



The bridge to possible

SRv6 + Network Service Mesh

SRv6のNetwork Programmingをもっと活用するために

Teppei Kamata

Systems Architect, Cisco Systems G.K.

Nov 2021

自己紹介

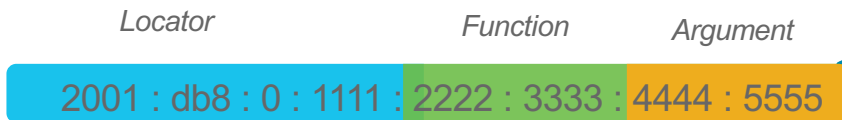
- 鎌田徹平 tkamata@cisco.com
- Cisco Systems 合同会社 2009年新卒入社
- Interop Tokyo ShowNet NOC team Member
- MPLS Japan **は2019年のSRv6 Service Chainingに続いて二回目**

Agenda

- ・はじめに
- ・ NSMとSRv6を使った事例
 - ・ QoEサービス
 - ・ L3VPN
- ・ デモ
- ・ 終わりに

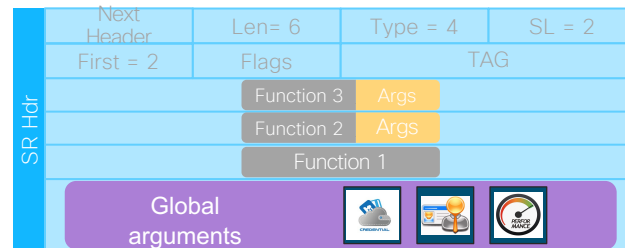
はじめに

- SRv6 Network ProgrammingがRFC8986になりました



Functionは自由に定義可能
In-Network Computing

- SRv6 SID は128ビットのIPv6アドレス表記
 - Locator: セグメントをペアレントノード*へRouteするためのビット
 - Function: ペアレントノード*において取られるActionを示すビット
 - Argument [optional]: 最後のビットはFunctionで参照される指数
- SID Functionはペアレントモードにおいてローカルに定義される
 - つまり、なんでも定義できる
- SRヘッダーが”Network Program”の情報を含んでいる



“SRv6 Network Programming”のコンセプトは、アプリケーションが複雑なプログラムを「ネットワーク上に分散する個々の機能の組み合わせ」としてエンコードすることを可能にする。

SRv6はいま何に使われているのか？

- 4.1. Requirements Language
- 3. SRv6 SID
 - 3.1. SID Format
 - 3.2. SID Allocation within an SR Domain
 - 3.3. SID Reachability
- 4. SR Endpoint Behaviors
 - 4.1. End: Endpoint
 - 4.1.1. Upper-Layer Header
 - 4.2. End.X: L3 Cross-Connect
 - 4.3. End.T: Specific IPv6 Table Lookup
 - 4.4. End.DX6: Decapsulation and IPv6 Cross-Connect
 - 4.5. End.DX4: Decapsulation and IPv4 Cross-Connect
 - 4.6. End.DT6: Decapsulation and Specific IPv6 Table Lookup
 - 4.7. End.DT4: Decapsulation and Specific IPv4 Table Lookup
 - 4.8. End.DT46: Decapsulation and Specific IP Table Lookup
 - 4.9. End.DX2: Decapsulation and L2 Cross-Connect
 - 4.10. End.DX2V: Decapsulation and VLAN L2 Table Lookup
 - 4.11. End.DT2U: Decapsulation and Unicast MAC L2 Table Lookup
 - 4.12. End.DT2M: Decapsulation and L2 Table Flooding
 - 4.13. End.B6.Encaps: Endpoint Bound to an SRv6 Policy with Encapsulation
 - 4.14. End.B6.Encaps.Red: End.B6.Encaps with Reduced SRH
 - 4.15. End.BM: Endpoint Bound to an SR-MPLS Policy
 - 4.16. Flavors
 - 4.16.1. PSP: Penultimate Segment Pop of the SRH
 - 4.16.2. USP: Ultimate Segment Pop of the SRH
 - 4.16.3. USD: Ultimate Segment Decapsulation
 - 5. SR Policy Headend Behaviors
 - 5.1. H.Encaps: SR Headend with Encapsulation in an SR Policy
 - 5.2. H.Encaps.Red: H.Encaps with Reduced Encapsulation
 - 5.3. H.Encaps.L2: H.Encaps Applied to Received L2 Frames
 - 5.4. H.Encaps.L2.Red: H.Encaps.Red Applied to Received L2 Frames
 - 6. Counters
 - 7. Flow-Based Hash Computation

FRR
OAM等

L2L3VPN

Traffic
Engineering

Segment Routing for IPv6 (SRv6) is the implementation of Segment Routing over the IPv6 dataplane.

