

IP anycast + SRv6によるNetwork APIの提供

2019/2/22 (15分発表+5分質疑)
ENOG55@味覚天国たまや, 燕三条

KDDI総合研究所

宮坂 拓也

自己紹介

名前：宮坂拓也

所属：KDDI総合研究所

会社での活動：

2011年	入社 & KDDIネットワーク運用 (バックボーン)
2012年～2018年	KDDI ネットワーク開発 (バックボーン)
2018年～	ネットワーク関連の研究開発

社外での活動：IETF, JANOG, AECC (ちょっとだけ：3GPP,ETSI)

はなすこと



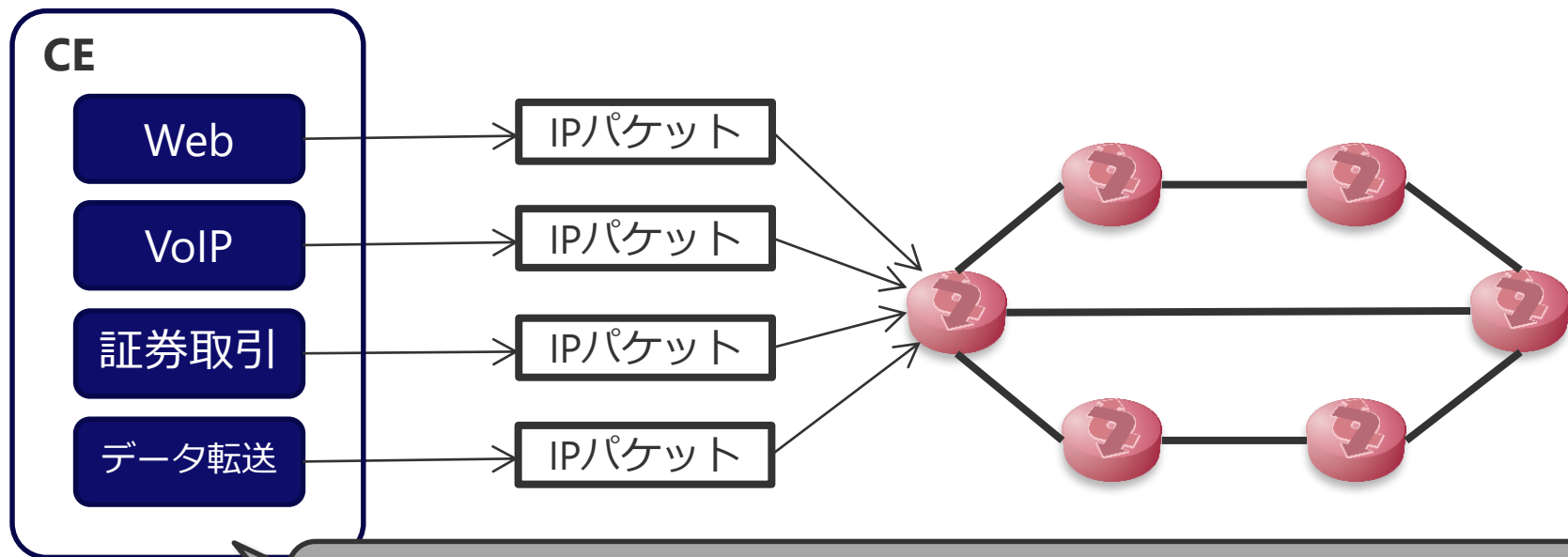
1. ユーザー・ネットワークオペレーターが困っていること
2. SRv6でやりたいこと : Network API
3. コントロールプレーンをどうするか : IP Anycast + SRv6

はなすこと



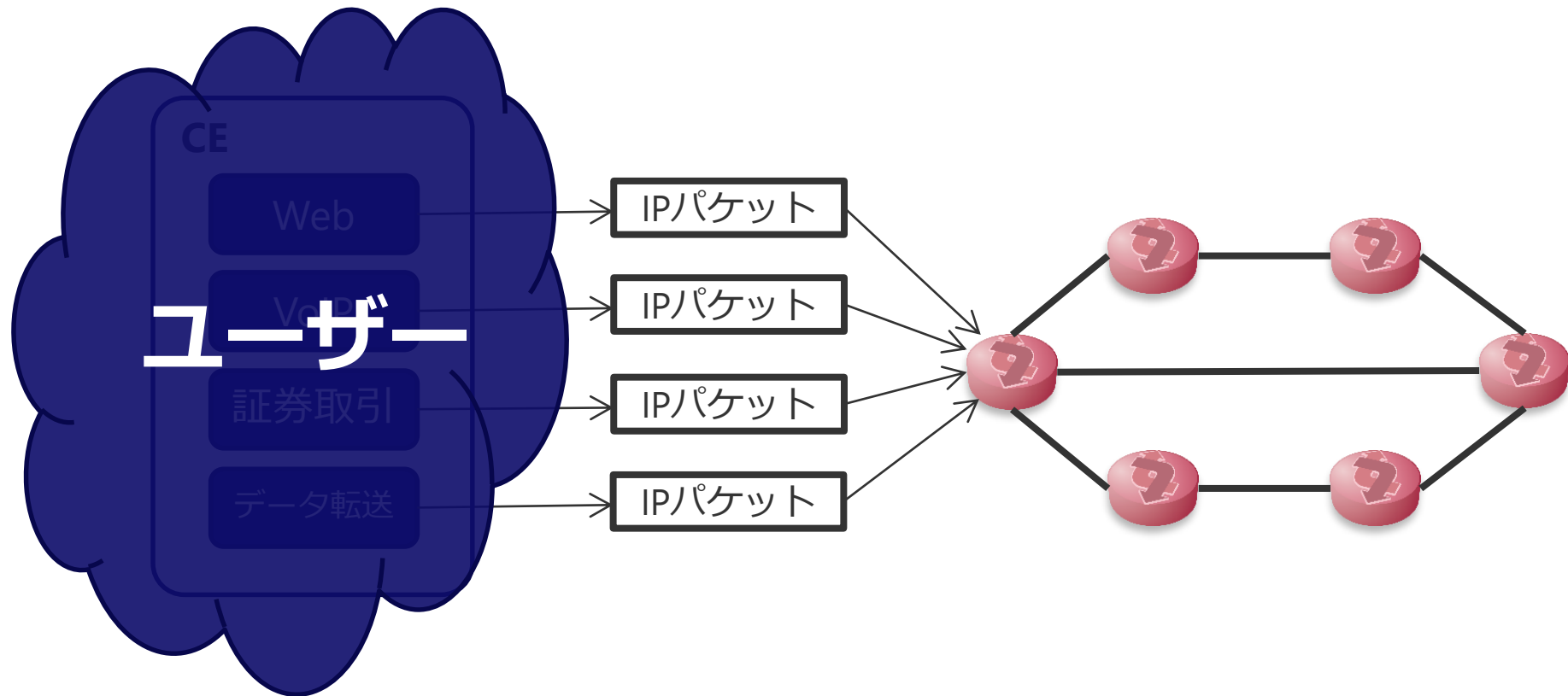
1. ユーザー・ネットワークオペレーターが困っていること
2. SRv6でやりたいこと : Network API
3. コントロールプレーンをどうするか : IP Anycast + SRv6

