

SRv6@ShowNetとプログラマビリティ

東京大学/ShowNet NOCチームメンバー
中村 遼

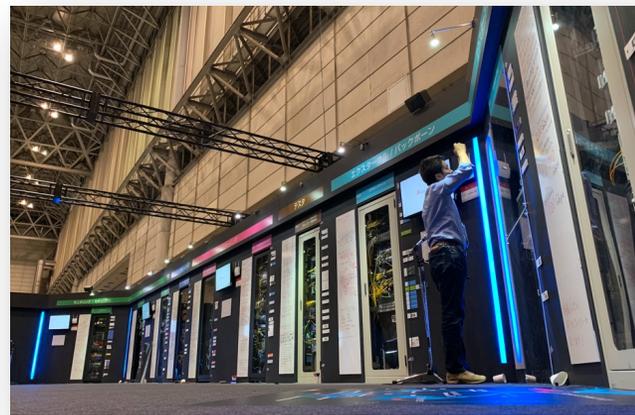
Open Networking Conference 2022

自己紹介

- 東京大学
 - 情報基盤センター ネットワーク研究部門 准教授
 - ネットワークアーキテクチャ、OSネットワークサブシステム
 - 東京大学キャンパスネットワークの運用
- Interop Tokyo ShowNet NOCチームメンバー
 - 主にL2/L3を担当
 - SDNとかもやっていました

Interop Tokyo **SHOWNET** Over the Premise

- **Interop:** 世界最大のネットワーク機器と技術の展示会
 - 毎年6月に幕張メッセで開催、2022年は来場者約9万人
- **ShowNet:** 世界最大のデモンストレーションネットワーク
 - その年々の最新技術/プロトコルの相互接続性検証やデモを実施
 - 出展社や来場者へのネットワーク接続性を提供するLive Network



Device ID: X
Loopback: 45.0.VPN_ID.X/32
Address: 2001:3a8::VPN_ID.X/128
P2P Link ID: X

VLAN ID: VPN_ID x 100 + Link_ID
VPN ID 0 1
V4: 45.0.(100 + VPN_ID).(Link_ID x 4/30
V6: 2001:3a8:0:(VLAN_ID)::/64
VPN ID 2 0
V4: 10.0.(100 + VPN_ID).(Link_ID x 4/30
V6: 2001:3a8:0:(VLAN_ID)::/64
Mgmt Address
172.16.X.Y/16 or 172.16.0.X

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

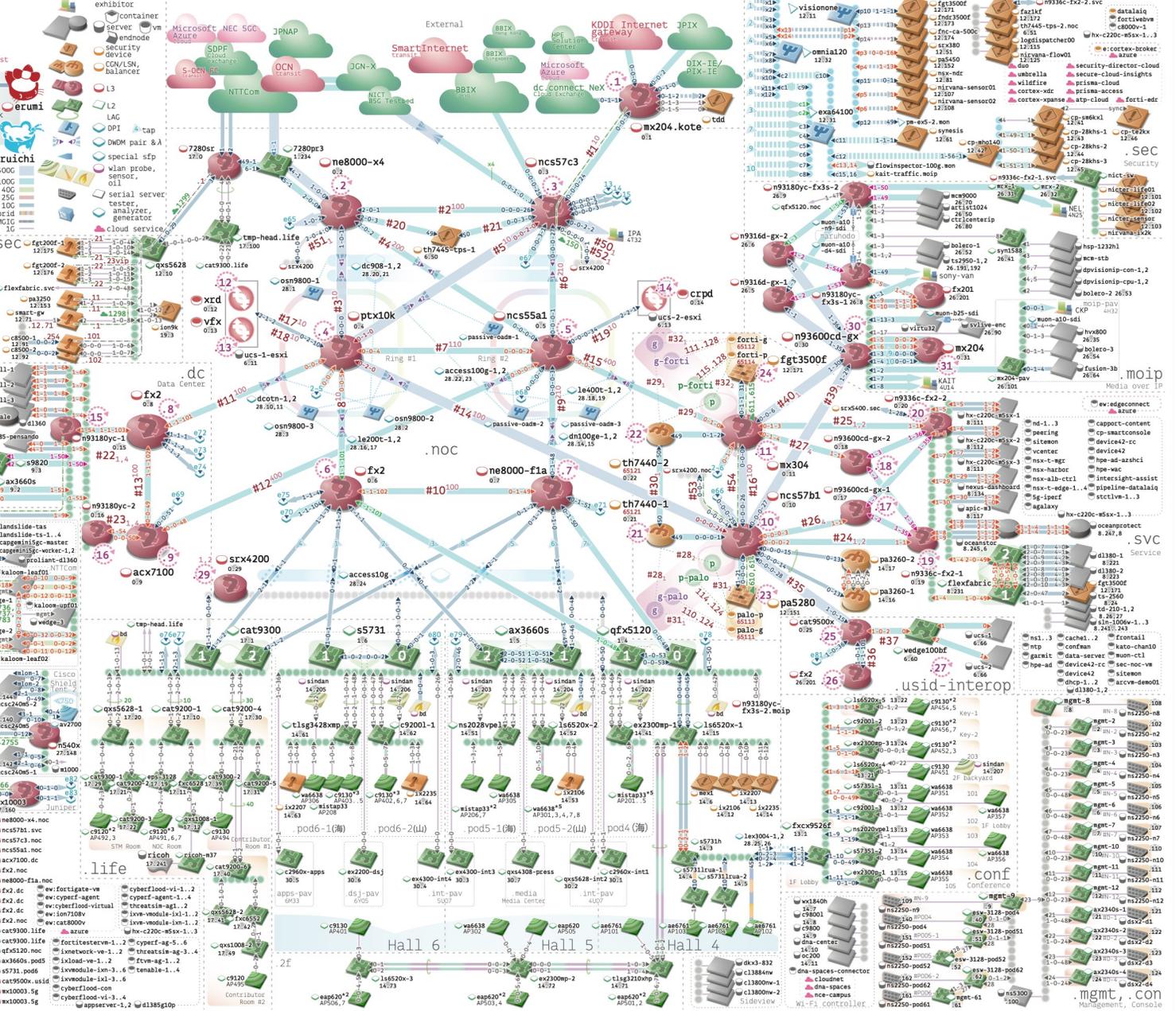
FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer

FlowInspector
FlowCollector
Zabbix
Nessus
GNS3
Vagrant
Docker
Kubernetes
Istio
Prometheus
Grafana
ELK
Ansible
Terraform
Packer



ShowNetのアイコンは
Interop Tokyoの
ShowNet 2022のページから
フリー (CC-BY-SA 4.0)で
ダウンロードできます！



<https://www.interop.jp/2022/shownet/concept/>
「ShowNetアイコン」の画像をクリック

ShowNetにおけるSegment Routingの取り組み

- 2019年: SRv6 Service Chaining
 - 6機種5ベンダーのSRv6機器で構築
 - draft-upa-srv6-service-chaining-exp, draft-eden-srv6-tagging-proxy

現時点で実現可能なアーキテクチャ

- SRv6ヘッダを解釈できるService Functionは限られているため、SRv6 ProxyによってSRv6の世界と普通のIPv4/IPv6の世界を橋渡しする必要がある

(図は上りトラフィックの場合)
 fx201-[1,2].noc (Furukawa Networks FX201), kamuee.fp1 (NTT Communications kamuee), ne40e-x (Huawei NE4),
 fx201.fp2 (Furukawa Networks FX201), xdp-sho (ShowNet NG), vpp.fp2 (Cisco/Infra VPP), seg6-sh (ShowNet NG)



Function Poolへの誘導

- Function Poolから広告されるSIDによってトラフィックを誘導

ShowNet 2019 Backbone

- SRv6によるサービスチェイニング
 - IP RoutingにSegment Routingによる柔軟性の追加
 - SR-ProxyによるSR-unawareファンクションへの対応

サービスチェインの始点

へ通すためのSRv6ヘッダを付加する

- SRv6には2つのモード(ヘッダの付加方式)があり、ShowNet 2019ではアドレスタイプに応じて使い分け
 - IPv6の場合はInsert mode
 - IPv4の場合はEncap mode
 - 決定要因はSRv6 Proxyの方式

ShowNetにおけるSegment Routingの取り組み

- 2021年: SR-MPLS/SRv6バックボーン
 - SR-MPLS: Flex-Algo, BGP Egress Peer Engineering
 - SRv6: draft-ietf-bess-srv6-servicesベースの相互接続検証

APNIC

Get IP - Manage IP - Training - Events - Insights - Community - Blog - Help Centre - About - Contact

Measuring the potential benefit of egress traffic engineering with Segment Routing

By Ryo Nakamura on 10 Mar 2022

Category: Tech matters

Tags: BGP, Guest Post, Japan, measurement, segment routing

Border Gateway Protocol (BGP) best paths on which the Internet relies are not always the 'best' when it comes to bandwidth and latency. This is due to several factors including BGP not considering congestion on links and AS-PATHs using prepending no longer reflect closeness.

