NWトポロジー可視化から時系列化へ:
BGP-LS+BMP+αによるトポロジー把握の時間軸拡張

ソフトバンク株式会社

内海、伊藤、魏

2026/07/17

自己紹介

SoftBank

内海 友輔
Yusuke Uchimi

経歴
■ 社内基盤開発（2018~2022）
オンプレ / クラウド（AWS/Azure）上の
インフラ設計や自動化開発、SRE的な業務
■ NW異常検知（2022~）
NW機器の早期異常検知、
サイレント故障検知、及び基盤運用

趣味
■ マラソン（フルマラソン3時間ぐらい）

JANOG参加歴
■ 登壇2回目
前回: JANOG54@奈良（登壇）

伊藤 達真
Tatsuma Ito

経歴
■ NWデータ分析 (2019-2023, 2026-)
Streaming Telemetry/TWAMP/…
Splunk/AWS上の分析基盤の構築、運用
■ プライベートクラウド開発 (2023-2026)
オーケストレーション、Kubernetes基盤

趣味
■ ドラム/パーカッション/編曲

JANOG参加歴
■ 初参加

魏 亮
Liang Wei

経歴
■ ISPネットワーク開発 (2021-2024）
ISPバックボーンNW PE、CGN/FW装置の
設計/導入/検証
■ NW異常検知（2024～）
Streaming Telemetryコレクタ内製化開発
NW機器シスログ分析

趣味
■ フットサル・サッカー/読書

JANOG参加歴
■ 初参加

組織の紹介

SoftBank

■ 組織の立ち位置

NW機器の導入・運用部門とは異なる
少し特殊な立ち位置の部署。

■ 主な目的

- NW機器の異常検知、大量データ収集、自動検知の仕組み化
- 様々なデータソースの利活用検討およびトライアルの実施

ネットワーク主管部門

導入主管部

運用主管部

NW機器の検証導入や保守運用を担当

依頼, 相談

実装, 提案

所属組織



自動化や異常検知、データ分析

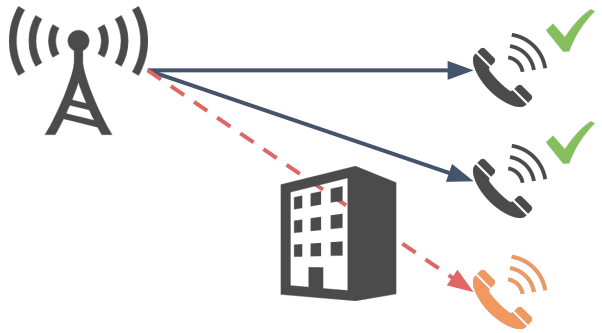
クローズドループの取り組み展開
自動化検討、新規データの取得活用

そもそもNetwork Digital Twinとは？

無線の世界 (RAN/モバイル)
におけるNDT

■ **イメージしやすい物理を模した世界（と想定しています）**

- 例えば、3Dマップ上にビルを建て、電波の遮蔽や伝搬をシミュレーション
- チルト角や出力などのパラメータ変更で、エリアカバーがどう変わるかが視覚的に分かる



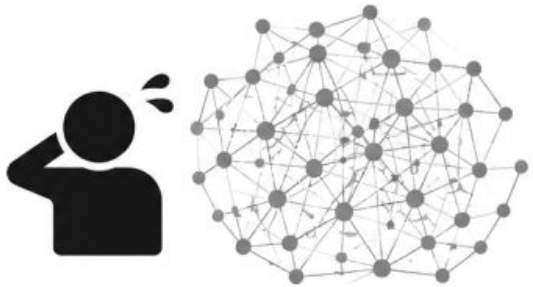
伝送網・IP網
におけるNDT

■ **イメージしにくい論理の世界**

- 物理的な遮蔽物はない
- Configやルーティング、ポリシー等の集合

■ **そもそも、何を指すのか？**

- Configのバックアップ
- トポロジーを描く、Metricsを収集する
- **どのパラメータを変更して、何を検証、シミュレーションすればいいのか？**



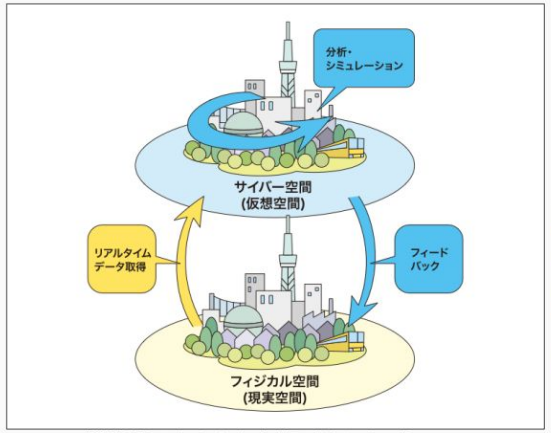
直近のJANOGにおける、NDTやNW監視に関する言及

SoftBank

ネットワークのデジタルツインに求める要件は何ですか？
～理想的な仮想環境への期待と現状～

（JANOG57）

デジタルツイン



情報通信白書 for Kids：インターネットの活用：デジタルツインって何？
https://www.soumu.go.jp/hakusho-kids/use/economy/economy_11.html

ネットワーク・デジタルツイン(NDT)は、
これをネットワークシステムに対して
応用するもの

IETF(NMRG)やITUのNDT関連資料では現実-仮想「マッピング」みたいな言い方

デジタルツイン（Digital Twin）とは
現実世界から集めたデータを基に
デジタルな仮想空間上に双子（ツイン）を構築し、
さまざまなシミュレーションを行う技術である。

総務省｜令和6年版 情報通信白書｜デジタルツイン
https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r06/html/nd217530.html

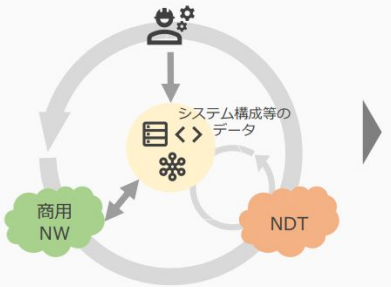
	デジタルツイン	シミュレーション
再現環境の違い	現実世界をコンピュータ上の仮想空間に再現する	コンピュータ以外(模型など)でも実行されることがある
リアルタイム性の違い	現実世界の状況に合わせて、その都度変化するためリアルタイム性が高い	「ある時点でのモデル」を活用する
精度の違い	実在する現実世界を再現する	いくつかの仮説をベースに実行する

デジタルツインとは？わかりやすく最新技術を徹底解説。メリットや導入事例も！｜SCSK IT Platform Navigator
https://www.scsk.jp/sp/itpnavi/article/2024/11/digitaltwin.html

NDT: 従来の"ラボ環境"との違い?