ユーザーとネットワークオペレーター

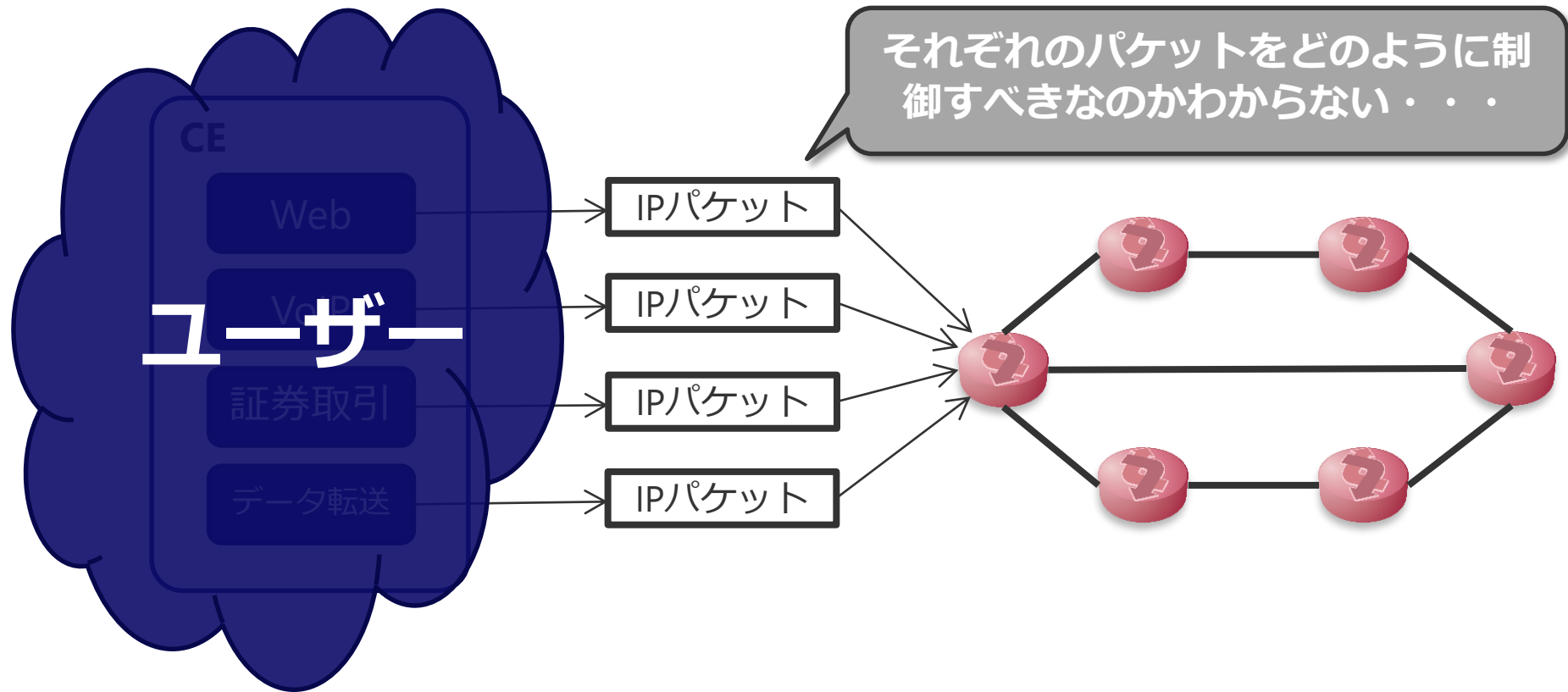


ユーザーが利用するアプリケーションは多種多様で、それぞれのアプリケーション毎に実施すべきネットワーク制御が異なる

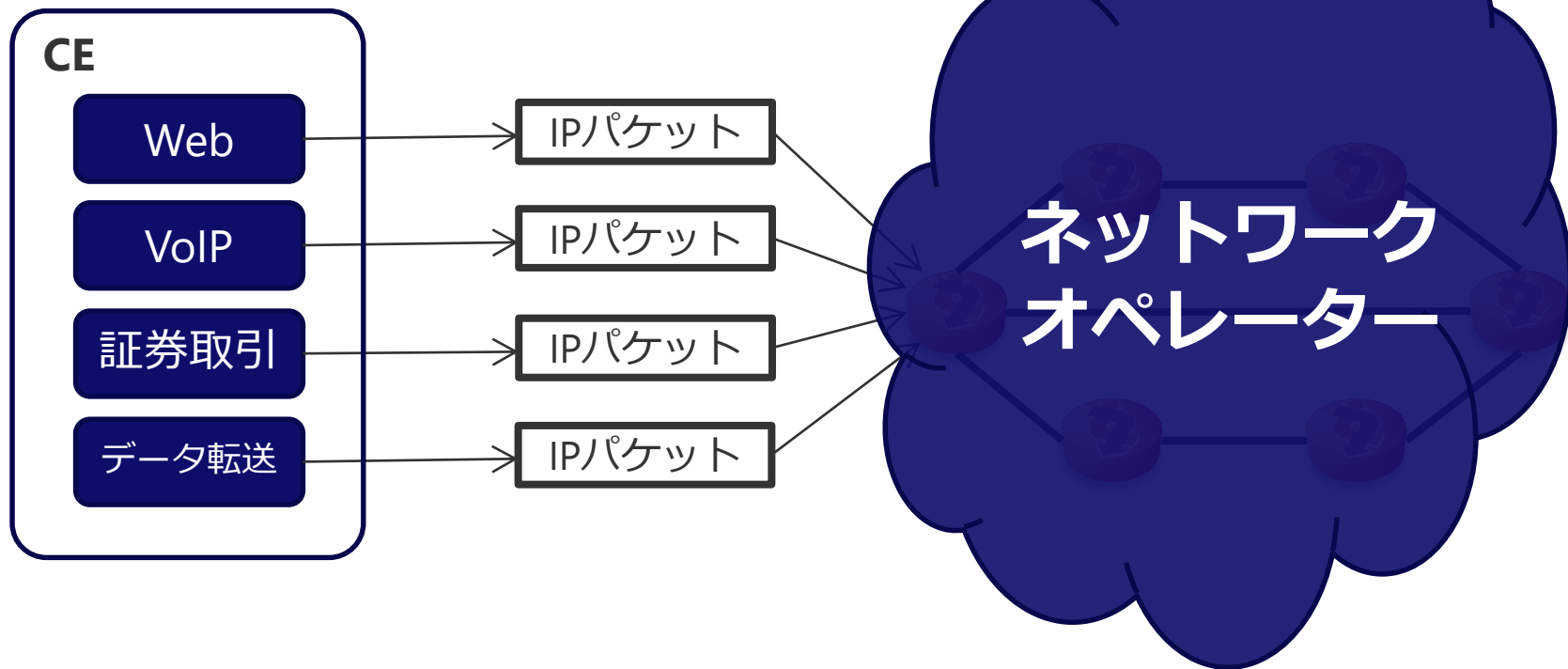
ネットワークオペレーターの悩み



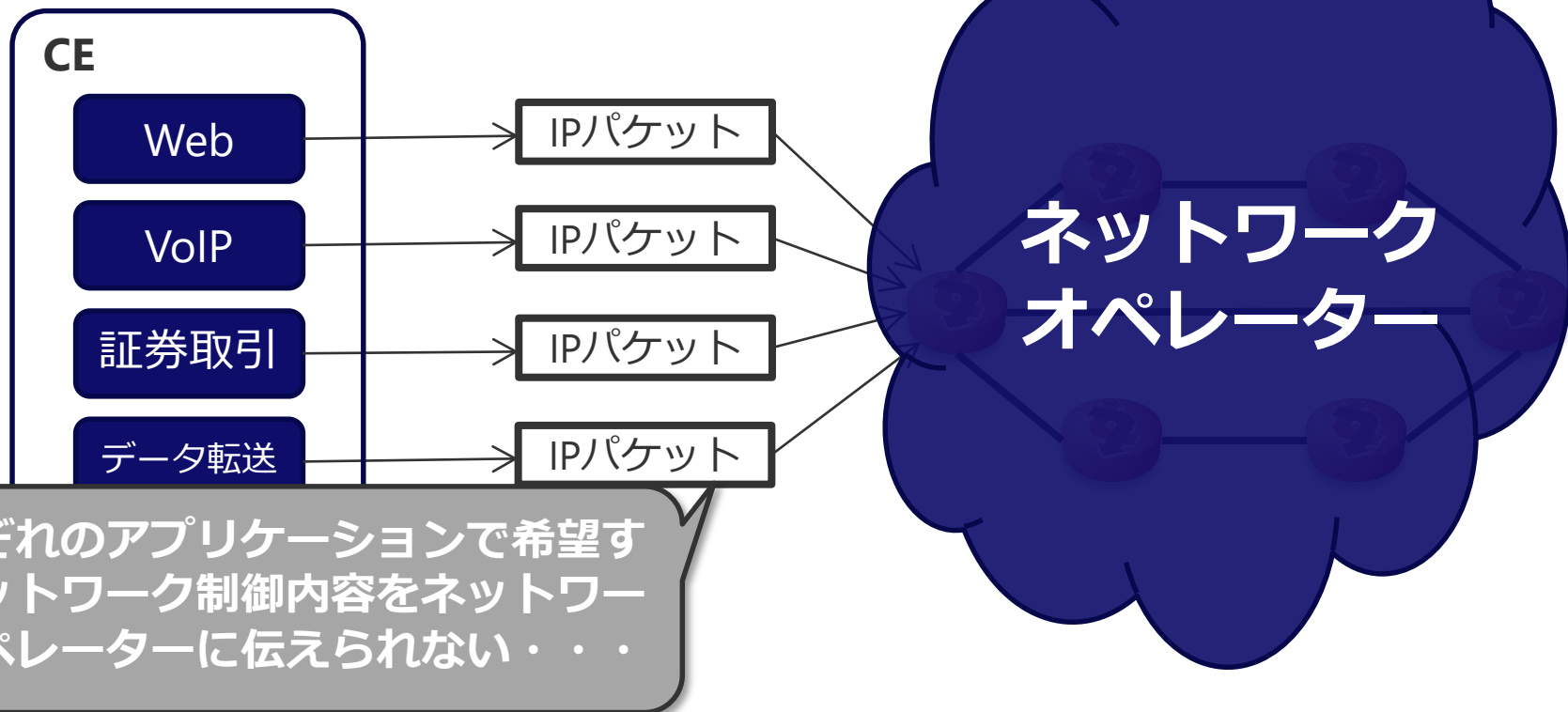
ネットワークオペレーターの悩み



ユーザーの悩み

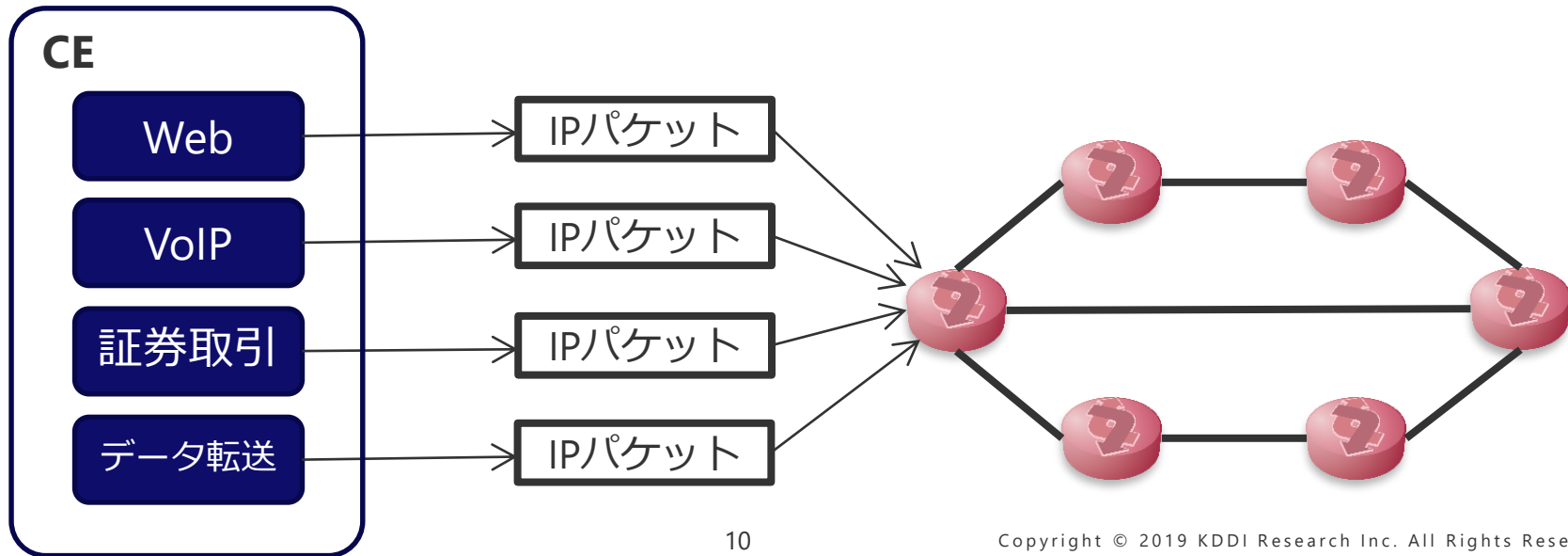


ユーザーの悩み

