

# IETF105報告会

## SRv6+について

---

2019年8月31日

NTTコミュニケーションズ株式会社  
上手 祐治

# SRv6+について

## ■ 動機： 現在のSRv6仕様が抱える問題(パケット長)

### ■ Segment(SID)の数が増えると...

- オーバーヘッドが増え、ネットワークの帯域を浪費する  
(SID 1段につき16 byte増加)

-ショートパケットになればオーバーヘッドが無視できない(設備コスト増加)

- 従来のASICを搭載する装置では、十分な転送性能を発揮できない

### ■ ルーティングを織り交ぜSID段数を少なく使う限りは大きな問題にならないが

、  
大規模展開<sup>0</sup>や高度なSteering<sup>1</sup>処理<sup>2</sup>を行う<sup>3</sup>上<sup>4</sup>では制約となる

```
+-----+-----+-----+-----+
| 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)
//
+-----+-----+-----+-----+
```

従来のSRv6 Segment Routing Header  
(Segment1個あたり128bits IPv6 addr)

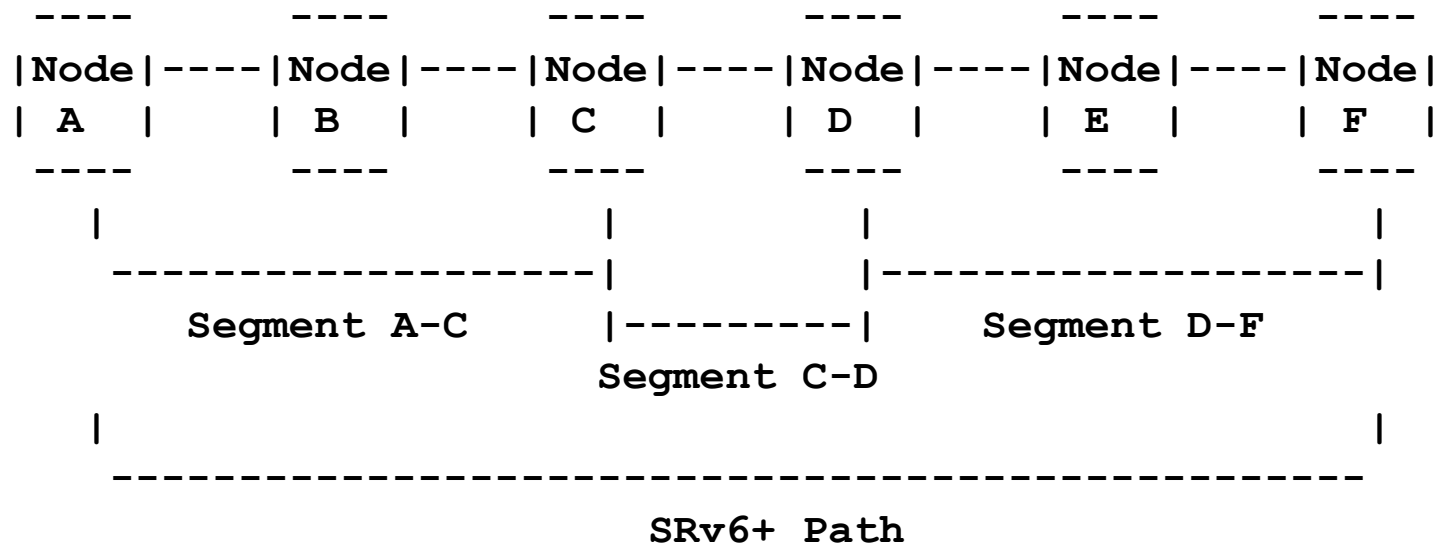
# SRv6+

SRv6+ ... SRv6を活用し、パケット長の懸念を無くした技術の提案

- 全体アーキテクチャ
  - draft-bonica-spring-srv6-plus-05 \*
- Compression Header
  - draft-bonica-6man-comp-rtg-hdr-06 \*
- Per-Segment Service Instruction
  - draft-bonica-6man-seg-end-opt-04
- Per-Path Service Instruction
  - draft-bonica-6man-vpn-dest-opt-06
- IS-IS Extension
  - draft-bonica-lsr-crh-isis-extensions-00
- BGP-based VPN
  - draft-ssangli-idr-bgp-vpn-srv6-plus-02

