

ShowNet SDNへの取り組み @2017-2019

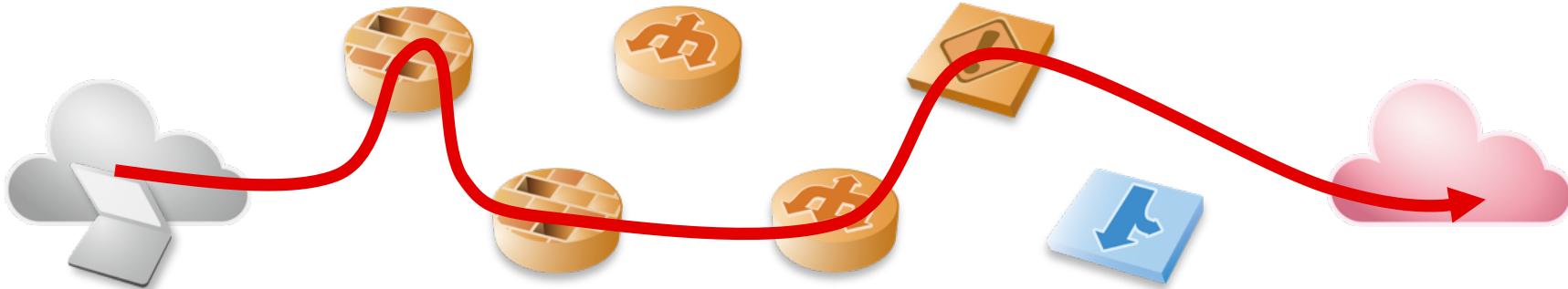
NTTコミュニケーションズ /
Interop Tokyo ShowNet NOCチーム
上野 幸杜

SDN/NFV@2017-2019

- このパートのカバー範囲
 - 2017: IPベースサービスチェイニング
 - 2018: IPベースサービスチェイニング改
 - 2019: SRv6ベースサービスチェイニング

サービスチェイニング

- ・ネットワークの機能(Service Function)を鎖のように連携させるアーキテクチャ
 - ・サービス構成とネットワーク構成の分離
 - ・ユーザごとのきめ細かなサービスの適用と制御
 - ・自動迂回・フェイルオーバーなどの耐障害性



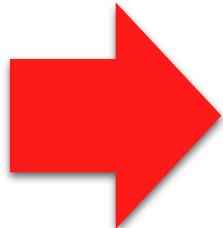
5年間で得られた知見

- IPと比べると、SDN/NFVは構築・運用が難しい
 - 開発コスト: データプレーン、コントロールプレーン
 - 運用コスト: 専用のパケット転送/制御ロジックへの精通
 - とりわけShowNetでは、2週間で構築、検証を行う