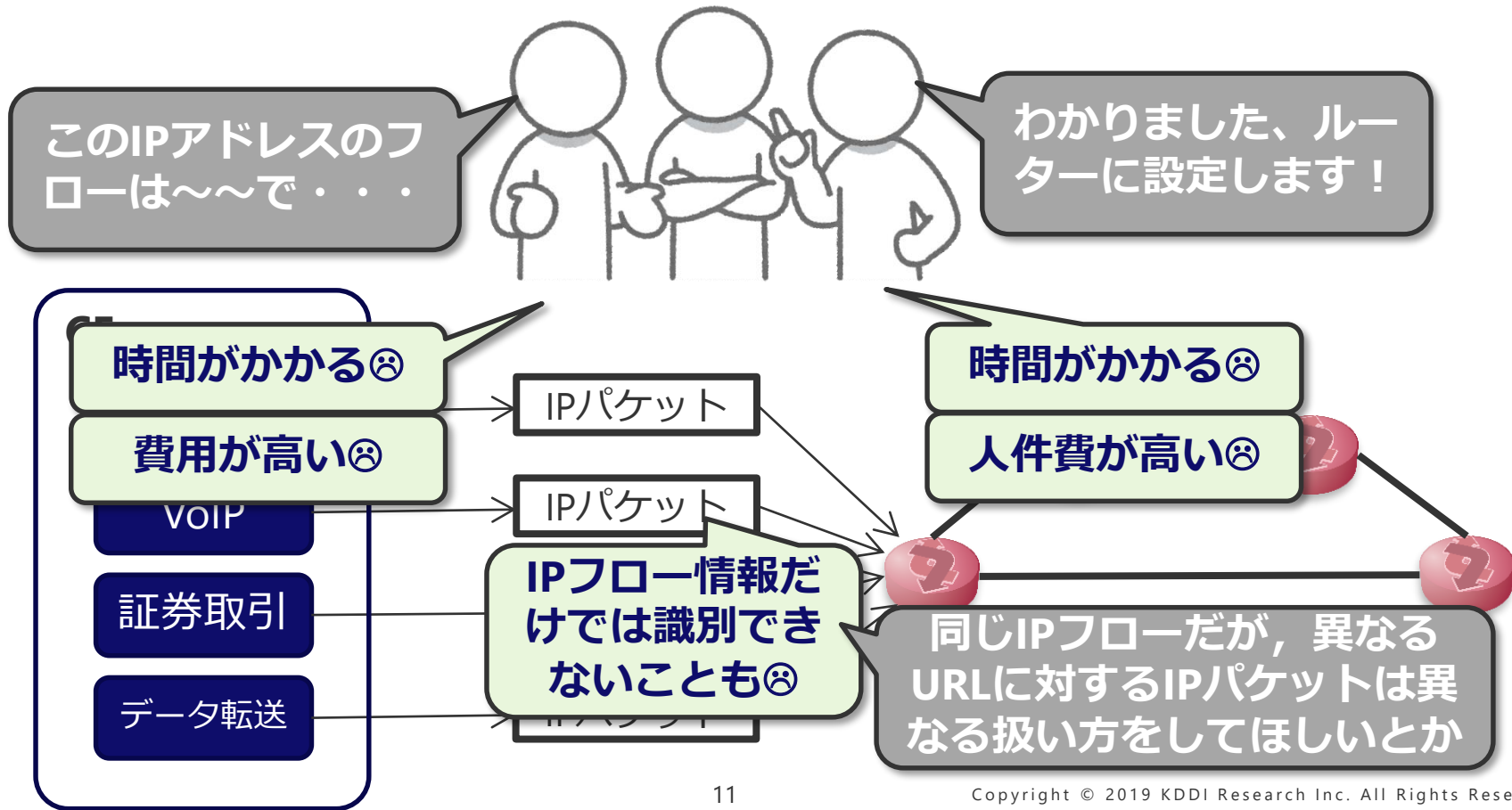


それぞれのアプリケーションで希望するネットワーク制御内容をネットワークオペレーターに伝えられない・・・

よくある解決方法？



よくある解決方法？

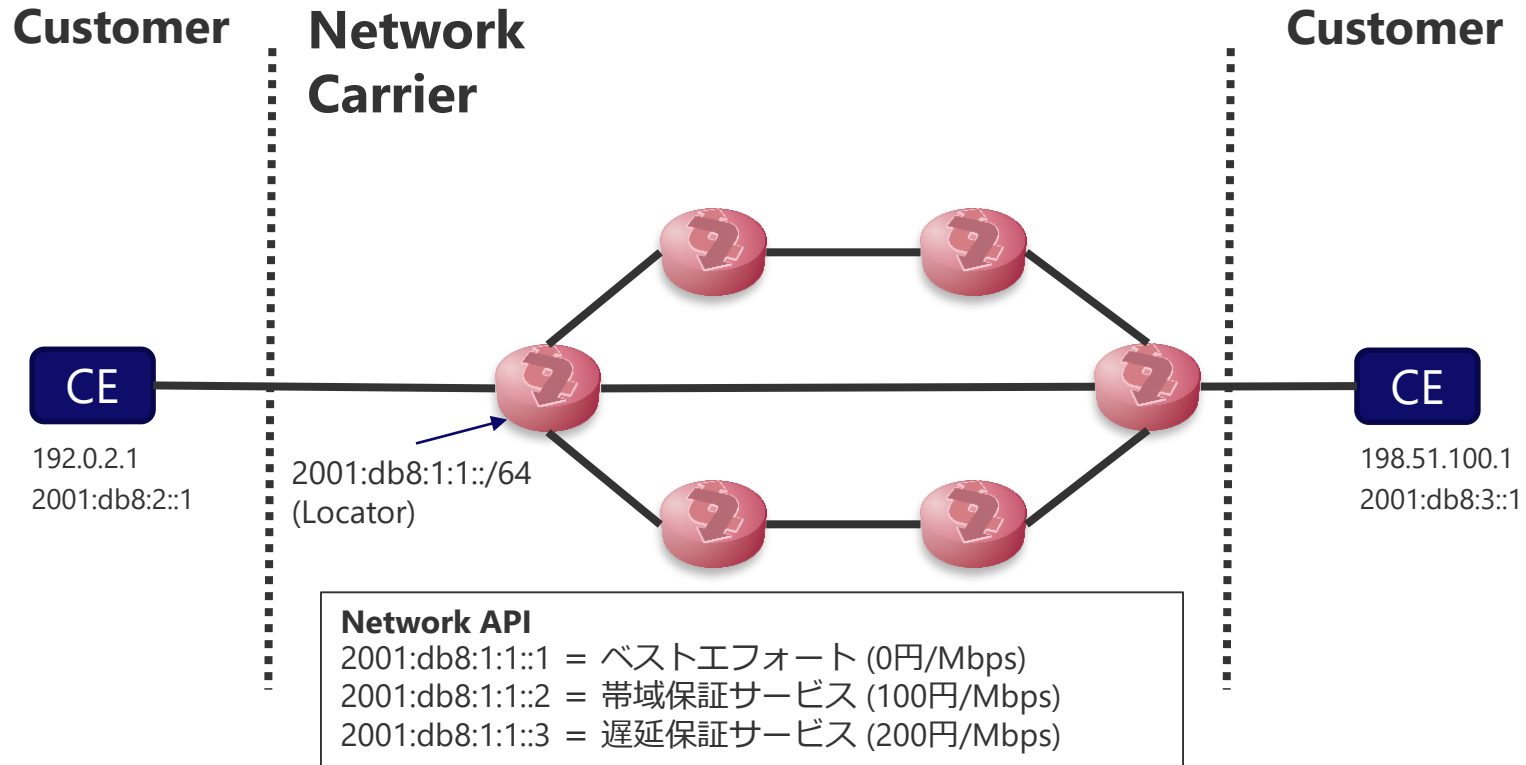


はなすこと

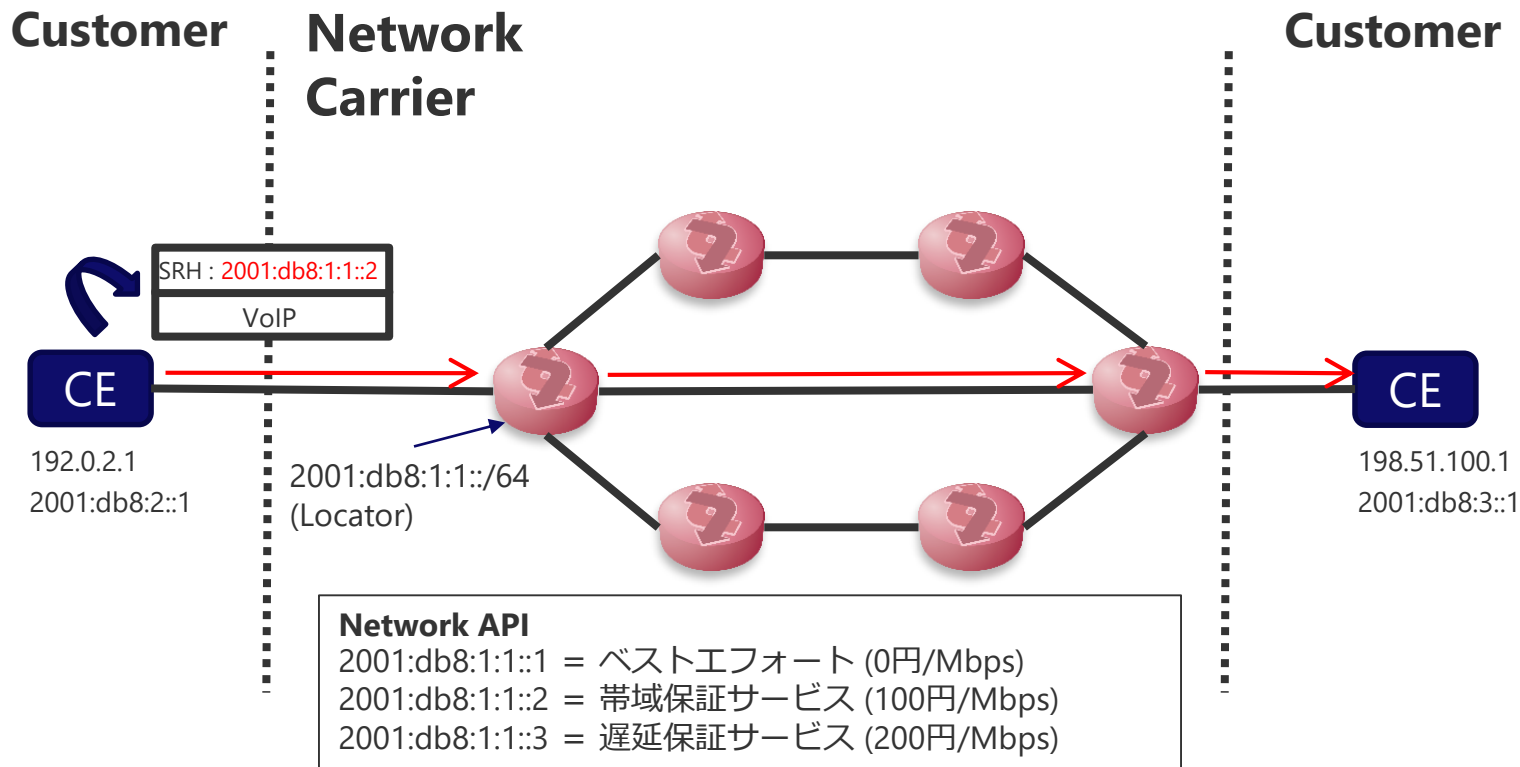


1. ユーザー・ネットワークオペレーターが困っていること
2. SRv6でやりたいこと : Network API
3. コントロールプレーンをどうするか : IP Anycast + SRv6