\*) 8/30時点の最新draftでは、IETF105発表時から改版あり

# PathとSegment

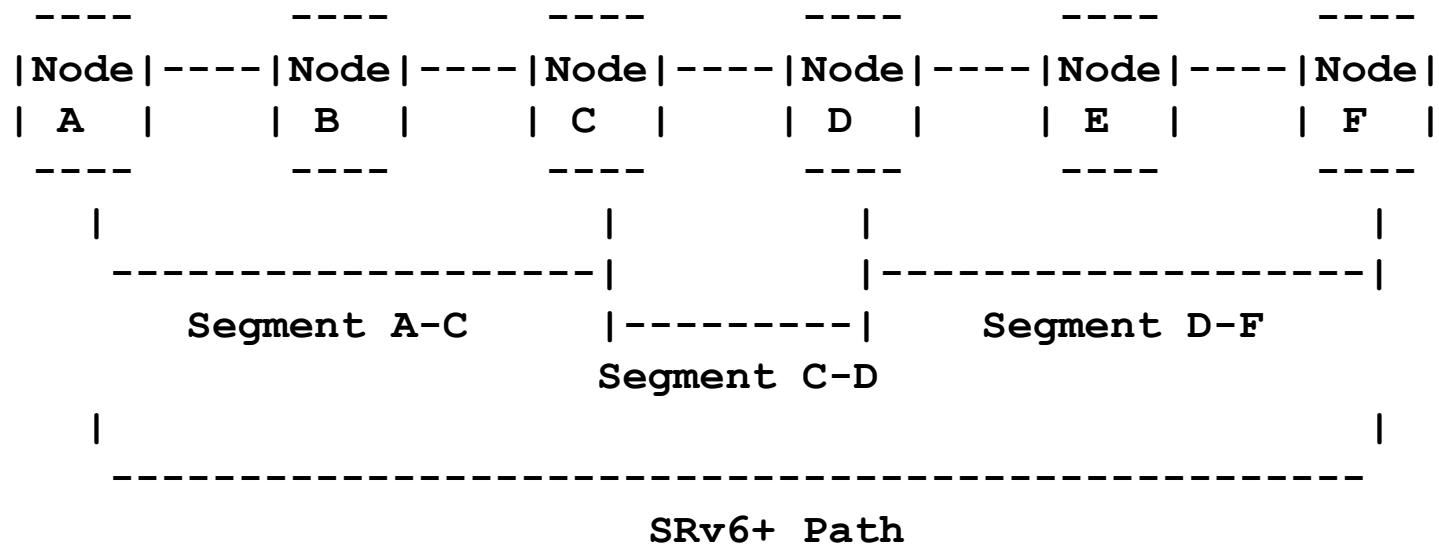


# PathとSegment

## ■ Path

•PathのIngress

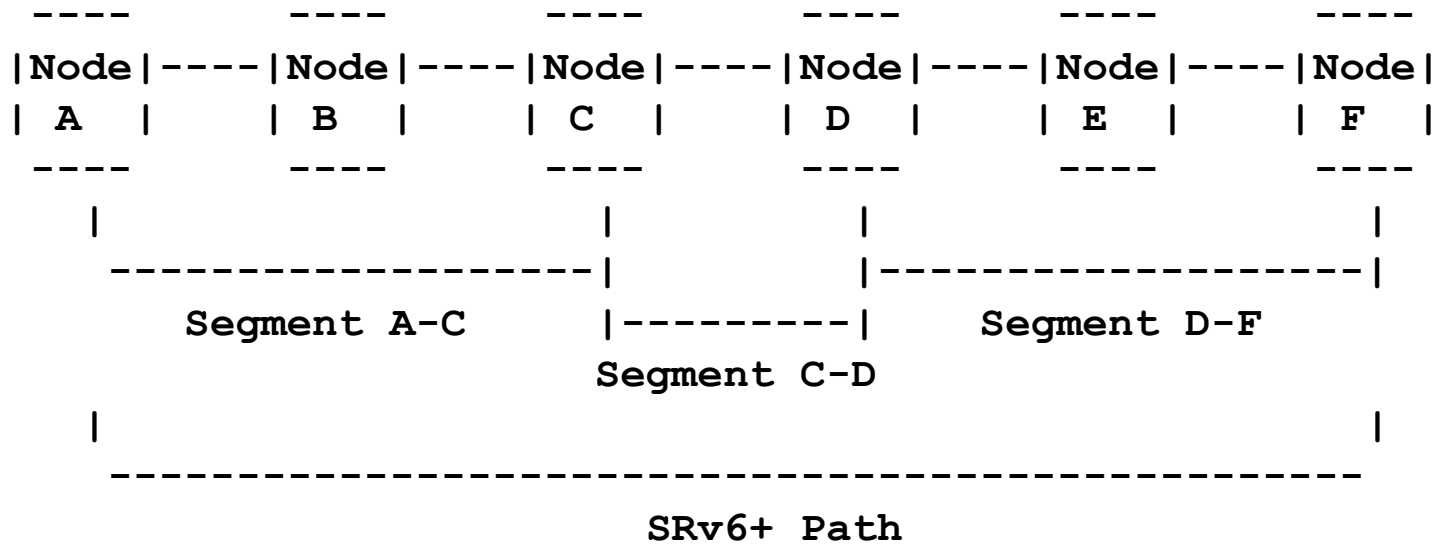
•PathのEgress



# PathとSegment

## ■ Segment

- ・A-CのIngress
- ・Transit
- ・A-CのEgress
- ・C-DのIngress
- ・C-DのEgress
- ・D-FのIngress
- ・Transit
- ・D-FのEgress



# Topological/Service Instruction

---

- 従来のSRv6でのSIDが担っていた内容を、2つの概念に分離する
  - Topological Instruction
    - 通過トポロジを指定する
    - パケットとしてはRouting Headerで表現
      - Compression Header(CRH)
  - Service Instruction
    - セグメントやパスの終わりでの追加処理を指示する
    - パケットとしてはDestination Option Headerで表現

# SRv6+ Segment

---

## ■ Segment は2種類

- Adjacency Segments (\*1)

\*1 ... Strictly Routed Segmentsから改称 (draft -05→06)

- Node Segments (\*2)

\*2 ... Loosely Routed Segments から改称(draft -05→06)

## ■ IS-ISでSegment情報(SID)を広告

- Segmentのingress nodeが、adjacency segmentをフラッド
  - 始点側がつくる
- Segmentのegress nodeが、node segmentをフラッド
  - 終点側がつくる