- Segment Routing over IPv6 Overview
- Configuring SRv6 under IS-IS
- Configuring SRv6 IS-IS Flexible Algorithm
- Configuring SRv6 IS-IS TI-LFA
- Configuring SRv6 IS-IS Microloop Avoidance

- SRv6 Services: IPv4 L3VPN
- SRv6 Services: IPv6 L3VPN
- SRv6 Services: IPv4 L3VPN Active-Standby Redundancy using Port-Active Mode
- SRv6 Services: BGP Global IPv4
- SRv6 Services: BGP Global IPv6
- SRv6 Services: EVPN VPWS – All-Active Multi-Homing
- SRv6 Services: SRv6 Services TLV Type 5 Support

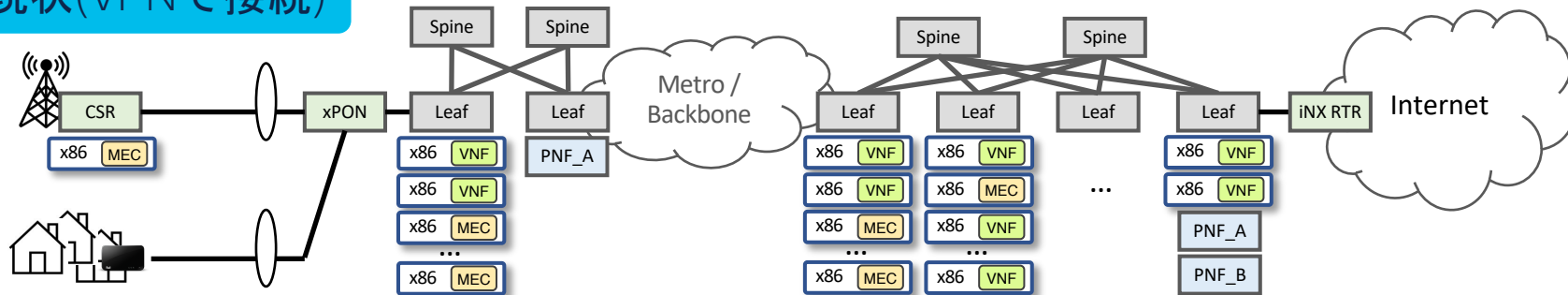
- SRv6/MPLS L3 Service Interworking Gateway
- SRv6/MPLS Dual-Connected PE
- SRv6 SID Information in BGP-LS Reporting
- SRv6 OAM – SID Verification

Cisco IOS-XR Configuration guide

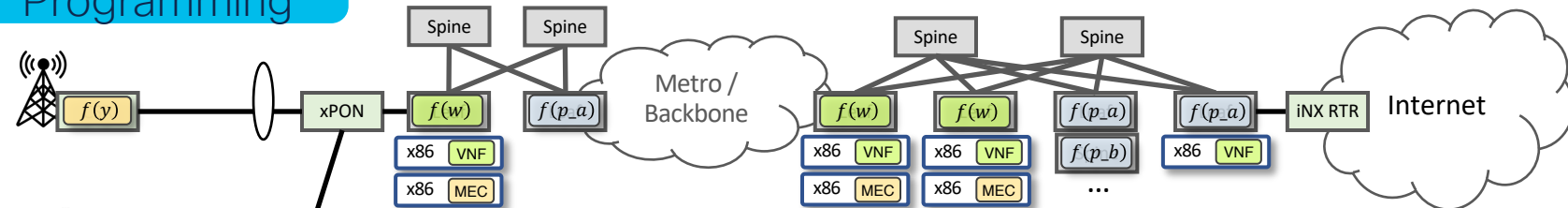
RFC8986 Table of contents

SRv6で本当にやりたかったこと

現状(VPNで接続)

