

Cisco Service Provider Technology Webinar
「Interop Tokyo 2022 におけるシスコの学びダイジェスト」

SRv6 最新情報

Miya Kohno, Distinguished Systems Engineer, Cisco Systems

2022年7月21日





Agenda

- Interop Tokyo Conference について
- SRv6最新情報
 - 標準化動向
 - Network as a Platformに向けて

Interop Tokyo カンファレンス (有料)

開催概要 タイムテーブル ▾ FAQ (よくあるご質問) 関連リンク ▾

日本のインターネット技術の **発展・検証の場**
Interop だからこそ得られる重要技術情報

Interop® Tokyo Conference

オンデマンド登録開始は6月下旬より >

そんな”Interop Tokyoだからこそ伝えられる有益な情報”を
15のキーワードで構成された全41セッションでご紹介します。

ネットワーク

Wi-Fi

5G/Beyond 5G

Segment Routing

IOWN

データセンター

セキュリティ

IT戦略

クラウド

デジタル基盤

モビリティ

モバイル

量子テクノロジー

ID

法律・規約

SRv6 特集 @ Interop Tokyo Conference

1. Tutorial

YA1-01

Segment Routing (1): SRv6チュートリアル

Speaker



シスコシステムズ (同)
情報通信産業事業統括
システムズエンジニアリング本部
システムズアーキテクト
鎌田 徹平

ネットワーク

Segment Routing

2. 商用deployment

YA1-02

Segment Routing (2): 商用デプロイメント

Speaker



ソフトバンク (株)
パケットネットワーク部モバイル
IPネットワーク課
課長
渡邊 孝也

Speaker



LINE (株)
シニアソフトウェアエンジニア
城倉 弘樹

Speaker



楽天モバイル (株)
IPTX戦略、アーキテクチャ&技術
部 部長
Praveen Kumar

Chair



シスコシステムズ (同)
Distinguished Systems Architect
業務執行役員
河野 美也

ネットワーク

Segment Routing

3. 標準化 + Advanced Topics

YA1-03

Segment Routing (3): 標準化動向と実装状 況

Speaker



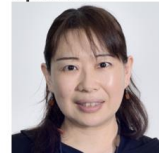
ソフトバンク (株)
城嶋 聡

Chair



LINE (株)
シニアソフトウェアエンジニア
城倉 弘樹

Speaker



シスコシステムズ (同)
Distinguished Systems Architect
業務執行役員
河野 美也

ネットワーク

Segment Routing

SRv6 特集 2. 商用deployment session @ Interop Conference

Interop Tokyo 2022

**SRv6 -Application and Benefits
from Mobile Network Operator's Perspective**

Praveen Kumar

Department Manager, IPTX Strategy, Architecture & Engineering Department
Rakuten Mobile, Inc.



皆さん、こんにちは。SRv6セッションによろこそ。私は楽天モバイルのIPTX
戦略アーキテクチャとエンジニアリングを率いているPraveen Kumarと申し

**Rakuten
Mobile**

LINE


SoftBank



これからの新モバイルアーキテクチャ - SRv6 MUP と ICN -

YE2-03

これからの新モバイル アーキテクチャ ～SRv6 MUPとICN～

Speaker



ソフトバンク (株)
コアネットワーク本部
副本部長
KHAN ASHIQ

Speaker



(国研) 情報通信研究機構
ネットワークアーキテクチャ研究
室
室長
朝枝 仁

Speaker



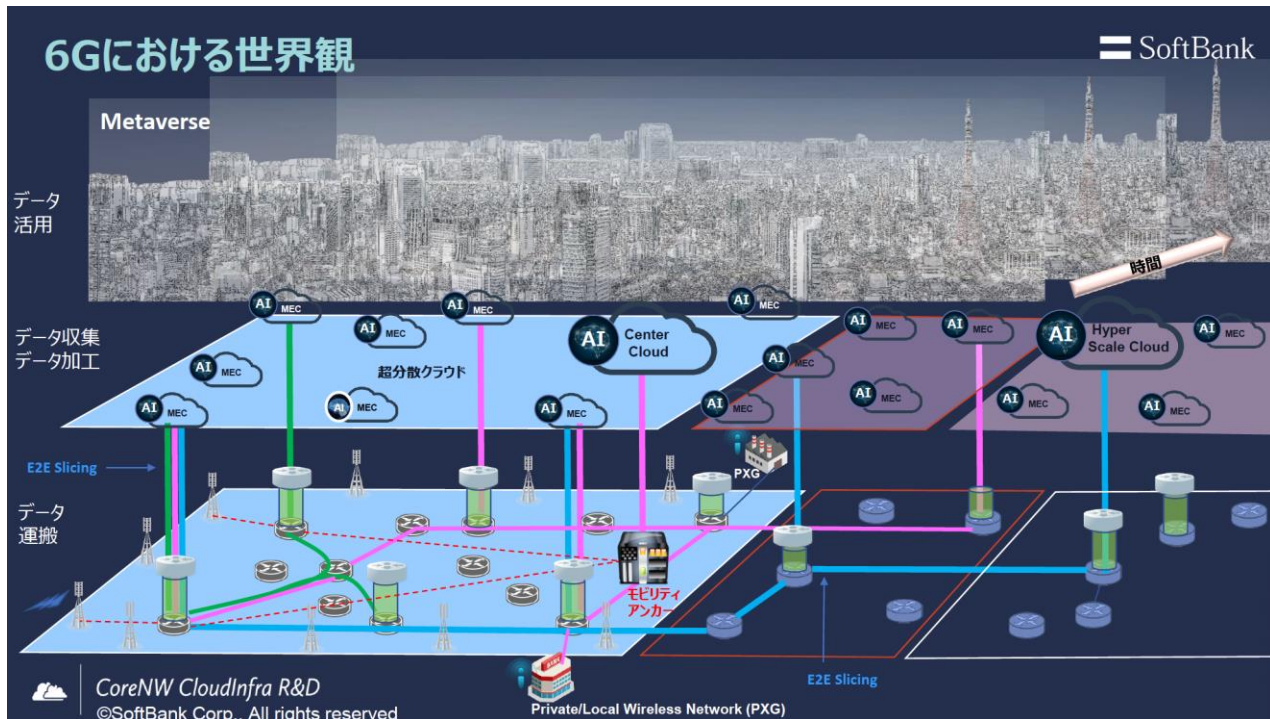
ソフトバンク (株)
テクニカルマイスター
松嶋 聡

Chair



シスコシステムズ (同)
Distinguished Systems Architect
業務執行役員
河野 美也

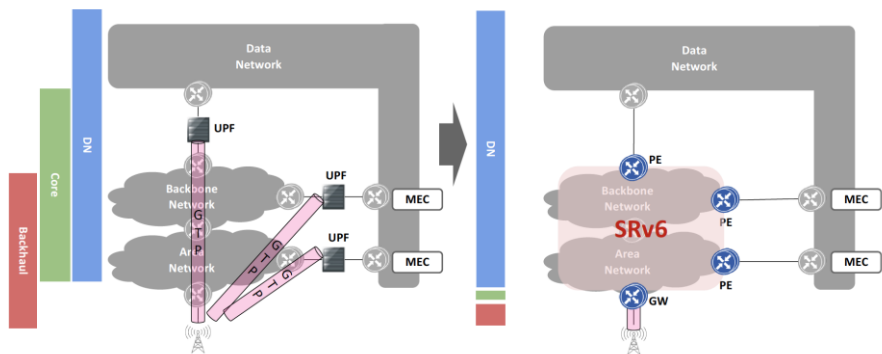
Keynote



これからの新モバイルアーキテクチャ - SRv6 MUP と ICN -

SRv6 MUP (Mobile User Plane)

