



IETF106報告会

SR 最新動向

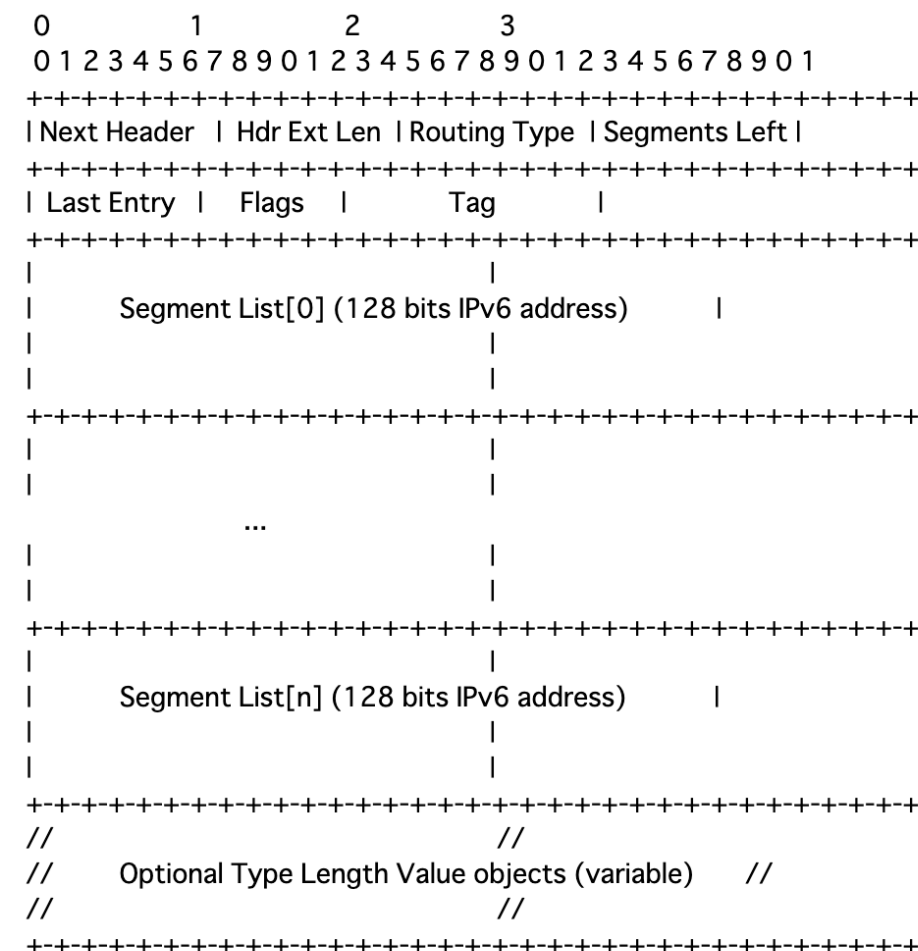
Teppei Kamata
Cisco Systems G.K.
9th Jan 2020

本日の発表内容

- Spring WGを中心としたSRの最新動向
 - SRv6の進展
 - SR Control Planeなど
 - uSID/CRHなど
- SRv6 Service Chainingに関連してSpring WGに提案したI-D 2本の紹介
 - draft-upa-srv6-service-chaining-exp
 - draft-eden-srv6-tagging-proxy

Spring WGを中心としたSRの最新動向

- まず最初に(いきなりSpringではないですが)
 - SRv6のDataplaneを定義するSRHが
Proposed Standardとなりました
draft-ietf-6man-segment-routing-header
 - 最初にDraftが提出されたのが2014年なので
5年越し
- 最終的なSRH形式は右のように



Spring WGを中心としたSRの最新動向

- **以下のDraftに関してもLast Call/Proposed Standardへと進捗中**
 - SRv6 Network Programming
draft-ietf-spring-srv6-network-programming
 - Control Plane(IS-IS, BGP)
draft-ietf-lsr-isis-srv6-extensions
draft-ietf-idr-bgp-ls-segment-routing-ext
 - SR Policy
draft-ietf-idr-segment-routing-te-policy
 - OAM
draft-ietf-6man-spring-srv6-oam
 - YANG
draft-ietf-spring-sr-yang
- **SR-MPLSについても主要なDocumentがRFCとなるなど標準化は最終段階へ**
 - RFC 8660 Segment Routing with the MPLS Data Plane
 - RFC 8661 Segment Routing MPLS Interworking with LDP