従来: 設計上想定される機能、動作パターンの確認

NDT: 想定しきれない現実世界の構造・状態に基づく
- 「現実世界」の構造や動き(ふるまい)の再現を基準にする
- 現実世界の時間軸(リアルタイム性)を考慮する



データモデリング・データを基にした効率化
- 対象を詳細に表現するデータモデル
- モデル上での検証(verify)・シミュレーション・分析
- 自動化・宣言的アプローチ (仮想NW型)

クラウドネイティブ的な考え方・運用プラクティス
- CI/CDプロセス
- Reconciliation Loop
- 一時的(ephemeral)な検証環境 (仮想NW型)
  - APIベースのリソース操作

- ネットワークをマルチレイヤで構成される
  グラフとしてモデル化することで
  仮想環境上に構成をコピーする取り組み
- 現実世界から集めたデータを基に再現環境を
  構築し、シミュレーションを行う技術

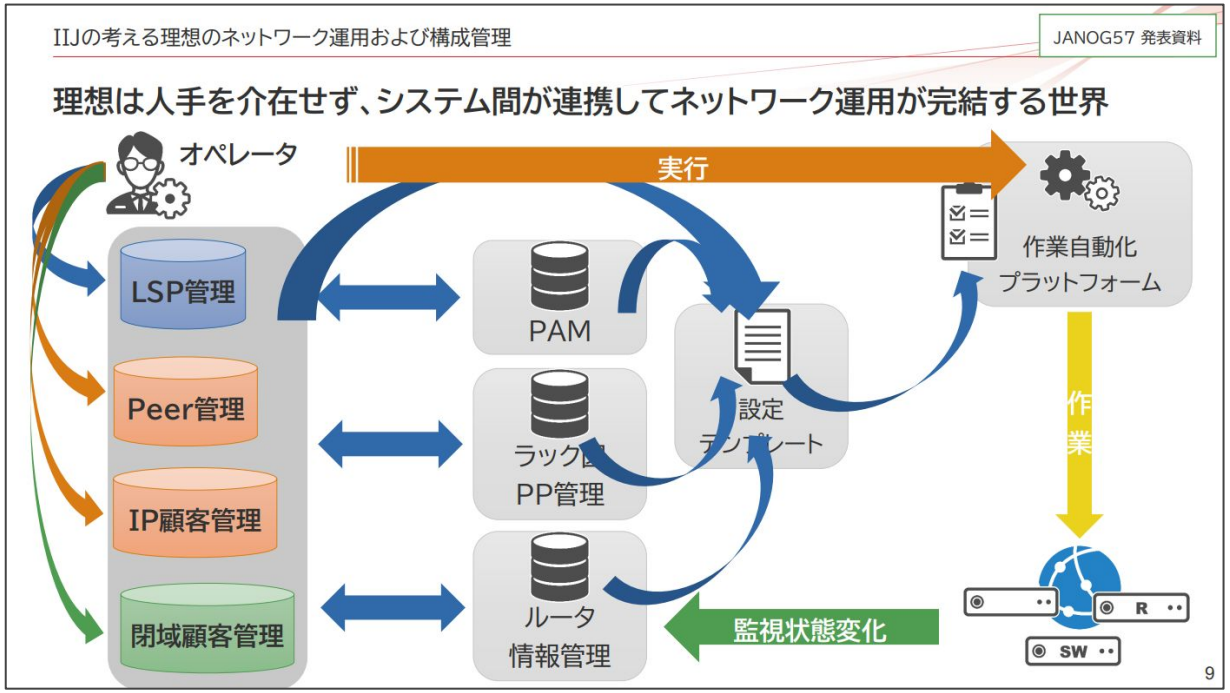
- 従来のラボ環境は想定機能、動作パターンを再現
- NDTでは、想定しきれない現実世界の構造を再現
  - 現実のふるまいの再現
  - リアルタイム性も考慮

[1]: https://www.janog.gr.jp/meeting/janog57/digital-twin/

直近のJANOGにおける、NDTやNW監視に関する言及

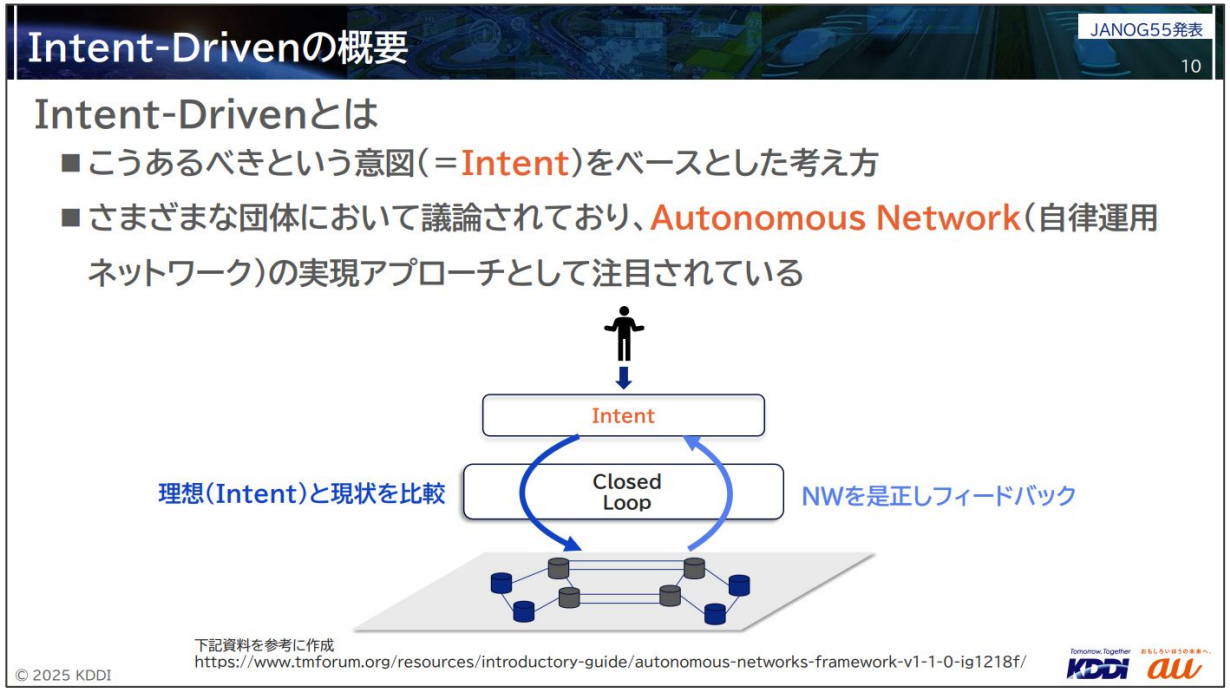
SoftBank

構成管理はどこまでやるべきか？ ネットワーク運用の未来を見据えた検討と実践 （JANOG57）



- NW要件の多様化により手動運用は限界に。構成情報を適切なデータ形式で管理する必要有り
- 理想は人手を介在せず、システム間が連携してネットワーク運用が完結する世界