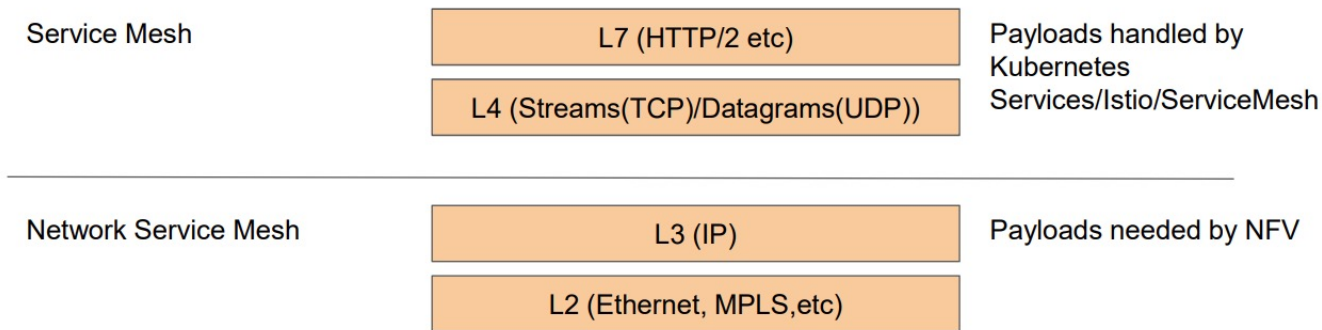


SRv6 Network Programming



In-Network Computing

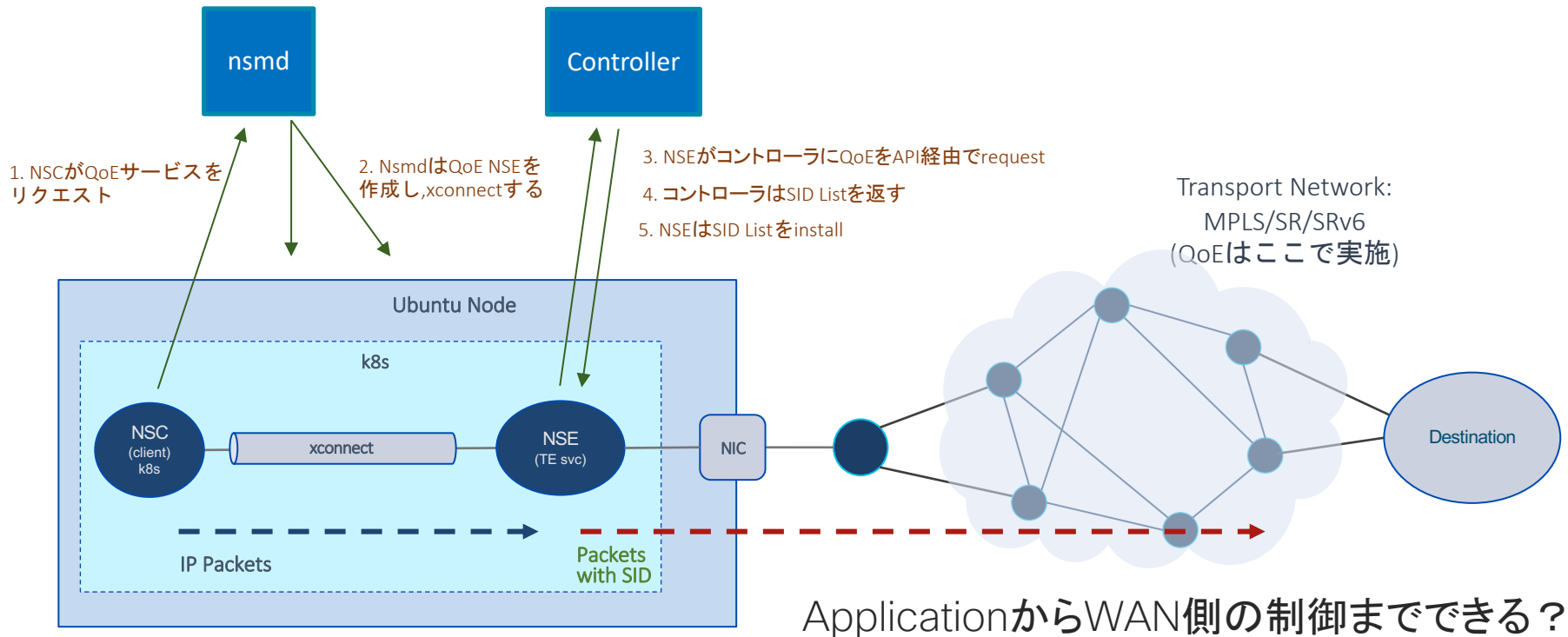
Application側からのアプローチ...NSM?



- Network Service Mesh(NSM)はService MeshのMissing Layerと言われている複雑なL2/L3ユースケースを解決するためのアプローチ
- Istioに触発されたService Meshの概念をL2/L3ペイロードにMappingする
- L2/L3ペイロードにMappingする情報をSRv6を用いてより高価値なものにできないか?
→Cloud nativeな世界だけではなくWANも含めて一緒に制御可能になるはず