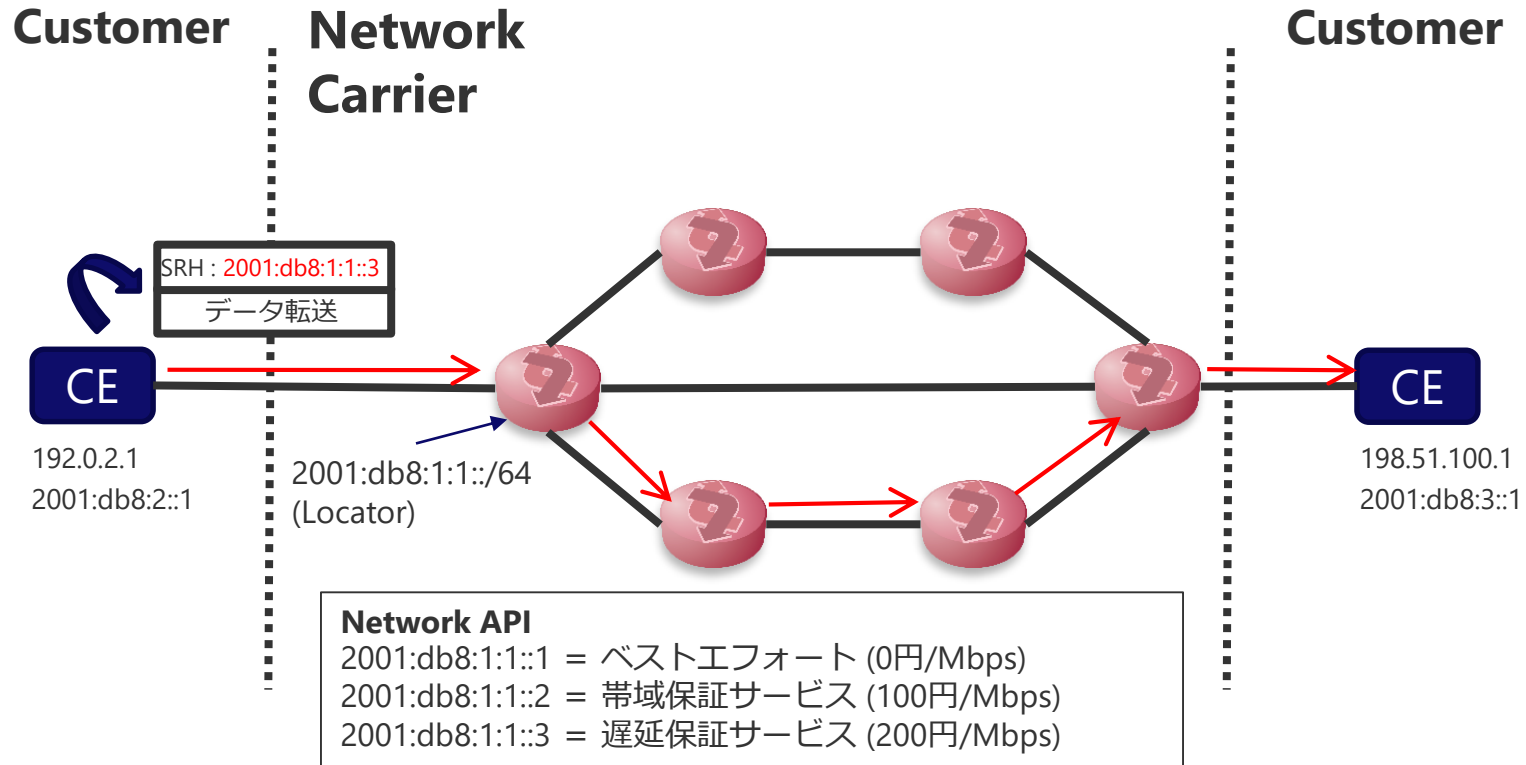
SRv6でユーザーにNetworkのAPIを提供したい



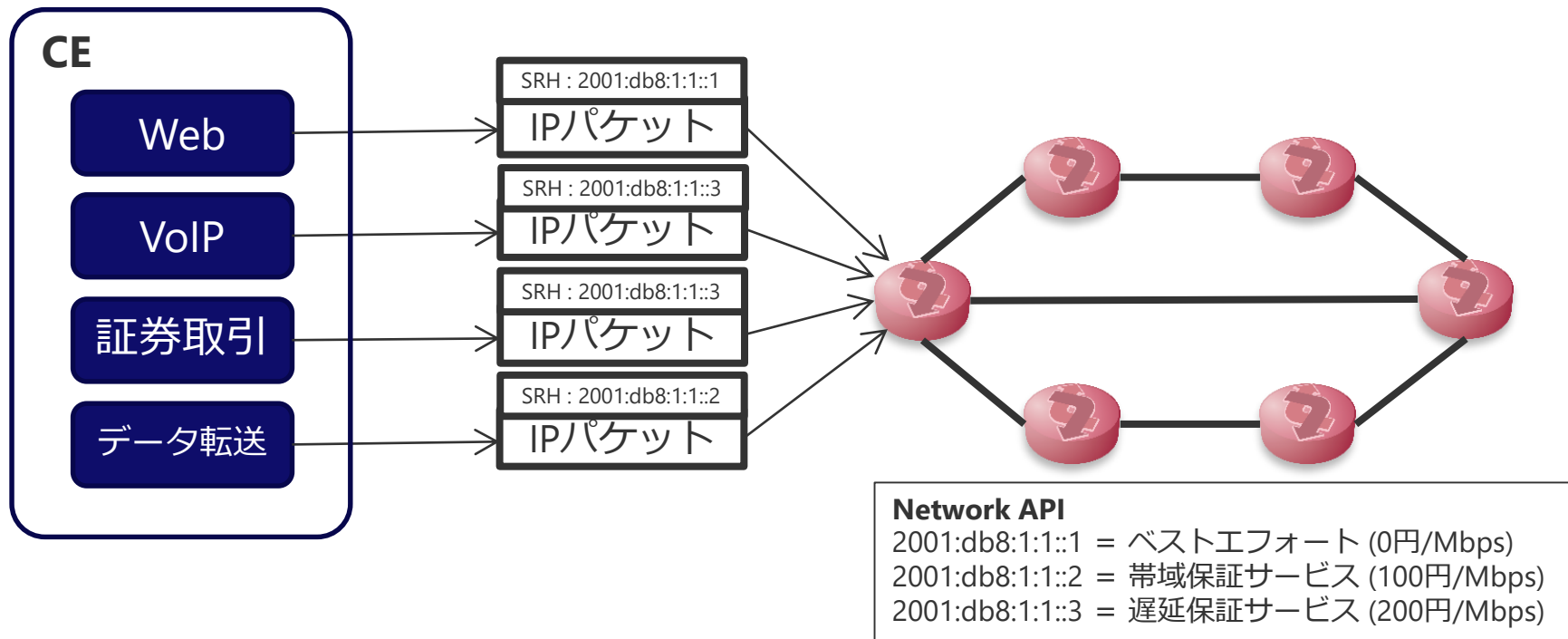
SRv6でユーザーにNetworkのAPIを提供したい



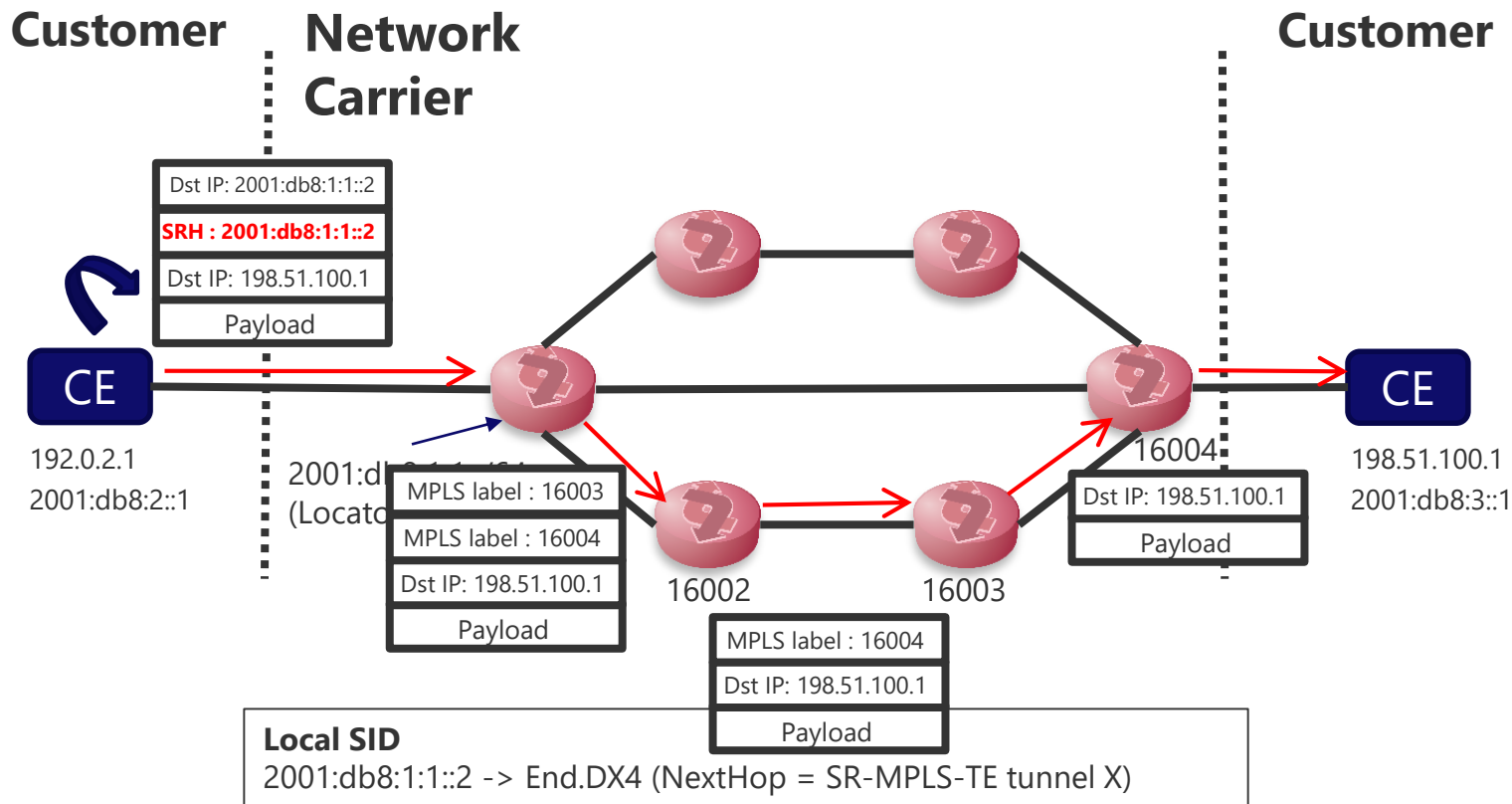
SRv6でユーザーにNetworkのAPIを提供したい



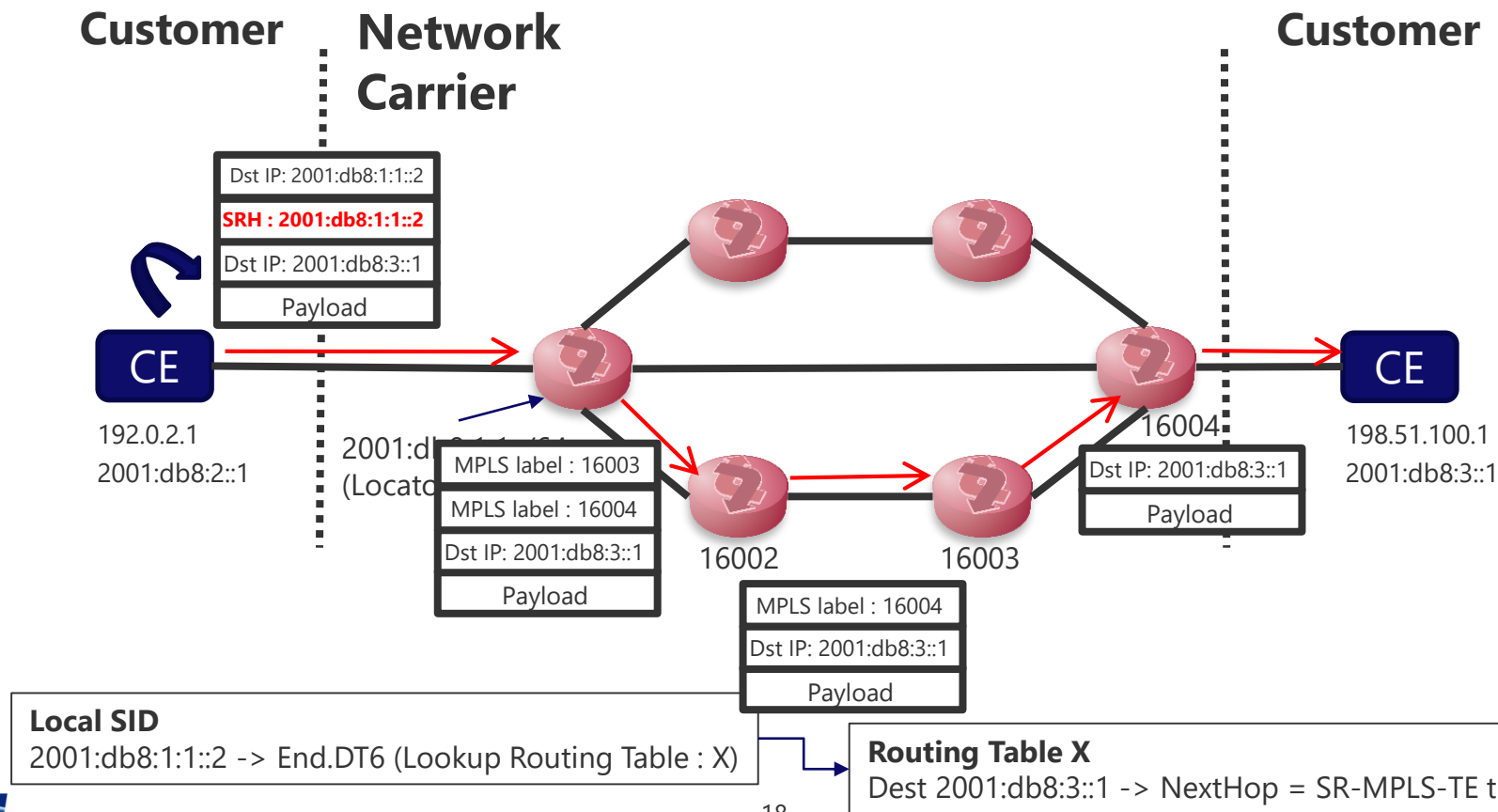
ユーザーとネットワークオペレーター



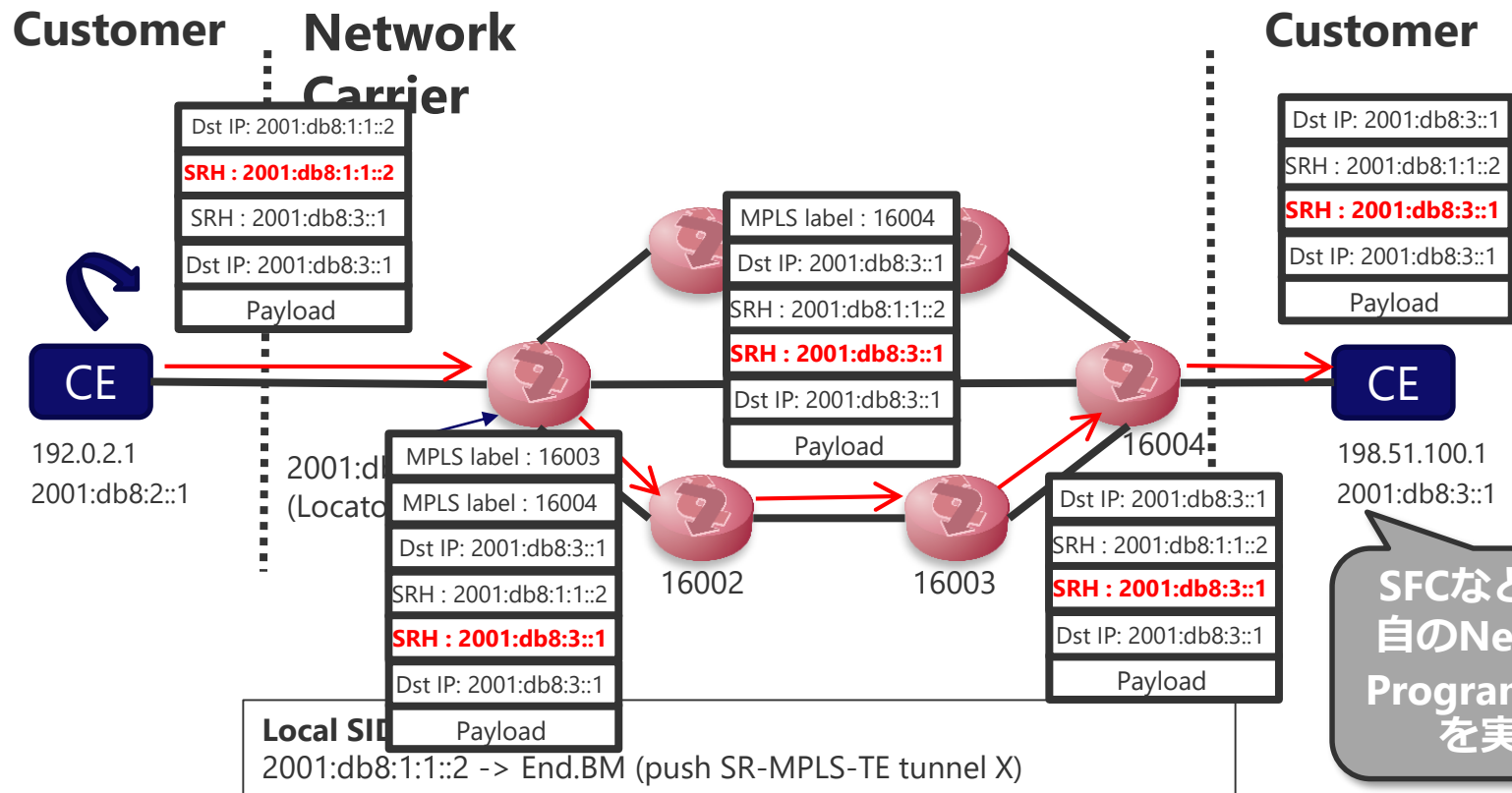
たとえば : End.DX4でやると



たとえば : End.DT6でやると