日中



明け方



ShowNetに必要なものは 本当にプログラマビリティか？

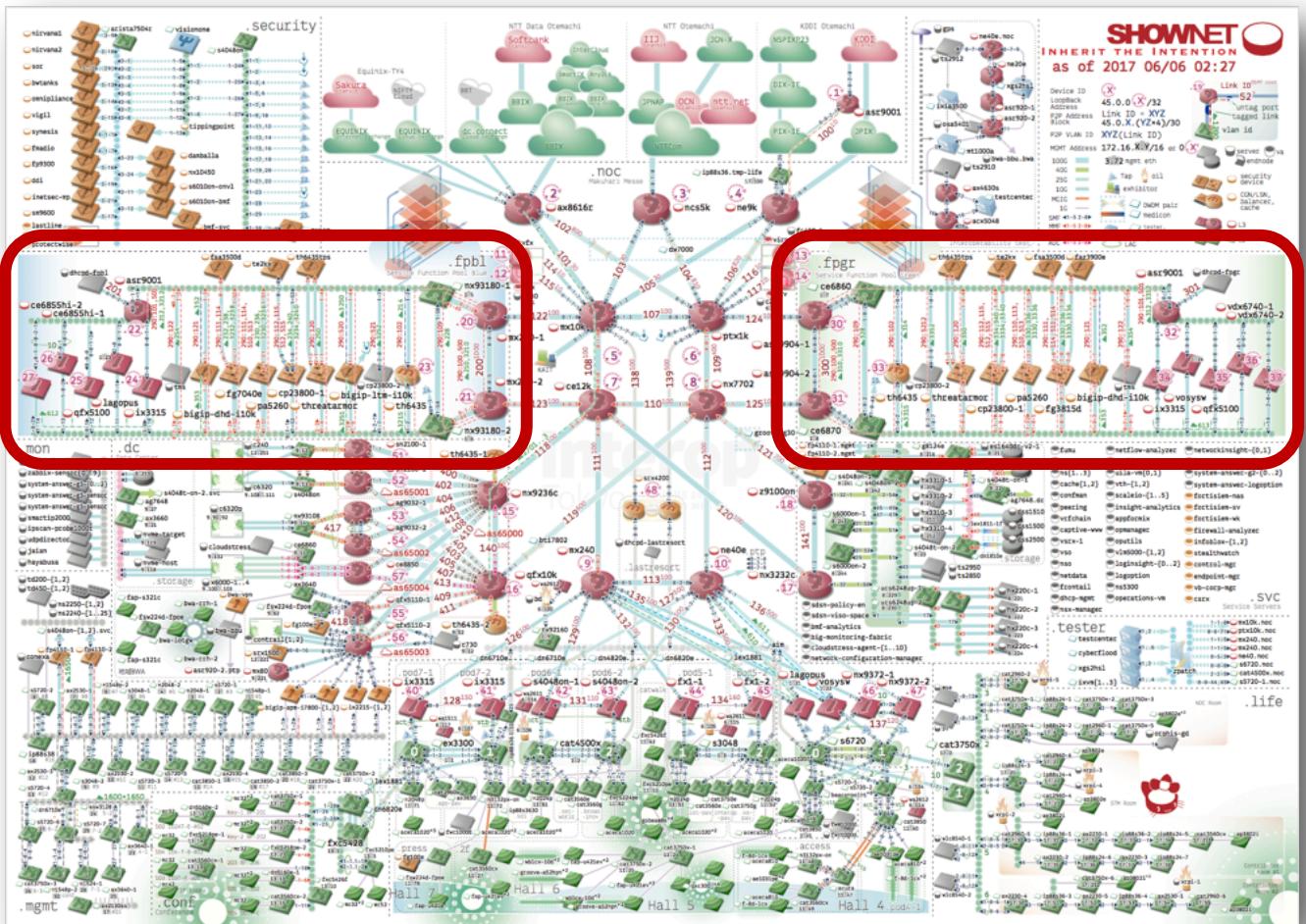
- もちろん、プログラマビリティが必要とされるケースは昨今多数ある
 - SDN/NFV@2014-2016はその具体化
- しかし短期決戦かつ、ショーケース的意義を持つShowNetでは
 - 現場でゆっくりデバッグしている暇はない
 - 開発の属人化を回避したい
 - 万人がアレンジして運用できるものを見せたい
- サービスチェイニング実現のための機能の最小セット (=IP+ソースルーティング) を、標準的な技術のみで実現できれば良いのではないか
 - 当時実装があり”経路”として5-tupleでのルーティングを実現するBGP Flowspecに目を付ける

BGP Flowspec (RFC5575)

- BGPでフィルタールールを伝搬する技術
 - BGPのNLRIを使って細かなパケットのタイプを指定し、適用するアクションを伝搬する
 - 2016年までもSDN/NFVに使用し、“きちんと動く”ことがわかつていた