- Transform 5G user plane architecture from telco's legacy circuit-switching to the latest IP routing. (i.e., SRv6)



SRv6 MUP

ICN

活用例1: スマートシティのための高レスポンス通信

- 車車間、車-モノ(信号、路肩、路面沿いのビル・お店など)間、モノ-モノ間通信
 - 周辺の**事故情報**、**路面状態**(凍結など)、**渋滞情報**などを車車間、車-モノ間でリアルタイムに交換
 - 重要な情報は集約後にクラウドへアップロード
- デジタルサイネージやAR、CPS (Cyber Physical System)
 - 必要な情報(例: **観光情報**、**レストラン情報**)を、場所や条件にマッチするヒト・モノのみに配信

Interest (ccnx:/navigation/**事故**)
(T=Location, V=位置, 範囲)(T=Time, V=時間範囲)

Interest (ccnx:/AR/**観光**)
(T=Location, V=位置, 範囲)(T=Time, V=50年前)



Multi Cloud Networking と NaaS

YA2-01

マルチクラウドネット
ワーキングへの希
望・欲望と現状

Speaker



日商エレクトロニクス (株)
エンタープライズ事業本部
クラウドサービス事業推進室
チーフコンサルタント
坂口 武生

Speaker



PacketFabric
CEO
Dave Ward

Speaker




(株) ミクシィ
開発本部
本部長
吉野 純平

Chair



シスコシステムズ (同)
Distinguished Systems Architect
業務執行役員
河野 美也

再生 (k)




PACKETFABRIC

Agile Cloud Core as a Fully Automated Private Alternative to the Internet

InterOp Tokyo Conference 2022

Dave Ward
Ken Gray





Agenda

- Interop Tokyo Conference について
- SRv6最新情報
 - 標準化動向
 - Network as a Platformに向けて

SRv6 - 標準化動向

- **RFC 8402** * – Proposed Standard
 - SR-MPLS with MPLS dataplane and Label SID's
 - SRv6 with SRH and SRv6 SID's
- **RFC 8754** – Proposed Standard
 - SRv6 DataPlane: SRH and SRv6 SID
- **RFC 8986** – Proposed Standard
 - Network Programming (END, END.X, END.DX/DT, H.Encaps)
- **RFC 9252** – Proposed Standard
 - BGP Overlay Service Based on SRv6
- **RFC 9259** – Proposed Standard
 - OAM in SRv6
- **もう間もなく RFC**
 - SR Policy (draft-ietf-spring-segment-routing-policy-22) *
 - IS-IS extension (draft-ietf-lsr-isis-srv6-extensions-18)
 - BGP-LS extension (draft-ietf-idr-bgpls-srv6-ext-09)



RFC 8986

SRv6 Network
Programming

RFC 8754

IPv6 Segment
Routing Header

100M *live subscribers*
over SRv6

SIMPLICITY ALWAYS PREVAILS

* SR-MPLS/SRv6 共通

SRv6 micro SID (uSID)

当初は方式が乱立

- uSID (Cisco)
- CRH / SRm6 (Juniper)
- G-SRH (China Mobile)
- CSID (Huawei)
- ...

IETF SPRING WGにおいて、SID 圧縮方式の分析・検討を行う Design Teamが発足

- draft-ietf-spring-compression-analysis
 - > Cisco, China Mobile, Huawei は、SRv6 (RFC8754/8986) との互換性を強調し、CSIDとしてMerge
 - > CSID に 2つの Flavor : NEXT (uSID), REPLACE (G-SID)

SRv6 micro SID (uSID)

- SRv6 との 互換性、SRv6機能 のサポートが重要とされる

	CRH	CSID	UIDSR	VSID
Encapsulation Header Size				
Forwarding Efficiency				
State Efficiency				
SRv6 Based				
SRv6 Functionality				
Heterogeneous SID Lists				
SID List Length				
SID Summarization				
Operational Requirements				
Scalability Requirements				
Protocol Design Requirements				
Security Requirements				

Legend

Requirement is met
Shade shows relative degree based on conclusion text.

Most Least

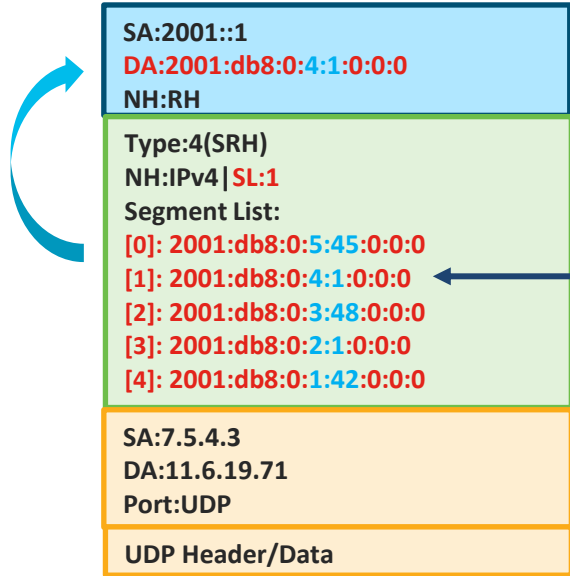
Requirement is not met

Source: SRv6 Compression Design Team Status Report, IETF-111

- 現在は、Huaweiも uSID (NEXT) を実装を表明

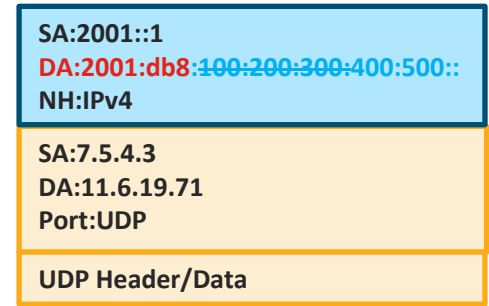
SRv6 micro SID (uSID)