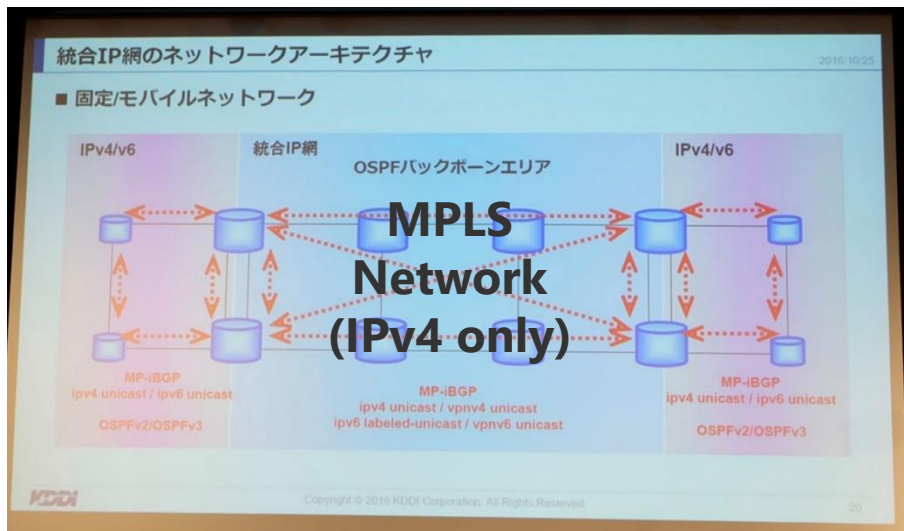
たとえば：End.BMでやると



TEはMPLS前提？SRv6でもできるよね？

なんでMPLS前提にしている？

- ▶ 現在のKDDIバックボーンがMPLSで動いているから
- ▶ SR-MPLSの方がオーバーヘッドが小さいから
 - MPLSラベル：20bit IPv6アドレス：128bit



<https://techplay.jp/column/14>

はなすこと

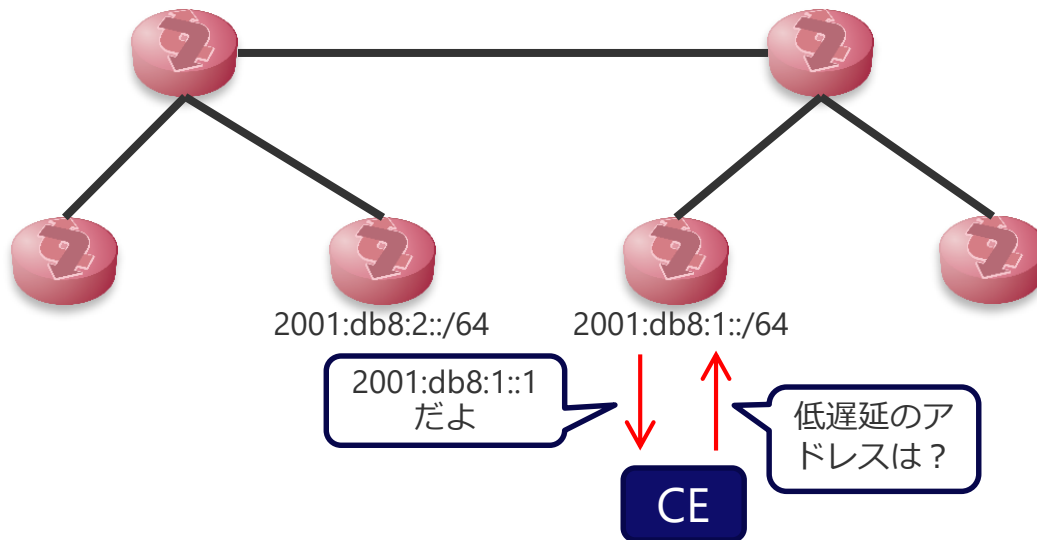


1. ユーザー・ネットワークオペレーターが困っていること
2. SRv6でやりたいこと : Network API
3. コントロールプレーンをどうするか : IP Anycast + SRv6