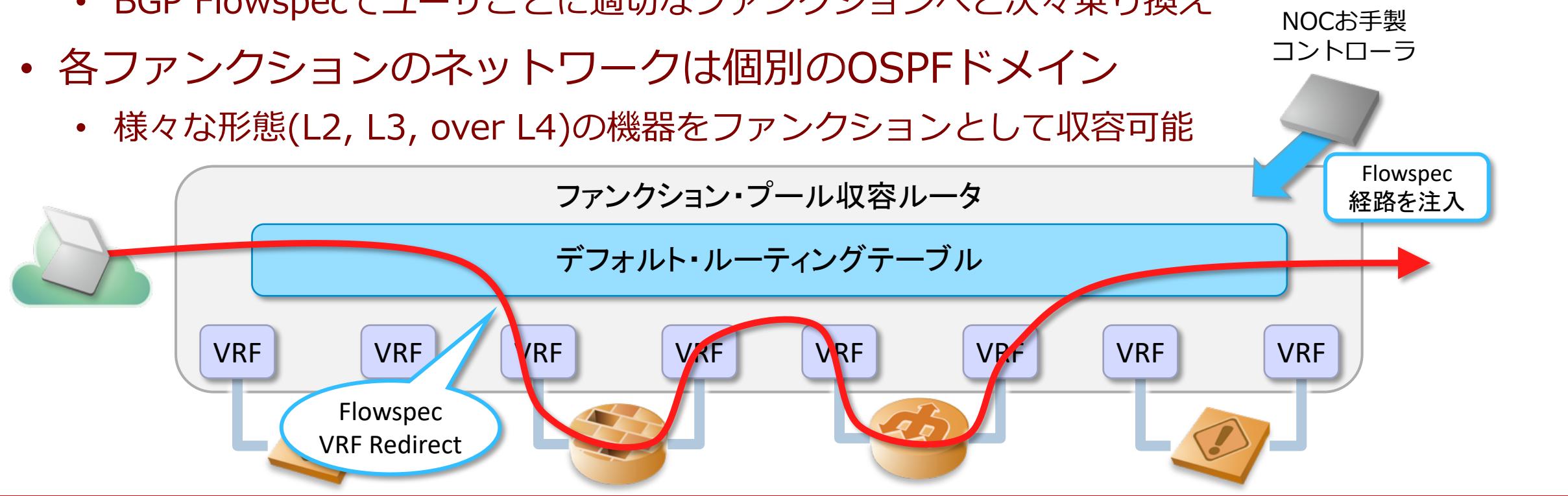
2017: IPベースサービスチェイニング

- BGP Flowspecにより、
チェインを上書きすることでサービスチェイニングを実現
- ユーザトラフィックにデフォルトで適用



BGP Flowspecを使ったサービスチェイン

- 小さいIPネットワークをBGP Flowspecによって”乗り換え”させるアプローチ
 - VRFを用いてファンクションごとに個別のネットワークを構築
 - BGP Flowspecでユーザごとに適切なファンクションへと次々乗り換え
- 各ファンクションのネットワークは個別のOSPFドメイン
 - 様々な形態(L2, L3, over L4)の機器をファンクションとして収容可能