NSM x SRv6

先行事例① NSM QoEサービス

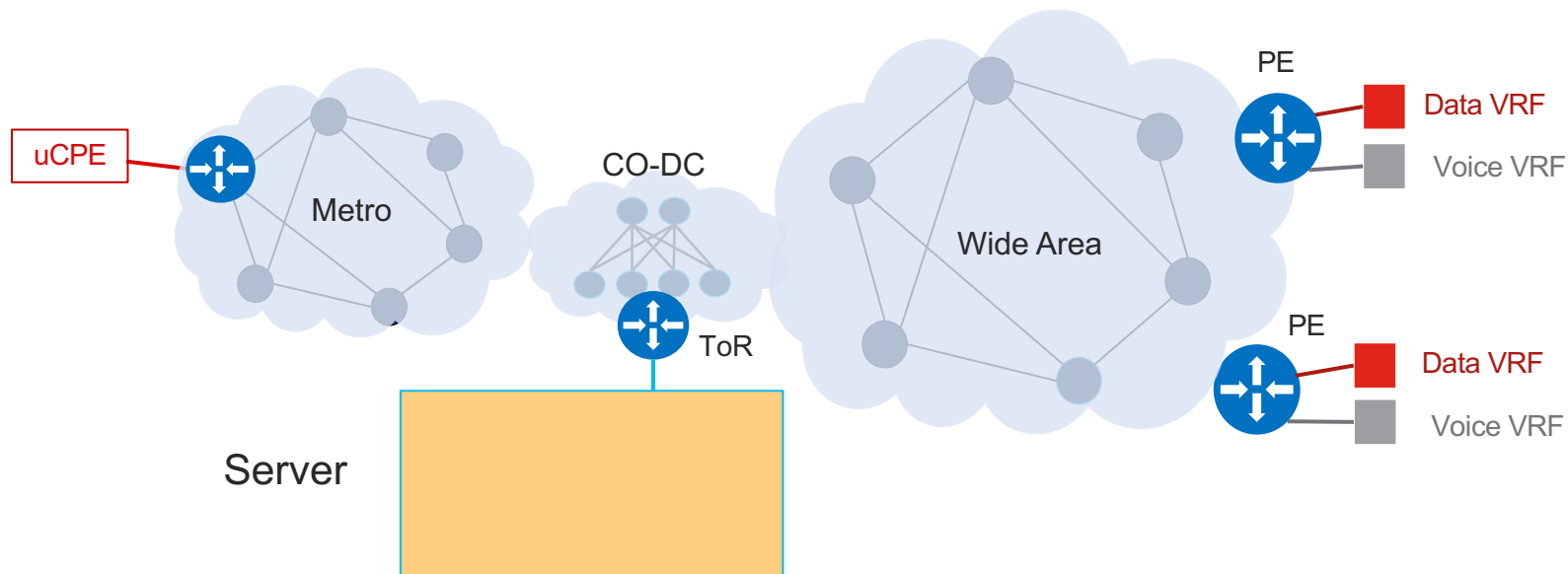


ApplicationからWAN側の制御までできる？

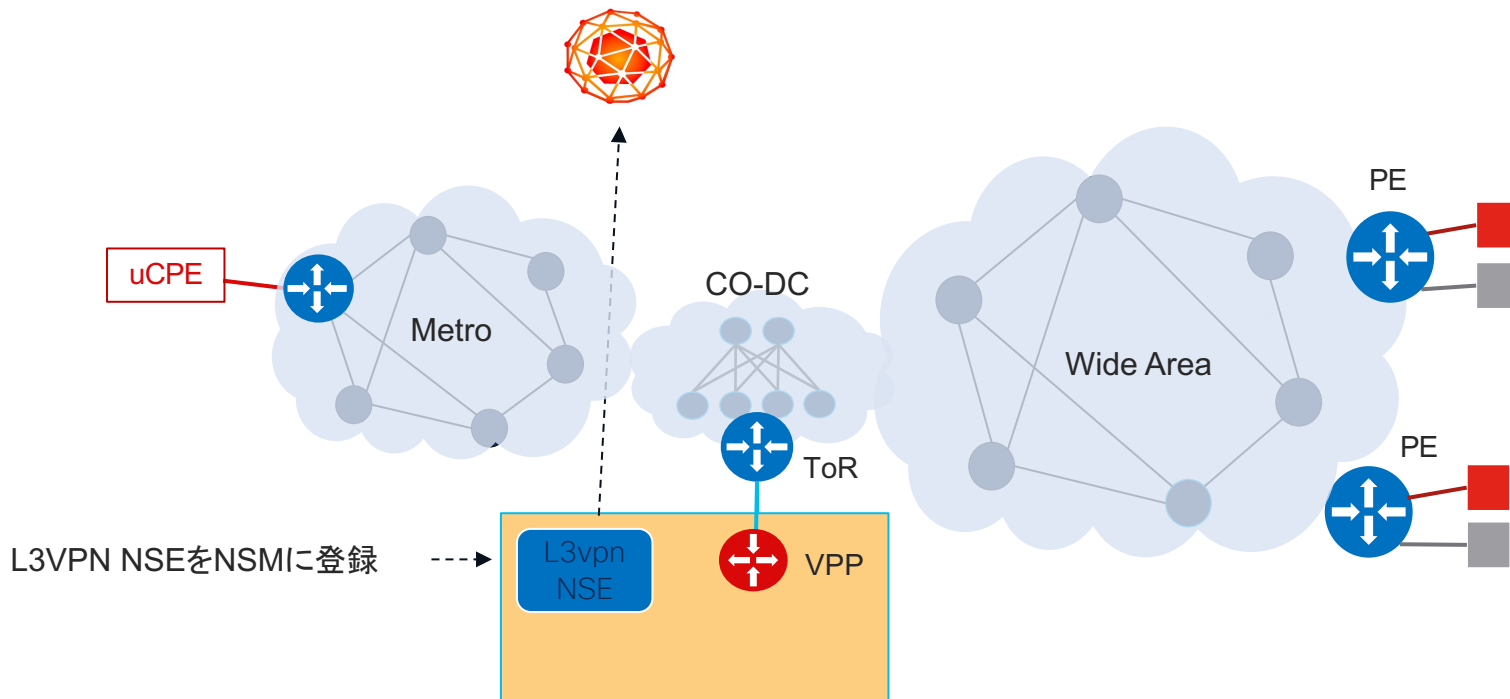
先行事例②

NSM+SRv6 L3VPN

Containerized VNFをSRv6 L3VPN VRFに紐付ける

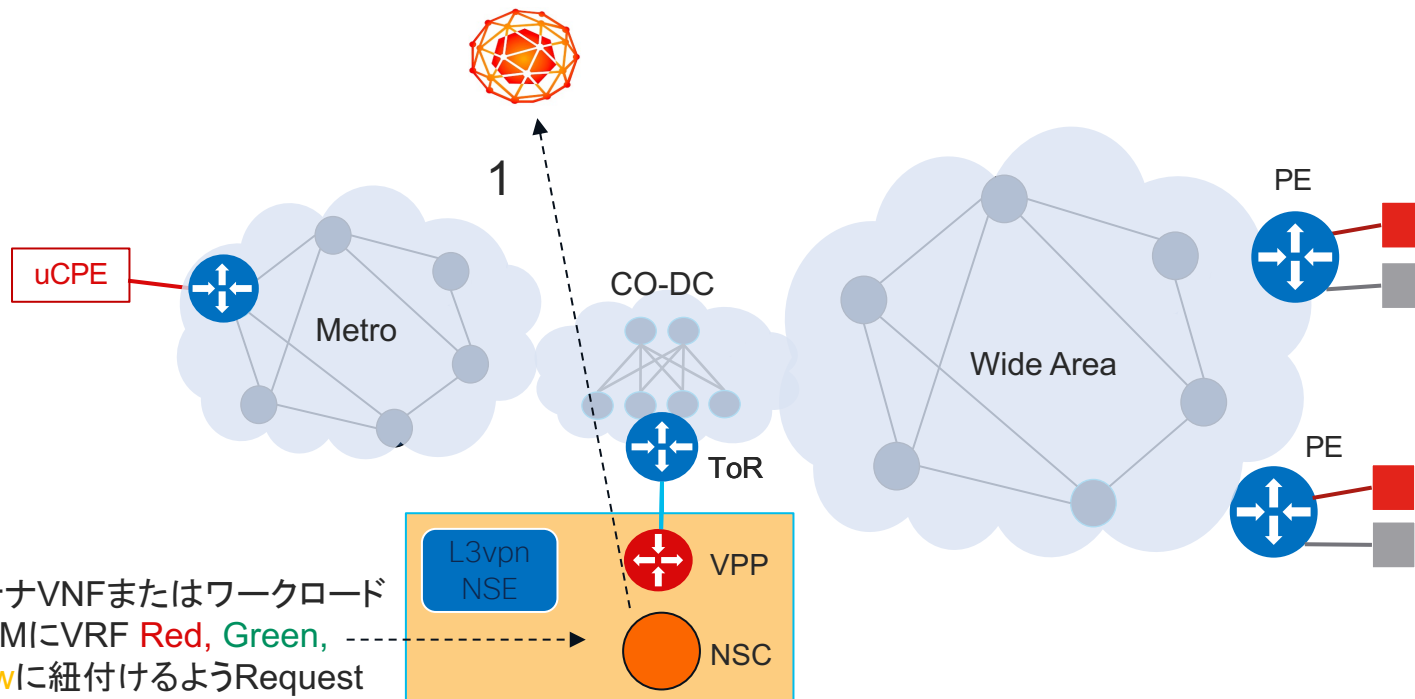


先行事例② NSM+SRv6 L3VPN



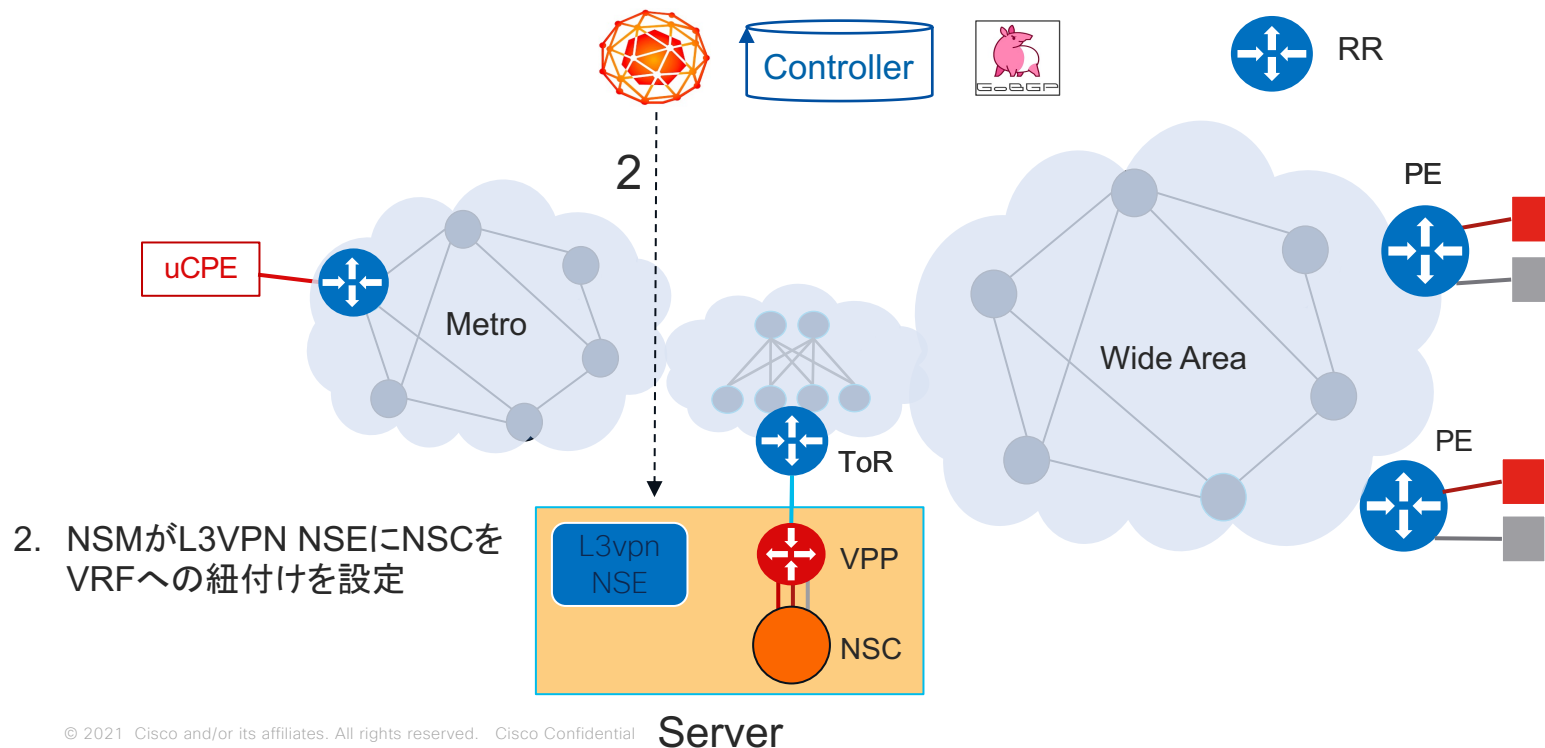
先行事例②

NSM+SRv6 L3VPN



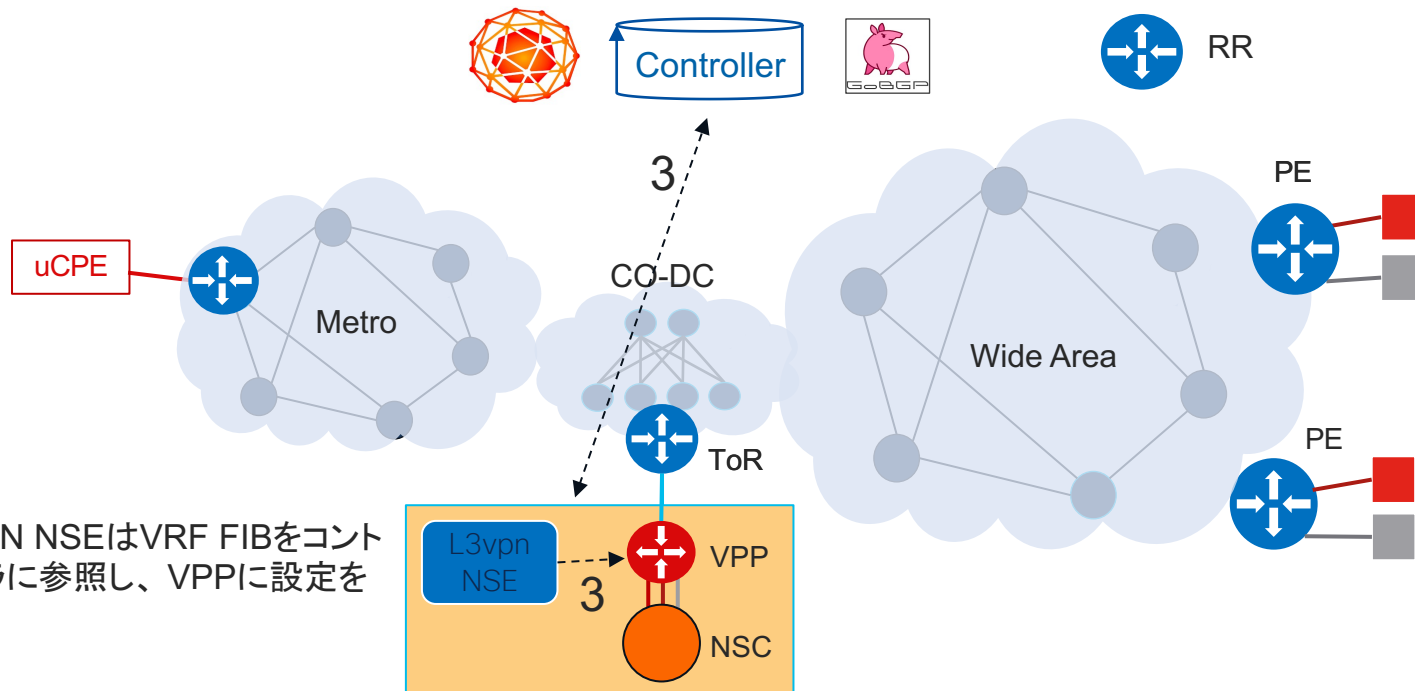
先行事例②