コントロールプレーンの問題

ユーザーが様々な拠点を持っていたり、ユーザー端末が移動する場合(スマートフォン・自動車)、各地点でそれぞれのSRv6 SIDを取得する必要がある

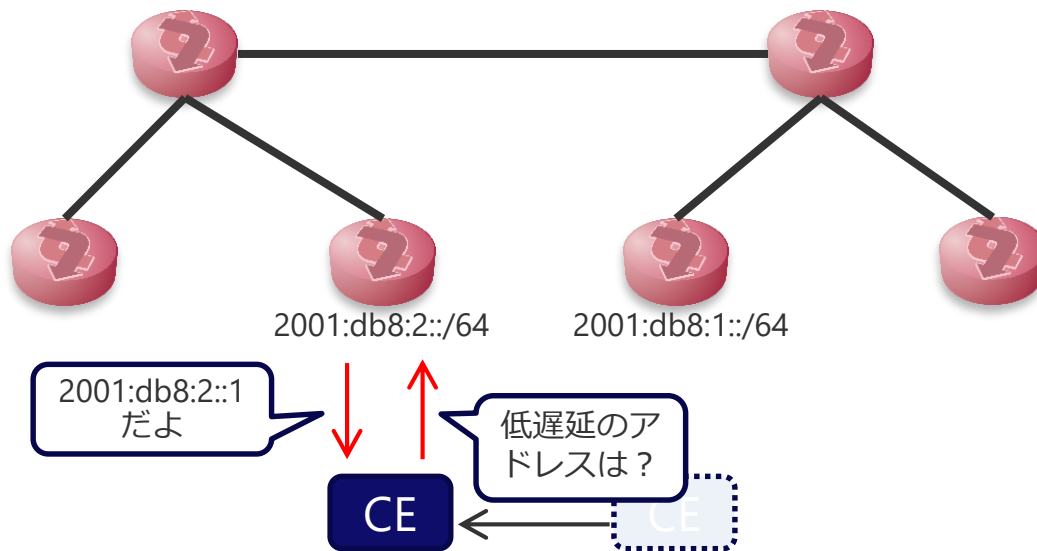
- ▶ 拠点毎にIPv6アドレスを変更する実装が必要
- ▶ 端末側において、定期的なアドレス解決機構の実装が必要
- ▶ アドレス解決の時間分オーバーヘッドとなりアプリケーション遅延増大・電力消費増大



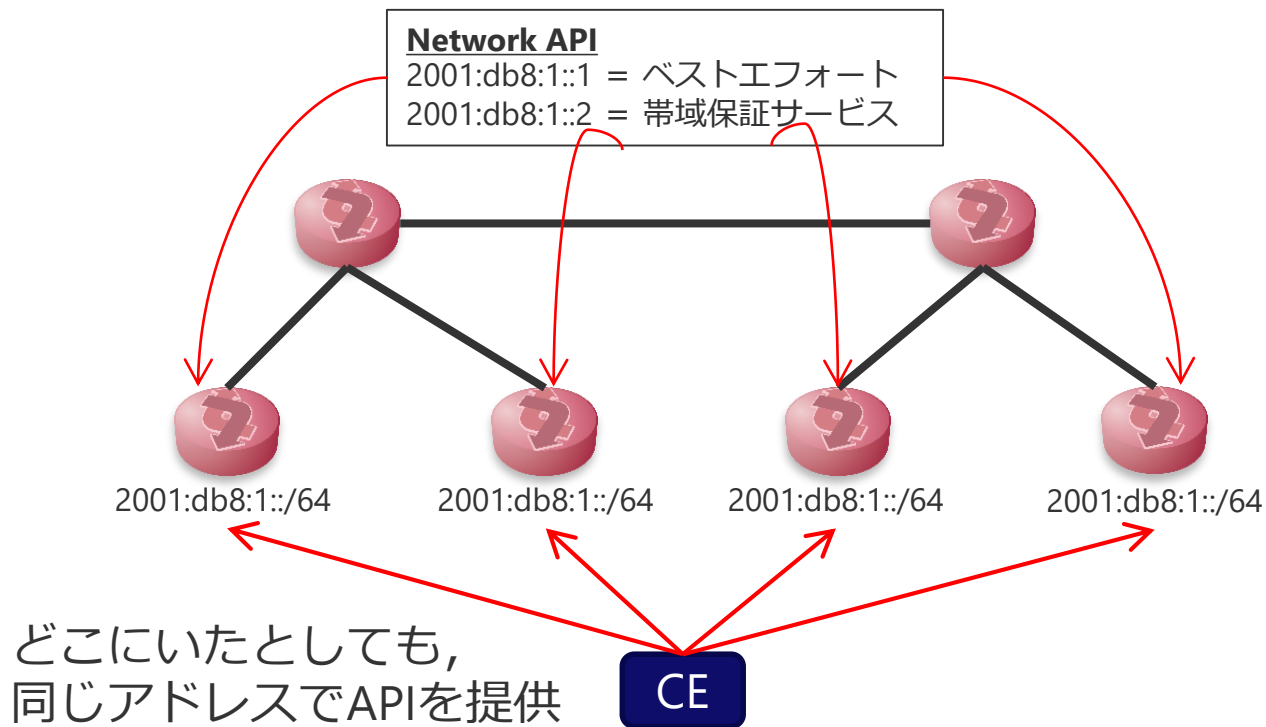
コントロールプレーンの問題

ユーザーが様々な拠点を持っていたり、ユーザー端末が移動する場合(スマートフォン・自動車)、各地点でそれぞれのSRv6 SIDを取得する必要がある

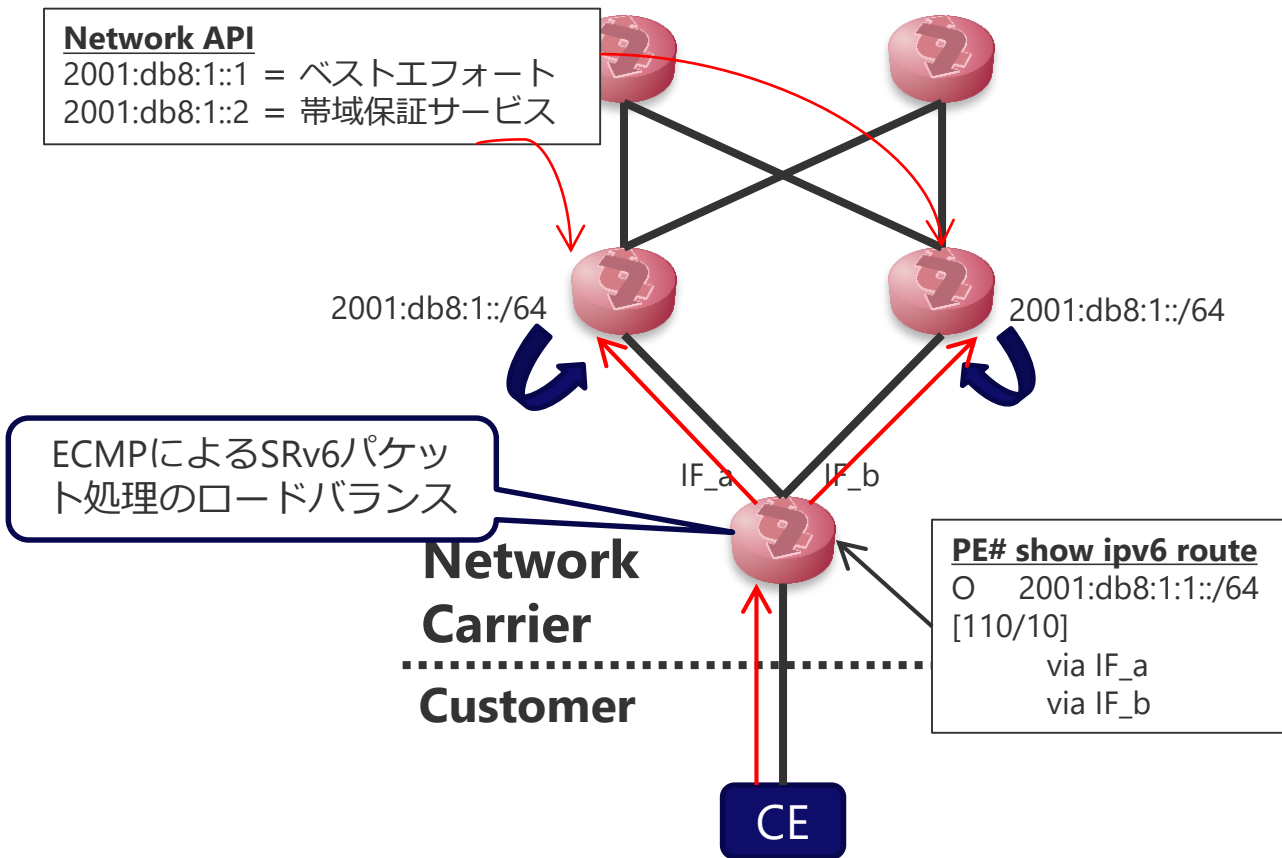
- ▶ 拠点毎にIPv6アドレスを変更する実装が必要
- ▶ 端末側において、定期的なアドレス解決機構の実装が必要
- ▶ アドレス解決の時間分オーバーヘッドとなりアプリケーション遅延増大・電力消費増大



IP anycast + SRv6 = Network API



Load Balance(ECMP) + Resiliency



Load Balance(ECMP) + Resiliency

Network API

2001:db8:1::1 = ベストエフォート
2001:db8:1::2 = 帯域保証サービス

2001:db8:1::/64

2001:db8:1::/64

機器障害時には自動で
L3 Routingで切り替え
ることができる

**Network
Carrier**

Customer

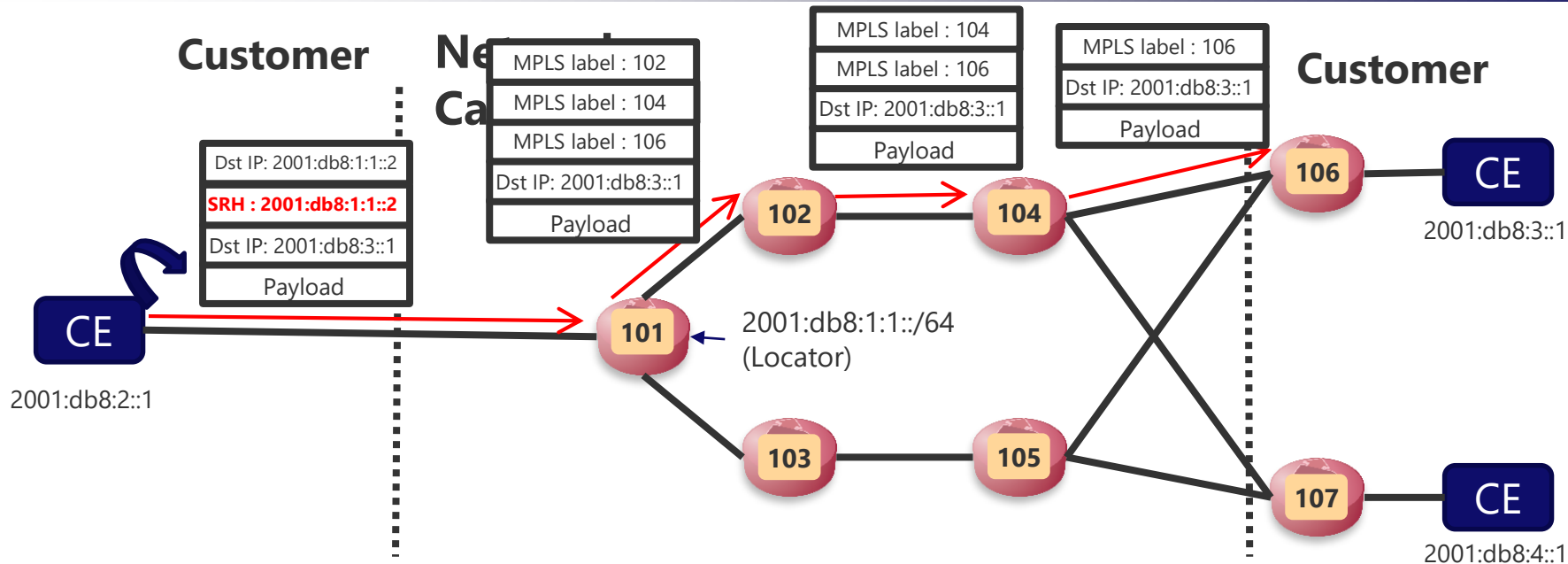
IF_a IF_b

CE

PE# show ipv6 route

```
O 2001:db8:1:1::/64  
[110/10]  
via IF_a  
via IF_b
```