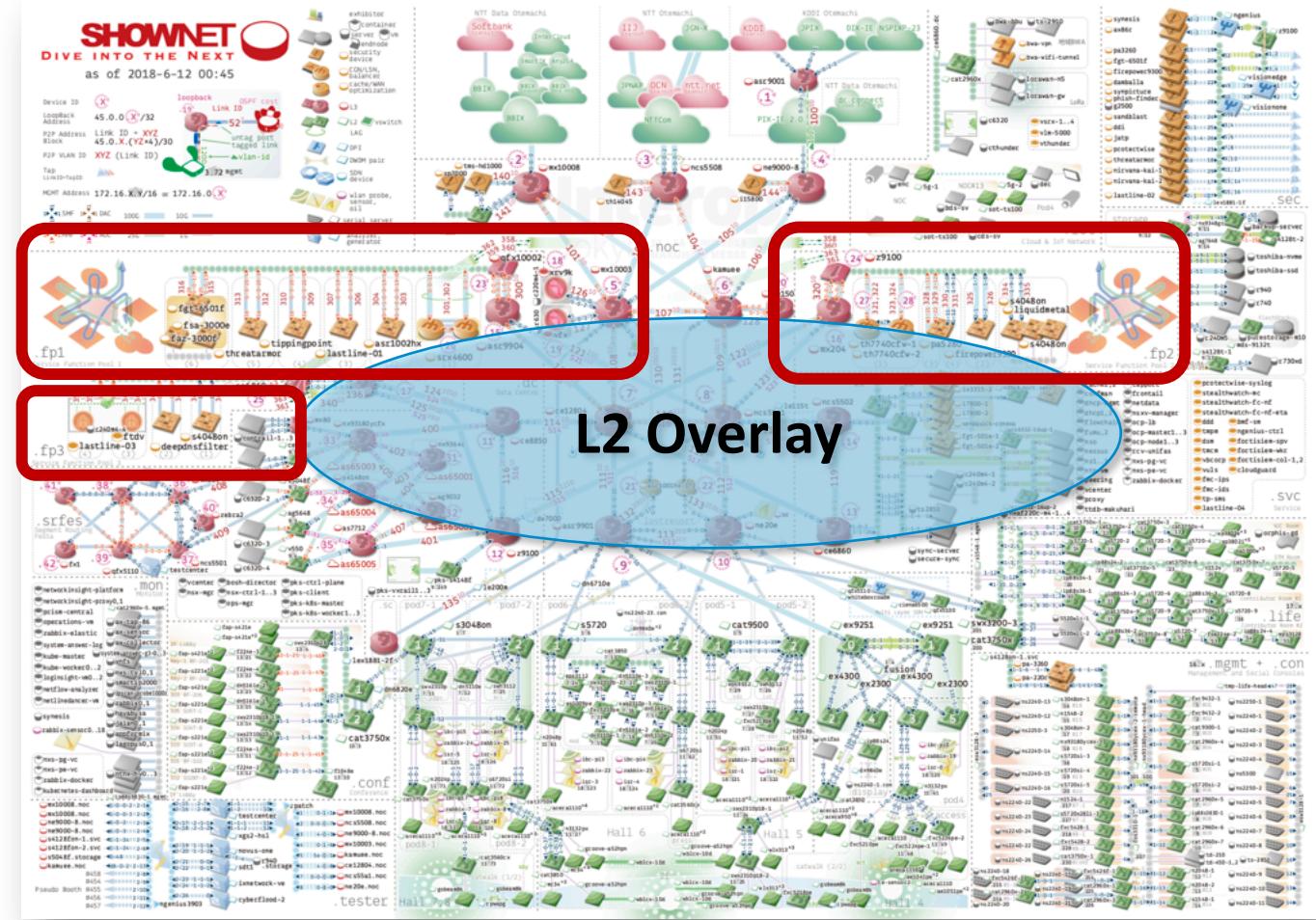
運用結果

- 本番では大きなトラブルなくサービス提供に成功
 - 21 VRF / Device
 - 5804 Flows / Device
 - 200 Tenants
 - 出展社、来場者無線等のユーザトラフィックを全て収容
- もちろん課題もたくさん
 - 実際のユースケースでは1対の完全に同一なFunction Poolを用意するのは難しい, etc.



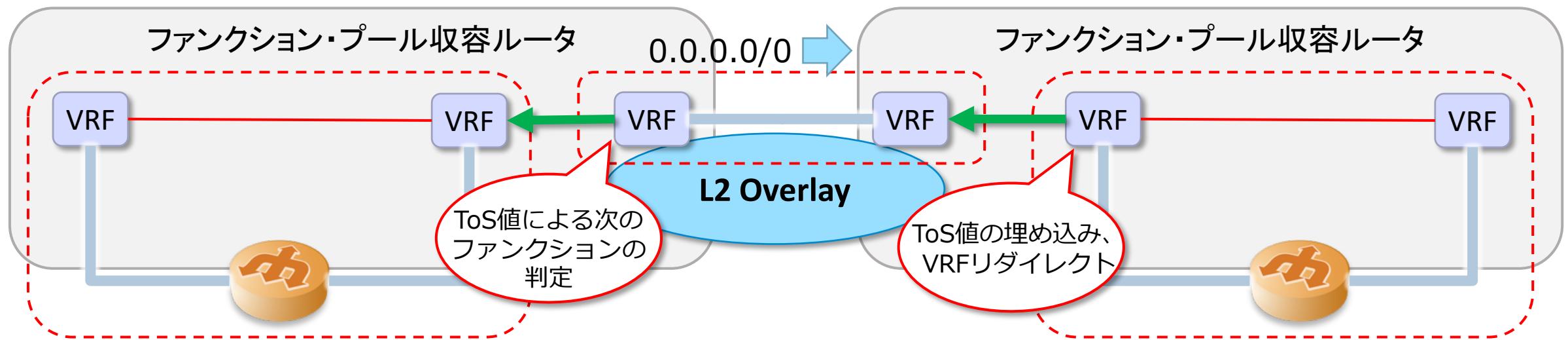
2018: IPベースサービスチェイニング改 (拠点間チェイン)

- 2017をベースに
EVPN/VXLANによる
拠点間サービスチ
エイニングを実現
 - 拠点数が3つに増え、規
模が非対称に
 - 実際のユースケースを意
識した改良



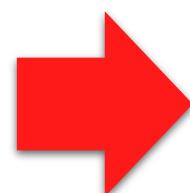
Function Poolをまたぐチェインの構築

- Function Pool間に仮想的なP2Pリンクを作り、OSPFでデフォルトルートを広告
- BGP Flowspecで各Function Pool行きのVRFに乗り換え
- 到着先のFunction PoolではToS値によって次のFunctionを判定



2018までの取り組みとさらなる改良への壁

- BGP Flowspecを使用することで、サービスチェイニング網はIPとBGPの知識があればデバッグ可能に
 - 細かい挙動まで理解してトラブルシュートできる人数が増えたことで、構築スピードが向上、結果チャレンジできる内容も増えた
- BGP Flowspecの本来の想定用途はサービスチェイニングではないため、フェイルオーバーなど機能的な限界が見え始めた