Spring WGを中心としたSRの最新動向

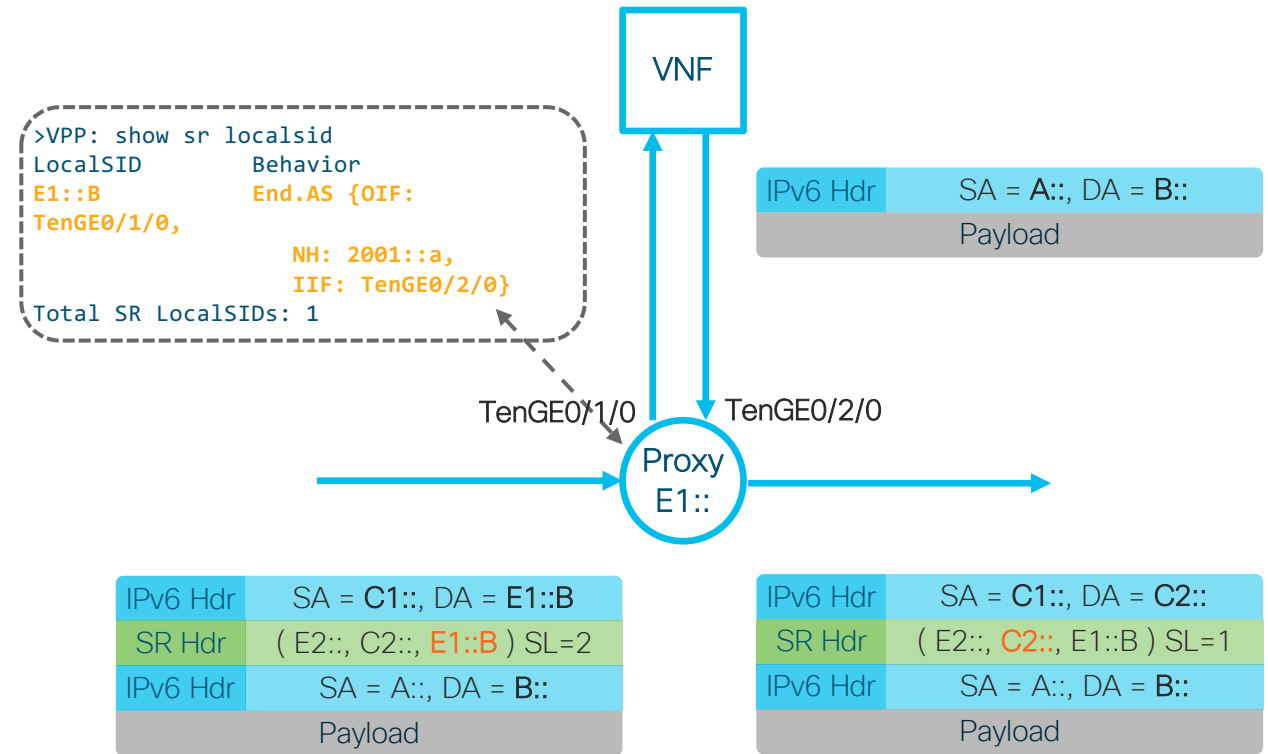
- IETF105の報告会で話題になったuSID/CRHについて
 - uSIDについてはIETF106では目立ったアップデートは特になし
 - CRHについてはSRv6とは別提案という扱いになり、SRv6+からSRm6(Segment Routing Mapped IPv6)へとRenameすることになった
draft-bonica-spring-srv6-plus
→draft-bonica-spring-sr-mapped-six