Kubernetesを活用した、Intent-Drivenアプローチに基づくネットワーク設計・作業自動化の取り組み （JANOG55）



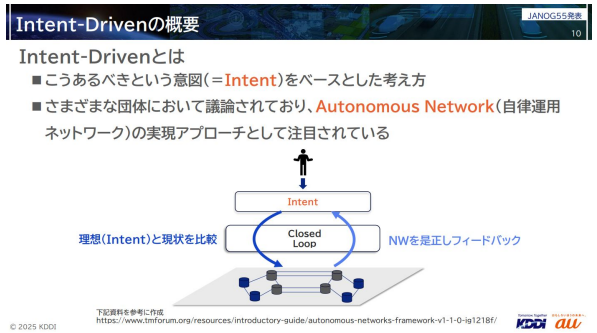
- 抽象的な指示の入力から必要なConfigの生成、及びConfig投入までの自動化を実現
- 将来的には機器から取得したテレメトリーを基にした自律的なネットワーク制御を実現

[2]: https://www.janog.gr.jp/meeting/janog57/p-mgr/
[3]: https://www.janog.gr.jp/meeting/janog55/k8s/

本報告での共有内容について

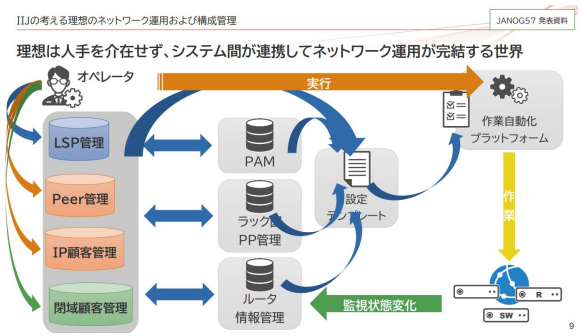
SoftBank

我々はL3トポロジーに特化したリアルタイム、及び時間軸を考慮した観測用のシステムを検討した。
そちらの取り組みを基に、今後の拡張やNDTに対して、どのように検討/運用されているか議論したい。



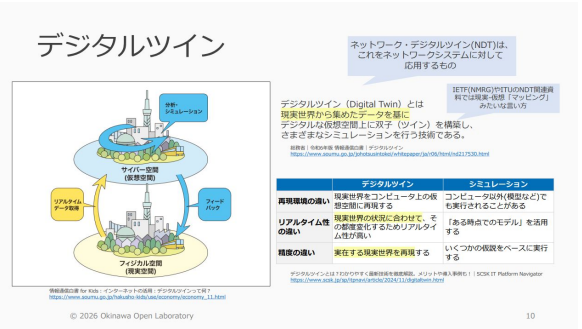
- **抽象的な指示の入力から必要なConfigの生成**
- **及びConfig投入までの自動化を実現**

実際にL3のルーティングトポロジー上で
意図通りのトラフィック分散を生み出したか、
想定外の迂回を生んでいないかを確認する
動的な可視化の仕組みが必要になると想定。
（= IntentとStateの整合性があっているか）



- **手動運用は限界に。**
- **構成情報を適切なデータ形式で管理する必要有り**

L3状態変化の反映を手動ではなく
動的でリアルタイムに反映したい。
トポロジーはGraphDBとして扱うが、
あくまで隣接情報のみ。他データは他DBで管理。



- **NDTでは、想定しきれない現実世界の構造を再現**
- **リアルタイム性も考慮**
- **結局見たいのは、複雑なNW環境の動きやふるまい**

過去事象再現から信用積み上げという話は仰る通り。
商用NWのある時点のトポロジー = snapshotを
把握する仕組みも必要と想定し、そちらを検討した。
静的な構成情報は、後からマッピングする必要がある
※ 今回はスコープ外

なぜ L3トポロジー特化（GraphDB）を検討したのか

SoftBank

トポロジー表現にはGraphDBが適している

その一方でネットワークエンジニアにとって
グラフ用のクエリやグラフ理論の
学習コストが高かった。

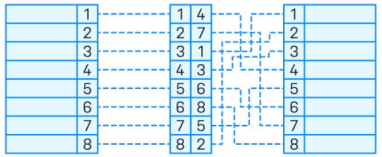
学習コストをAI-Agentである程度吸収でき得る

■ 自然言語からGraphDBのクエリを自動生成

LLMを活用したAI Agentが、自然言語や
ネットワークの文脈からグラフクエリを自動生成。

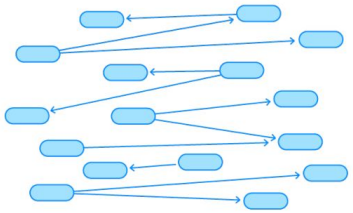
そこまで深いクエリ知識がなくても、
ネットワーク運用の知見（ドメイン知識）で
GraphDBを扱うことができるように。

RDBMS



- 学習コストが比較的低い
- ネットワークのような多対多の
  接続関係をSQLで書くと、
  大量のJOINになり非常に複雑化

GraphDB



- ルータ（Vertex）と
  リンク（Edge）の表現が
  ネットワークそのものであり、
  探索や再構成に最適
- クエリ言語が特殊で、
  学習コストがやや高い

参考）自然言語からGraphDBのクエリを自動生成

現時点での結論

SoftBank

NDTの常時稼働、完全なミラーの可否について

- 実網を完全に模した、常時稼働させる様なNDTを作るのは現実的なのか？
- 実際はそちらが望ましいと思う一方で
  リソース的（コスト的）・技術的の両面でも課題はたくさん残っている

代替案

必要な部分を必要なタイミングで、切り出して（再現して）使うのが良いのではないか

- NWレイヤの観点:
  - 例えばL3レベルの検証を行う場合、完全な商用模倣である必要はない。
  - 今回は変化が激しいL3のルーティングトポロジーに対して
    リアルタイム観測と時系列化に特化し、BGP-LS+BMPによるNDTもどきを試した。
- リソースの観点:
  - 仮想環境 + シミュレーションを検討する場合にも、全網再現ではなく
    GraphDBから必要時刻の必要な範囲だけを抽出。（例: 障害時刻前後の影響ノード群）
  - 静的な構成管理データ（L2/L1）とマッピングして、オンデマンドに切り出す