As such, engineering egress traffic toward neighboring Autonomous Systems (ASes) is a constant challenge for network operators seeking better paths.

To begin with, the potential benefit of egress traffic engineering is unclear. Measuring the performance of a non-best path to a destination requires preferring the non-best path. However, using local preference to do this impacts all user traffic to the

SR-MPLS Egress Peer Engineering

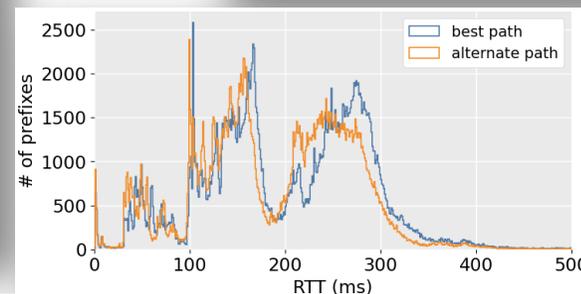
- パケットをLongest Prefix Matchによらずに狙ったピアへ
- draft-ietf-spring-segment-routing-central-edge
- ASBRがBGP PeerごとにSRのセグメント(ラベル)をアサイン
 - BGP Peering Segments
 - そのセグメント行きのパケットは、そのピアへ送信される
 - これを利用して送信方向のトラフィックを制御できる

Copyright © Interop Tokyo 2021 ShowNet NOC Team

SR-MPLS網とSRv6網の相互接続

- インターワークゲートウェイにて、SR-MPLSのラベルとSRv6のSIDを相互変換

Copyright © Interop Tokyo 2021 ShowNet NOC Team



SRv6 SID in ShowNet2021

- SRv6 SID Design: 2001:3e8:6:deviceID:function::/64

```

RP/0/RSP/CR# show ip bgp summary segment-routing srv6 sid
Tue Apr 13 14:24:37.475: %
*** Location: "ShowNet" ***

```

Prefix	Neighbor	Contact	Status	RW
2001:3e8:6:1:1	End (PSP)	defaultr11	Stale	Y
2001:3e8:6:1:1:1	End (OP)	defaultr11	Stale	Y
2001:3e8:6:1:1:1:1	End (X (PSP))	[100/100/212, Link-Local]	Stale	Y
2001:3e8:6:1:1:1:1:1	End (X (PSP))	[100/100/212, Link-Local]	Stale	Y
2001:3e8:6:1:1:1:1:1:1	End (X (PSP))	vr6-gigabit	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1	End (X (PSP))	[100/100/214, Link-Local]	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1	End (X (PSP))	[100/100/214, Link-Local]	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1:1	End (OP)	vr6-gigabit	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1:1:1	End (OP)	vr6-private1	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1:1:1:1	End (OP)	vr6-private2	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1:1:1:1:1	End (OP)	vr6-private3	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1:1:1:1:1:1	End (OP)	vr6-super-cc	Stale	Y
2001:3e8:6:1:1:1:1:1:1:1:1:1:1:1:1:1:1	End (OP)	vr6-super-cc-srv6	Stale	Y

Copyright © Interop Tokyo 2021 ShowNet NOC Team

課題が多かったのはBGP... draft-ietf-bess-srv6-services

- BGP Prefix-SID TLVにてSIDの広告を行うが、Draft上変更が非常に多い
- もともと使っていたType-4 TLVはすでにdeprecated
- Reference RFCが5549から8950に変更
- SRv6 SID Structure Sub-Sub-TLV という新しいTLVが登場、これ以前のインオペが取れず
- Structure Sub-Sub-TLVに対応していても解釈の違い
- 結果として、全ベンダがHotstage中にVersion up
- 無事にサービスの疎通を確認

Copyright © Interop Tokyo 2021 ShowNet NOC Team

そして2022年

- ShowNetバックボーン全体がフルSRv6化

- 4ベンダーから9機種のSRv6対応ルータと3種の仮想RR

- Cisco、Huawei、Juniper、古河電工

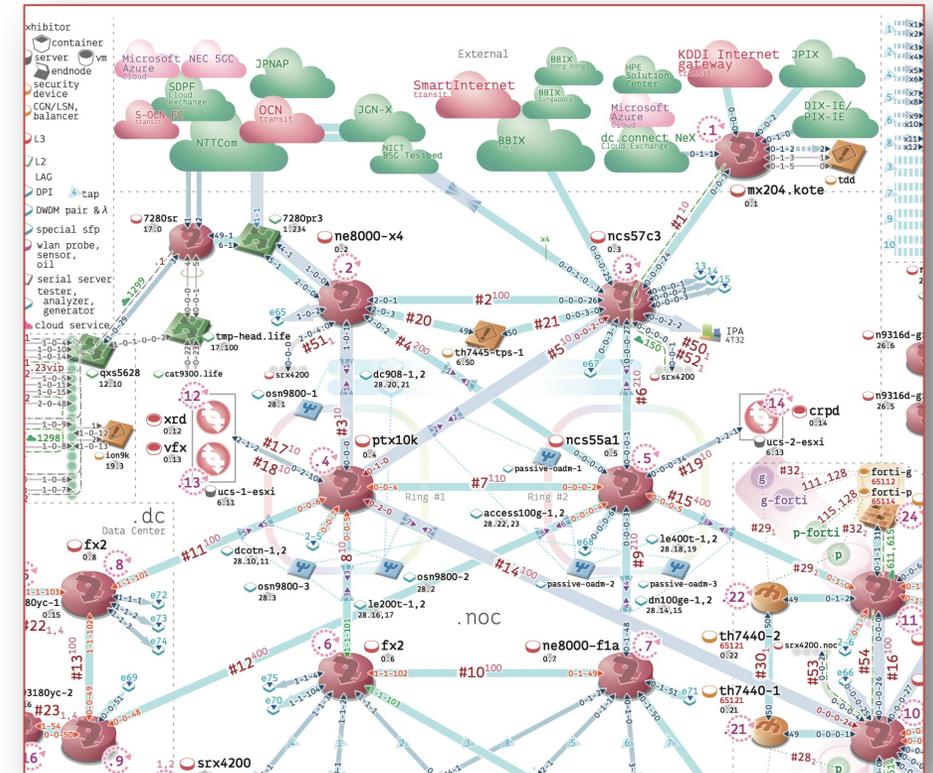
- BackboneはIPv6シングルスタック

- トラフィックは全てover SRv6で転送

- 相互接続検証および実験

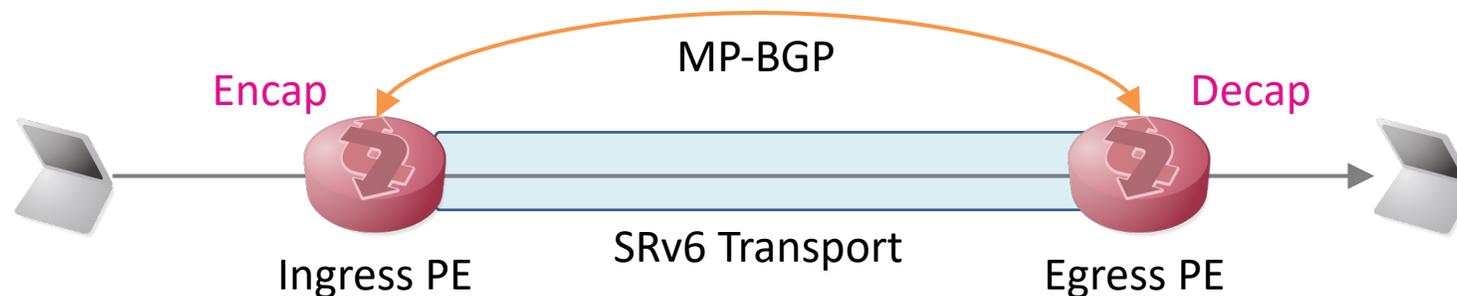
- SRv6 L3VPN、SRv6 Flex Algo、uSID

- SRv6による長距離映像伝送実験



SRv6 Layer-3 VPN

- SRv6 トランスポートのL3VPNという基本的なユースケース
 1. Ingress PEがH.EncapsでCEからのパケットをカプセル化
 2. PはただのIPv6パケットとして転送 or Traffic Engineering
 3. Egress PEはEnd.DT4/DT6でカプセル化を解除し転送
- MP-BGPによるPE間のL3VPNシグナリング
 - PE間でお互いの経路とそれに紐づくSIDを交換する
 - RFC9252で2022年7月に標準化完了(Interop Tokyo 2022よりも後)



SRv6 L3VPN 相互接続

- SIDと紐づくFunctionによってL3VPN/L2VPNが構築される
 - ルータ同士がBGPでSIDを交換しSRv6オーバーレイを構築
 - H.EncapsはPE内で完結、End.DTのSIDを対向PEへBGPで広告

```
RP/0/RP1/CPU0:ncs57c3.noc#show segment-routing srv6 sid  
Wed Jun 15 08:39:24.999 JST
```

```
*** Locator: 'shownet' ***
```

