Segment Routing v6

Technical Overview

Jakub Horn

Technical Marketing Engineer

CCIE#7543

Content

- **SRH**

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

- **Network Programmability**

  draft-filsfils-spring-srv6-network-programming-05

- **IGP**

  draft-bashandy-isis-srv6-extensions-03

- **BGP**

  draft-dawra-idr-srv6-vpn-04

  - **DEMO**

    service chaining

-

Segment Routing v6 Header

[Docs] [txt|pdf|xml|html] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

Versions: (draft-previdi-6man-segment-routing-header)
00 01 02 03 04 05 06 07 08 09 10 11
12 13 14

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 30, 2018

C. Filsfils, Ed.
Cisco Systems, Inc.
S. Previdi
Individual
J. Leddy
Comcast
S. Matsushima
Softbank
D. Voyer, Ed.
Bell Canada
June 28, 2018

IPv6 Segment Routing Header (SRH)
draft-ietf-6man-segment-routing-header-14

Abstract

Segment Routing can be applied to the IPv6 data plane using a new type of Routing Extension Header. This document describes the Segment Routing Extension Header and how it is used by Segment Routing capable nodes.

Requirements Language

key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

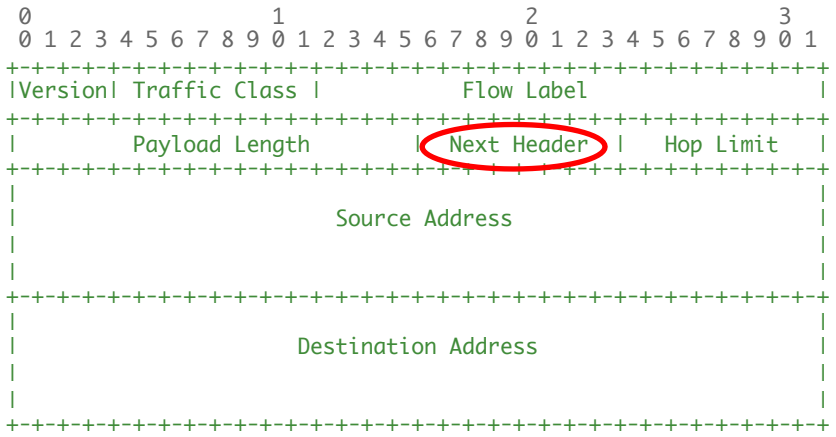
Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 30, 2018.