## ■ CRHにエンコード

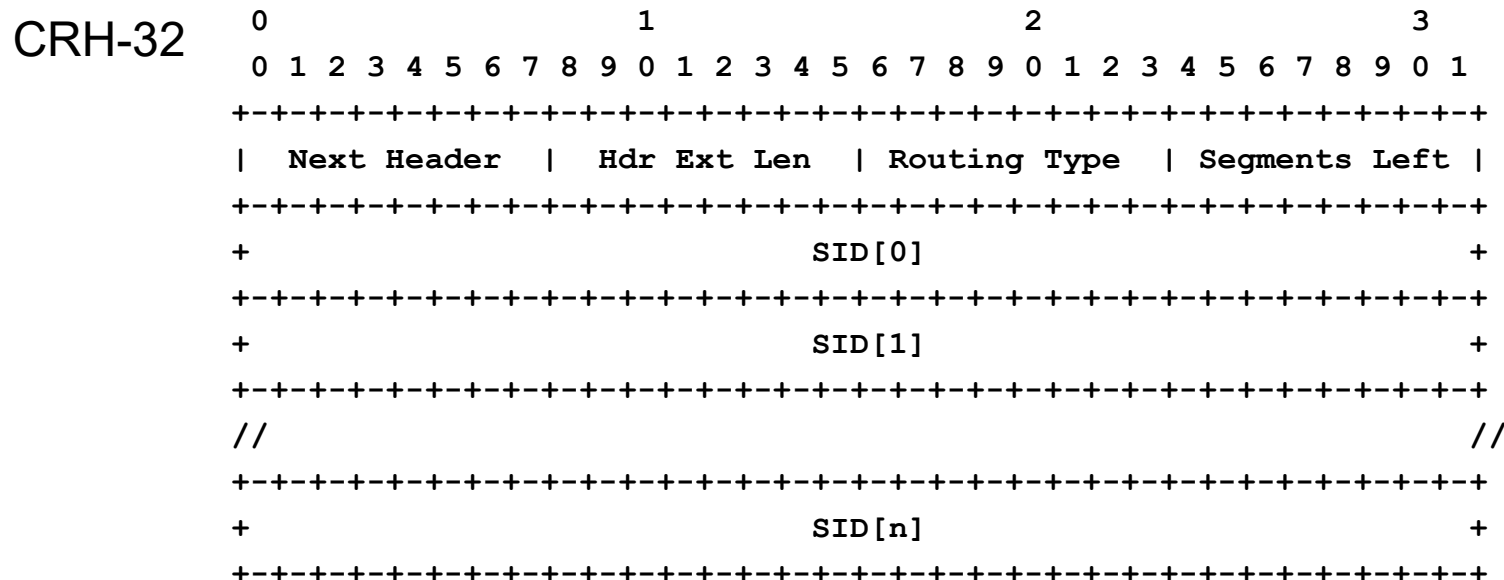
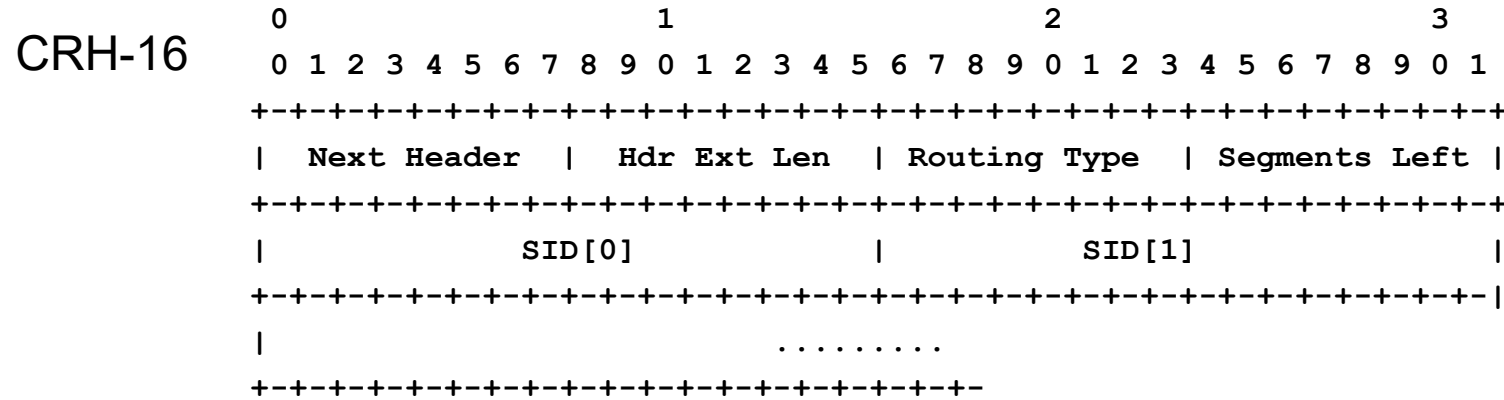
- Compression Header
  - 従来のSRHではSIDあたり128bitだったのが、CRHでは16bit or 32bitになる



# SRv6+ Segment -- CRH Format

## ■ SID長が16bit (CRH-16)の場合と、32bit(CRH-32)の場合に対応

draft-bonica-6man-comp-rtg-hdr-06



---

# SRv6+ Service Instruction

---

## ■ Per-Segment Service Instruction

- セグメントのEgressでサービス指示
- Routing Header (CRH) の\*前にある\*Destination Options headerにエンコード