2001:3e8:fa00:3:1::	End (PSP/USD)	'default':1	sidmgr	InUse	Y
2001:3e8:fa00:3:40::	End.X (PSP/USD)	[Hu0/0/0/27.6, Link-Local]	isis-290	InUse	Y
2001:3e8:fa00:3:41::	End.X (PSP/USD)	[Hu0/0/0/26.2, Link-Local]	isis-290	InUse	Y
2001:3e8:fa00:3:42::	End.X (PSP/USD)	[Hu0/0/0/24.1, Link-Local]	isis-290	InUse	Y
2001:3e8:fa00:3:43::	End.DT6	'vrf-global'	bgp-290	InUse	Y
2001:3e8:fa00:3:44::	End.DT4	'vrf-global'	bgp-290	InUse	Y
2001:3e8:fa00:3:45::	End.DX2	310:310	l2vpn_srv6	InUse	Y
2001:3e8:fa00:3:46::	End.X (PSP/USD)	[FH0/0/2/0.5, Link-Local]	isis-290	InUse	Y

IPv6 Link Localアドレスの利用

- バックボーンの間はIPv6 Link Localアドレスのみ
 - Link Localアドレスは自動生成される
 - ルータ間のリンクにはアドレスを設定する必要はない
 - descriptionとmtu、あとは機器によってipv6 enableやfamily isoなど
 - IGPで交換したLoopbackのGlobal UnicastアドレスでRRとBGP

```
interface FourHundredGigE0/0/0/24.14
description fhg-0-2-0.ptx10k.noc
mtu 9021
ipv6 enable
```

SRv6でのFlex Algo

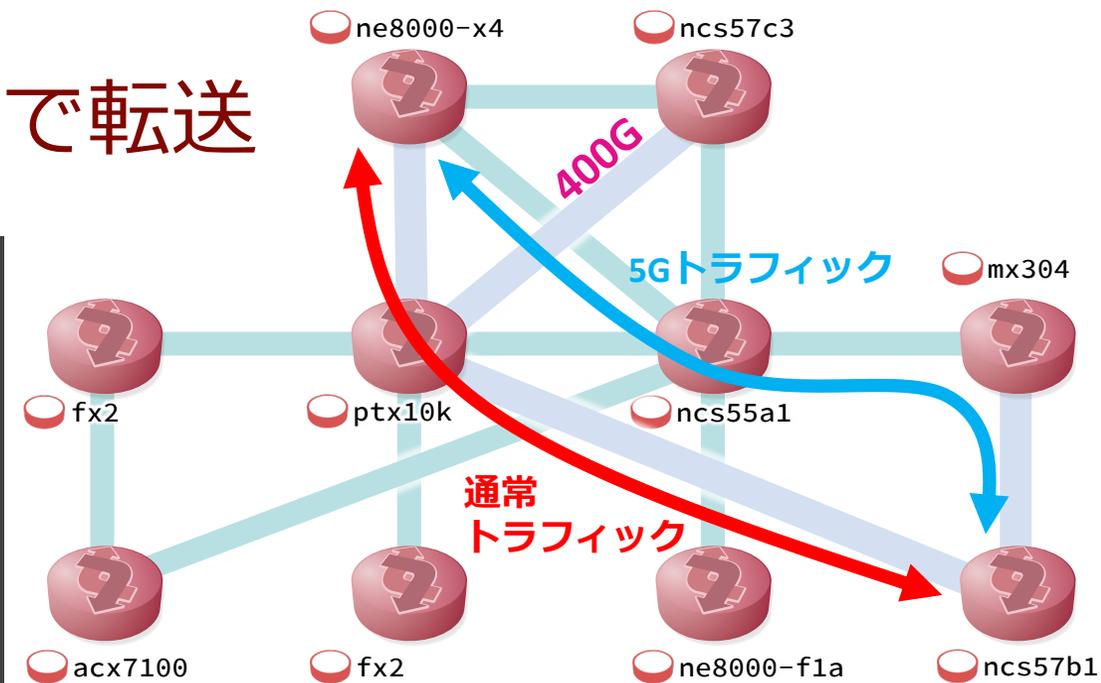
- SRv6ではAlgoごとにLocatorをIGPに広告
 - 広告したAlgo(面)のSPF計算に含まれる
- Flex Algoで5Gのトラフィックを通常トラフィックとは別のパスで転送



通常トラフィックが通っていたリンクは400G-ZR+

```
mx304.svc# show protocols isis source-packet-routing
flex-algorithm [ 128 129 130 ];
srv6 {
  locator loc-mx304;
  locator loc-128;
  locator loc-129;
  locator loc-130;
}
```

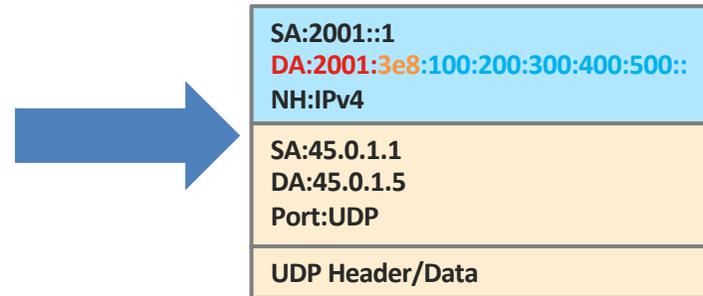
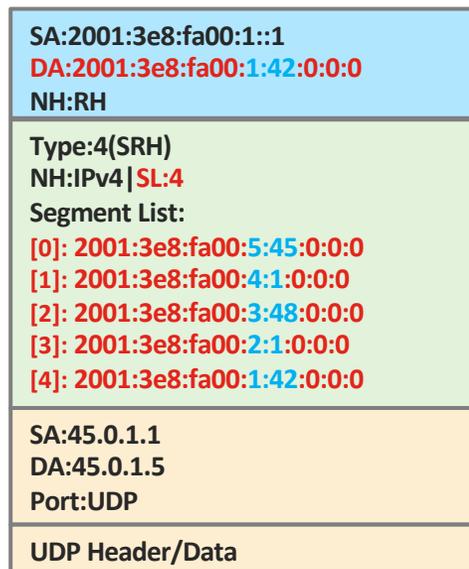
それぞれ別のIPv6 Prefix (Locator)



SRv6 uSID Interoperability Test

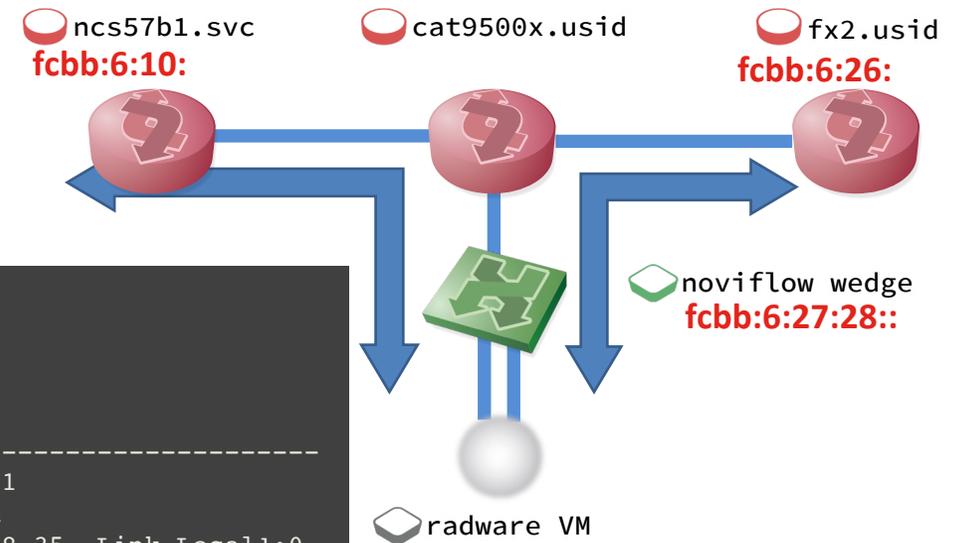
draft-filsfils-spring-net-pgm-extension-srv6-usid

- SRv6 uSIDの相互接続試験を実施
 - NCS57b1とFX2間でF3216 formatのuSIDをISIS/BGPにて広告、疎通を確認
 - ExplicitにPath制御することでNoviflow社SWのuSIDなService Chaining Function(End.SC)の通せることを確認



```
RP/0/RP0/CPU0:ncs57b1.svc#show segment-routing srv6 sid
(snip)
*** Locator: 'usid' ***
```

SID	Behavior	Context
fcbb:6:10::	uN (PSP/USD)	'default':1
fcbb:6:10:e000::	uDT4	'srv6-usid
fcbb:6:10:e001::	uA (PSP/USD)	[FH0/0/0/28.35, Link-Local]:0