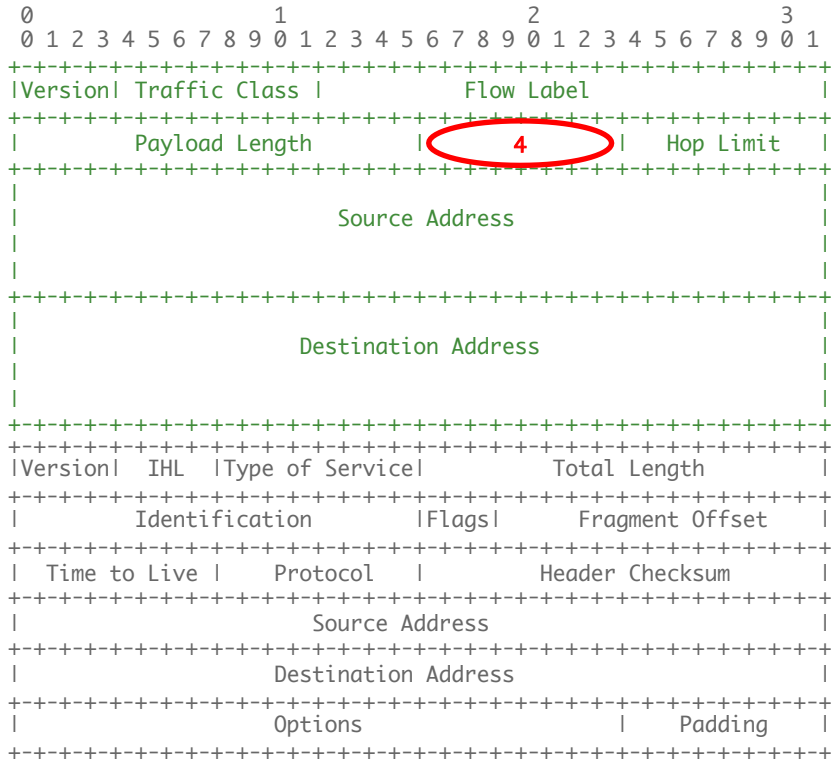
IPv6 SR Header

- IPv6 header
  - Next header field: *Indicates what comes next*



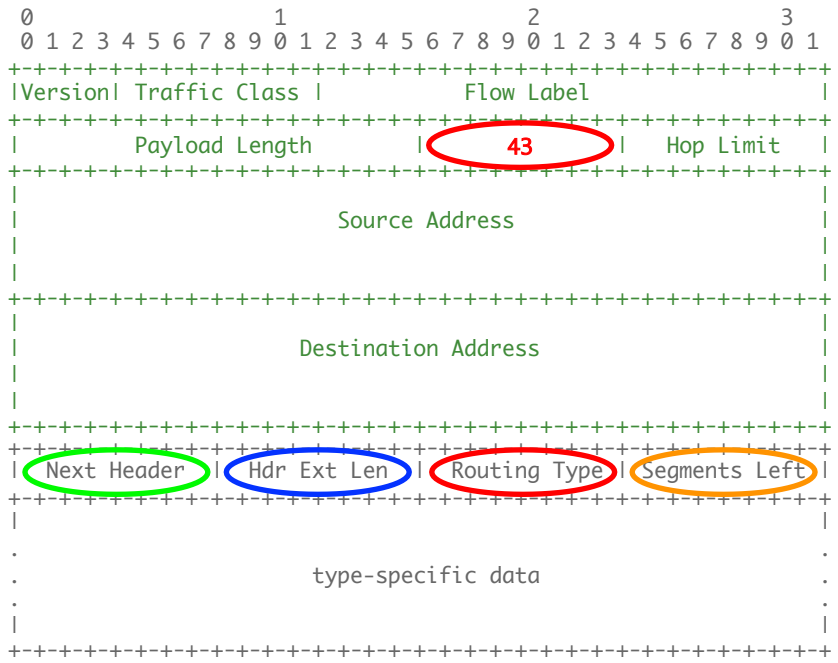
IPv6 SR Header

- IPv6 header
  - Next header field: 4 → IPv4



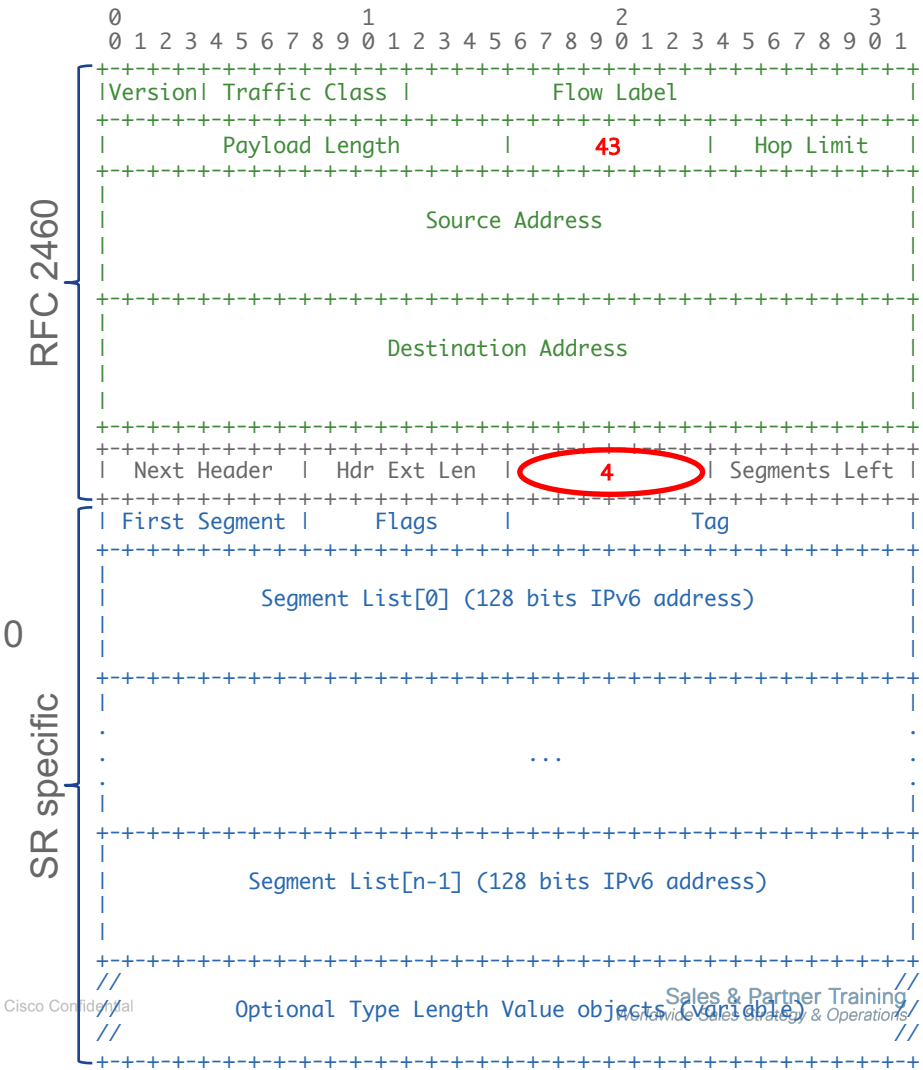
IPv6 SR Header

- IPv6 header
  - Next header field: **43** → **Routing**
- IPv6 Routing extension header
  - Generic header format defined in RFC 2460
    - 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)



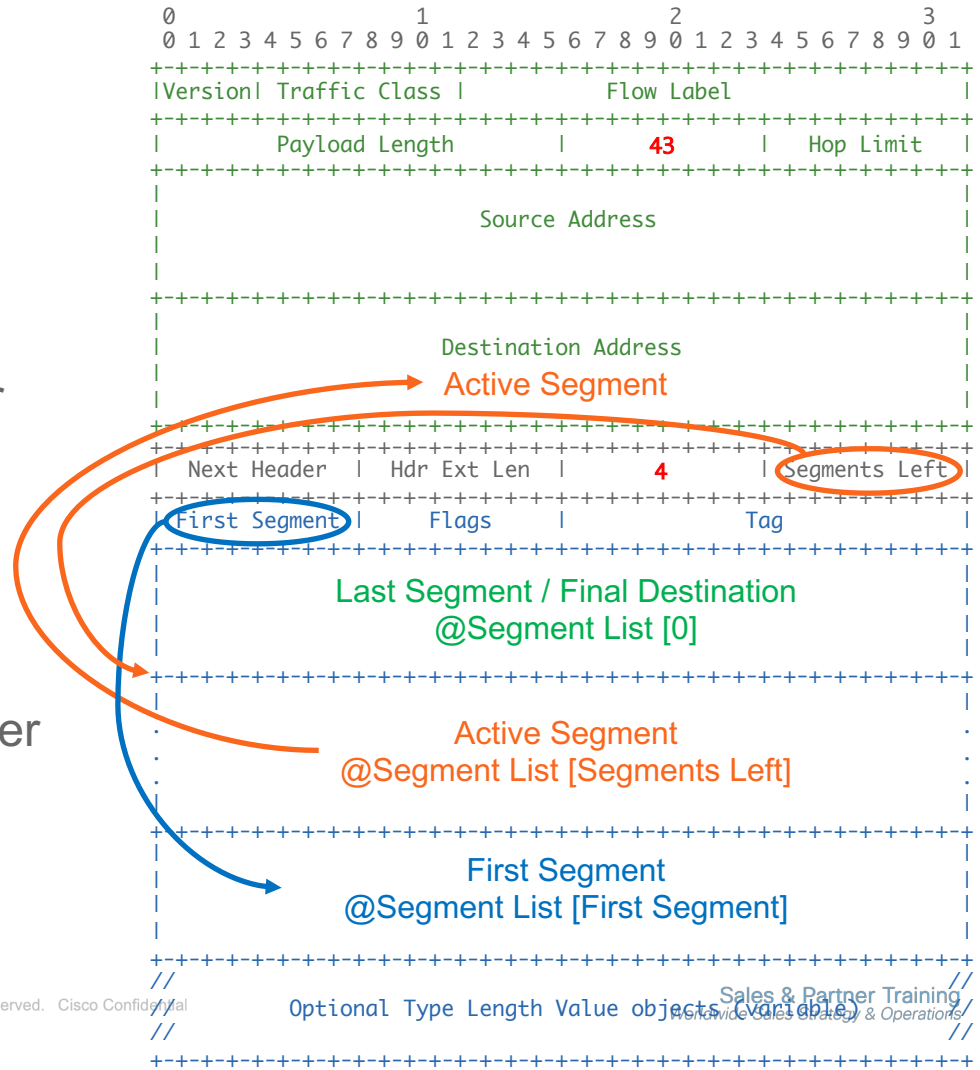
IPv6 SR Header

- IPv6 header
  - Next header field: **43** → **Routing**
- IPv6 Routing extension header
  - Generic header format defined in RFC 2460
    - 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** (tentative)