SRv6 Base Encapsulation



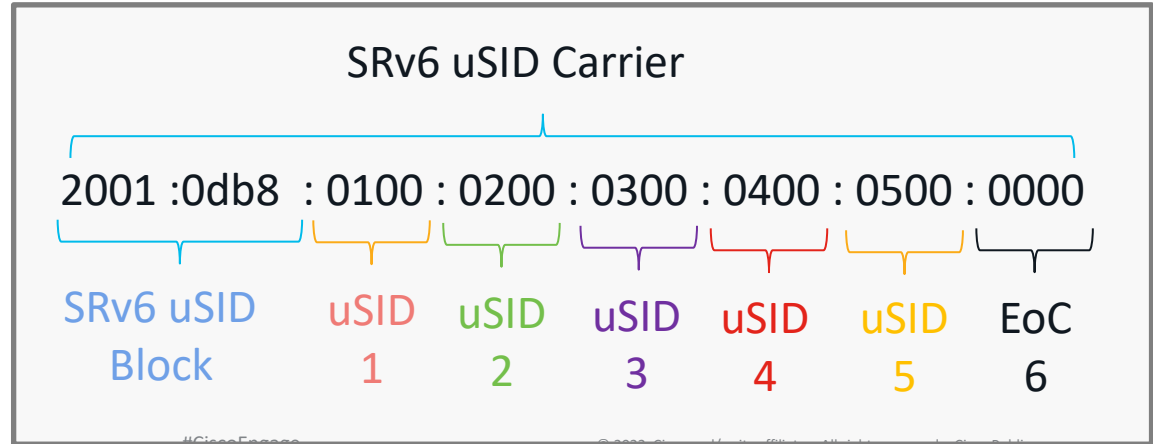
SID = [Locator + Function (+ Args)]

SRv6 uSID Encapsulation



Longest match
Shift & Forward !!

SRv6 uSID Carrier



SRv6 micro SID (uSID)

- Basic SID/uSID の混在

SA:2001::1
DA:2001:db8:100:200:300:400:500:600
NH:RH

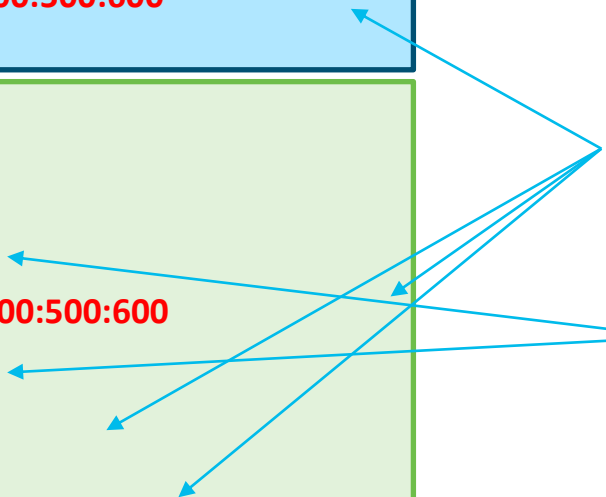
Type:4(SRH)
NH:IPv4 | SL:1
Segment List:
[0]: 2001:1:0:1:44::
[1]: 2001:db8:100:200:300:400:500:600
[2]: 2001:1:0:2:1::
[3]: 2001:db8:700:f111::
[4]: 2001:db9:500:800:900::

SA:7.5.4.3
DA:11.6.19.71
Port:UDP

UDP Header/Data

SRv6 uSID Carrier

SRv6 SID



SRv6 と IPv6 アドレスアーキテクチャとの関係

提示された疑問:

- SRv6 SID は IPv6 アドレスアーキテクチャに違反しているのではないか？
- uSID の操作は、IPv6 data plane の概念を変えるものではないか？

SID 定義の明確化:

- SID は RFC4291 IPv6 アドレスとは異なる
- エンドホストのインターフェースに割り当てられるものではない
- IPv6 DA に現れるが、SRv6-non-aware である中間ノード (transit node) のルーティングのみに使用される
- RFC7608 (variable prefix length) が適用可能

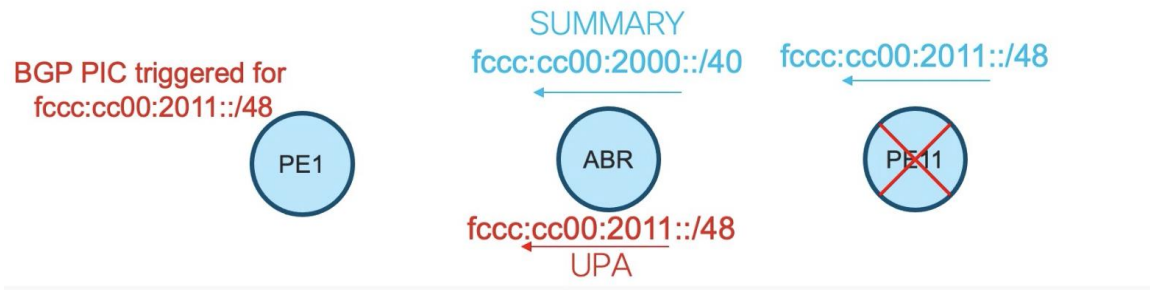


Unreachable Prefix Announcement

- SRv6 の強みは IPv6 サマライゼーションによるスケーラビリティ
- しかし、inter-area/domain サマライゼーションにより、特定のprefix障害時にコンバージェンスできない

(e.g. Remote domain の PE に障害が起きても、BGP PIC が動作しない)

→ Unreachable Prefix Announcement



- I-D : draft-ppsenak-lsr-igp-ureach-prefix-announce
- Demo : <https://www.youtube.com/watch?v=i28RgnUcD4w>

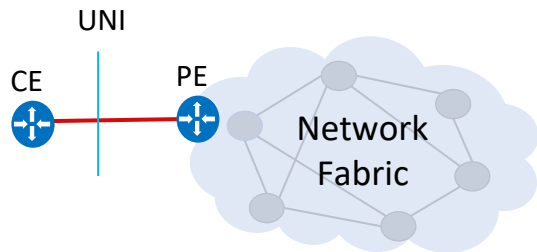


Agenda

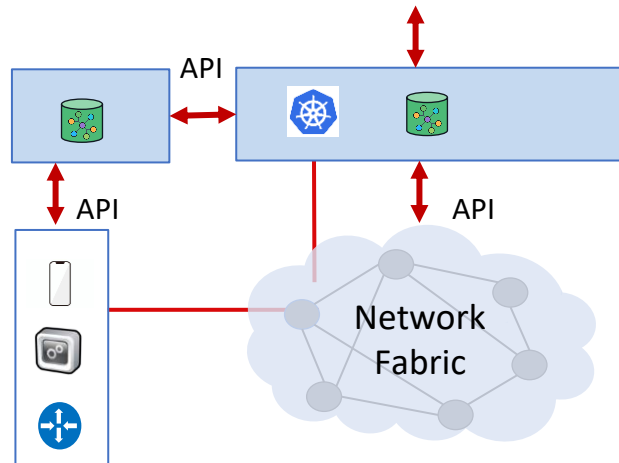
- Interop Tokyo Conference について
- SRv6最新情報
 - 標準化動向
 - Network as a Platformに向けて