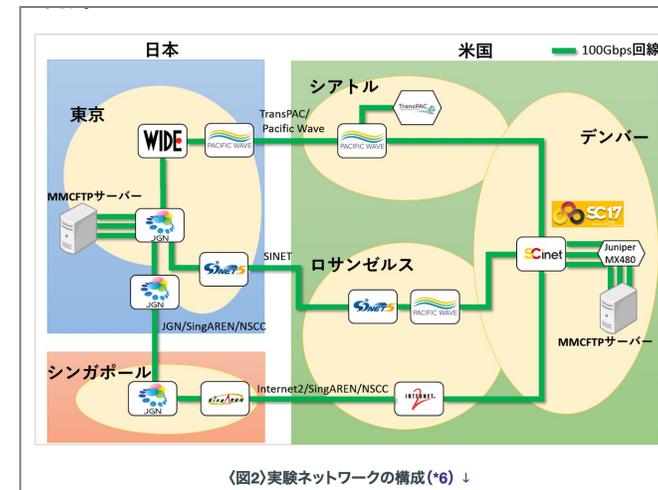


SRv6による長距離伝送路の構築実験

- 背景: 長距離大容量伝送実験
 - 本当に性能が出るかはやってみないとわからないことが多い
 - 一方で、時には海外に渡る長距離伝送路を確保するのは大変な労力(調整/構築/運用)を伴う



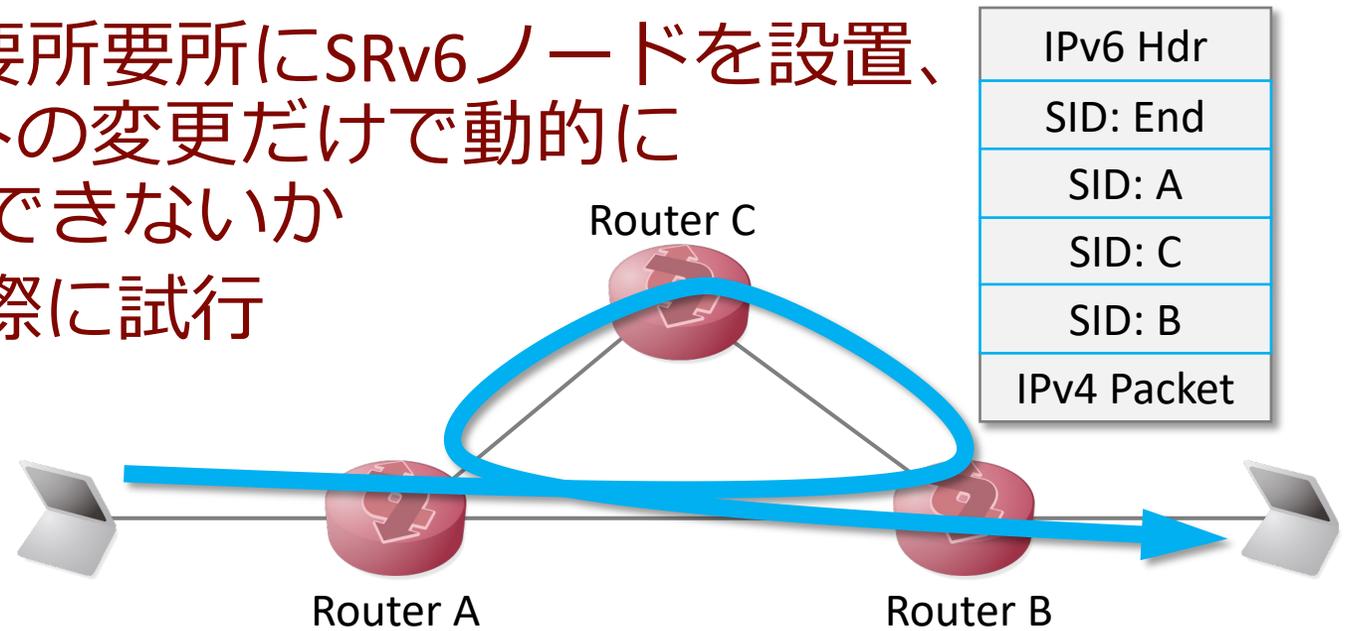
NICT, 2本の非圧縮フル解像度8K映像による超高精細立体映像伝送に成功,
<https://testbed.nict.go.jp/event/yukimatsuri2020-press.html>



NII, ファイル転送プロトコルMMCFTPで転送速度231Gbpsを達成/長距離データ転送の世界記録を更新, <https://www.nii.ac.jp/news/release/2017/1214.html>

SRv6を使った実験

- SRv6のIPv6透過性を活かした仮想/動的な長距離伝送路
 - SRv6であれば、Headendで挿入するSegmentリスト次第で、パケットに同じパスを往復させることも可能
 - IPv6ネットワークの要所要所にSRv6ノードを設置、あとはSegmentリストの変更だけで動的に長距離伝送路を構築できないか
 - ShowNetとWIDEで実際に試行



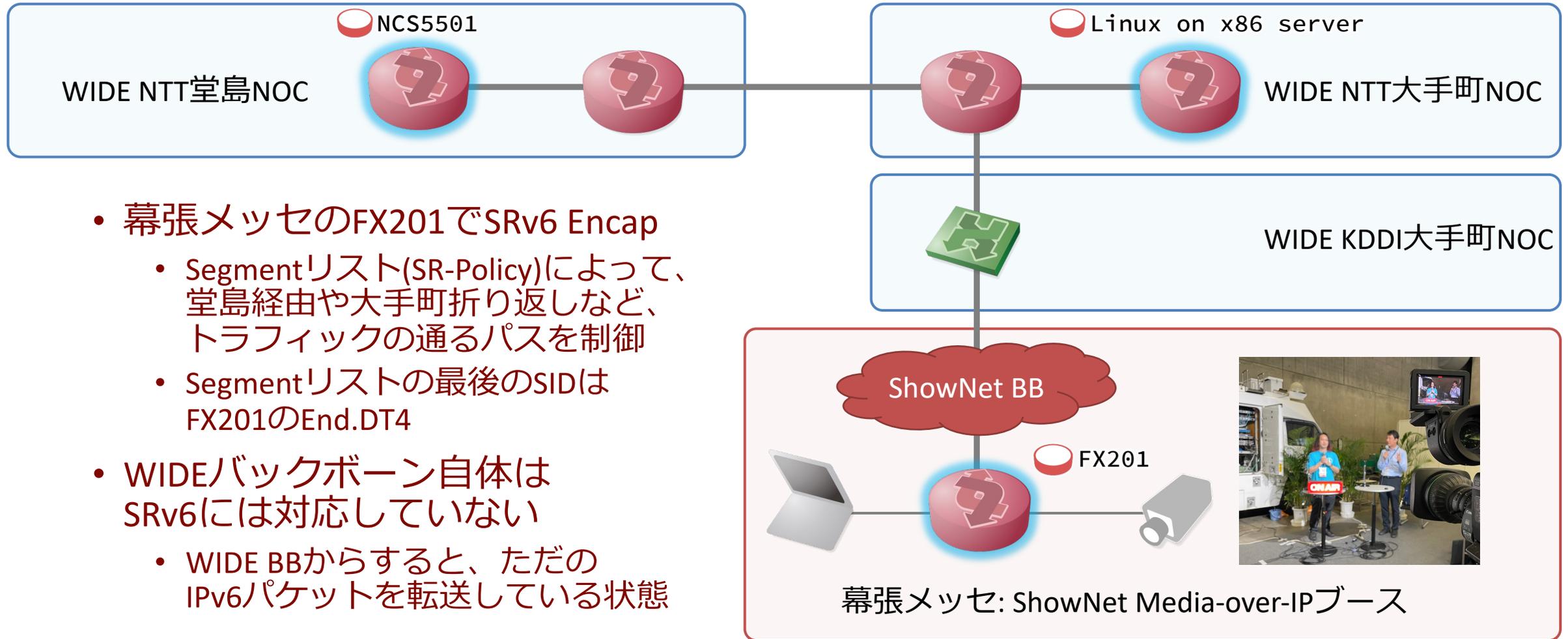
SRv6による長距離映像伝送を実施

- 東京と大阪にSRv6ノードを設置
- ShowNetのMedia-over-IPのデモ環境からSMPTE-ST2110のストリームをSRv6 Encap、東阪を往復して戻して幕張メッセで再生
- 展示期間中ブースの映像を継続して転送