IPv6 SR Header

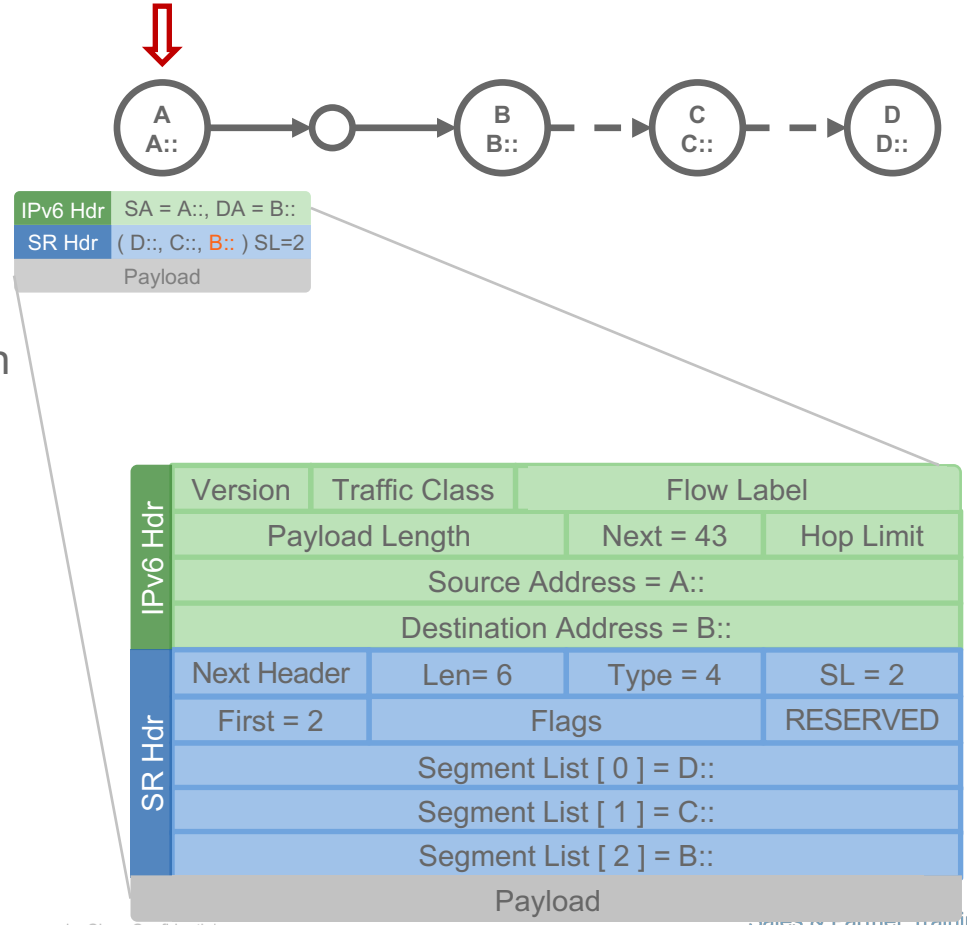
- Each segment is an IPv6 address
- Segments are encoded in reverse order
  - Last segment index is 0
  - First segment index is **First Segment**
  - Active segment index is **Segments Left**
- Active Segment is copied in the Destination Address field of the IP header
- Additional data can be stored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