Network as a Platform

Traditional SP Network



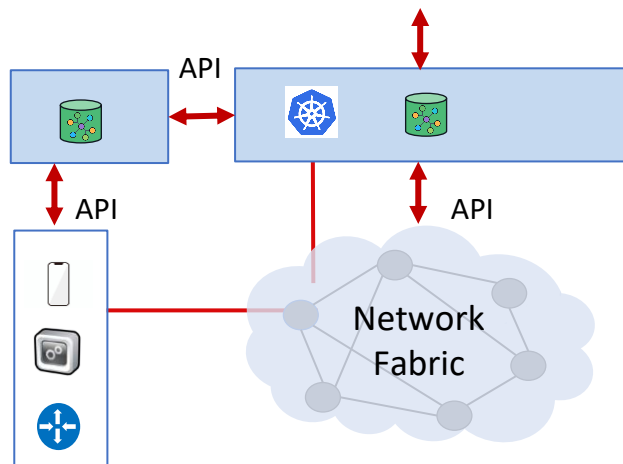
SP Network as a Platform



- ユーザとの境界は UNI ではなく **API**
- ネットワーク終端点は PE ではなく **host**
- ネットワークポリシーを制御するのも **host**

Network as a Platform

SP Network as a Platform



- 適切なAPIを提供するためには、ネットワークの状況を適切に把握する必要がある
- 必要なテレメトリデータを収集し、活用する

Path Tracing

Demand Matrix

Building Blocks for Network Slicing

Path Tracing

- IETF draft

[\[Search\]](#) [\[txt|html|xml|pdf|ized|bibtex\]](#) [\[Tracker\]](#) [\[Email\]](#) [\[Diff1\]](#) [\[Diff2\]](#) [\[Nits\]](#)
Versions: [00](#) [01](#)

SPRING
Internet-Draft
Intended status: Standards Track
Expires: 1 December 2022

C. Filsfils
A. Abdelsalam, Ed.
P. Camarillo, Ed.
Cisco Systems, Inc.
M. Yufit
Broadcom
T. Graf
Swisscom
Y. Su
Alibaba, Inc
S. Matsushima
SoftBank
M. Valentine
Goldman Sachs
30 May 2022

Path Tracing in SRv6 networks
draft-filsfils-spring-path-tracing-01

Abstract

Path Tracing provides a record of the packet path as a sequence of interface ids. In addition, it provides a record of end-to-end delay, per-hop delay, and load on each egress interface along the packet delivery path.

Path Tracing allows to trace 14 hops with only a 40-bytes IPv6 Hop-by-Hop extension header.

Path Tracing supports fine grained timestamp. It has been designed for linerate hardware implementation in the base pipeline.

- Demo : [Link](#)

- HW vendor + Open Source Support



Path Tracing - Packet Format

- IPv6 Header
 - SA, DA, DSCP, FL, ...
- Hop by Hop header
 - PT Option Type
 - MCD Stack
- SRH
 - SID List
 - SRH PT-TLV (TS, OIF ID, OIF Load)

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
SA			
DA			
Next Header	Hdr Ext Len	Option Type	Opt Data Len
MCD Stack			
Next Header	Hdr Ext Len	Option Type	OptData Len
Last Entry	Flags	TAG	
SID List			
Type	Length	OIF ID	OIF Load
64-bit Transmit Timestamp of Source Node			
Session ID		Sequence Number	

Path Tracing - Motivation

- ECMPが多いほど Resiliency が高いが、実際にどこを通過しているかは不明

- A → Mのパスは7種類ある！

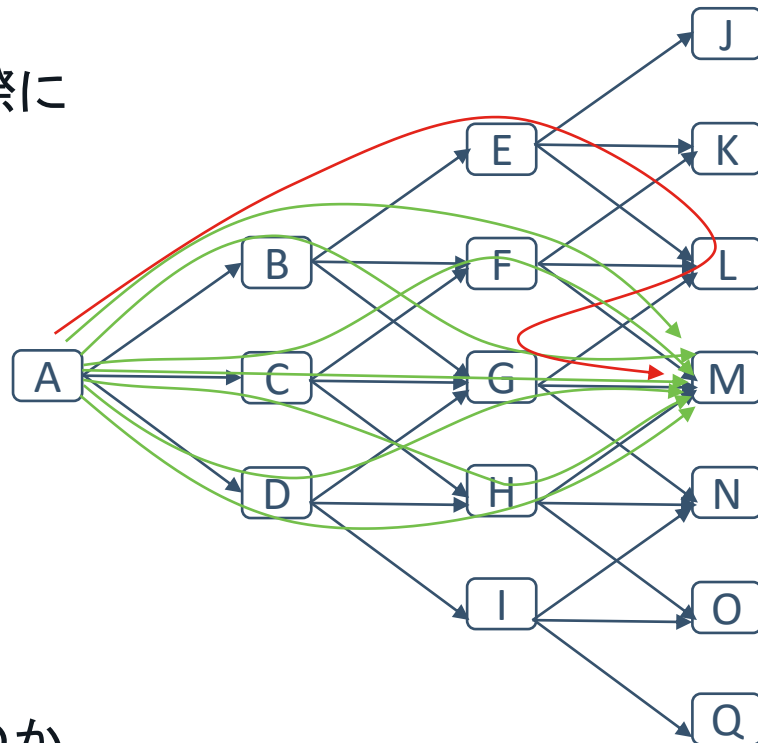
ABFM, ABGM, ACFM, ACGM, ACHM, ADGM, ADHM

- 誤りが含まれている可能性もある

ノードBのRIB/FIBが壊れた場合

- 遅延はどこで起こっているか

- 各ノードの負荷状況やポリシーはどうか



Path Tracing – 特徴

- パケット単位のdeterministic（決定論的）なトレース
- HWパイプラインにLinerateで実装
 - CPUへのpunt、Co-processorへのオフロードなし
- MTU効率の良さ：3 bytes / hop
 - Interface (12 bits), Timestamp (8 bits), Load (4 bits)
 - データ圧縮
(デコードは外部コントローラ・アプリケーションに任せる)
- SRv6/IPv6をサポート
- シームレスなデプロイメント
 - レガシーノードとのインターワーク
- まずは Probe 系から実装
 - In-band系も可能であるが、要望に応じて

Path Tracing - 収集するデータ

Path Tracing

- Source Node @A

Probeを生成するノード

- SRC.T64: 64-bit PTP Tx Timestamp
- SRC.OIL: 4-bit Outgoing Interface Load
- SRC.OIF: 12-bit Outgoing Interface ID