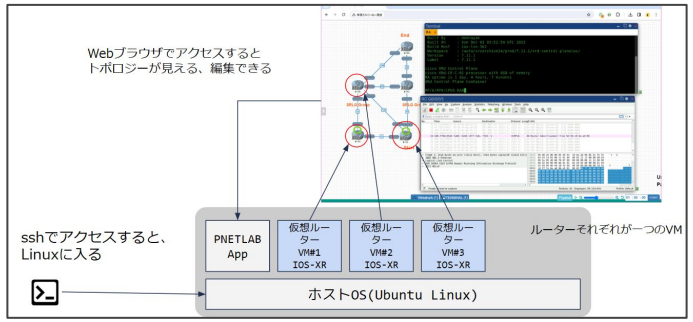
できる方が
望ましい

NDTの利用状況（一部）

用途特化型のNDTはすでに実在し、社内でも活用されている

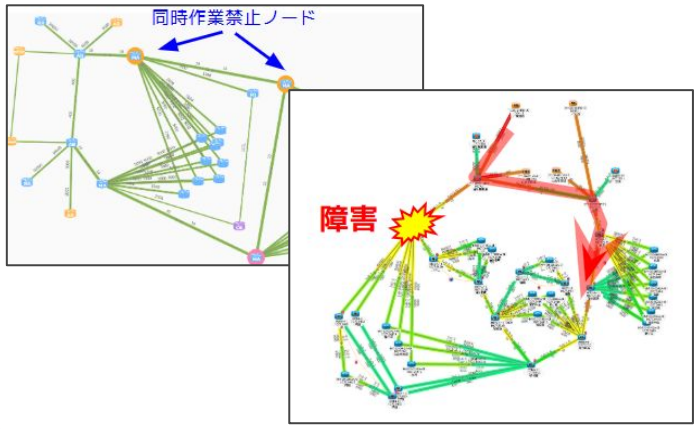
検証用途

PNETLAB（をラップしたもの）
仮想環境を自動で立ち上げ



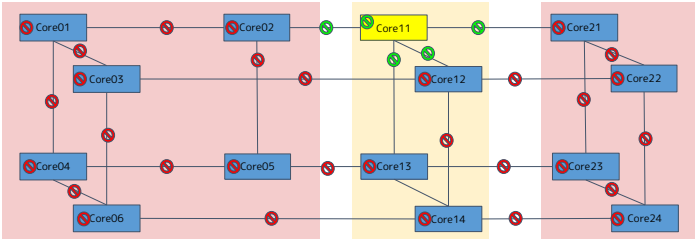
可視化/構成管理

静的情報ベースのNDT



運用判断

クローズドループにおける
対処可否判断のためのNDT



Autonomous Network に対する動的監視のアプローチ

SoftBank

Level 3: ルールベースの自動化

特定のドメインにおいて、事前に設定した
ポリシーや運用ルールに沿って
ネットワークの自動最適化を行う段階。

- **ルールベースの自動化がメイン**
  例）事前に定義されたポリシーの中で
  システムが最適化を実行

Level 4: 意図（Intent）ベースの自動化

ネットワークをIntentに沿わせる段階。
AIや予測モデルが、その都度最適な経路や
ルータのConfig変更を判断して適用。

- **抽象的な意図（Intent）のみシステムに指示**
  例）この法人顧客の通信遅延は
  常に20ms以下に維持というIntentのみ。
  上記を基にシステムが最適化を実行

以下を継続して実施する必要がある。

1. システムが人間の意図を解釈
2. 自動検証 / 適用
3. その後の継続的な監視
4. システムが人間の意図を解釈
5. ・・

本発表での
共有部分

Autonomous networks levels

Level Definition	L0: Manual Operation & Maintenance	L1: Assisted Operation & Maintenance	L2: Partial Autonomous Network	L3: Conditional Autonomous Network	L4: High Autonomous Network	L5: Full Autonomous Network
Execution	P	P/S	S	S	S	S
Awareness	P	P	P/S	S	S	S
Analysis	P	P	P	P/S	S	S
Decision	P	P	P	P/S	S	S
Intent/Experience	P	P	P	P	P/S	S
Applicability	N/A			Select scenarios		All scenarios

P: Personnel, **S:** Systems

[4]: Operationalizing intent-based autonomy at Level 4+

既存監視とトポロジー監視で監視すべき状態

既存監視の強み

- SNMP/Telemetryなど、単体のMetricsは装置やリソース監視にとても有用
- その一方で、NW全体の論理トポロジーや経路変化の前後関係を後から追うのは困難

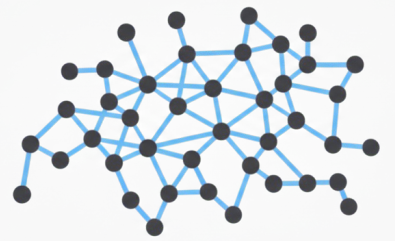
Metrics監視で把握する状態

- IFトラフィック、リソース状態
  → SNMP, Telemetry
- 通信量・フロー
  → NetFlow

トポロジー監視で把握する状態

- ルーティングの内部状態
- ノードやリンクの接続関係
- リアルタイムな経路変化

データの種別に応じて、それぞれ適した取得方式が存在



NW機器全台への設定投入前提では、
構成管理・運用負荷が大きい

▶

代表ルータから
情報を集約取得したい

BGP-LS+BMPを用いたシステムの構成及びユースケースについて

BGP-LS と BMP について

SoftBank

BGP-LS (RFC 7752 / 9552)

IGP LSDBの変換と送信

ルータが保持するIGP LSDB（Node, Link, Prefix）を変換し、
BGP Updateメッセージとしてコントローラ等へ送信。

主な用途

ネットワーク上のイベント（Link Down、Metric変更等）を
BGP経由で管理システムへ通知。

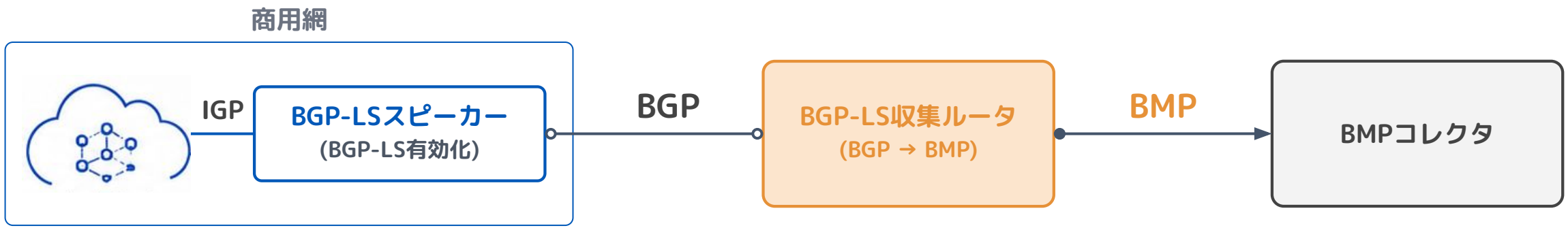
BMP (RFC 7854 / 9069)