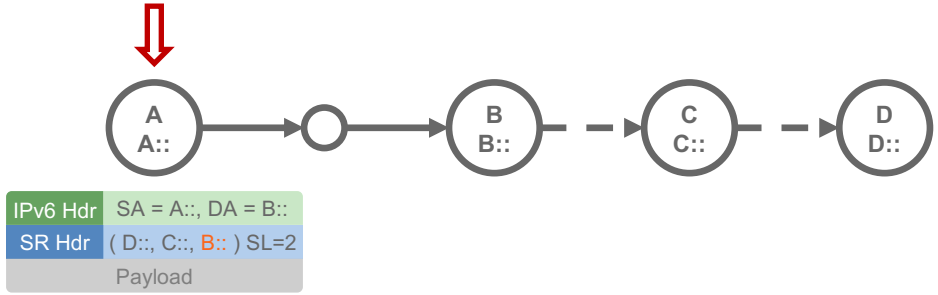
SR Header Processing

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



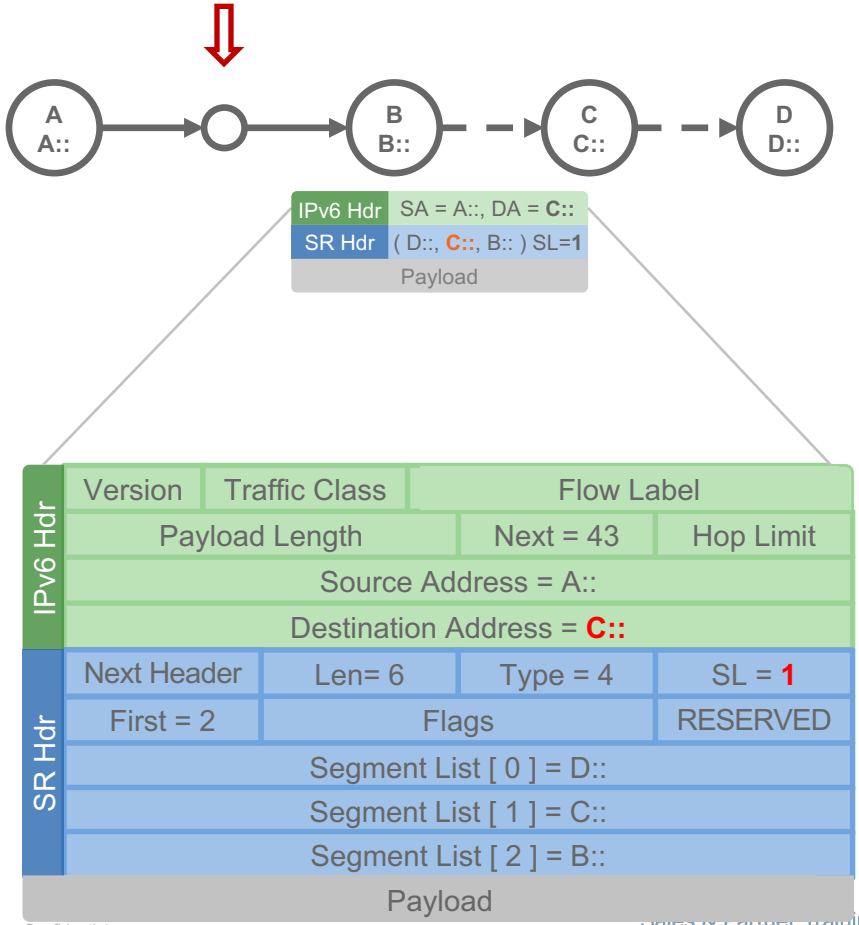
Non-SR Transit Node



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

SR Segment Endpoints

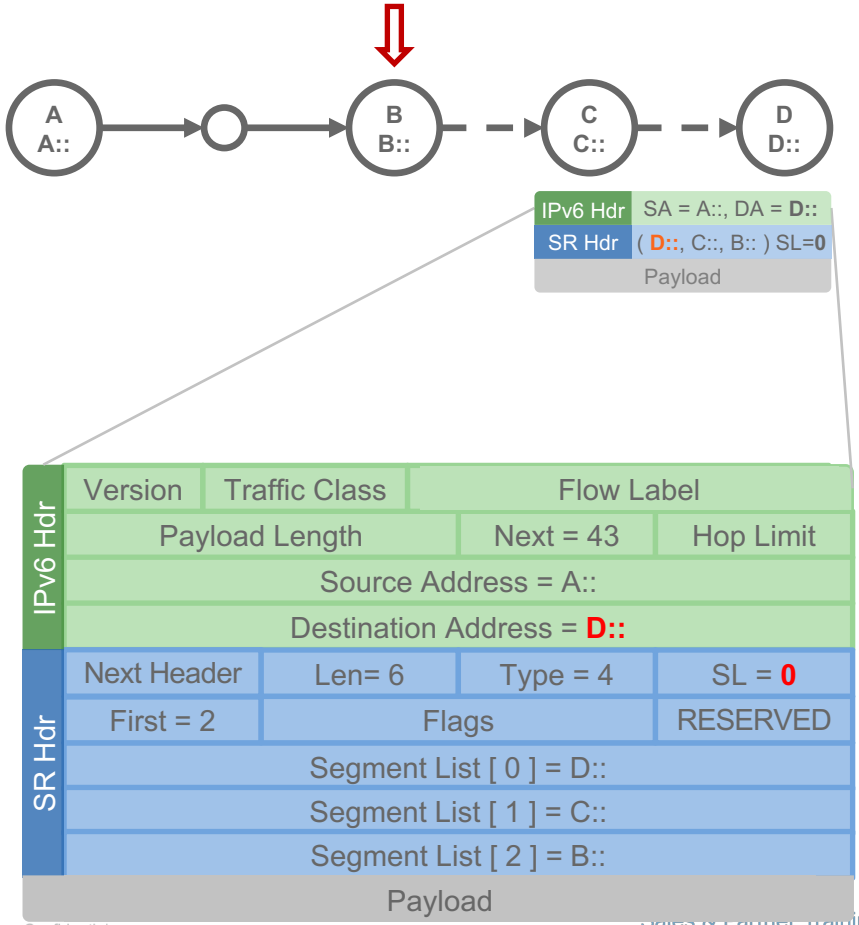
- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA



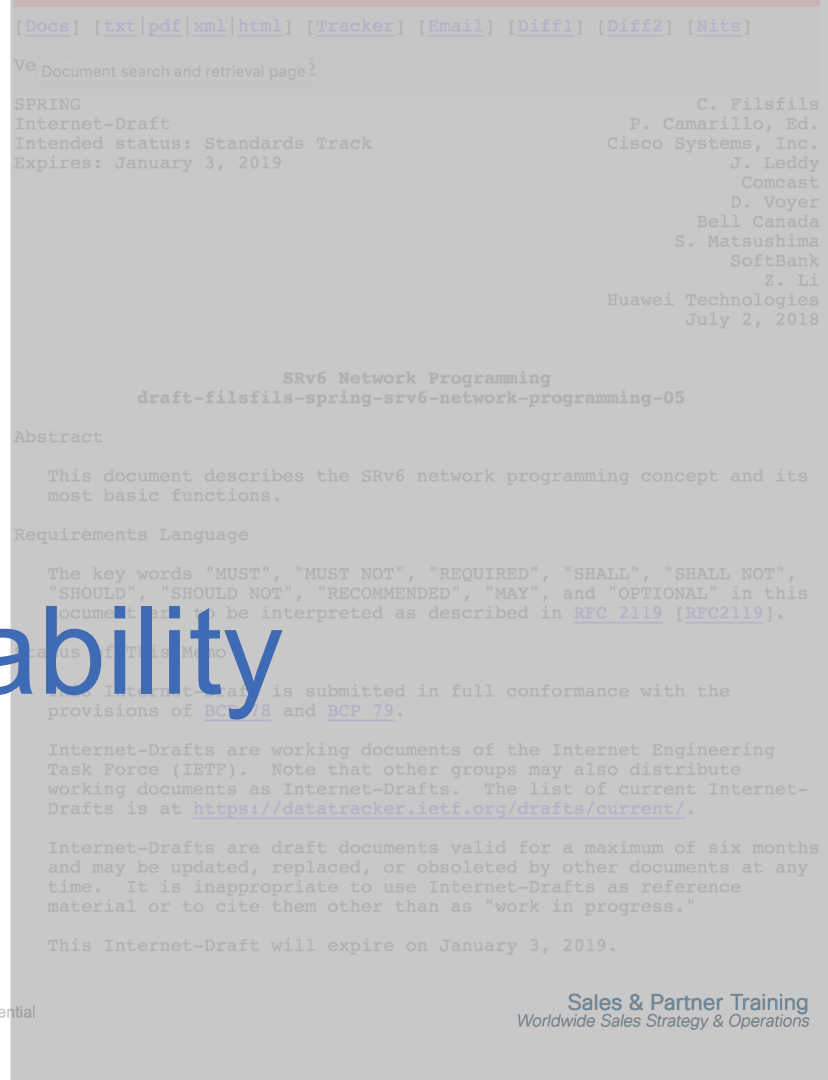
SR Segment Endpoints

- 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

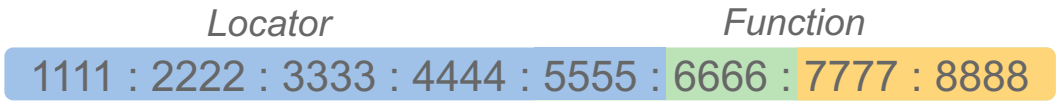
Standard IPv6 processing
The final destination does not have to be SR-capable.