- Midpoint (中間ノード) @B, G

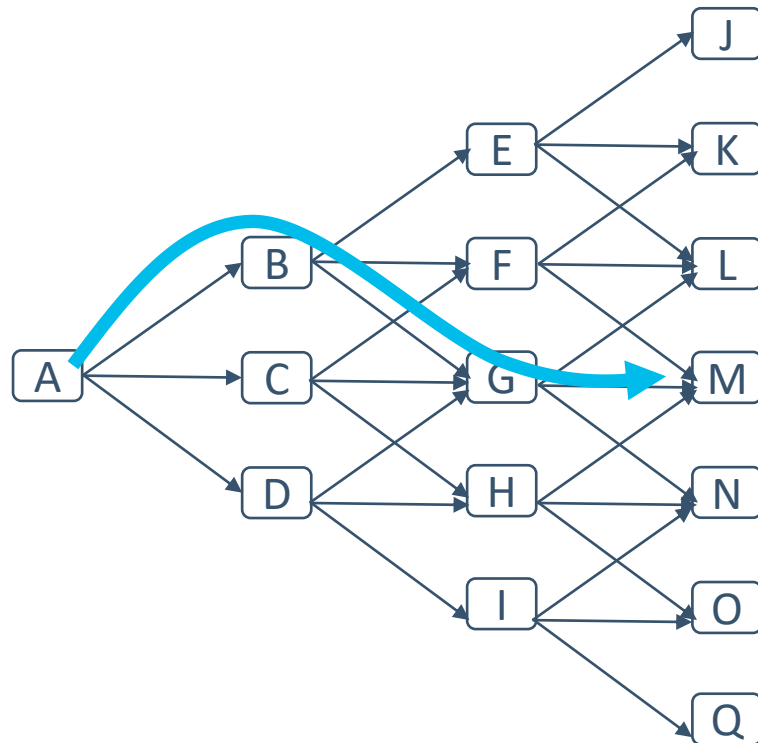
MCD: 24-bit Midpoint Compressed Data

- MCD.OIL: 4-bit Incoming Interface Load
- MCD.OIF: 12-bit Incoming Interface ID
- MCD.TTS: 8-bit Truncated PTP TimeStamp

- Sink Node @M

Probeを回収してコレクターに送信

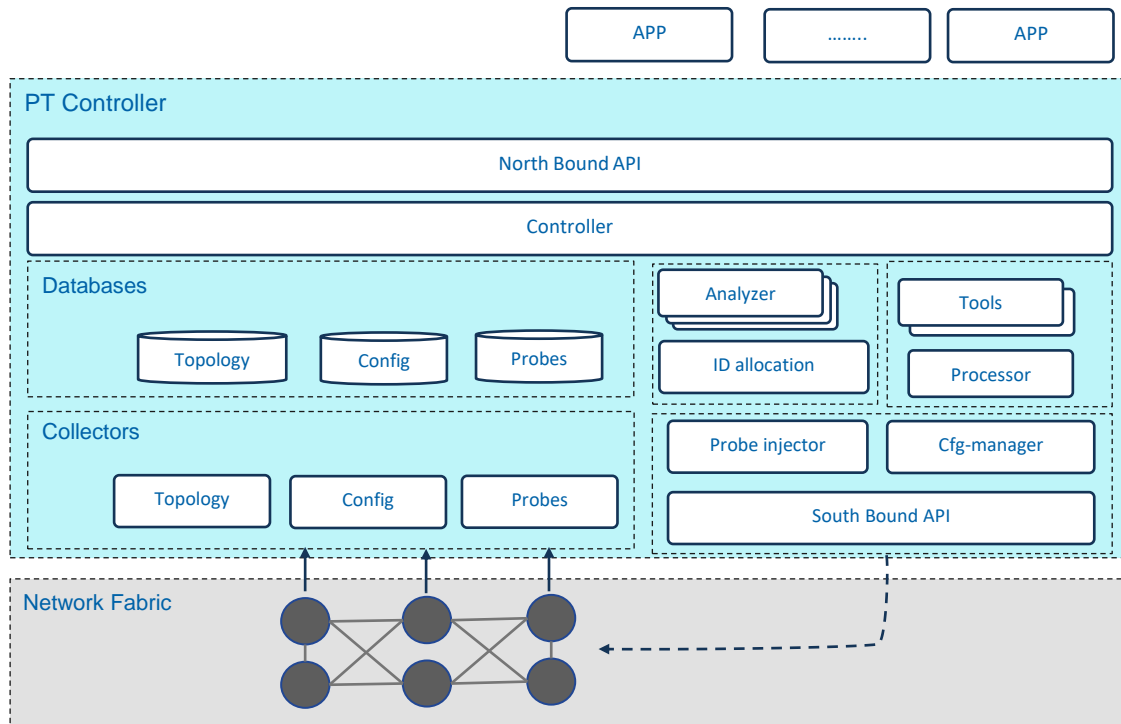
- SNK.T64: 64-bit PTP Rx Timestamp
- SNK.IIL: 4-bit Incoming Interface Load
- SNK.IIF: 12-bit Incoming Interface ID



Path Tracing - Controller

- **Controller:**
すべてのコンポーネントの呼出しや連携を制御
- **Collectors:**
トポロジー、ノードのコンフィグ、プローブデータを収集
- **Databases:**
Graph DB, TS(Timestamp) DB, Key Value Store
- **ID allocation:**
ID割り当てアルゴリズムの実装
- **Cfg-manager:**
SB-APIを使用しPT(Path Trace) configのプッシュ
- **Probe-injector:**
PT sessionの開始
- **Probe-processor:**
収集したプローブから実データを構築
- **Analyzer:** ユースケースに応じて、収集したプローブの分析

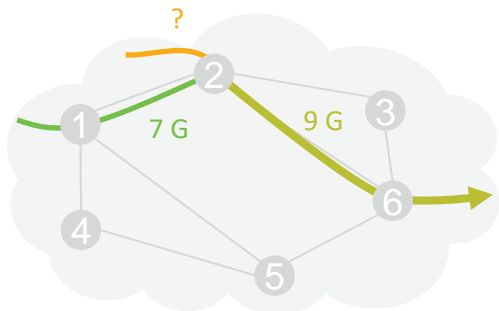
- **API:**
コンポーネント間、Network Fabricや外部ApplicationとのAPI



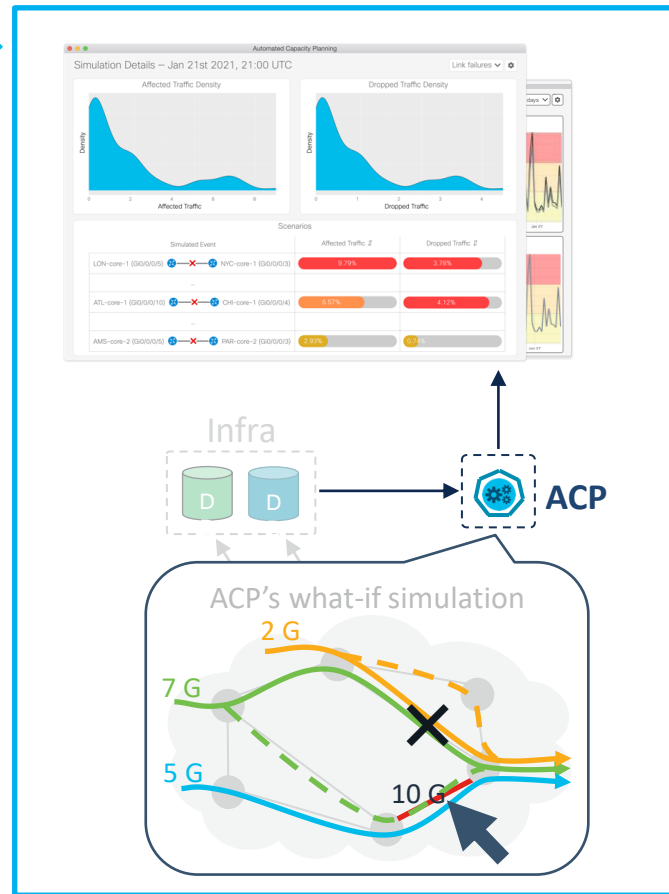
Demand Matrix + ACP (Automated Capacity Planning)