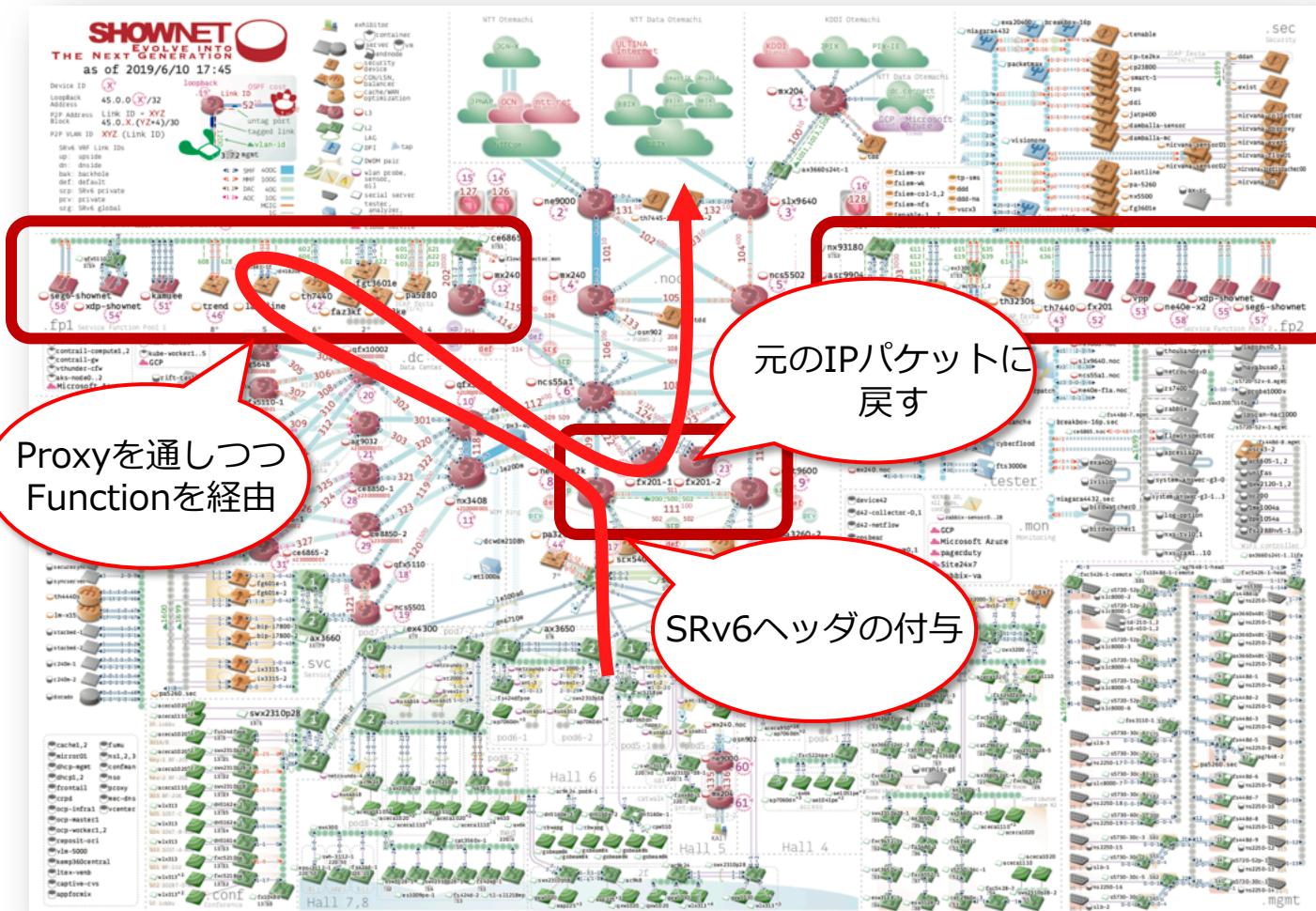


新しいアプローチ

- SRv6によってIPの手軽さとSDN的柔軟性の“いいとこ取り”を図る
 - IP網のデバッグのしやすさ、既存装置でのサポートを享受しながら、途中経路を制御可能
 - SRv6は標準技術であるため誰でも再現可能（ただし結果的には諸説アリ...）
- 2017-2018年には相互接続性試験を実施、2019年には各社の実装が揃ってきたため実戦投入
 - 相互接続性試験が実ユースケースにつながった事例の一つ