SRv6でService ChainingするFunction

- Service Chaining系のFunctionをまとめたService programmingも今回WGドラフトへ
 - draft-ietf-spring-sr-service-programming
- 定義されているのは以下のFunction
 - SR Aware Function (End.AN)
 - Static Proxy (End.AS)
 - Dynamic Proxy (End.AD)
 - Shared-memory Proxy (NEW!!)
 - Masquerading Proxy (End.AM)

End.AS – Endpoint to SR-unaware app via static proxy

- Ingress:
 - Active SIDは"E1::B" (Function 0xBは End.AS)
 - 外側のIPヘッダー (SRヘッダー含む) を Popする
 - End.ASに関連付けられたVNF (OIF, NH) へ転送する
- Egress:
 - IPヘッダー (SRヘッダー含む) をPushする (Static)
 - Next Segmentへ転送する
- **Valid for IPv4 and IPv6 traffic**
- **Per-chain static configuration**



End.AD – Endpoint to SR-unaware app via dynamic proxy

- Ingress:

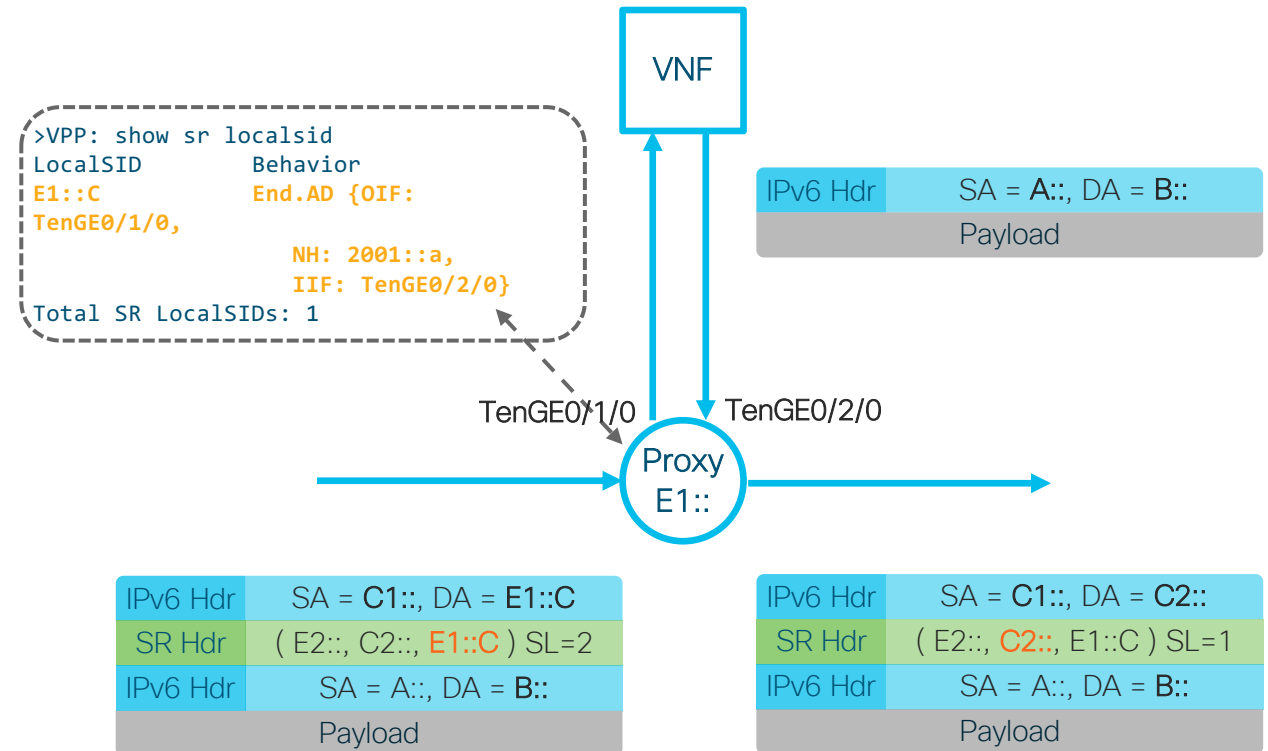
- Active SIDは"E1::C" (Function 0xCはEnd.AD)
- 外側のIPヘッダー (SRヘッダー含む) をCashしてPopする
- End.ADに関連付けられたVNF (OIF, NH) へ転送する