## ■ Per-Path Service Instruction

- パスのEgressでサービス指示
  - 上位レイヤパケットの\*前にある\*Destination Options headerにエンコード
-

# Per-Segment Service Instruction

- セグメントの終わりで、追加処理を指示できる

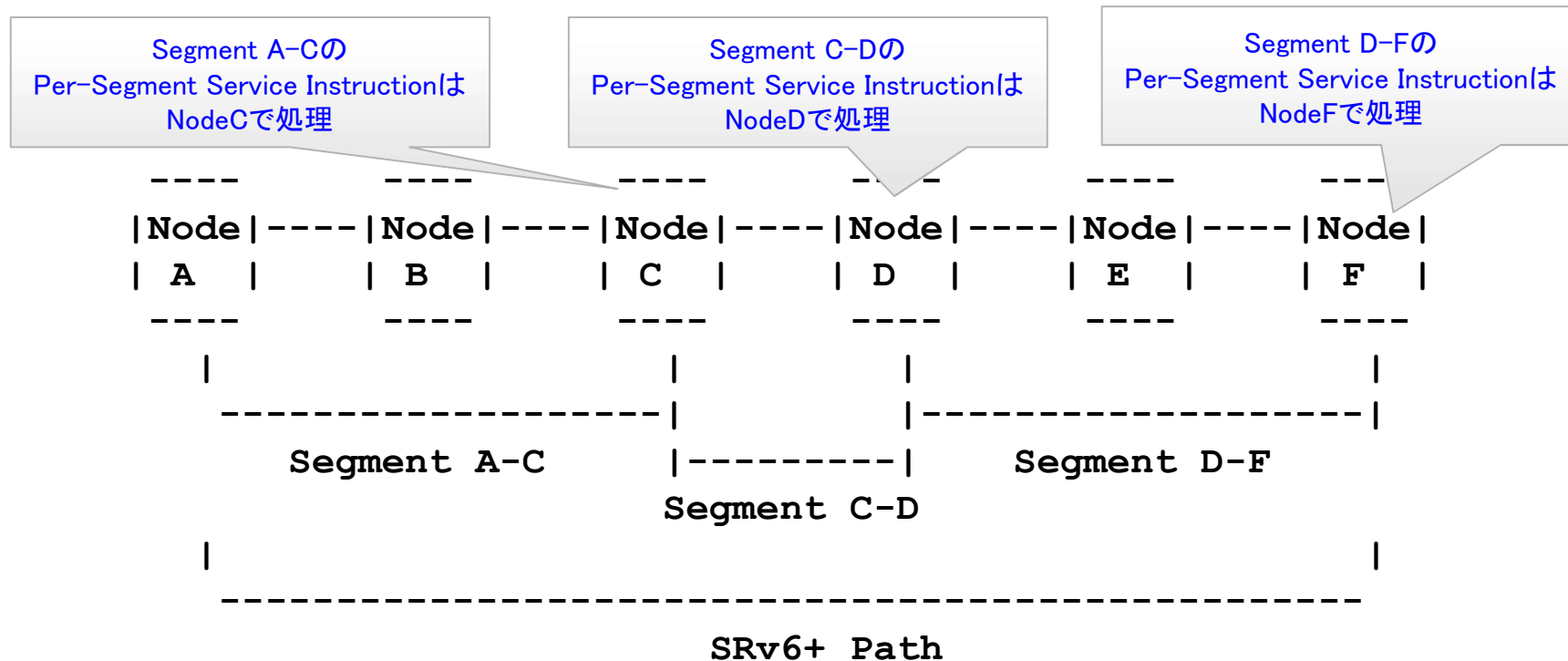


Figure 1: Paths, Segments And Instructions

例えば、Segmentが終わるごとに、Firewall policyや Sampling Policyを適用するなどの用途に使う

# Per-Path Service Instruction

## ■ Per-Path Service Instruction

- パスの終わりで、追加処理を指示できる