Demand Matrix : 対地ごとの、Algoごとの Traffic Matrix を計算

$$DM(u, v, a) = V_{out}(u, v, a) - V_{in}^*(u, v, a)$$

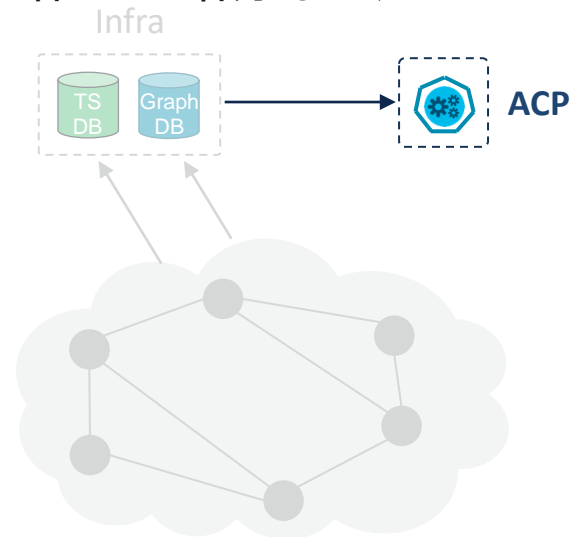


- $V_{out}(u, v, a)$: ノード u が、ノード v の Algo ロケータ a に対して送信したトラフィック量
- $V_{in}^*(u, v, a)$: ノード u が、ノード v の Algo ロケータ a に対して受け取ったトラフィック量



“what-if” 分析の完全自動化

- バックグラウンドで定期的に行（例：1時間ごと）
- ネットワークスナップショットを取得（トポロジーとトラフィック）
- Demand Matrix を計算
- 予想されるトラフィックの増加をシミュレーション（一律X%の増分など）
- 各リンク、ノード、SRLGの障害をシミュレート
- リンク飽和の可能性を報告



ライブ・ダッシュボード

動的な時系列レポート

- 時間Tの値=時間Tにおけるネットワーク上のwhat-if解析の全シリーズにおけるワーストケース
- 2つの重要な指標

> 影響を受けたトラフィック

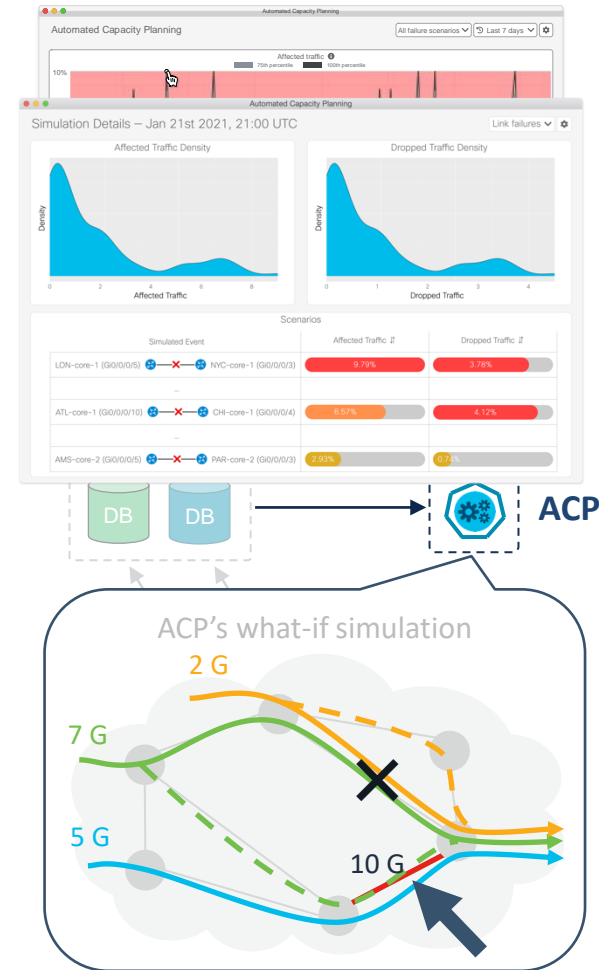
飽和リンクの総トラフィックに占める割合

> ドロップされたトラフィック

リンク容量を超える総トラフィックのシェア

時刻Tでの "ズーム・イン"

- 全シナリオの主要メトリクスの分布
- 各シナリオの詳細
- WAEでオフライン分析するためのネットワークスナップショットのダウンロード



定期的なレポート

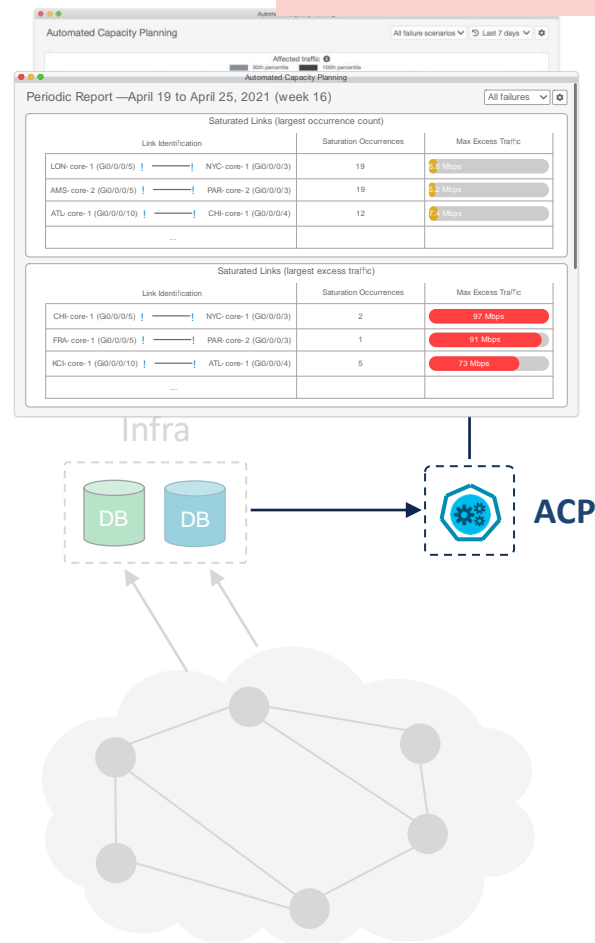
週次・月次統計の集計

・飽和リンク

- 最も頻繁に混雑するリンクのリスト（出現回数でソート）
 - > “靴の中の小石”: 影響は限定的だが、混雑の可能性は高い
- 最も混雑しているリンクのリスト(超過トラフィックでソート)
 - > “火山噴火”: 発生頻度は低いが、発生すれば影響が大きい
- → 容量アップのための候補