Network Programability



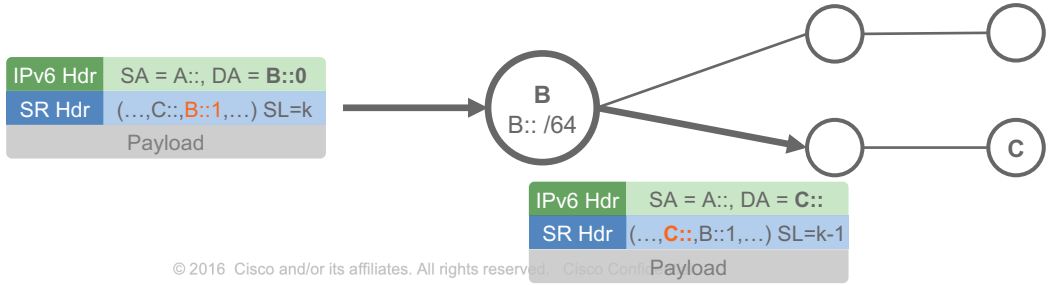
Segment format



- 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

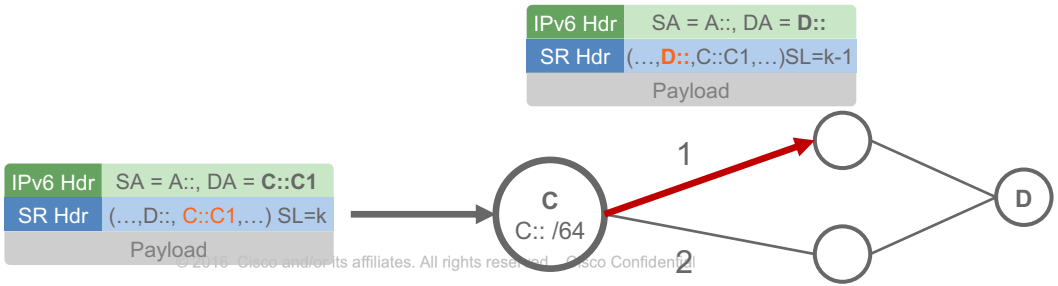
END – Default endpoint

- *Default endpoint* behavior (node segment)
  - Decrement Segments Left, update DA
  - Forward according to new DA
- Node B advertises prefix B::/64 (B::/64 is the SID **locator**)
  - Packets are forwarded to B along the default routes (shortest path)
- On B, the *default endpoint* behavior is associated with ID 0 (0 is the **function**)
- The SID corresponding to the *default endpoint* behavior on node B is B::1



END.X – Endpoint then Xconnect

- *Endpoint xconnect* behavior (adjacency segment)
  - Decrement Segments Left, update DA
  - **Forward on the interface associated with the Xconnect segment**
- Node C advertises prefix C::/64
  - Packets are forwarded to C along the default routes (shortest path)
- On C, the *endpoint xconnect* behavior for link (C, E) is associated with ID CE
- The SID corresponding to *endpoint xconnect-(C,E)* behavior on node C is **C::CE**

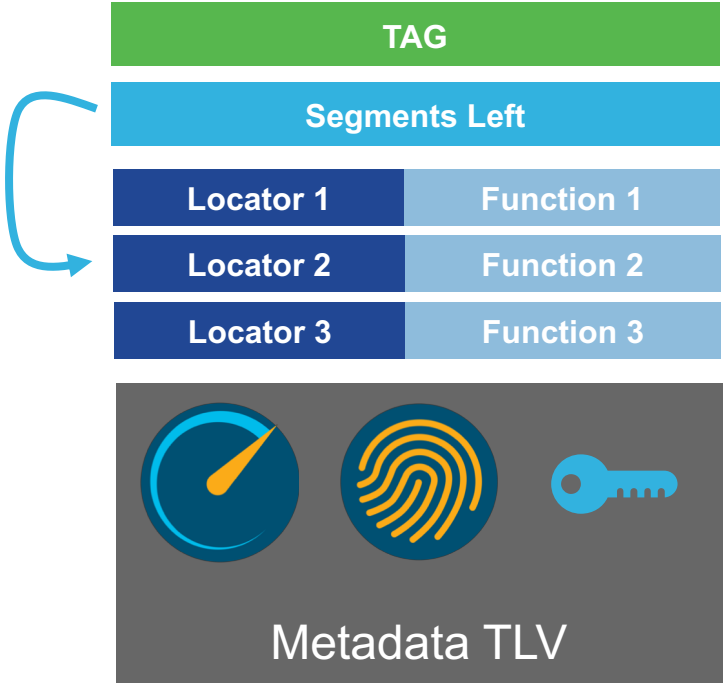


Functoins Defined in Net Programming

- **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.DT46** Endpoint with decapsulation and IP table lookup IP 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

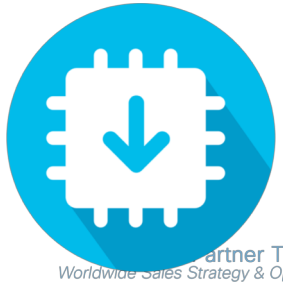
- **T.Insert** Transit behavior with insertion of an SRv6 policy
- **T.Insert.Red** Transit behavior with reduced insert of an SRv6 policy
- **T.Encaps** Transit behavior with encapsulation in an SRv6 policy
- **T.Encaps.Red** Transit behavior with reduced encaps in an SRv6 policy
- **T.Encaps.L2** T.Encaps behavior of the received L2 frame
- **T.Encaps.L2.Red** Transit with reduce encaps of received L2 frame

SRv6 for anything



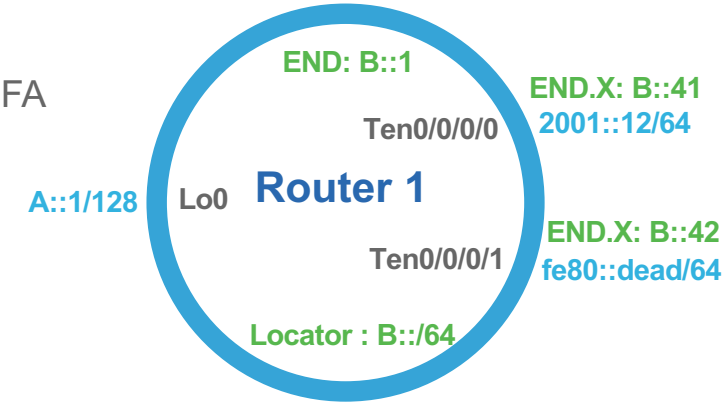
Optimized for HW processing
e.g. Underlay & Tenant use-cases

