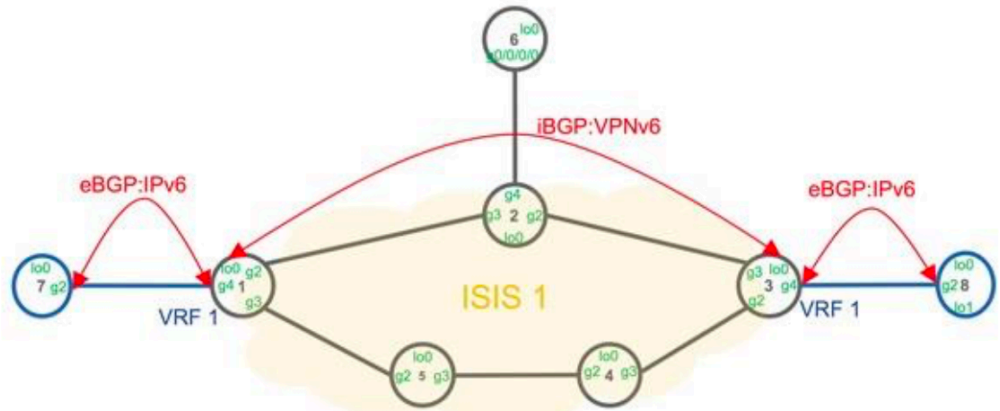


Figure 1: SRv6 IPv6 Routing

SRv6 Lab at Cisco dCloud

- In this lab you will explore how SRv6 works.
- In the first part, you will see how BGP extensions allows you to use SRv6 for L3 VPN across the IPv6 network
- In the second part, you will explore the possibilities of the On Demand Nexthop configuration to provide a latency-aware path for L3 VPN traffic.

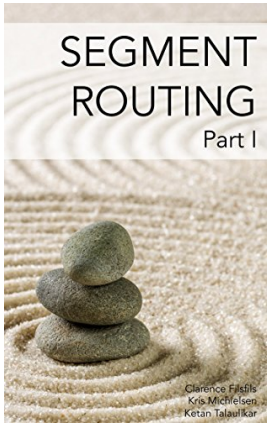




Stay up to date on Segment Routing

segment-routing.net

amzn.com/B01I58LSUO



ask-segment-routing@cisco.com

linkedin.com/groups/8266623

twitter.com/SegmentRouting

facebook.com/SegmentRouting/

SEGMENT ROUTING

NEWS DEMOS TUTORIALS CONF IETF OPEN SW ACAD

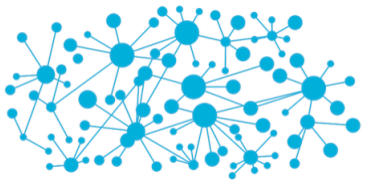
segment-routing.net

Segment Routing

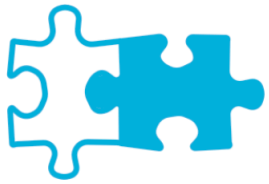
A source-routing architecture that seeks the right balance between distributed intelligence and centralized optimization.
The application steers its packets through an ordered list of instructions and realizes end-to-end policy without creating any per-flow state in the network.

Get Started! | SRv6 video | SR Customer Quotes

Simple



Scalable



Seamless deployment

IETF Docs SRv6

- RFC 8354: Use Cases for IPv6 Source Packet Routing in Networking
- IPv6 Segment Routing Header (SRH) draft-ietf-6man-segment-routing-header
- IS-IS Extensions to Support Routing over IPv6 Dataplane draft-bashandy-isis-srv6-extensions
- OSPFv3 Extensions for SRv6 draft-li-ospf-ospfv3-srv6-extensions
- BGP Link State extensions for IPv6 Segment Routing(SRv6) draft-dawra-idr-bgpls-srv6-ext
- Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6) draft-ali-spring-srv6-oam
- SRv6 Network Programming draft-filsfils-spring-srv6-network-programming
- SRv6 interoperability report, draft-filsfils-spring-srv6-interop-00

CISCO