ShowNet構築期間中にラック前でテスト映像が無事再生された際の様子



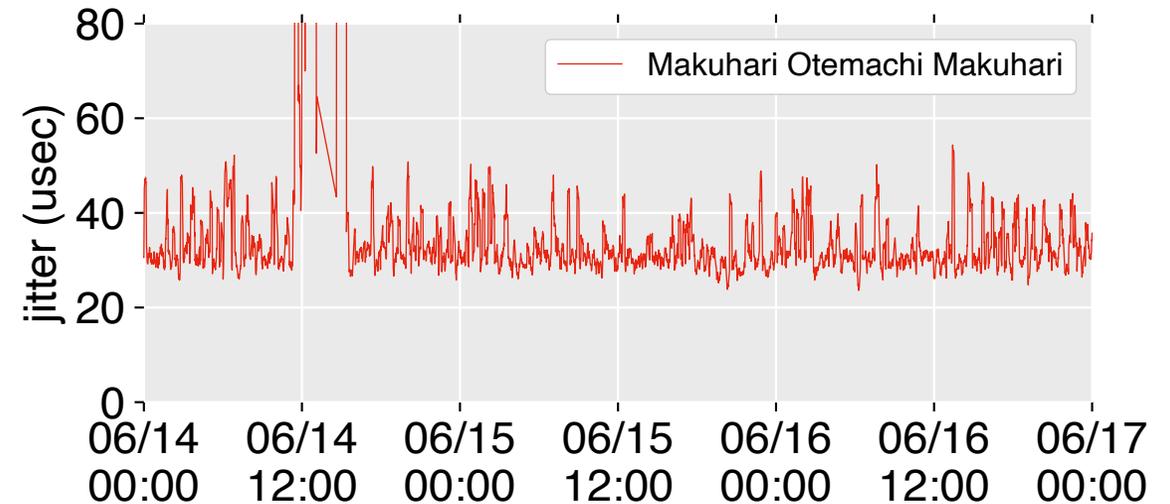
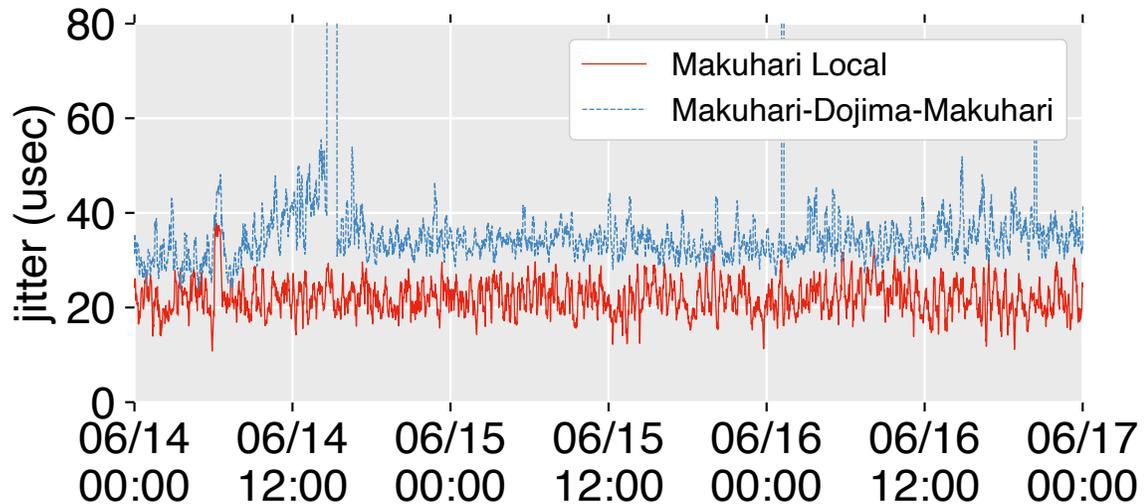
実際の構成



- 幕張メッセのFX201でSRv6 Encap
 - Segmentリスト(SR-Policy)によって、堂島経由や大手町折り返しなど、トラフィックの通るパスを制御
 - Segmentリストの最後のSIDはFX201のEnd.DT4
- WIDEバックボーン自体はSRv6には対応していない
 - WIDE BBからすると、ただのIPv6パケットを転送している状態

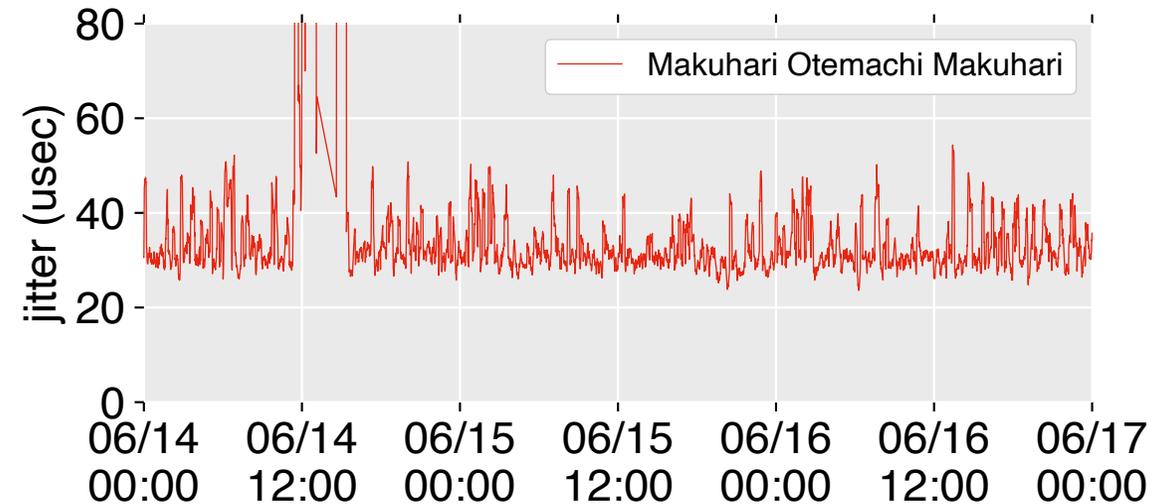
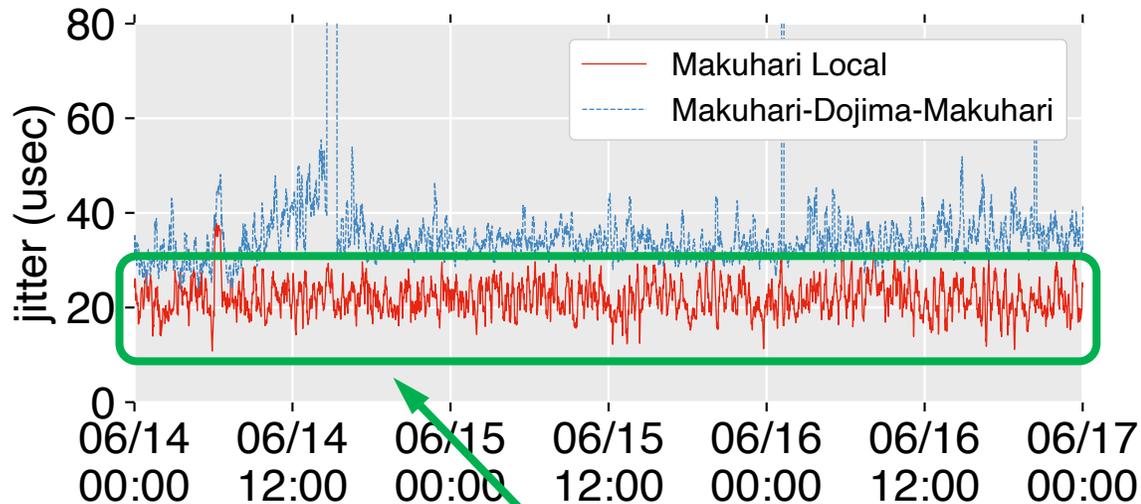
ジッターの計測