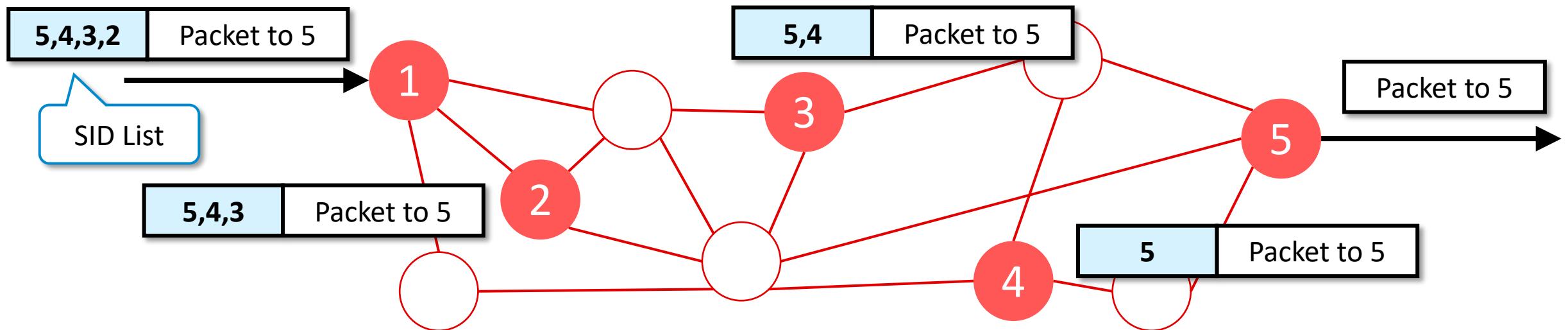
2019: SRv6ベースサービスチェイニング

- ユーザトラフィックにSRv6ヘッダを付与し、左右のFunction Poolへ誘導
- Functionの手前でProxyがSRv6のパケットをIPパケットに戻すことで既存Functionをサポート