カプセル化と転送

受信したBGP Updateメッセージをカプセル化し、
解析用サーバへ転送。

対応プロトコル

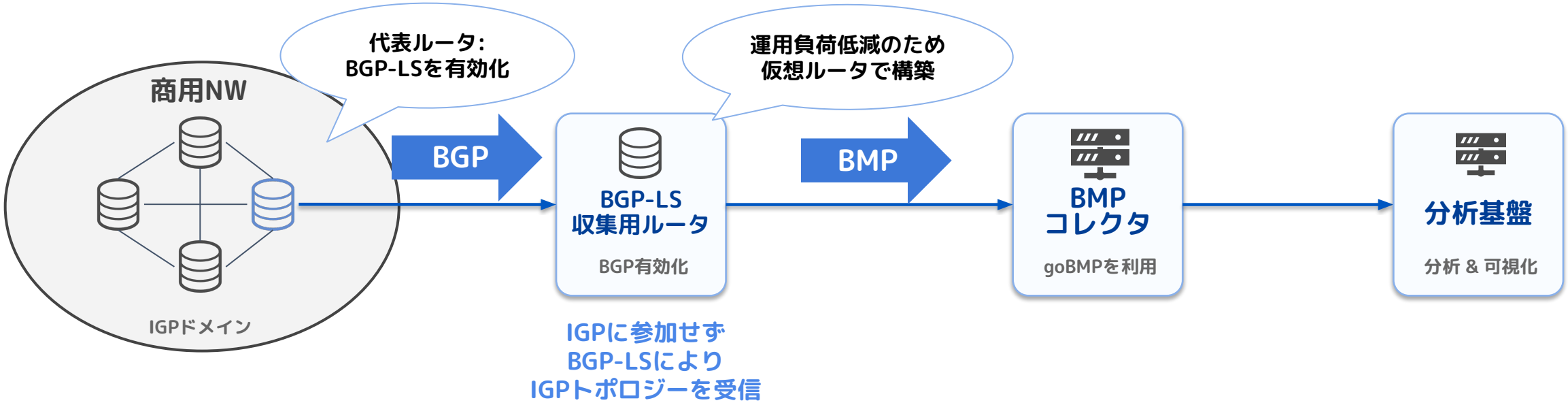
BGP-LSはBMPで転送されるBGPのAFの一種である。
VPNv4/v6やIPv4/IPv6 Unicast等の情報もBMPで転送可能。



BGP-LSデータ取得構成

構成要素

- **BGP-LSスピーカー**：IGPに参加している商用網のルータ。属しているIGP情報を保持。
  - BGP-LS収集用ルータとiBGP neighborを確立するのはこのルータのみ。
- **BGP-LS収集用ルータ**：スピーカーとiBGP neighborを確立し、IGPトポロジー情報を受信。
  さらにその情報をBMP経由でBMPコレクタへ送信。
  - 取得タイミング：IGPトポロジーに変更が発生した際、BGP-LS経由で差分情報を随時取得
- **BMPコレクタ**：収集した経路・トポロジー情報を後段の分析・可視化基盤へ送信。



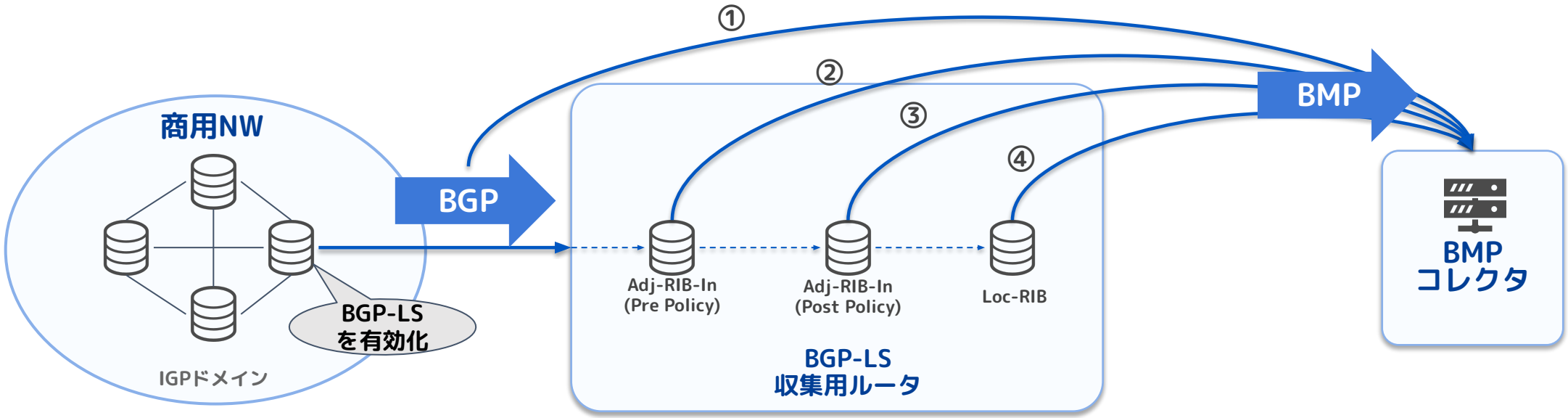
BMP転送ポリシー選定

SoftBank

BMP転送ポリシー候補

① **Route Mirroring**：peerから受信したBGP UPDATEメッセージを、受信時の内容に近い形で複製して送信。
② **Pre-policy Adj-RIB-In**：Inbound policy適用前のAdj-RIB-Inの経路状態を送信。
③ **Post-policy Adj-RIB-In**：Inbound policy適用後のAdj-RIB-Inの経路状態を送信。
④ **Loc-RIB**：BGP best path選択後にLoc-RIBへ反映された経路状態を送信。