- 映像伝送とともに、伝送路のジッター計測も実施



ジッターの計測

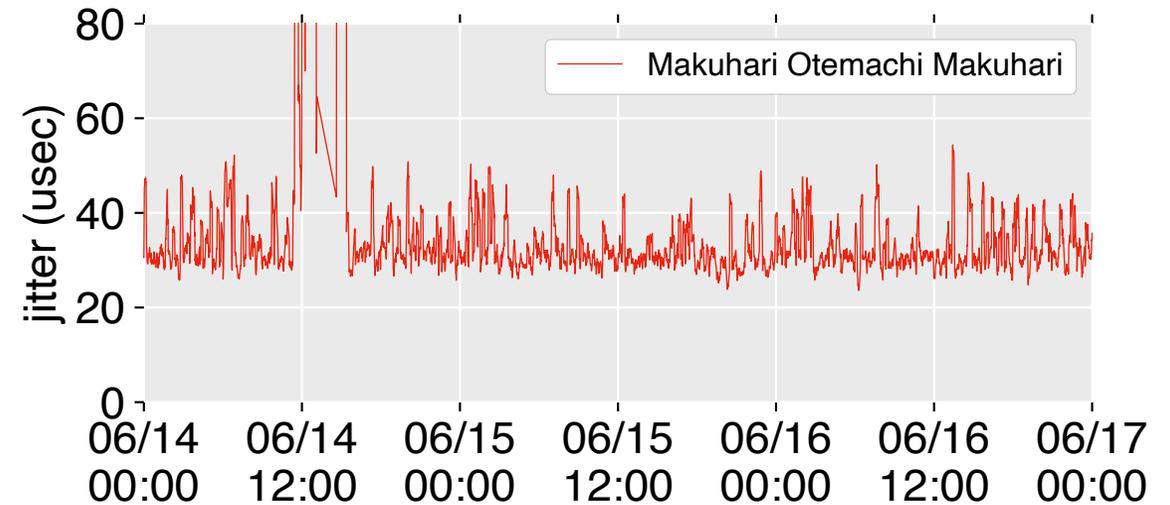
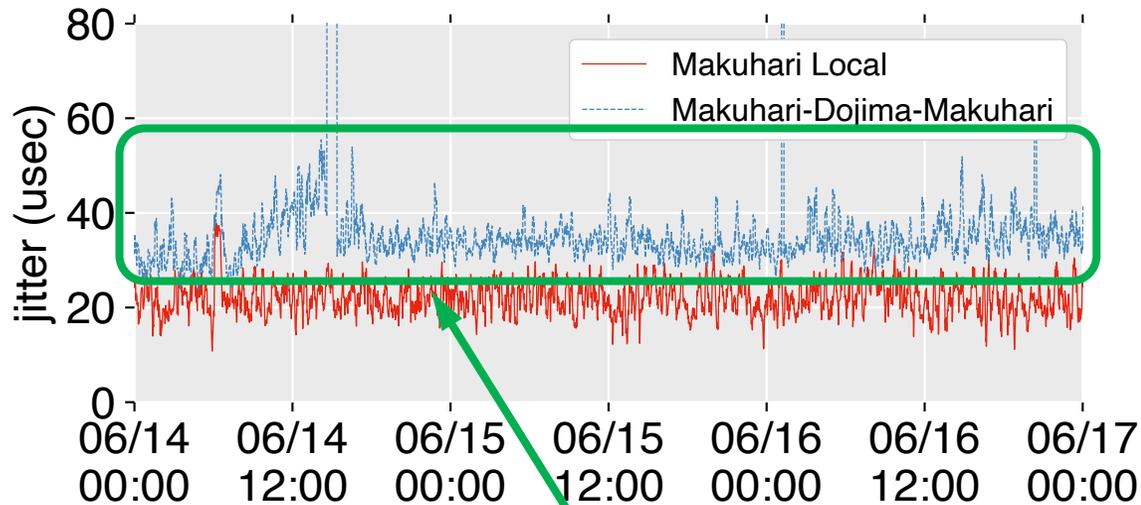
- 映像伝送とともに、伝送路のジッター計測も実施



ローカル環境での折り返しがもっともジッターが小さく20usecほど。
参考: SMPTE ST2110-30(AES67)が1Kpps = 1ms Packet Interval Time (PIT)

ジッターの計測

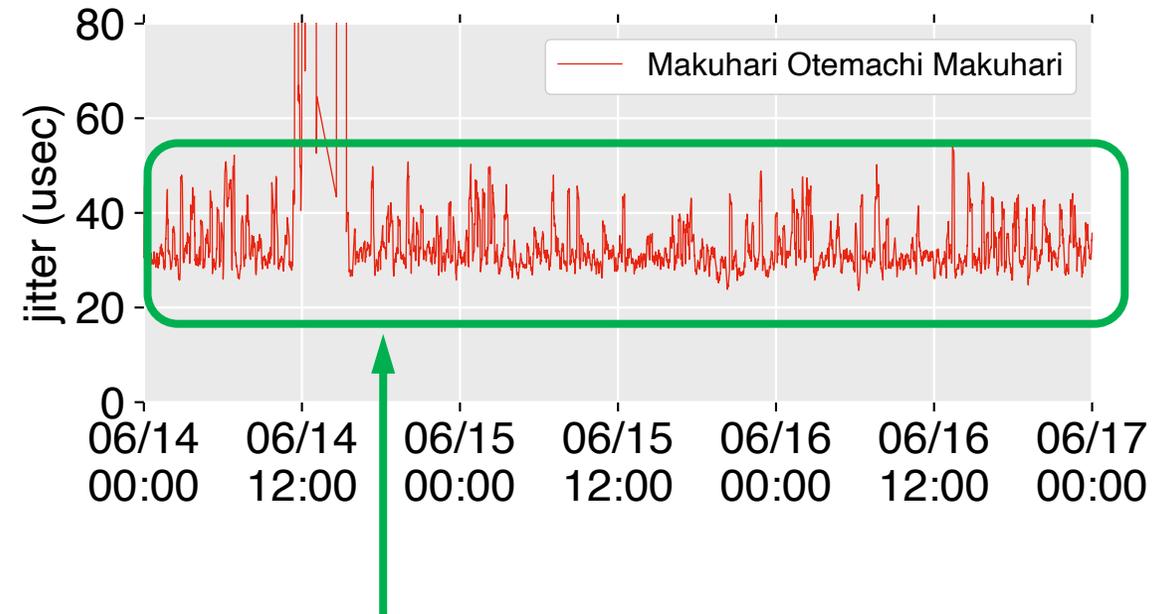
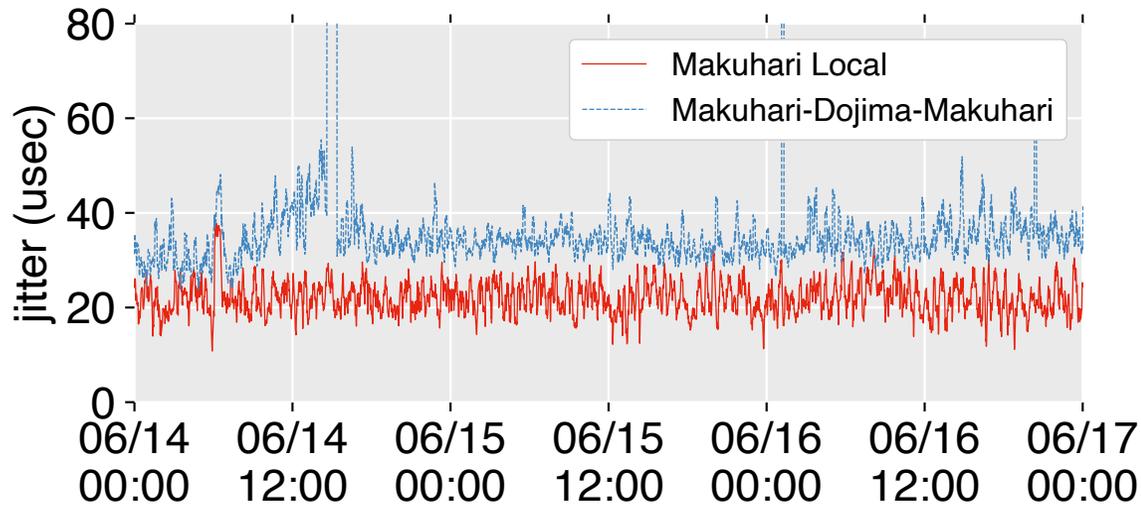
- 映像伝送とともに、伝送路のジッター計測も実施



幕張→堂島→幕張と東阪を往復すると30~40 usecと大きくなるが許容範囲。
WIDEバックボーンの中経路が輻輳していなかったことは確認

ジッターの計測

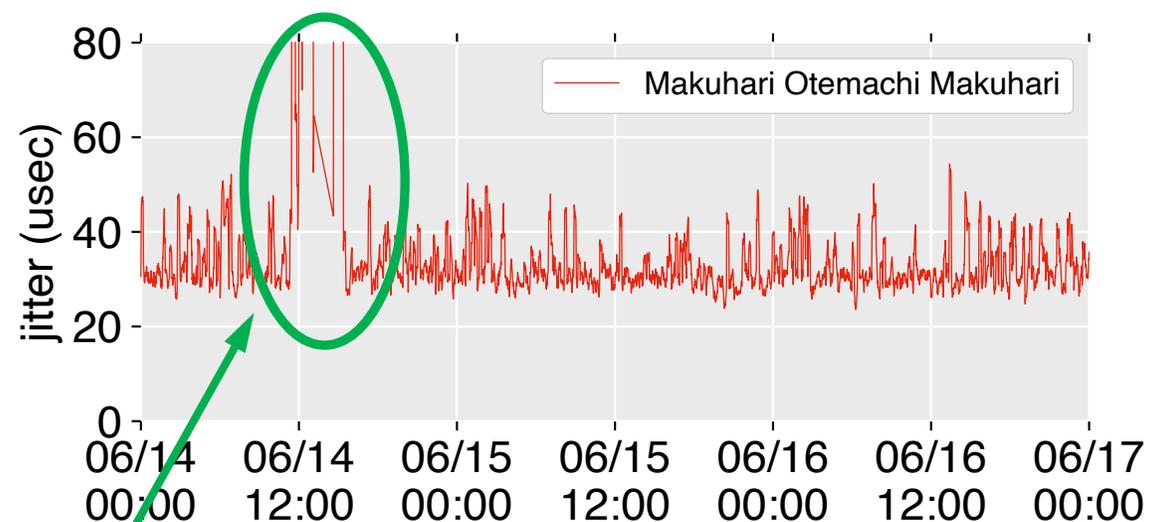
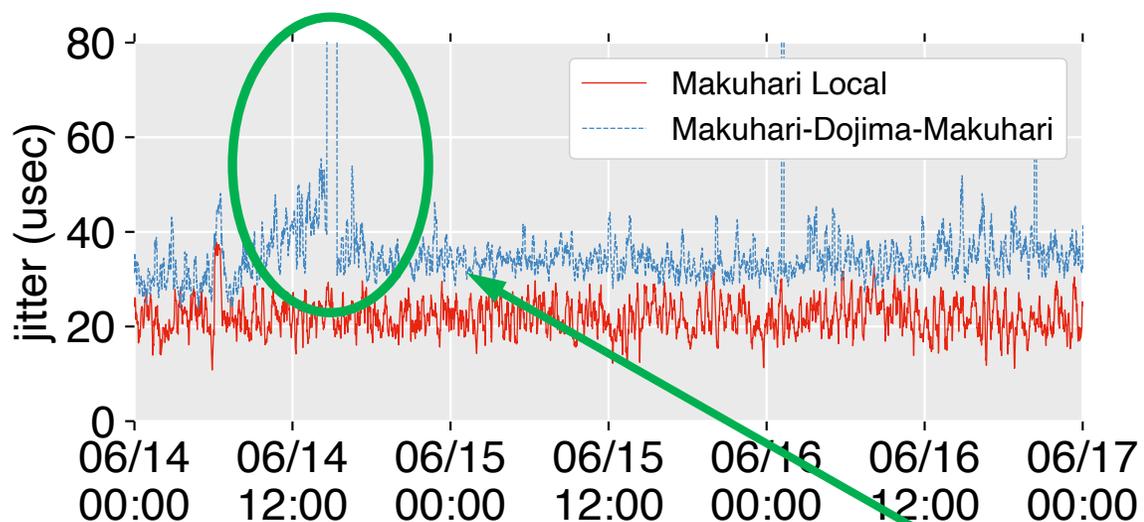
- 映像伝送とともに、伝送路のジッター計測も実施