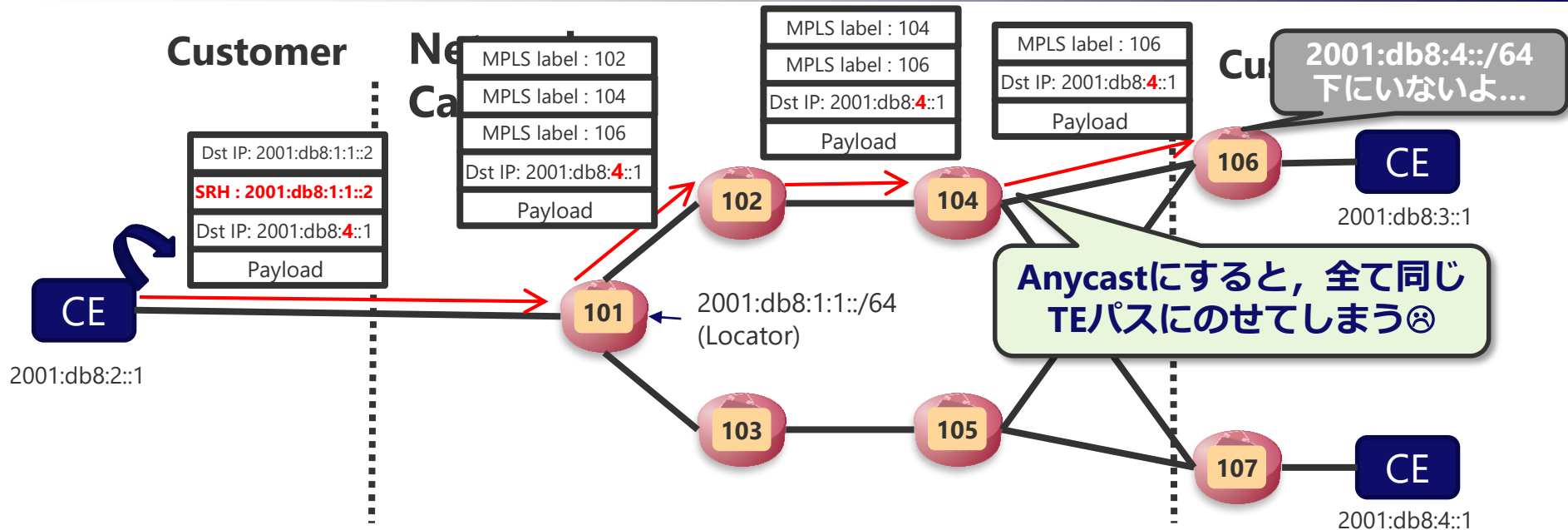
Anycastにしたときの制約：例)End.DX6



Local SID

2001:db8:1:1::2 -> **End.DX6** (NextHop = SR-MPLS-TE tunnel X (PUSH MPLS label : 102,104,106))

Anycastにしたときの制約：例)End.DX6



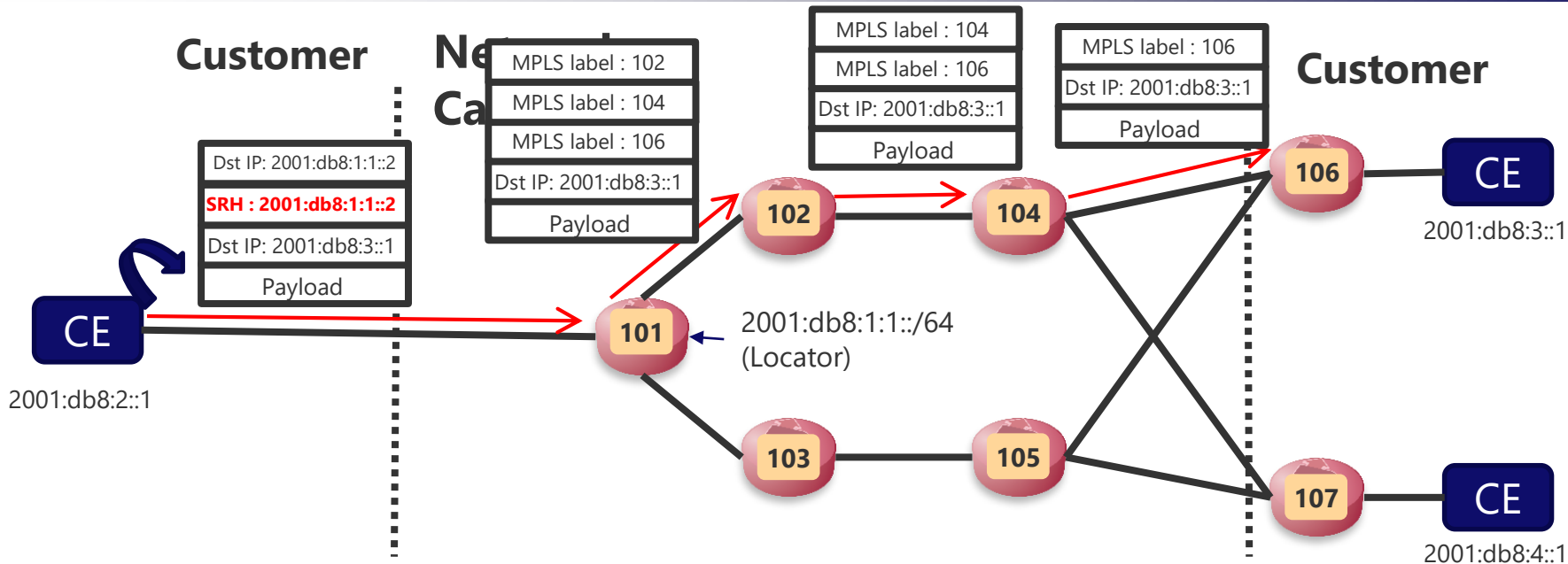
Local SID
 2001:db8:1:1::2 -> **End.DX6** (NextHop = SR-MPLS-TE tunnel X (PUSH MPLS label : 102,104,106))

Anycastにしたときの制約

SIDをAnycastにした場合…

- ▶ SIDと転送先のTEパスが1対1対応しているFunctionは利用できない！
 - End.X
 - End.DX6
 - End.DX4
 - End.BM
- ▶ TEパスを指定するのではなく、ルーティングテーブルを参照するFunctionが利用できる！？
 - End.T
 - End.DT6
 - End.DT4
 - End.DT46