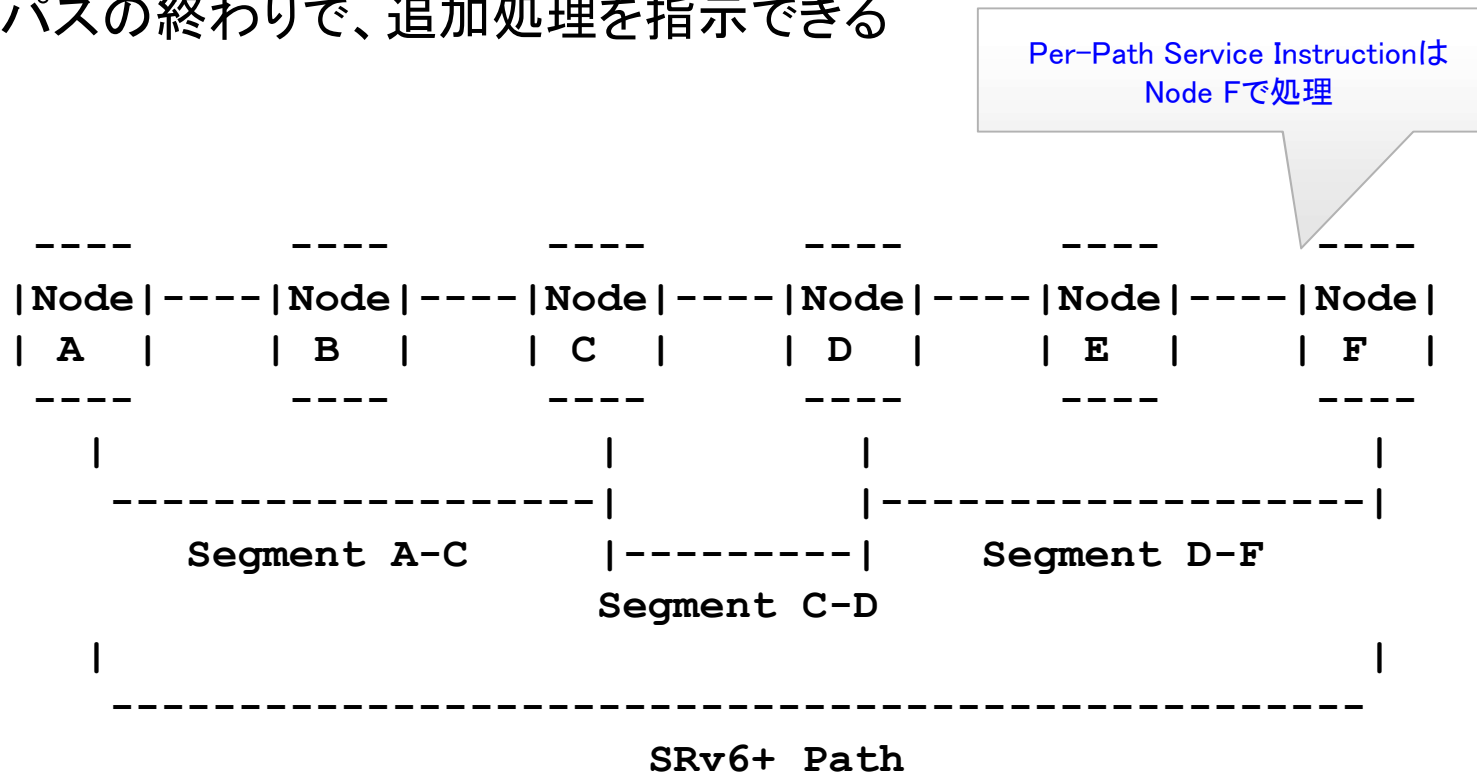


Figure 1: Paths, Segments And Instructions

例えば、カプセル化する/外す指示を入れることで、VPNを提供する用途に使う

# SRv6 vs SRv6+

## ■ 圧縮効果

SIDs	SRv6 SRH (128-bit SID)	SRv6+ CRH-16	SRv6+ CRH-32
1	24	8	8
2	40	8	16
3	56	16	16
4	72	16	24
5	88	16	24
6	104	16	32
7	120	24	32
8	136	24	40
9	152	24	40
10	168	24	N/A
11	184	32	N/A
12	200	32	N/A
13	216	32	N/A
14	232	32	N/A
15	248	40	N/A
16	264	40	N/A
17	280	40	N/A
18	296	40	N/A