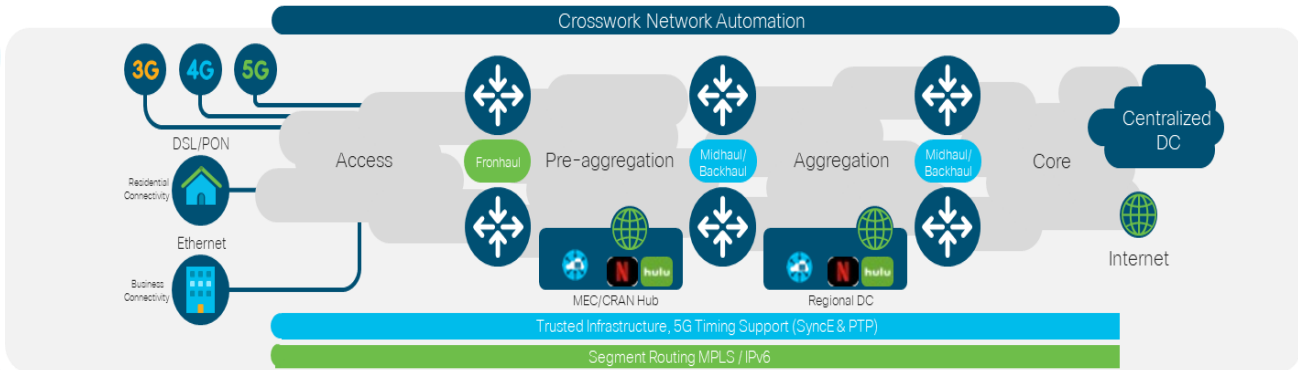
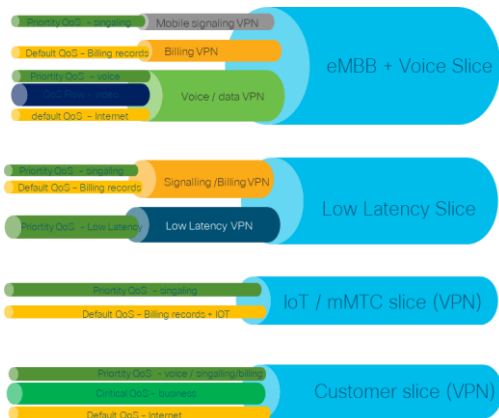
・クリティカルなネットワークコンポーネント

- ネットワークに最も頻繁に障害を起こすノード/リンク障害のリスト
- 最も頻繁にネットワークを破壊するノード/リンク障害のリスト
 - 冗長性を高めるための候補



5G Network End-to-End Slicing

ネットワークリソースとネットワーク機能を分割し、特定のビジネス目的のために、選択したアプリケーション／サービス／コネクションを互いに分離して実行できるようにする



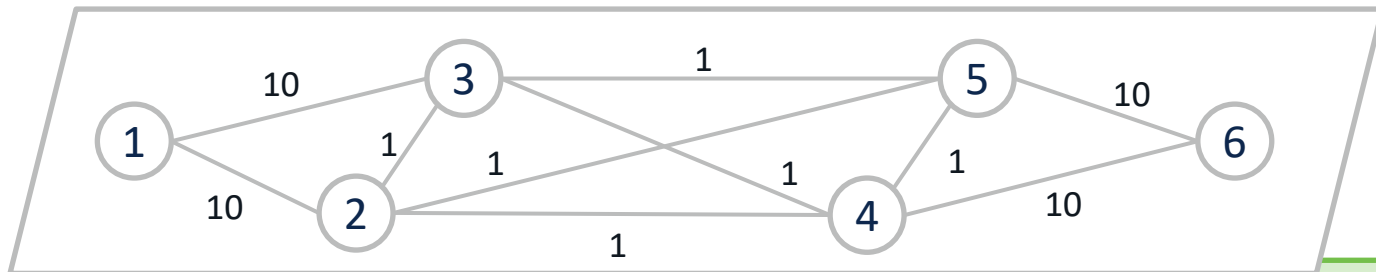
- Slice Intent achieved with Flexible Algorithm (and/or SRTE policy)
- TI-LFA with O(50 msec) protection in the slice underlay
- Automated Steering of VPN Services to slice
- SR Service Programming (NFV, SFC) and Slice Identifier
- Operation, Administration and Management (OAM) and Performance Management (PM)
- QoS using DiffServ
- Orchestration at the Controller