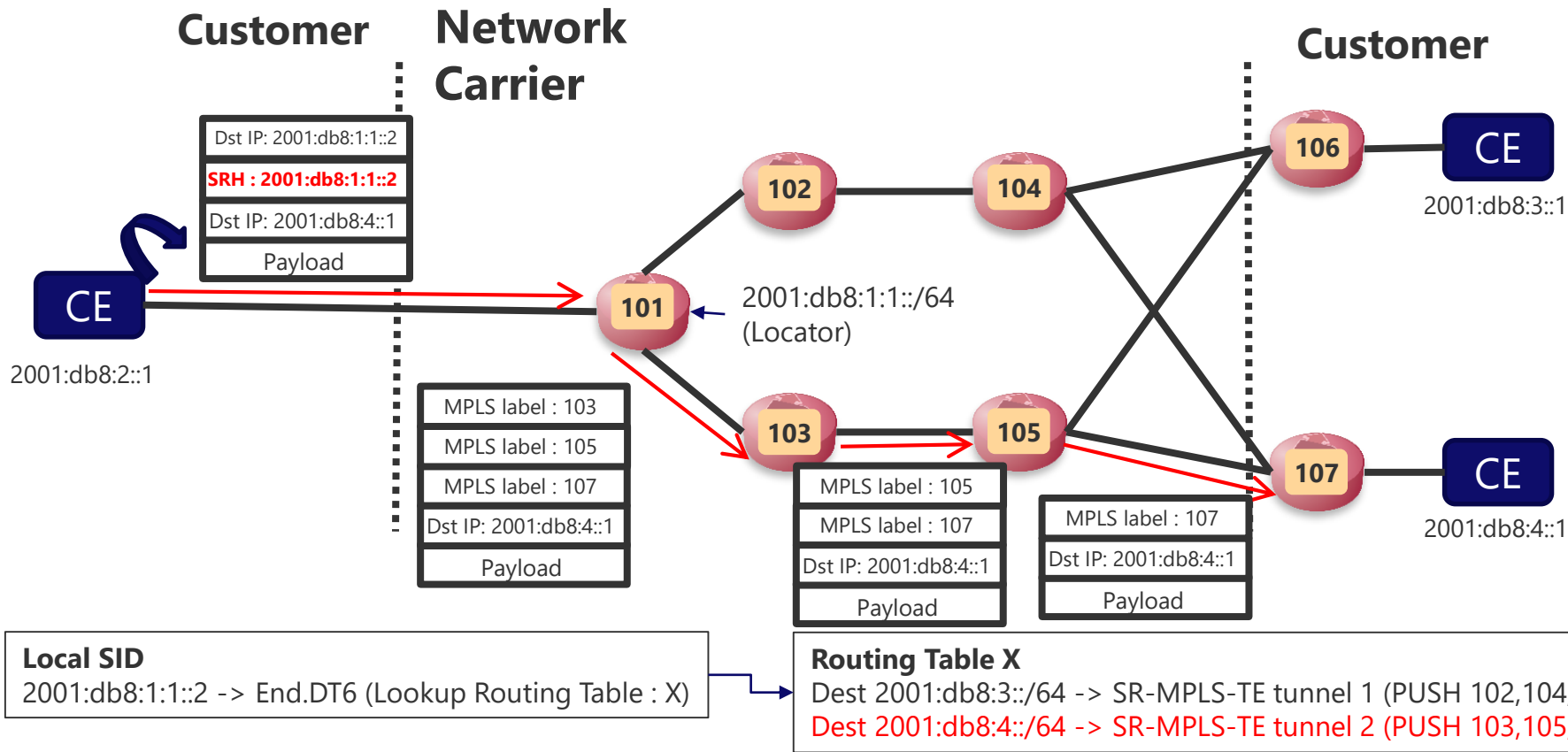
例) Anycast + End.DT6



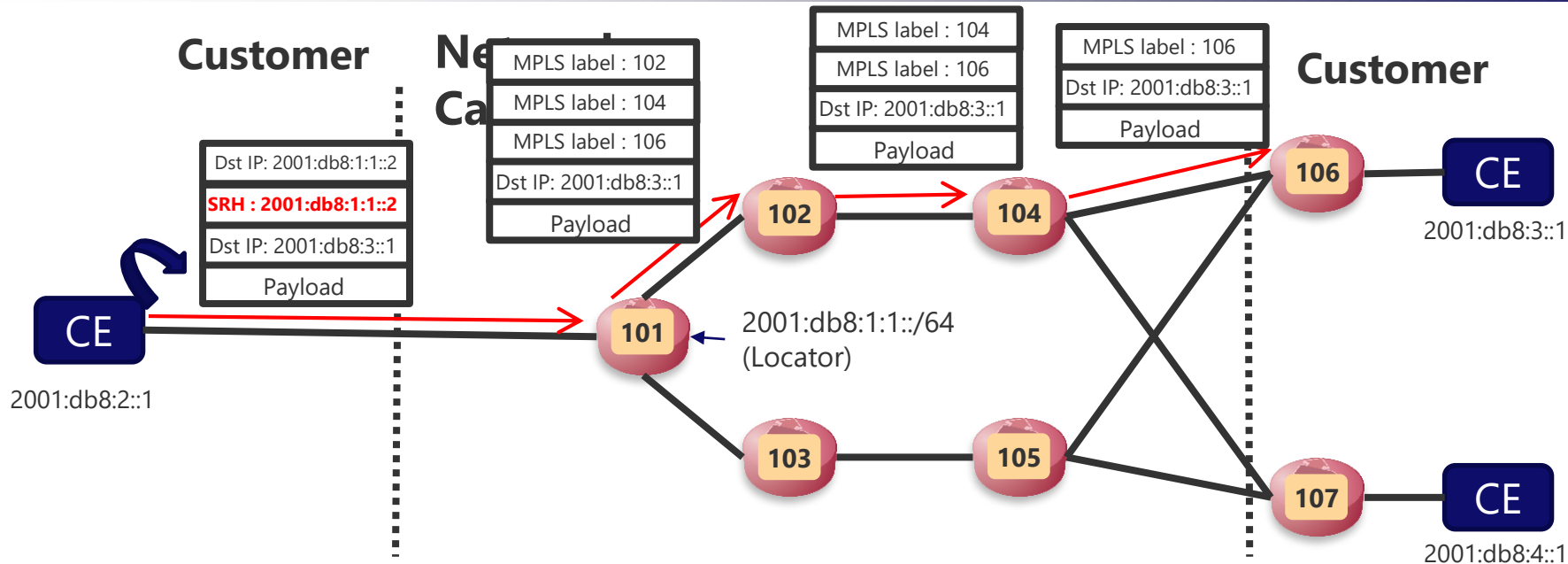
Local SID
2001:db8:1:1::2 -> End.DT6 (Lookup Routing Table : X)

Routing Table X
Dest 2001:db8:3::/64 -> SR-MPLS-TE tunnel 1 (PUSH 102,104,106)
Dest 2001:db8:4::/64 -> SR-MPLS-TE tunnel 2 (PUSH 103,105,107)

例) Anycast + End.DT6



例) Anycast + End.DT6



Local SID

2001:db8:1:1::2 -> End.DT6 (Lookup Routing Table : X)
 2001:db8:1:1::3 -> End.DT6 (Lookup Routing Table : Y)

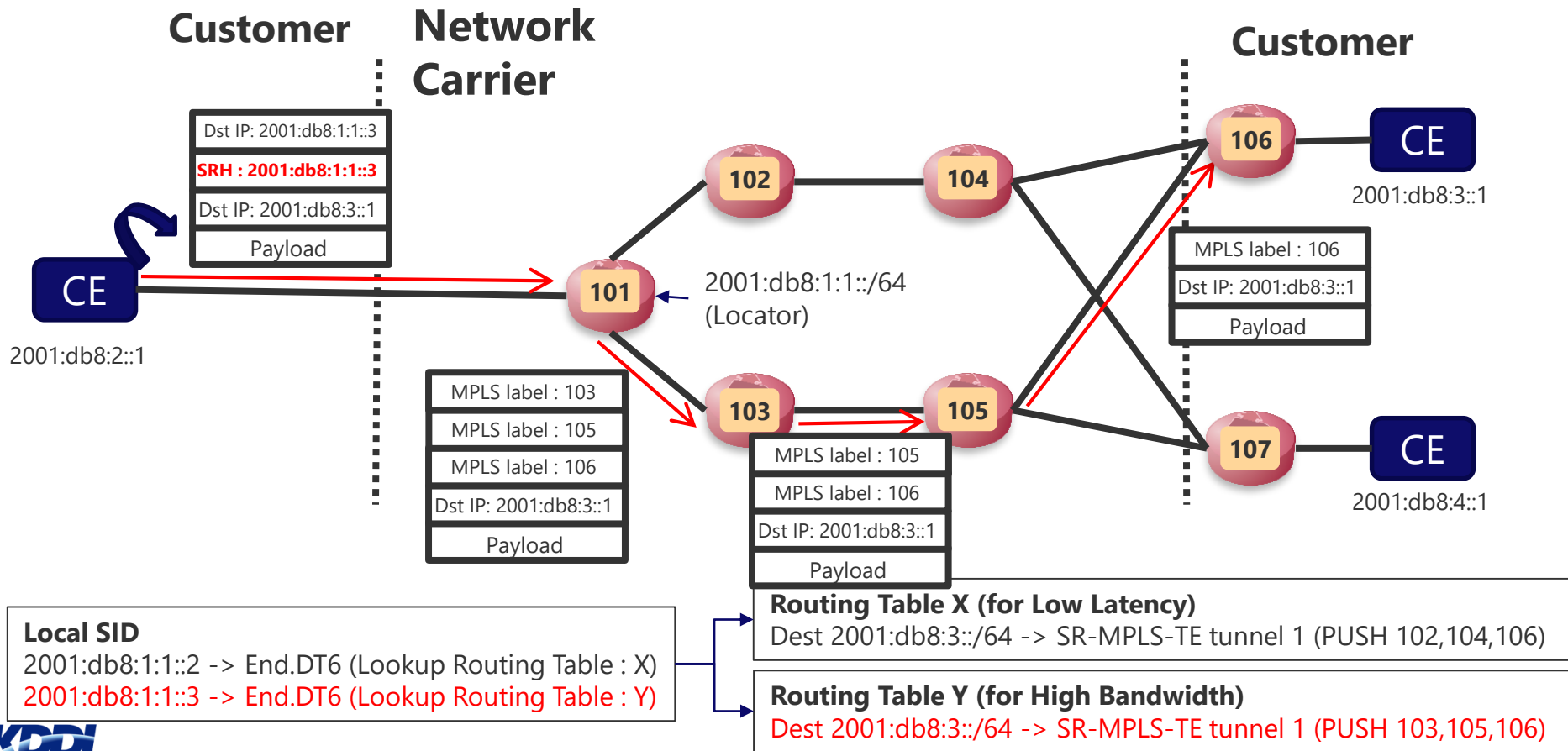
Routing Table X (for Low Latency)

Dst 2001:db8:3::/64 -> SR-MPLS-TE tunnel 1 (PUSH 102,104,106)

Routing Table Y (for High Bandwidth)

Dst 2001:db8:3::/64 -> SR-MPLS-TE tunnel 1 (PUSH 103,105,106)

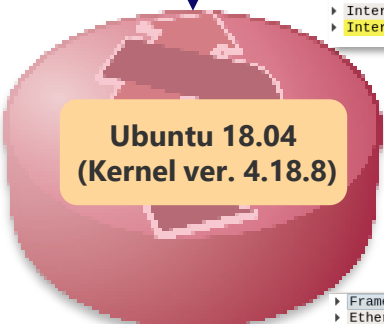
例) Anycast + End.DT6



Linux Kernelにおける動作を見てみた

```
▶ Frame 3: 182 bytes on wire (1456 bits), 182 bytes captured (1456 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_2a:bd:b4 (08:00:27:2a:bd:b4), Dst: PcsCompu_78:16:be (08:00:27:78:16:be)
▶ Internet Protocol Version 6, Src: 2001:db8:23:1::1, Dst: 2001:db8:2516::1
  0110 .... = Version: 6
  .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 0111 0010 0110 1111 1100 = Flow Label: 0x726fc
  Payload Length: 128
  Next Header: Routing Header for IPv6 (43)
  Hop Limit: 63
  Source: 2001:db8:23:1::1
  Destination: 2001:db8:2516::1
  ▶ Routing Header for IPv6 (Segment Routing)
    Next Header: IPv6 (41)
    Length: 2
    [Length: 24 bytes]
    Type: Segment Routing (4)
    Segments Left: 0
    First segment: 0
    ▶ Flags: 0x00
      Reserved: 0000
      Address[0]: 2001:db8:2516::1
    ▶ [Segments in Traversal Order]
      Address[0]: 2001:db8:2516::1
  ▶ Internet Protocol Version 6, Src: 2001:db8:12:1::1, Dst: 2001:db8:45:1::1
  ▶ Internet Control Message Protocol v6
```

SRv6パケット



MPLSパケット

```
▶ Frame 7: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_f8:0b:3a (08:00:27:f8:0b:3a), Dst: PcsCompu_b9:92:d1 (08:00:27:b9:92:d1)
▶ MultiProtocol Label Switching Header, Label: 101, Exp: 0, S: 1, TTL: 63
  0000 0000 0000 0110 0101 .... = MPLS Label: 101
  .... 0000 0000 0000 0000 0000 .... = MPLS Experimental Bits: 0
  .... 0000 0000 0000 0000 0000 .... = MPLS Bottom Of Label Stack: 1
  .... 0000 0000 0000 0000 0000 .... = MPLS Label Stack Length: 1
  .... 0011 1111 = MPLS TTL: 63
▶ Internet Protocol Version 6, Src: 2001:db8:12:1::1, Dst: 2001:db8:45:1::1
▶ Internet Control Message Protocol v6
```

```
~$ ip -6 route
2001:db8:2516::1 encap seg6local action End.DT6 table 123 via 2001:db8:34:1::2 dev enp0s9 proto zebra metric 1024 pref medium
```

```
~$ ip -6 route show table 123
2001:db8:45:1::/64 encap mpls 101 via 2001:db8:34:1::2 dev enp0s9 metric 1024 pref medium
```

参照 : Zebra SRv6 CLI on Linux Dataplane (ENOG#49)
<https://www.slideshare.net/kentaroebisawa/zebra-srv6-cli-on-linux-dataplane-enog49>

まとめ

1. SRv6を用いて、ユーザーにNetwork APIを提供する妄想を紹介しました
2. IP Anycastを応用することで、ユーザーがどこにいたとしても同一のSIDを利用可能
3. IP Anycastを利用した場合は、FunctionとしてはEnd.DT6などのテーブル参照型のものが利用可能である