⇒ BGP-LS UPDATEメッセージの受信内容を加工・状態圧縮せずに収集したい要件であるため
Route Mirroringを採用



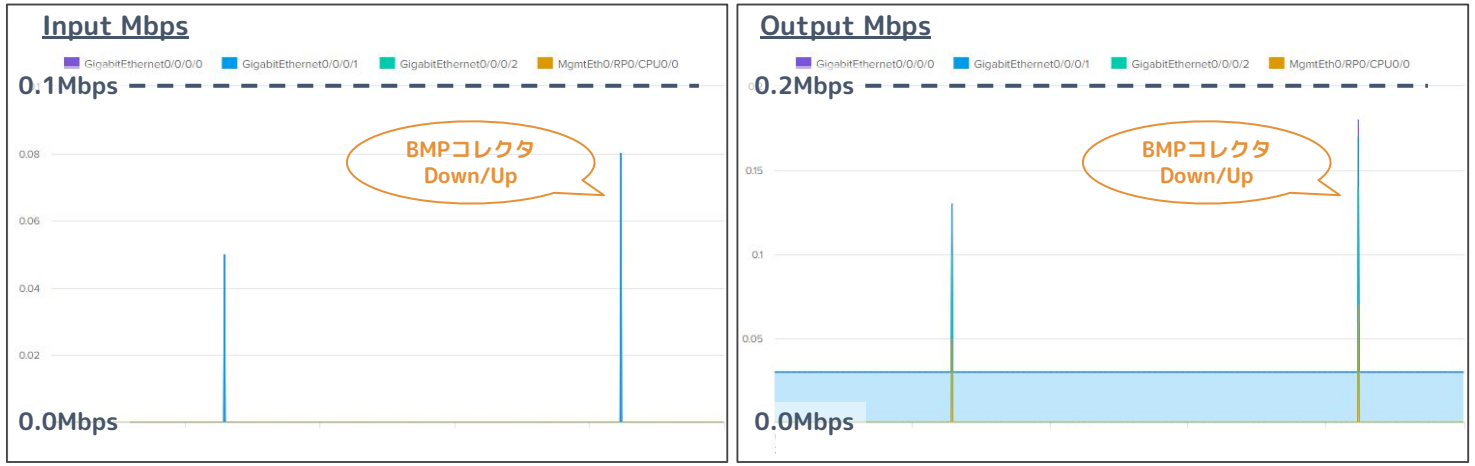
参考）BGP-LS収集用ルータのリソース変動

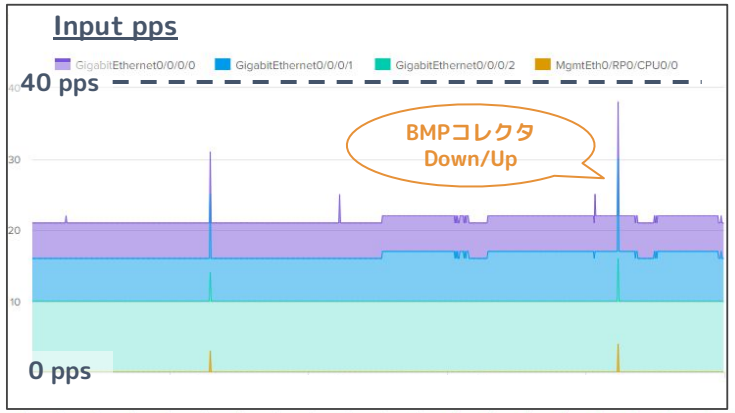
SoftBank

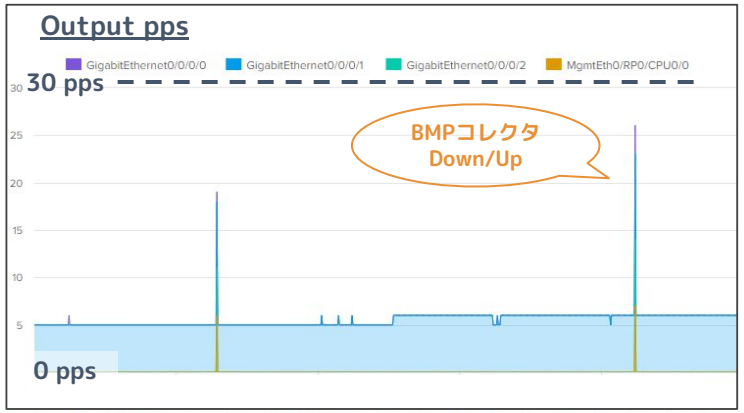
右の変動のトポロジー規模（概算）

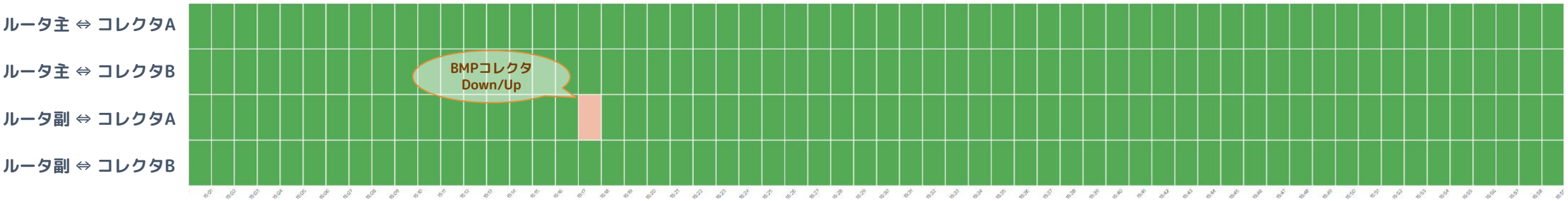
- Node:　2,000台
- Link:　6,000本
- Prefix:　20,000件