- Egress:

- IPヘッダー (SRヘッダー含む) をCashからPushする (Dynamic)
- Next Segmentへ転送する

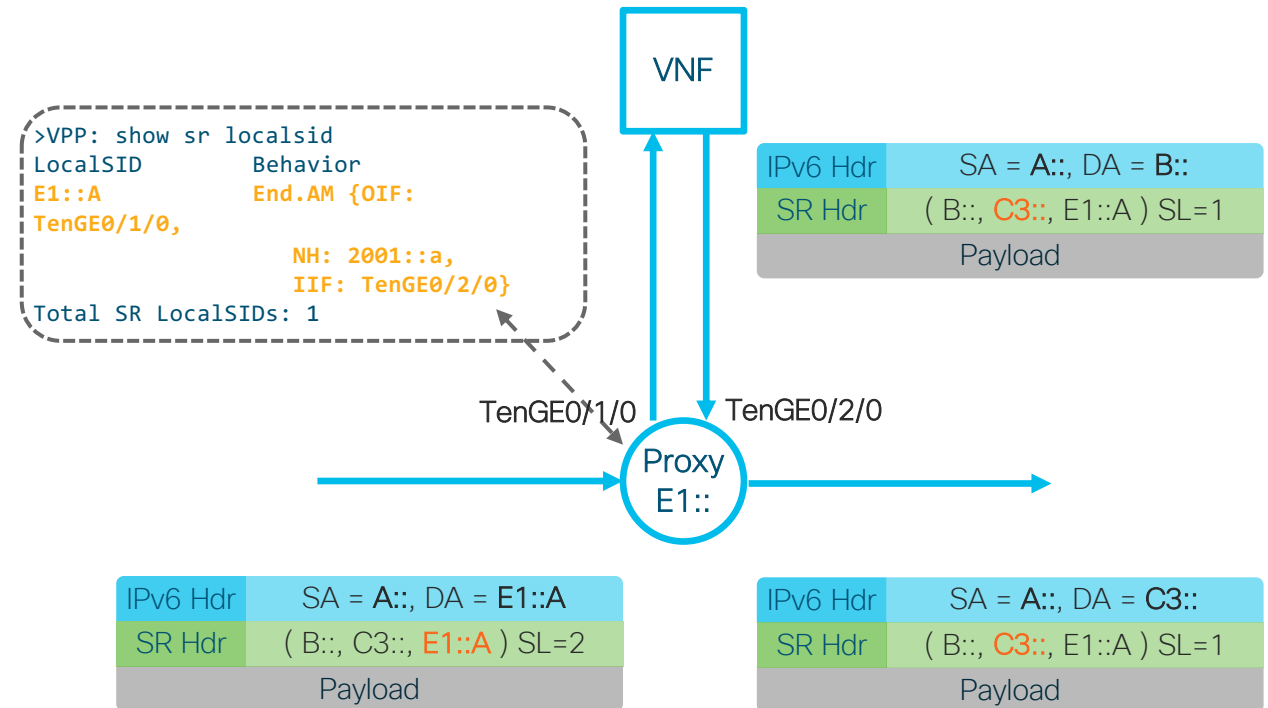
- **Valid for IPv4 and IPv6 traffic**

- **Cashは1Proxyにつき1枠**



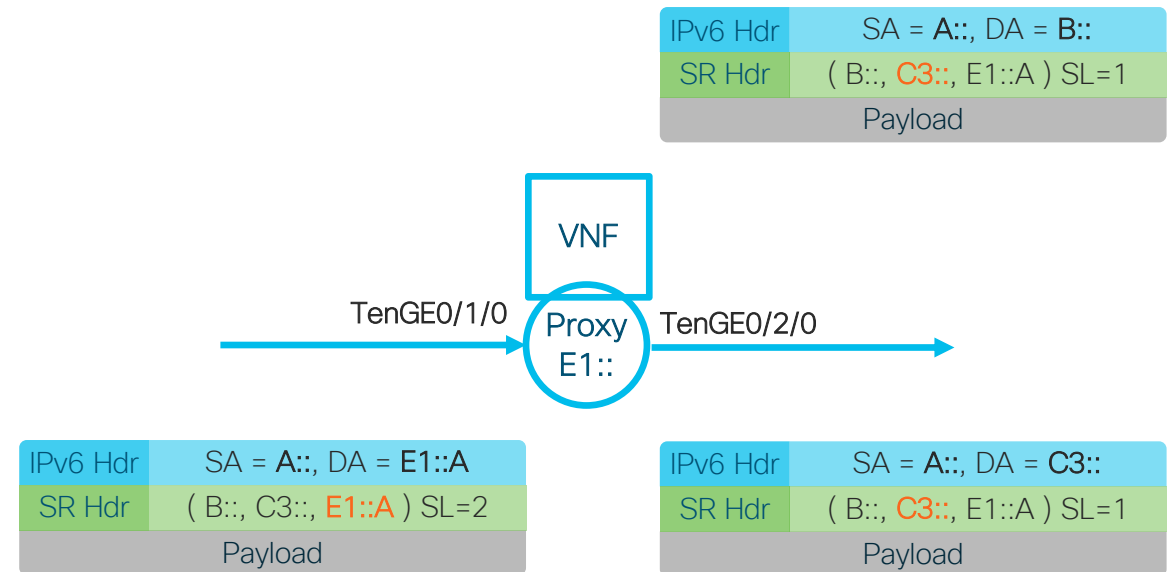
End.AM – Endpoint to SR-unaware app via masquerading

- Ingress:
 - Active SIDは"E1::A" (Function 0xAはEnd.AM)
 - 宛先をLast Segment B::に書き換える
 - End.AMに関連付けられたVNF (OIF, NH)へ転送する
- Egress:
 - SRHを確認し、宛先をActive Segment C3::に書き換える
- **Valid for IPv6 traffic only**
 - T.Insert方式のTrafficにしか適用できない
 - IPv4はT.Encapせざるを得ない
- **StateはPacketが保持したままなのでChainの重畳が可能**