一方、幕張→大手町→幕張では、ジッターは堂島往復よりは小さい。
x86サーバのLinux kernelのソフトウェア転送のジッターはそれほど大きくはない様子

ジッターの計測

- 映像伝送とともに、伝送路のジッター計測も実施



Peerが上がっていく過程で意図せずShowNetとWIDEの間に別のASが入ってしまい、急激にジッターが悪化

Lessons Learned

- SRv6でASを跨いだ動的な長距離伝送路の構築はできる
 - IPv6ネットワークの要所要所にSRv6ノードを設置するだけ
 - あとはHeadendルータで挿入するSegmentリストでパスを制御
- SRv6 Encapは $40 + 8 + \text{SID数} \times 16\text{-byte}$ のMTUオーバーヘッド
 - 3箇所通ると96-byte、ST 2110の最大パッケージサイズは1460-byte
 - そのため今回はjumbo frameの通るWIDE経由のみで実験を行なった
 - uSIDなどで緩和は可能
- トラフィックが意図しないネットワークを通過する事故
 - 今回はjumboの通るネットワークだったためスループットのみ落ちた
 - SRv6-awareなOAM機能(e.g., traceroute)がないと気付くのが難しい

Lessons Learned cont'd

- SRv6ヘッダ付きのパケットを透過的に転送しない機器
 - WIDEバックボーン中で大幅なスループットの減少を確認
 - 対応していないIPv6拡張ヘッダ(SRH)のパケットをCPUに上げるルータが存在した
- パケットに埋め込むSIDの数に関する制約
 - 機器によって処理できる1パケット中のSIDの段数が異なる
 - MTUが続く限りパスを構成するようなことはできない

まとめ

- ShowNetにおけるSRv6 Single Stack Backbone
 - 4ベンダー9機種によるSRv6 L3VPNの相互接続
 - IPv6 Link Localアドレスのみでバックボーンを構築可能
 - Flex-AlgoなどのTE機能もSRv6で相互接続しつつ動作
- SRv6による長距離映像伝送路の構築実験
 - 構築自体は少ない手間でできることを確認
 - 一方でMTUやOAM、SIDの段数やIPv6透過性に関する課題も
 - 今回は準備期間が短かったが、より緻密な実験/計測ができる可能性も



ちょっと待ってほしい

プログラマビリティ...? 🤔

- RFC8986 SRv6 Network Programming
 - *The concept of "SRv6 Network Programming" refers to the capability of an application to encode any complex program as a set of individual functions distributed through the network.*
- 2022年にできたのはL3VPNだけ
 - SDNやNFVでできたほどのプログラマビリティ(または柔軟性)は、まだ発揮できてはいない(と、個人的には思う)

プログラマビリティとの悪戦苦闘史: 導入編

- 2012年~2013年: OpenFlowの導入
- ネットワークオペレータがプログラミングもし始める
 - ネットワークの運用とプログラミングは以前は別個のものだった
 - ネットワーク側にもプログラミングできる"余地"が生まれ始める
- プログラミングする目的
 - 新しい機能/付加価値を実現する手段
 - 自動化

2013年の日中



とても大変でした...

2013年の明け方



SDN Japan 2014 #shownet ←
Scratch and Re-build the Internet

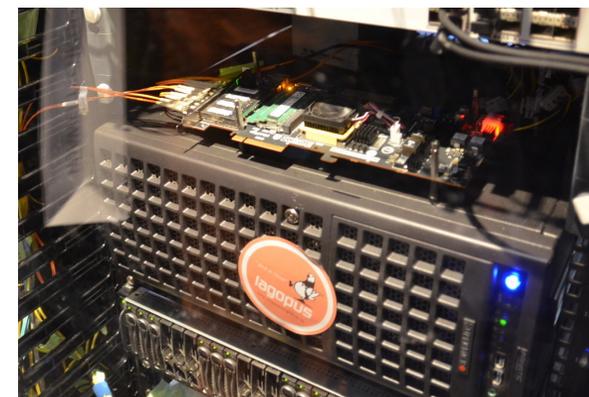
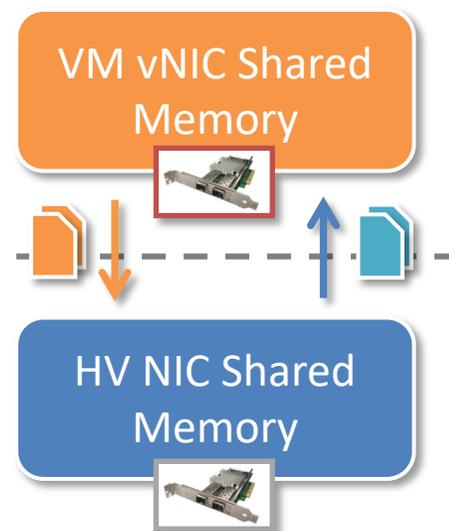
ネットワーク運用自動化の難しさ

- ソフトウェア制御で設定の変更を自動化
- ベンダーや製品、バージョンごとに異なるAPI
 - 実際の現場ではさまざま機器を利用している
 - 全ての機器を網羅するAPIは現状では存在しない
- エラーハンドリングの難しさ
 - Debuggabilityの欠如

Copyright © INTEROP TOKYO 2014 ShowNet NOC Team 57

プログラマビリティとの悪戦苦闘史: 仮想化編

- 2014年~2016年: 仮想ルータの登場
- ソフトウェアによるパケット転送が現実的に
 - 仮想ルータ自体をソフトウェア的にインテグレーションする
 - ハードウェアのコモディティ化
- 性能が課題に
 - ネットワークオペレータがコンピュータアーキテクチャも勉強し始める
 - 時にはFPGAにも手を出す



むき出しのNetFPGA-SUMEと、LagopusとVirNOSの動作するx86サーバ

プログラマビリティとの悪戦苦闘史: 回帰編

- 2017年~2018年: IPでもがんばれるのでは？
- IP Routingでサービスチェイニングを実現する
 - VRFとBGP Flowspecの合わせ技
 - Flowspec Routeを打ち込むコントローラをexabgpで実装
 - FlowspecをFlow Modのように使ったとも言える



振り返って: プログラムできる余地の拡大

- プログラマビリティは様々なレイヤーに存在する
 - 厳密には、プログラムから触れるAPIが各層に整備された
 - 下に行くほど大変だが、新しいことを実現できる
 - 上に行くほど(一見)簡単なインターフェースで制御を書ける

Orchestration	ONOS, Vendor Orchestrator, 必要な部分だけ自分で作ったり
Management	YANG, Netconf, SNMP, gNMI, gNOI, Ansible, etc
Forwarding Rule	Routing Protocols, PCEP, Flowspec, Netlink, Cisco XR Service Layer, Juniper JET, etc
Switch Chip	P4, SAI, Vendor SDK, etc

抽象化とその漏れ

- プロトコル/ネットワークスタックとは、抽象化そのもの
 - ある層はその下の層を隠蔽し、抽象化された機能を実現する
 - 例: TCPは下のIPv4やIPv6を隠蔽し、信頼性のある通信を提供する
 - 例: 機器のOSはASICを隠蔽し、ルーティングプロトコルによってネットワークにおけるノードとしてふるまうことを実現する
 - 現実には、抽象化は漏れる
 - 下のレイヤーのことを知らないと、上のレイヤーの問題を解決できない