Table 1: Routing Header Size (in Bytes) As A Function Of Routing Header Type and Number Of SIDs

- 例) Traffic Engineeringをして 12ホップさせたいときのオーバーヘッド
  - SRv6.... 240 byte (IPv6 Header 40 byte, SRH 200 byte)
  - SRv6+ ... 72 byte (IPv6 Header 40 byte, CRH-16 32 byte)

# まとめ

## ■ SRv6+ のポイント

- Topological Instruction と Service Instruction を分離 (下図参照)
- Topological Instruction (SID) は Routing Header で表現 → CRH
  - Routing Header 内の SID は、IS-IS で広告
  - SID の値そのものは、IPv6 アドレスのセマンティクスとは関係ない
  - CRH の SID は 16bit or 32bit → パケット長の圧縮効果
- Service Instruction は Destination Option Header で表現

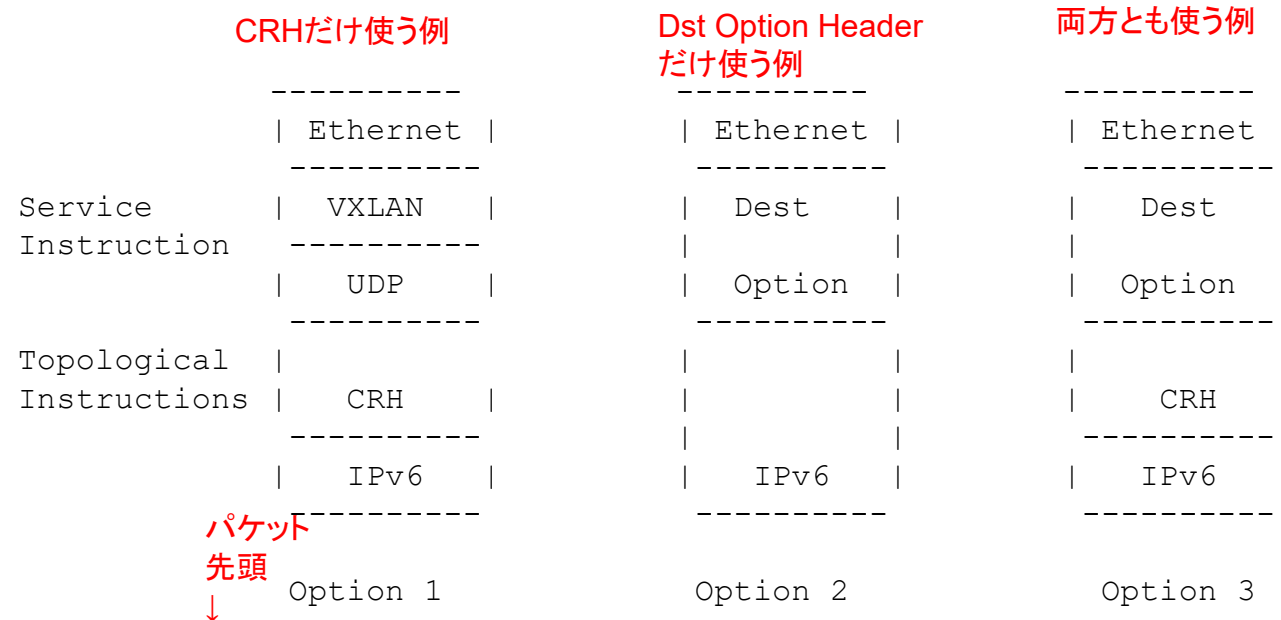


Figure 2: EVPN Design Alternatives