Shared-memory Proxy

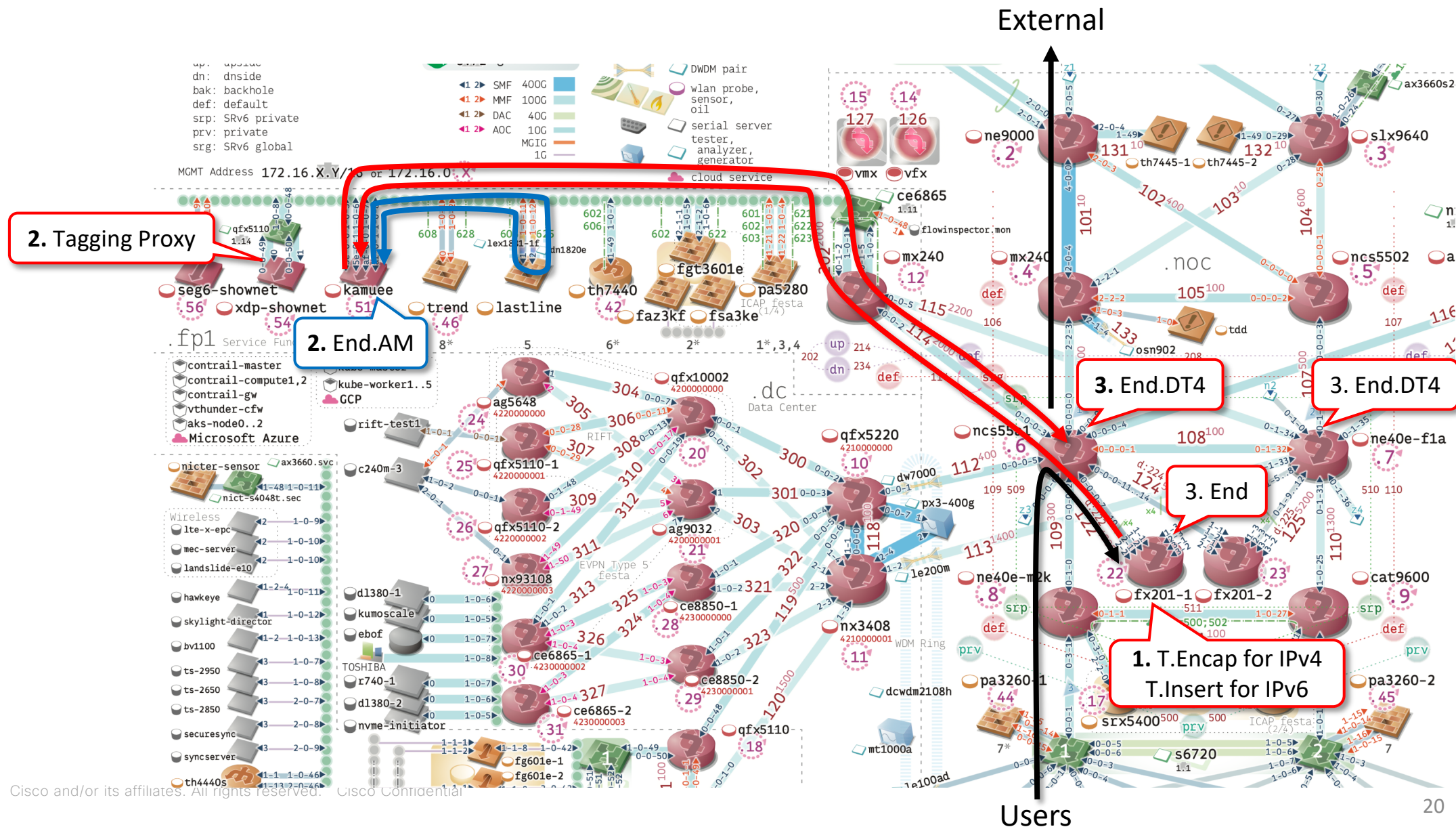
- VNFとProxyが同一Compute Node上で動作
- Shared-memory IFを用いてVNFに対してSRHを用いたままサービス提供する
- **More details will be added in a future revision of this document.**
- **Valid for IPv4 and IPv6 traffic?**
- **StateはPacketが保持したままなのでChainの重畳が可能?**
- **まだ実装方式も明らかではないので、今後要トラック**



Spring WGに提案したI-D 2本の紹介

- ShowNet2019でSRv6 Service Chainingを実施
 - End.ANのFunctionは一つだけしか実装がなかった
 - IPv6のTrafficはEnd.AMでProxy可能
 - IPv4に対して現実的に用いることができるProxy方式が存在していなかった
- 結論としてIPv4に対する複数Chainを重ねできるProxy方式End.ATを検討して実装したので、I-Dを出してみました。
 - <https://tools.ietf.org/html/draft-eden-srv6-tagging-proxy>
- せっかく ShowNetという大規模環境でSRv6 Service Chainingを実装したので知見を残そうと Experimental draft も出してみました。
 - <https://tools.ietf.org/html/draft-upa-srv6-service-chaining-exp>

ShowNet 2019でのSRv6 Service Chaining



draft-upa-srv6-service-chaining-exp

- このExperimental draftでは以下の知見を共有しました
 - Transparency of SRv6 header
 - SRHがついていてもVNFは処理可能
 - Services that cannot co-exist with End.AM
 - 自発パケットを生成するFunctionと一部Proxyは相性が悪い
 - Service liveness detection and conditional advertisement of service segments
 - Proxyの裏にいるFunctionの正常性確認の難しさ
(SRの世界ではIGPを使いたいですがVNFに求めるのは難しい)
 - Control Plane Capabilities (Control Planeは必須です。。。)
 - Match Condition for Applying SRv6 Functions
 - 今回はVRFを切ってPBRで引き込みを実施
 - SR PolicyはColor(Destination)ベースでTraffic Steering可能ですが、戻りパケットを考えると5TupleでSteeringが必要