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

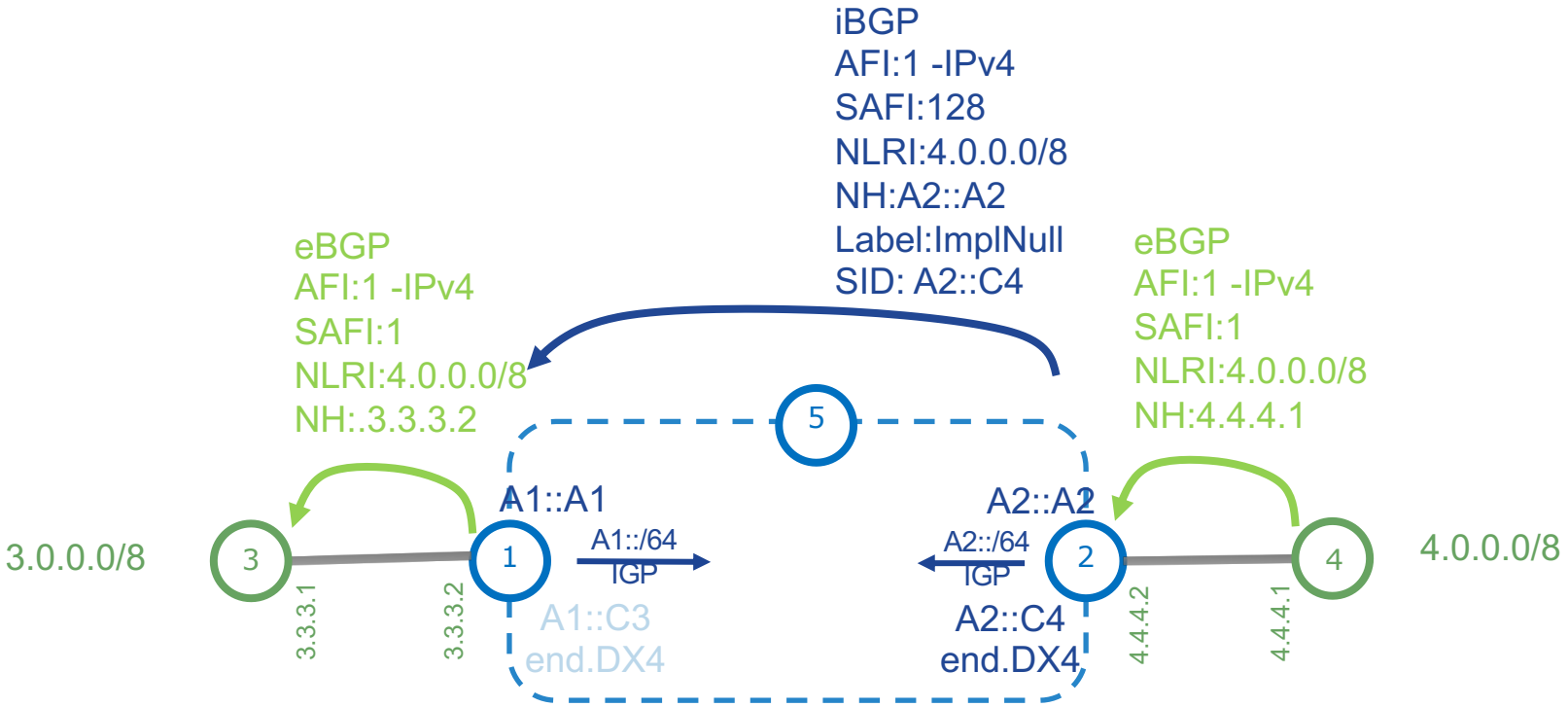


ISIS easily extensible

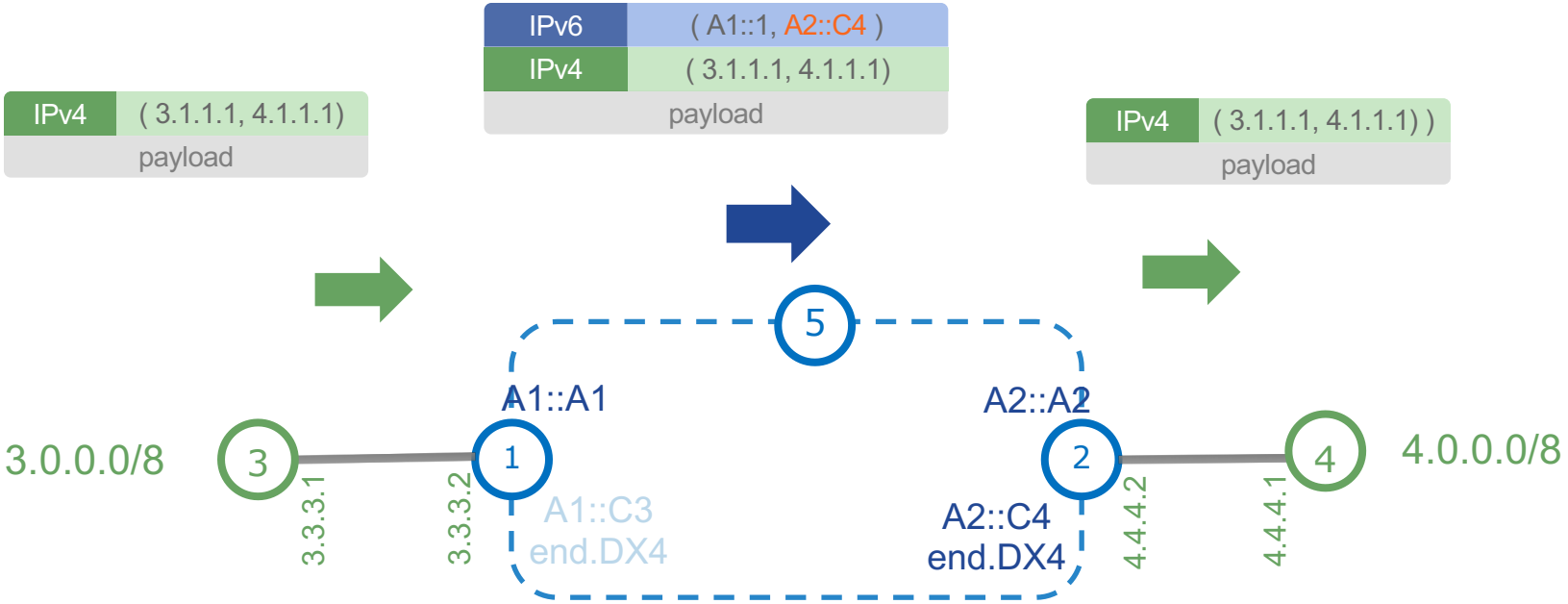
- Uses TLVs
- For Srv6:
  - Locator – for Reachability (twice for backward compatibility)
  - END function - TI-LFA and TE
  - END.X function for each interface in routing protocol TI-LFA and TE
  - Capabilities:
  - Max SID depth for different functions
- OSPF will follow