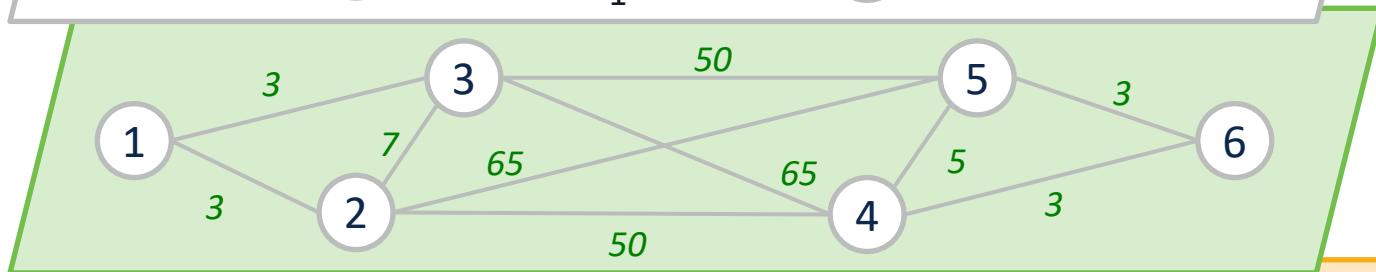
SR Flexible-Algorithmを使ったスライスの例

Building Blocks for Network Slicing

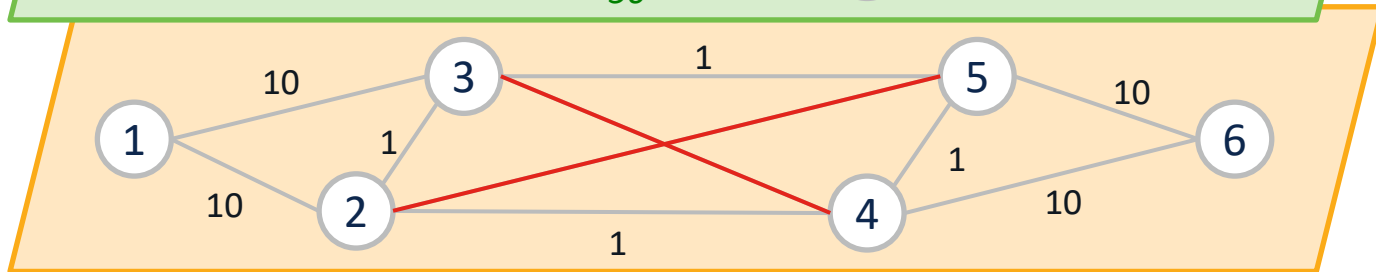
Default Slice
Algo 0



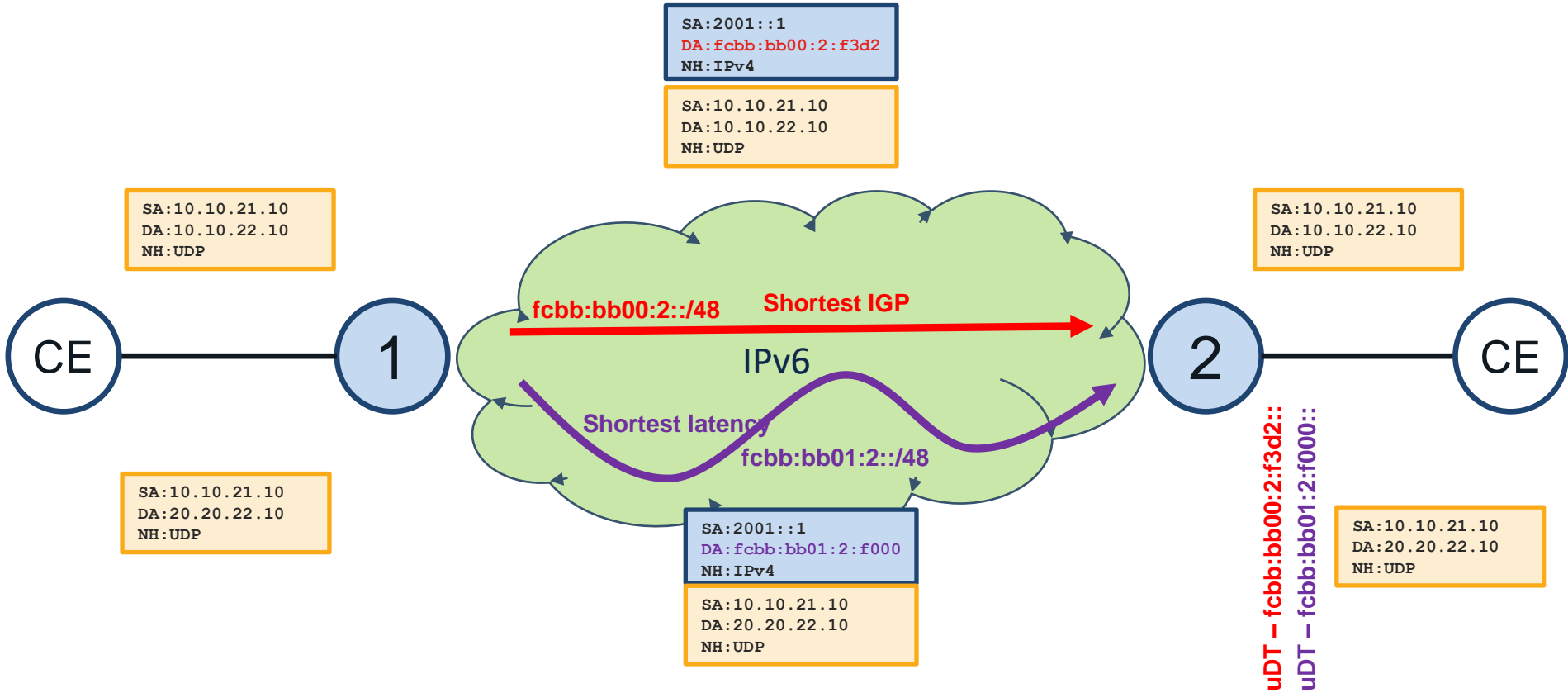
Low Latency Slice
Algo 128
(minimize delay metric)



Secure Slice
Algo 129
(minimize IGP, exclude
non-encrypted links)

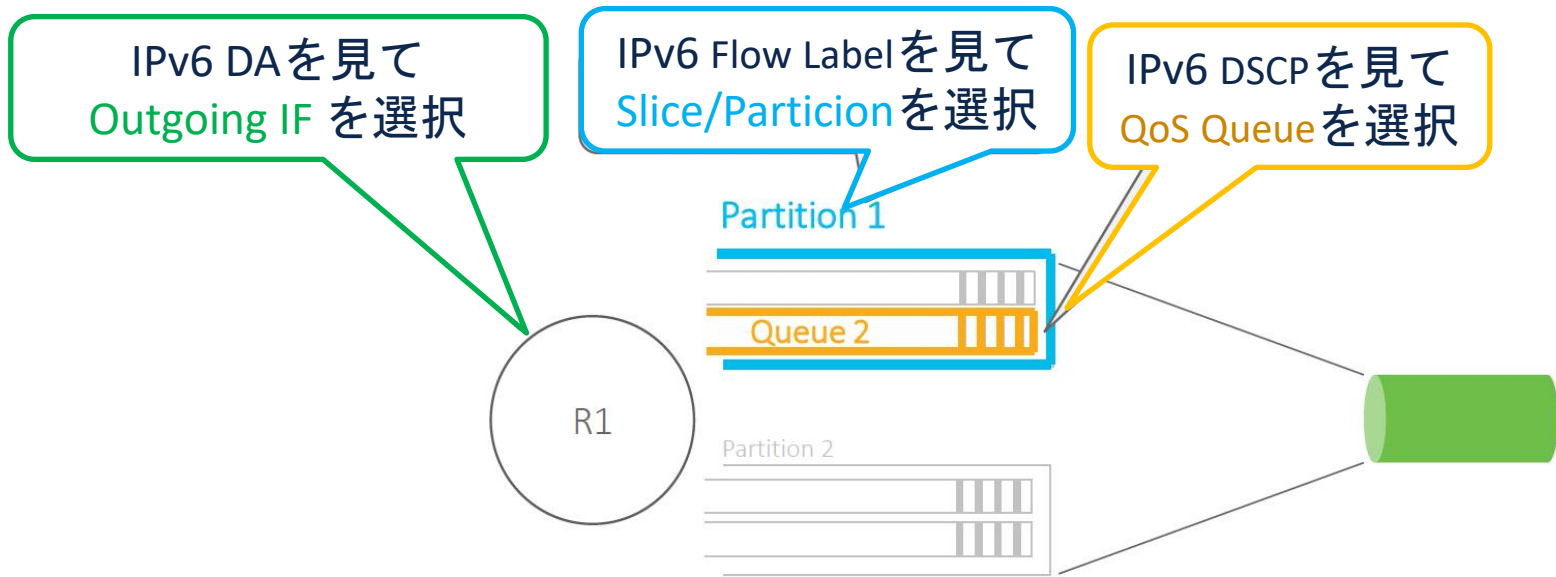


Flexible Algorithm データプレーン



ステートレス、Hop by Hopのキュー制御

Building Blocks for Network Slicing



- draft-ali-spring-network-slicing-building-blocks
- draft-filsfils-spring-srv6-stateless-slice-id



The bridge to possible

Thank you



CISCO *Engage*

#CiscoEngage