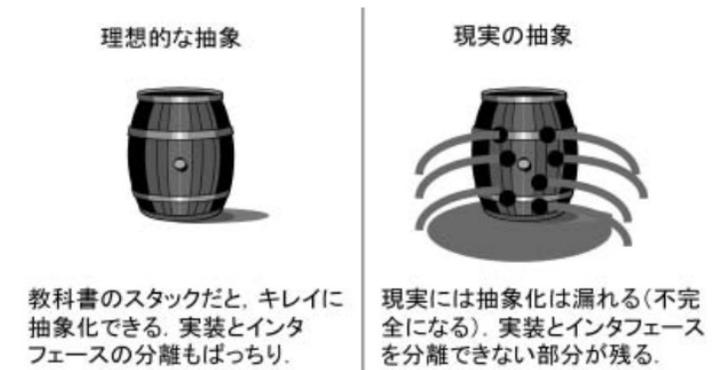


図 2 現実問題では抽象化は漏れる (不完全になる)。

なぜソフトウェア論文を書くのは難しい(と感じる)のか、
権藤克彦 et.al., コンピュータソフトウェア, 2009
原典はLaw of Leaky Abstractions, Joel Spolsky

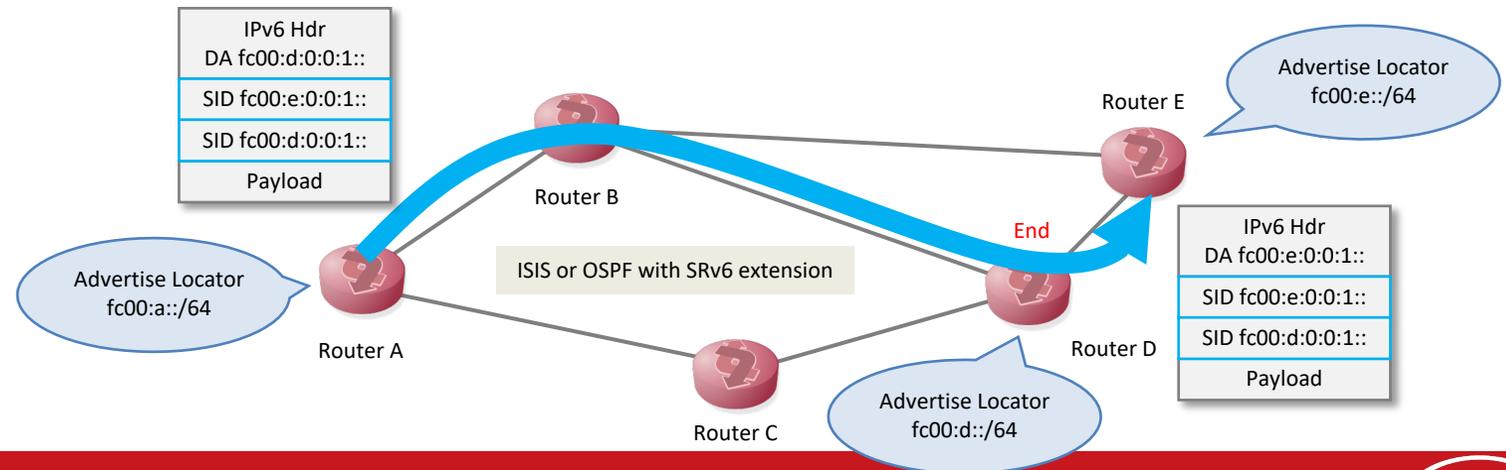
抽象化と漏れとデバッグと

- 抽象化の漏れ@ネットワーク
 - 例: 壊れているのは、自分が書いたOpenFlowアプリか？
コントローラか？ Secure Channelか？
はたまたOpenFlowスイッチか？
 - オペレータならおそらくみんな似たような経験しているのでは
- よく出来た抽象化とよく出来たその実装、
つまりプロトコル、ソフトウェアを見出す/生み出すことが大事
- ShowNetでの悪戦苦闘は、この部分の成熟度合いに起因(早すぎたんだ...)



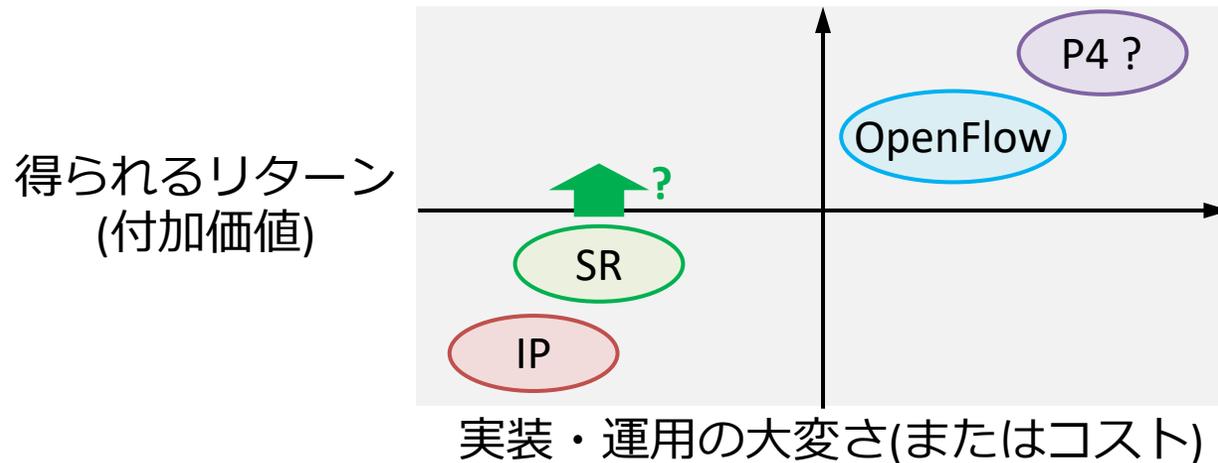
Segmentという抽象化

- Source Routingという機能と、実装としてのSR(v6)
 - その下のレイヤーは**信頼と実績のプロトコルと機器群**
 - 今ついにSRv6というレイヤーが動くところまで来た
 - 次はこのレイヤーの上に何を作るのか
 - SRv6でプログラミングする
余地はどこだ？



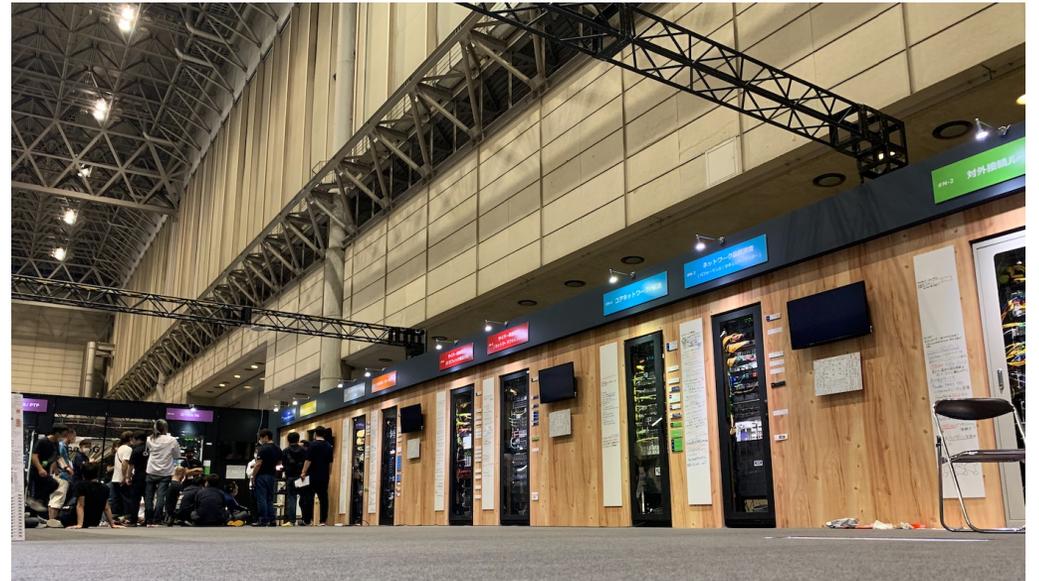
悪戦苦闘の果てに

- 柔軟なネットワークの実現に向けて@Interop Tokyo ShowNet
 - ネットワークがプログラマビリティを獲得する10年
 - OpenFlowはdrawすぎて、それだけを現場で使うにはハードルが高かった
 - ソフトウェアエコシステムのネットワークへの導入が始まったSDN/NFV
 - 実践的な運用の楽さと、柔軟性を実現する複雑さのバランスをとったソリューションとしてのSource Routingの再発見



おわりに

- 何のプログラマビリティをどう活かすか
 - プログラマビリティは様々なレイヤーに存在する
 - 開発・運用の大変さと、得られるリターンのバランス
 - SRv6には引き続き期待
- 来年もぜひShowNetに足をお運びいただければ幸いです
 - 先んじて悪先苦闘
 - そしてコミュニティへの還元



Interop[®]22

Tokyo JUNE 15-17
MAKUHARI MESSE, JAPAN

SHOWNET

Over the Premise