VPNv4 – Basic SRv6 VPN –Control Plane



VPNv4 – Basic SRv6 VPN -Data Plane



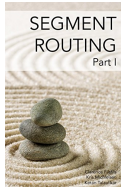
Stay Up-To-Date

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

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

https://twitter.com/SegmentRouting

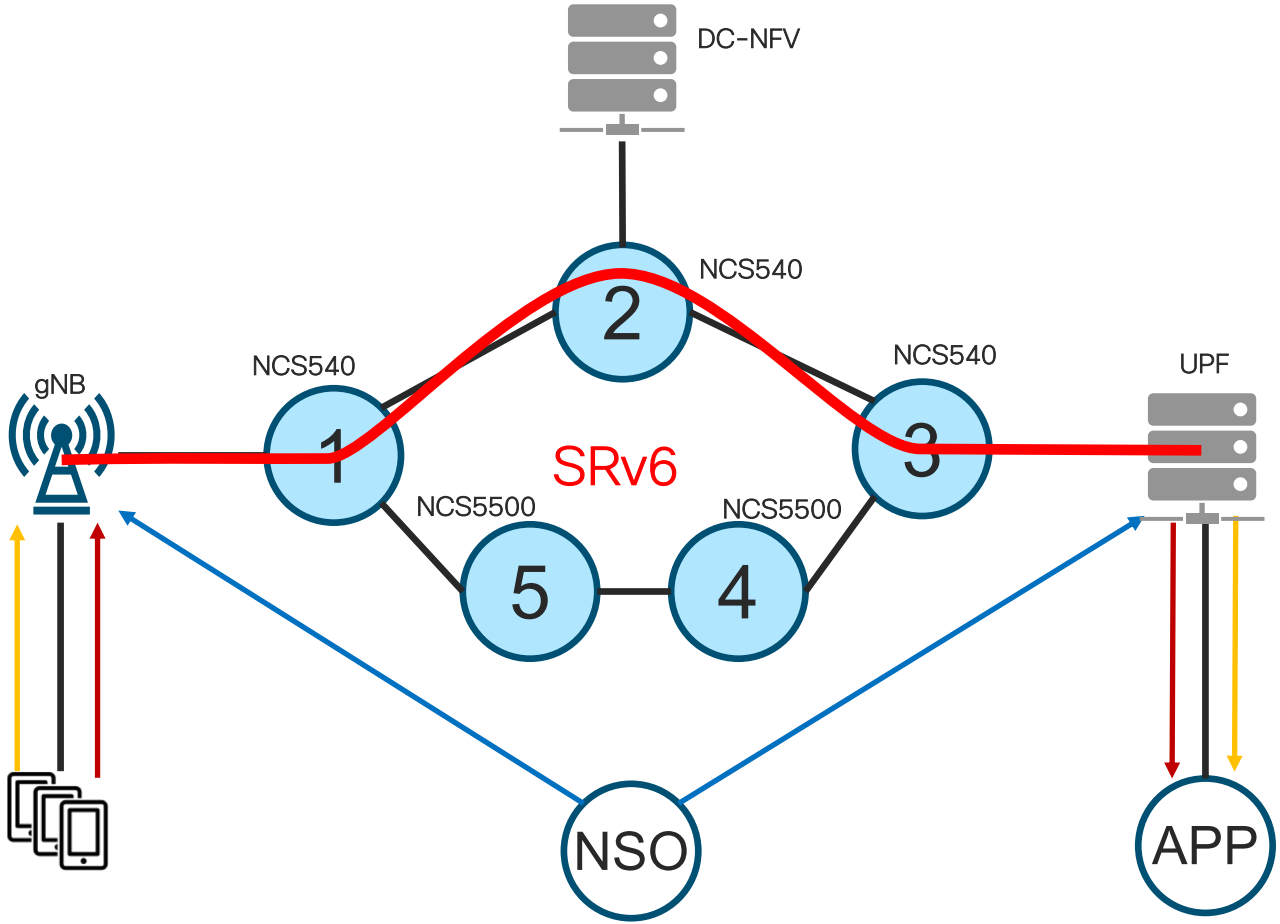
https://www.facebook.com/SegmentRouting/