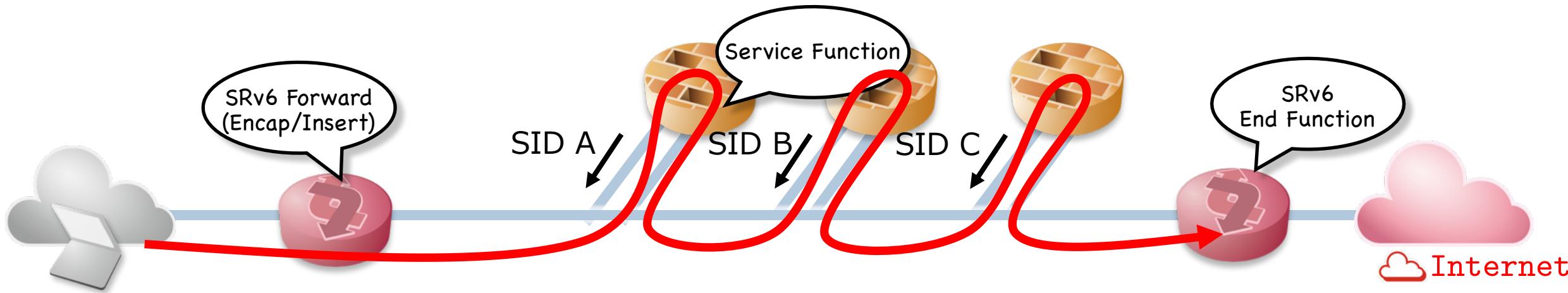
Segment Routing

- ネットワークをSegmentで表現し、ヘッダに埋め込まれたSegment Listによってパケットを転送する新しいルーティング技術



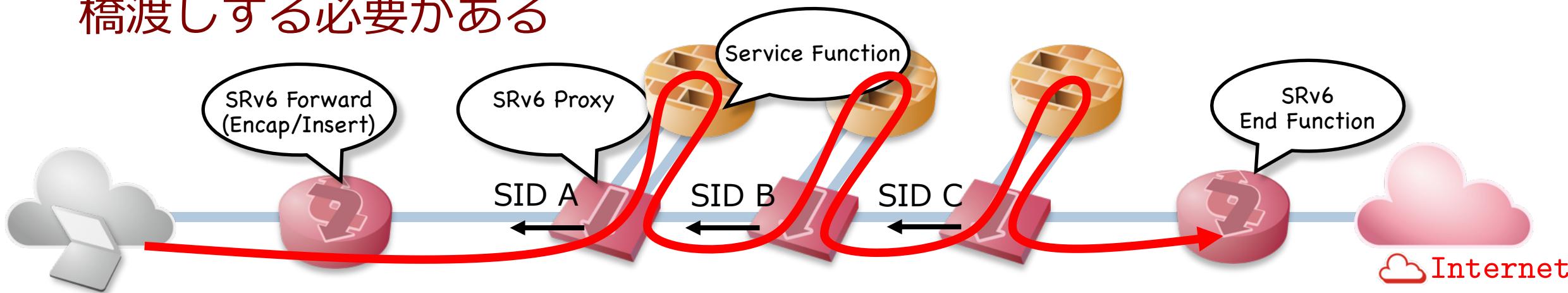
SRv6の理想的なアーキテクチャ

- 各Service FunctionがSIDを広告、SRv6ヘッダを解釈し、
それに基づいてトラフィックを転送する



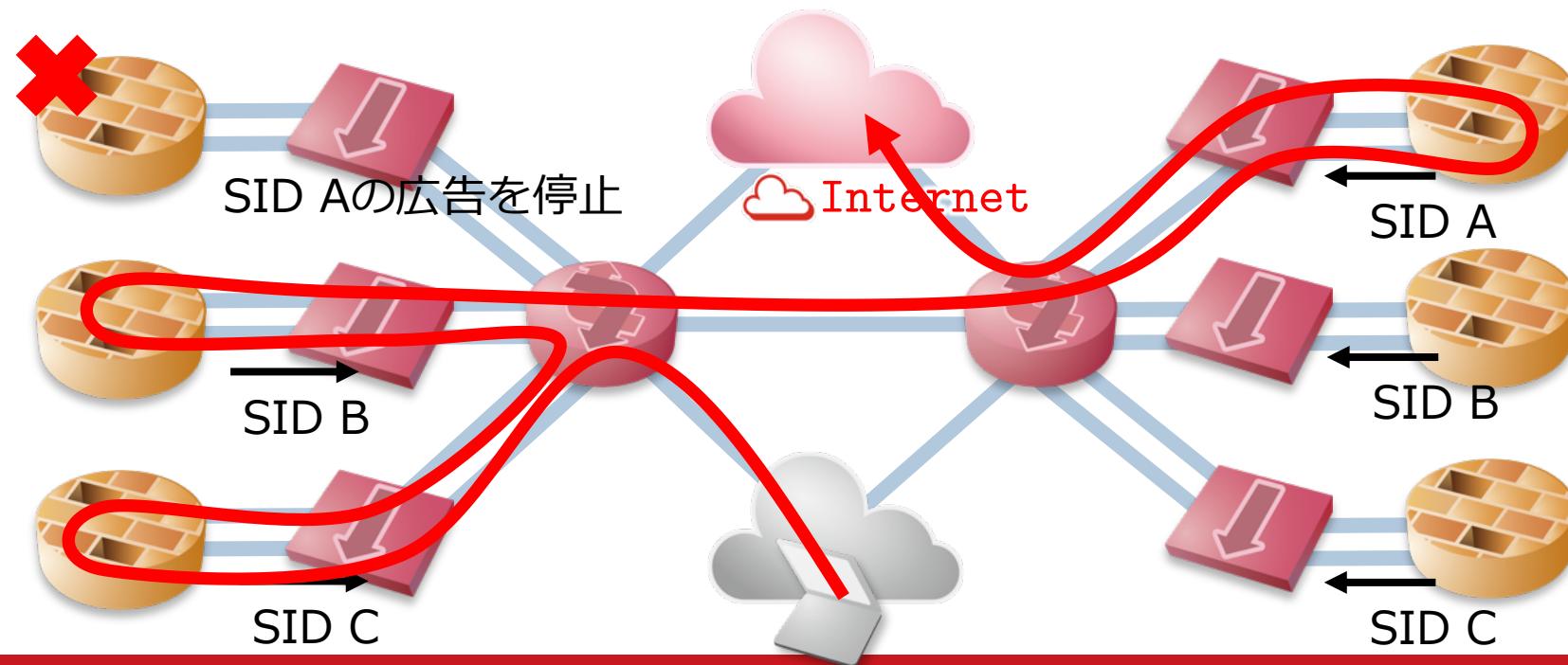
2019時点で実現可能だったアーキテクチャ

- 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



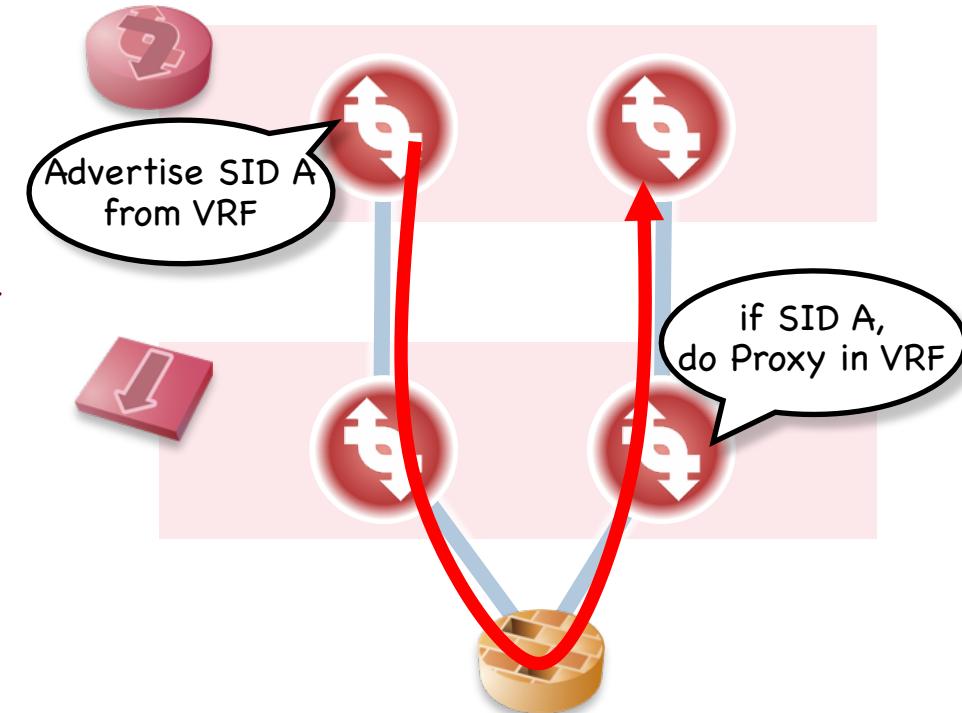
Service Functionの自動フェイルオーバー

- Service Functionの故障時は、SIDの広告停止によりバックアップのService Functionに自動フェイルオーバー
 - ただし、Service Functionの故障検知は難しい



Service Function故障時の自動SID廣告停止

- ShowNet 2019での手法:
OSPFv3 + VRF
 - ワークアラウンド的手法
 - 将来的にはService Functionが直接SRv6を解釈するか、SRv6 Proxyが死活監視機能を持つことが望ましい
- 2017-2018のサービスチェイニングのフェイルオーバー方式から応用されたアイデアの一つ



SID AがService Functionを経由して伝搬

SRv6 Proxyの方式

- いくつか種類があるが、機能的な観点でみると以下2種類に大別
- ShowNetのような大規模環境では、チェインそれぞれに SRv6 ProxyおよびFunctionを用意することが現実的に不可能
 - ただしEnd.AMはIPv6トラフィックのみに適用可能



IPv4 over SRv6のProxy方式: End.AT