NSM+SRv6 L3VPN



先行事例②

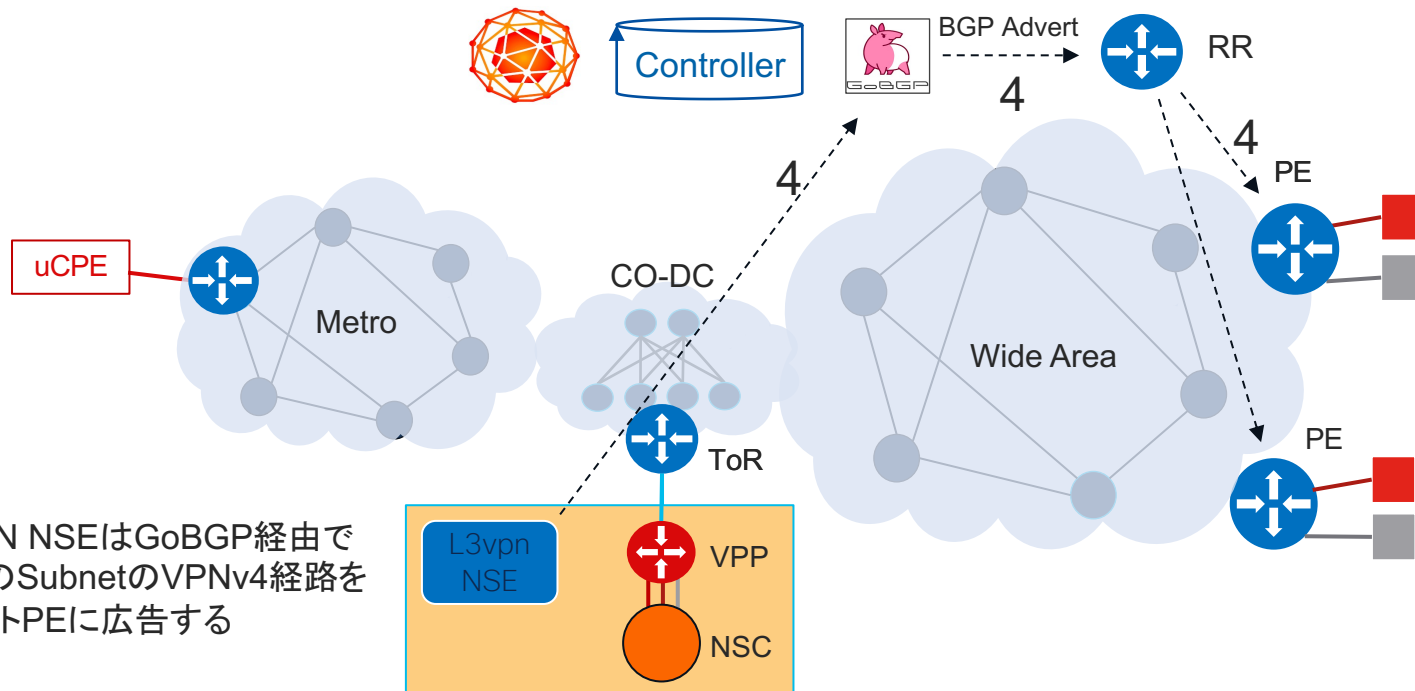
NSM+SRv6 L3VPN



3. L3VPN NSEはVRF FIBをコントローラに参照し、VPPに設定を行う

先行事例②

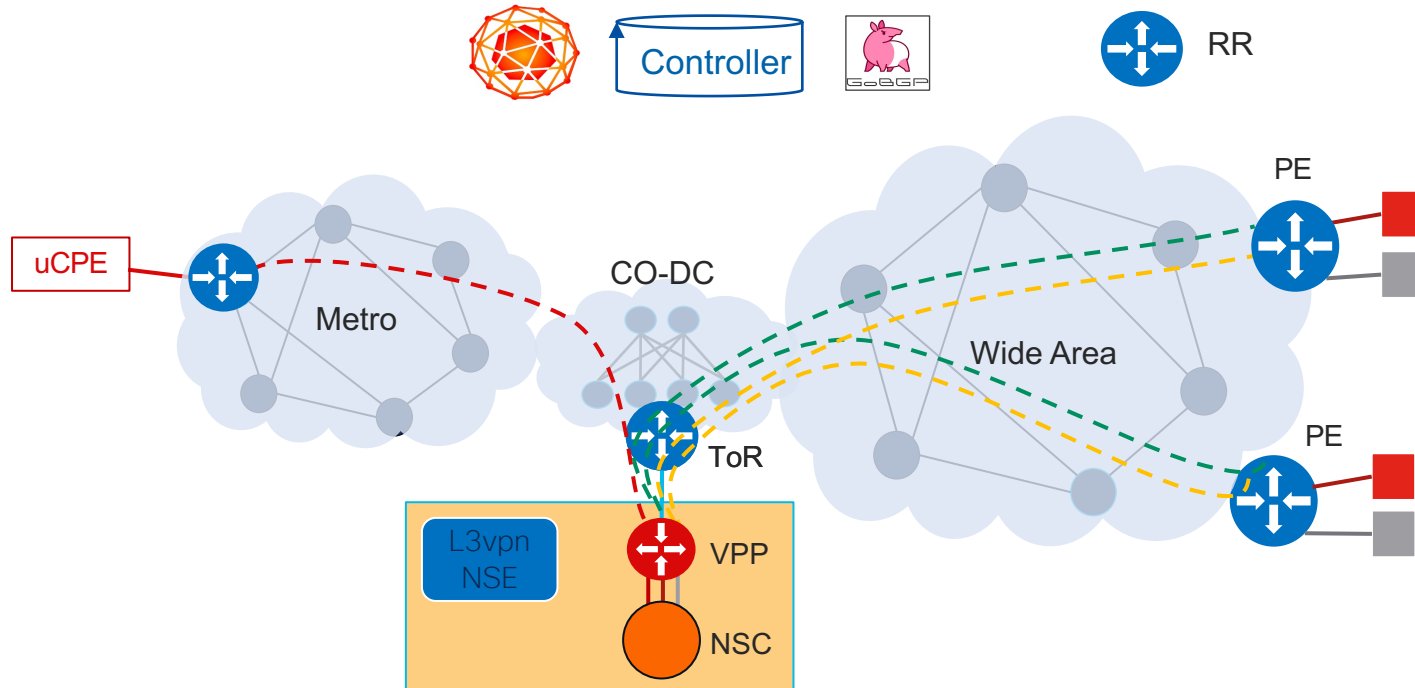
NSM+SRv6 L3VPN



4. L3VPN NSEはGoBGP経由でNSCのSubnetのVPNv4経路をリモートPEに広告する

先行事例②

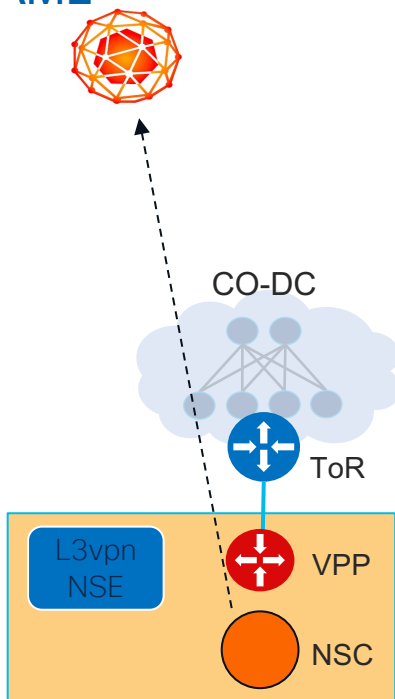
NSM+SRv6 L3VPN



先行事例②

NSM+SRv6 L3VPN

Example YAML



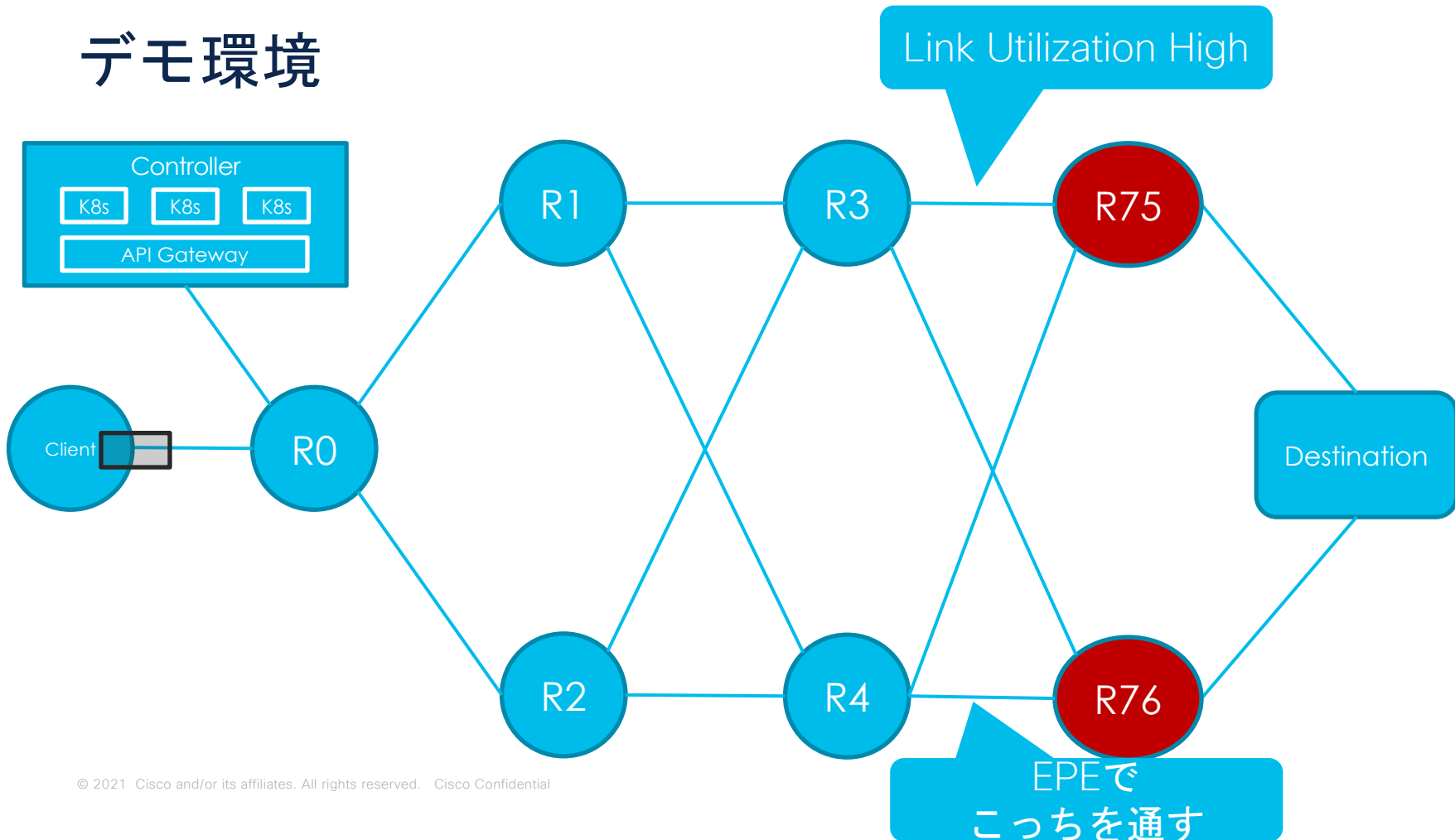
```