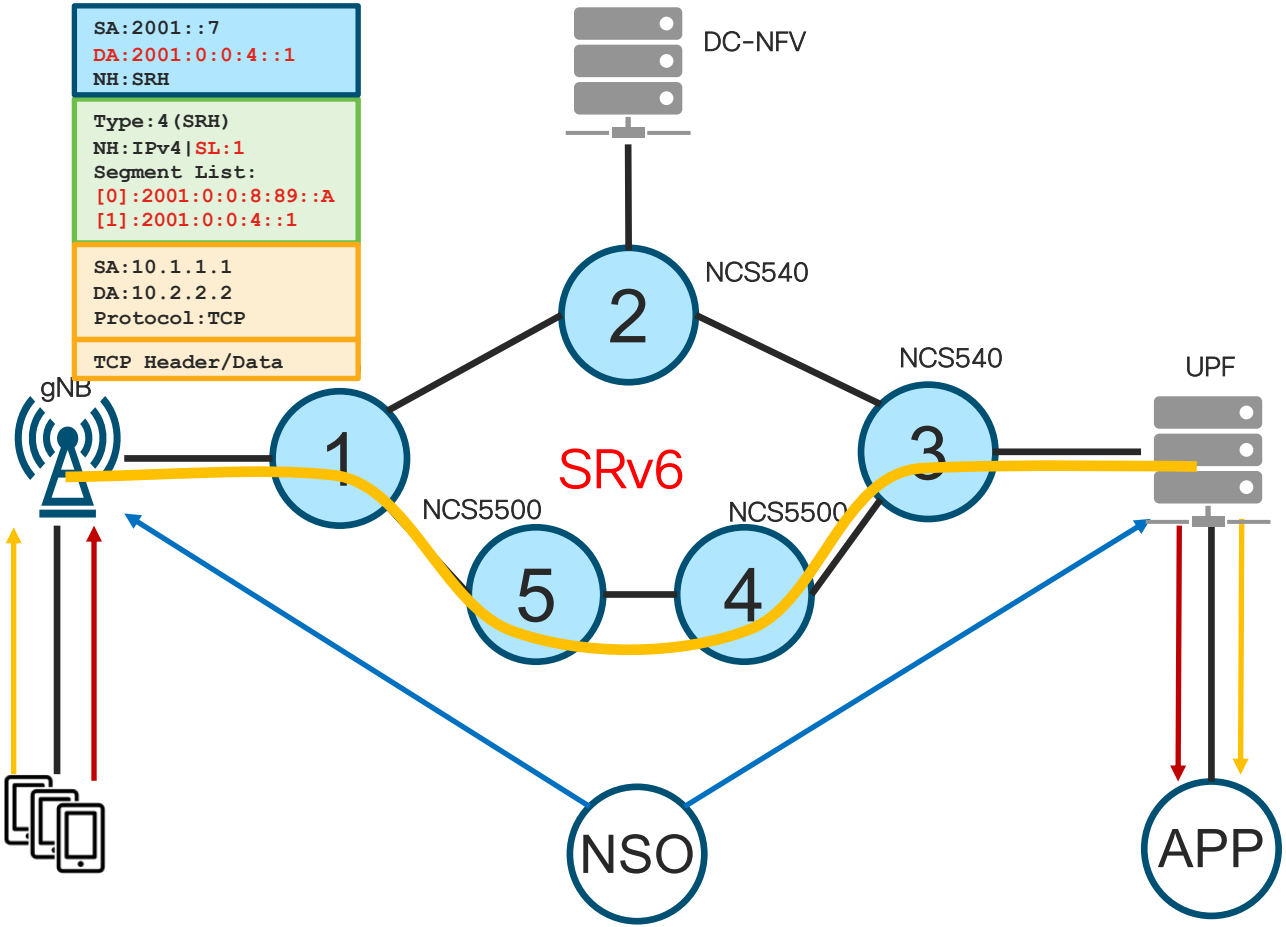
Segment Routing, Part I - Textbook

Demonstration

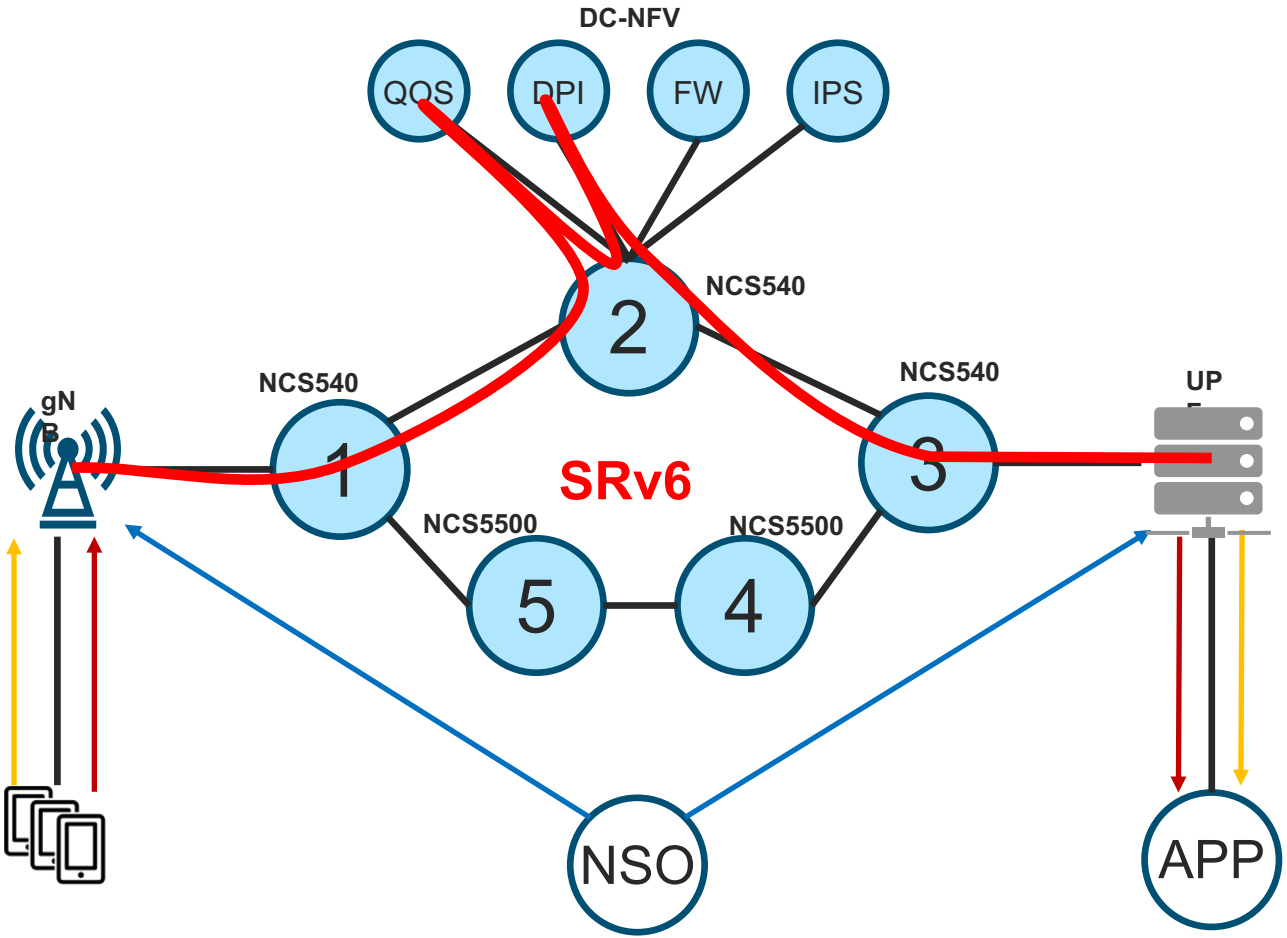
eMBB Service



URLLC Service



Service Chaining



Thank you

TOMORROW starts here.