- ShowNet 2019での要件を満たすためNOCチームで設計・実装
- 1つのSRv6 Proxyで複数のIPv4チエインを収容可能なProxy方式
 - IPv6+SRヘッダをキャッシュし、ToSに識別子を埋め込む
 - XDPによる実装:
<https://github.com/edenden/end.ac>



標準化コミュニティへの貢献

- 下記2本のInternet-DraftをIETF 106にて提案
 - ShowNet 2019のSRv6によるサービスチェイニング
 - SRv6 Proxyの新方式(End.AT)



I E T F®

出典: <https://www.ietf.org/blog/ietf-105-highlights/>

2019を終えて

- Segment Routingはサービスチェイニングに実際に適用可能、頑張ればFunctionごとのフェイルオーバーも実現できる
- 実装が“枯れる”にはまだ時間がかかりそう
 - コントロールプレーン: 2019時点では実装状況が間に合わず完全手動、設定量が膨大に
 - データプレーン: Locator長など設定可能な範囲、SID Policy選択の手法、SRv6 behaviorの実装状況など、各機器ごとに異なる細かい制約があり構築負荷が高い
 - OAM: 素のIPにおけるping, tracerouteのようにこなれた手法が通用しない場面も多く、インターフェースカウンター頼みになることも

まとめ

- 2017-2019の3年間はサービスチェイニングの実現に注力
 - 2017: IPベースサービスチェイニング
 - 2018: IPベースサービスチェイニング改
 - 2019: SRv6ベースサービスチェイニング
- プログラマビリティを駆使したオーダーメイドな構成から標準的なIPベースの構成へのシフト
 - 適用可能な製品の幅が広がり、構築に際してのマンパワーも増えた。
結果的に規模が大きかったり複雑な機能へのチャレンジも可能に
 - 得られた知見は他のユースケースへの応用も可能（→標準化への参画）
 - ただし、結局最先端のユースケースにつっこむので現場での苦労は大差なし…