apiVersion: apps/v1
kind: Deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: l3vpnnsse-client
  template:
    metadata:
      labels:
        app: l3vpnnsse-client
    spec:
      containers:
        - image: 192.168.80.240:4000/tools/centos-tool-box:latest
          name: debug
          securityContext:
            privileged: true
          imagePullPolicy: Always
          command:
            - /bin/bash
            - -xec
      metadata:
        name: l3vpnnsse-client
        namespace: default
        annotations:
          ns.networkservicemesh.io:
l3vpnnsse/vrf_red?vpn=5070:65033,l3vpnnsse/vrf_yellow?vpn=5070:65044,
l3vpnnsse/vrf_green?vpn=5070:65024
```

NSM x SRv6

- ApplicationがNetwork Serviceを必要とするケースにおいて、NWを認識して制御することは基本的には難しい
- NWを認識して制御するコントローラが必要になる
- コントローラがNWからデータをちゃんと集めることができていると空いているところを見つけて制御なども実施できる
- WAN側の制御はやはりBGP?
NW機器はBGPのAdvertisementを受ければSIDでいくらかでも制御できる

Demonstration

デモ環境



終わりに

- ApplicationがNetwork Serviceを必要とするケースにおいて、NWを認識して制御することは基本的には難しい
- Networkを抽象化してUser/Applicationから制御するにはDataを集めてくれるコントローラは必要で、NSMと親和性が高いとうれしそう
- DataplaneがSRv6であればNetwork serviceをそのままSIDにEncodeできる→In network computingに近づくとと思います



The bridge to possible