トラフィック負荷小







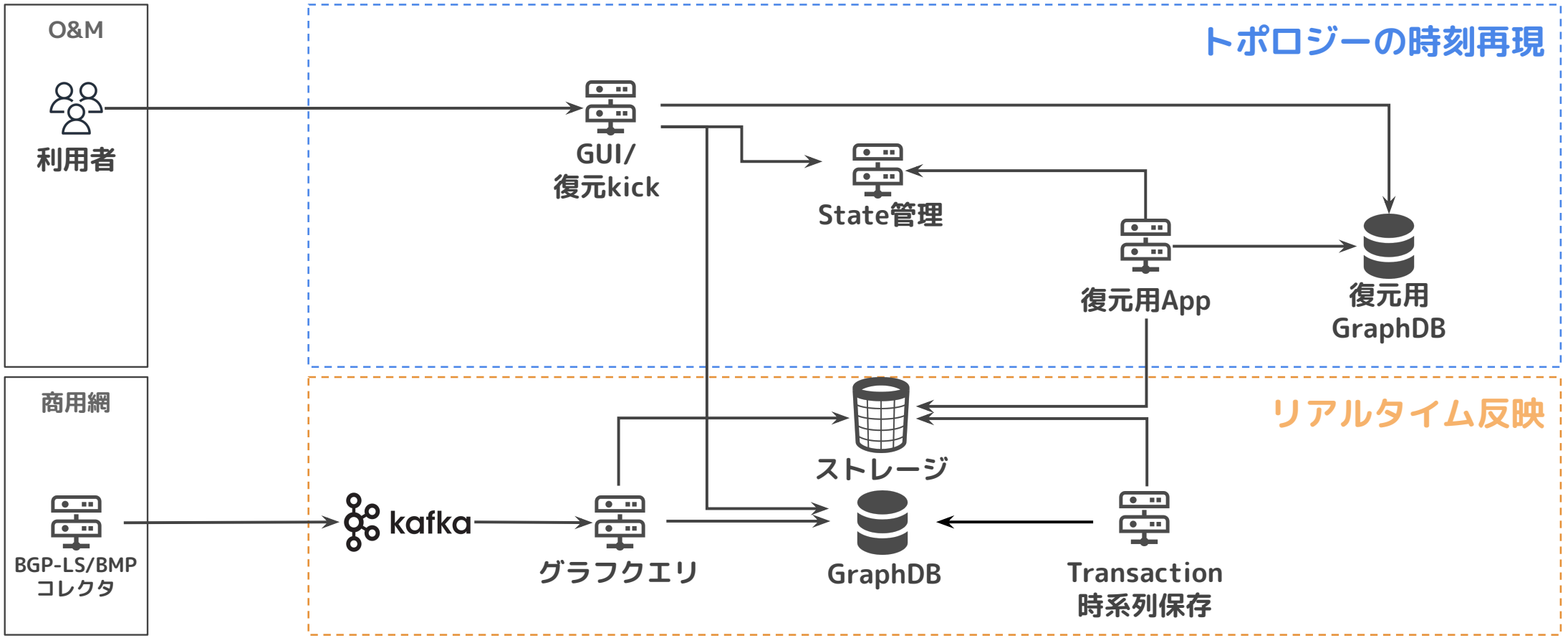


BMPコレクタとの接続状態例

分析・可視化基盤構成

SoftBank

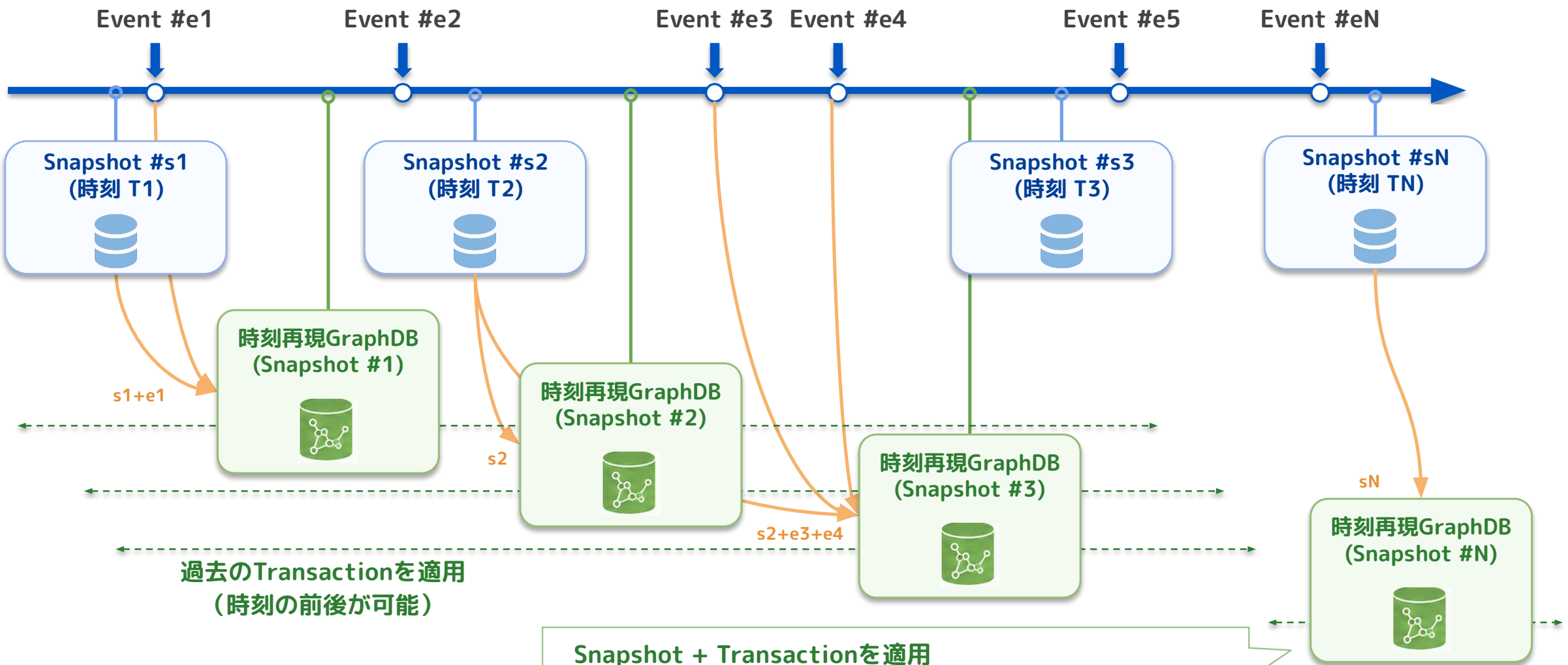
商用NWからの変更イベントを取り込み、リアルタイム反映と任意時刻の再現を両立



現在のネットワーク状態だけでなく、過去の任意時刻におけるトポロジー状態も再現可能

トポロジーのリアルタイム観測、時刻再現の考え方

SoftBank



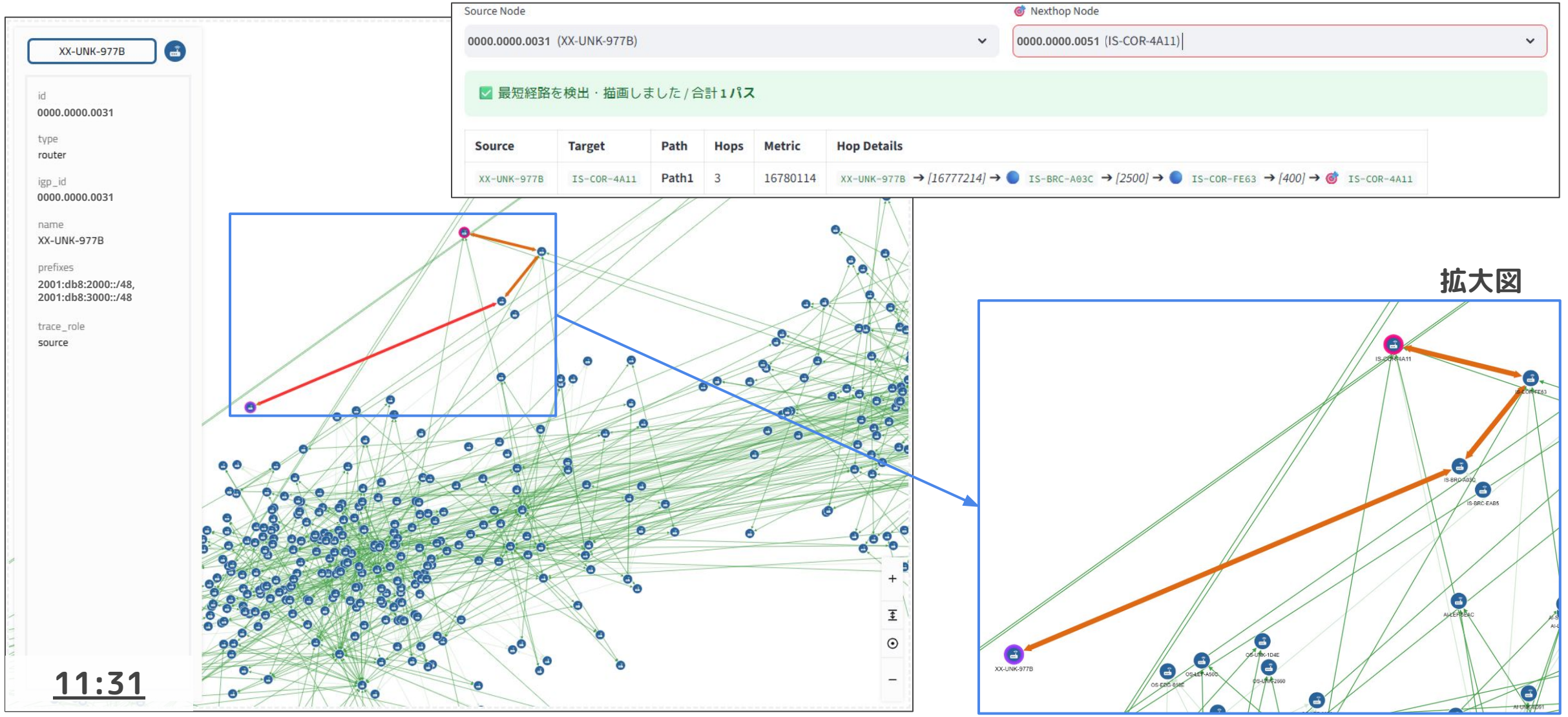
Snapshot + Transactionを適用

- 特定時刻のスナップショットを取得
- 過去のTransaction（≒イベント）を適用し、時刻的な前後を再現

ユースケース①: 特定時刻での最短経路探索

SoftBank

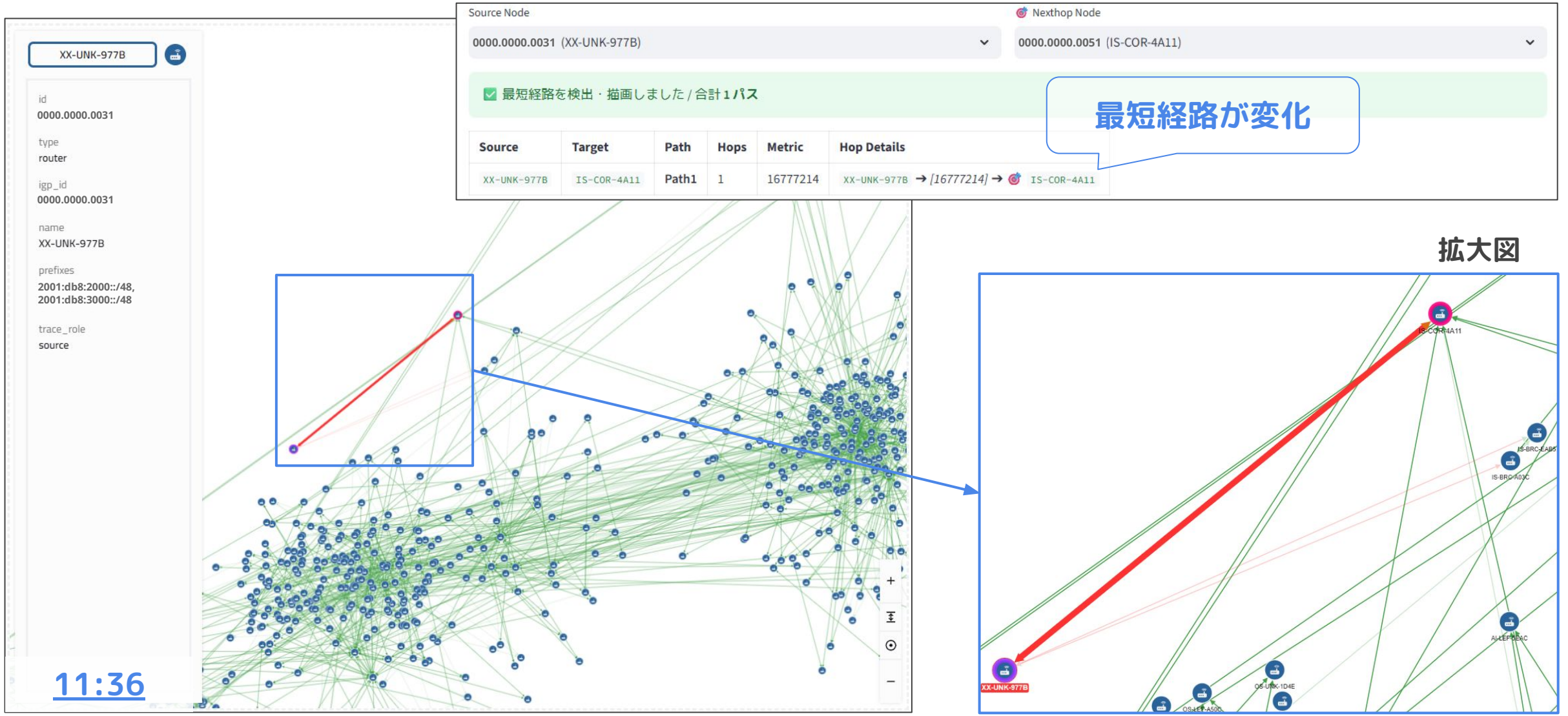
特定時刻におけるトポロジーを再現し、その時間軸での最短経路を探索（SPF）



ユースケース①: 特定時刻での最短経路探索

SoftBank

11:31（前ページ） → 11:36 のIGPトポロジー変動を反映



ユースケース②: 2つの時刻間におけるトポロジーdiff

SoftBank

例① 11:31 → 11:36 のdiffの一部【5分送り】

詳細イベント履歴 [2026-mm-dd 11:31:00 ➔ 2026-mm-dd 11:36:00 JST]

Time (JST)	Action	Type	Protocol	Target	Details
2026-mm-dd 11:31:13.465	ADD	Edge	ISIS	0000.0000.0033 → 0000.0000.0031	-
2026-mm-dd 11:31:13.465	ADD	Edge Property	-	6acf9aa1-f99c-d9d8-5ac4-dda15e967924	[igp_metric]: 5000000
2026-mm-dd 11:31:14.461	ADD	Edge	ISIS	0000.0000.0031 → 0000.0000.0033	-
2026-mm-dd 11:31:14.461	ADD	Edge Property	-	e2cf9aa1-fb8e-9f10-42a4-c05d61d54942	[igp_metric]: 16777214
2026-mm-dd 11:31:34.148	REMOVE	Edge Property	-	eecf9a9b-55fc-3907-a65d-6e3d19dc7423	[igp_metric]: 16777214
2026-mm-dd 11:31:34.148	ADD	Edge Property	-	eecf9a9b-55fc-3907-a65d-6e3d19dc7423	[igp_metric]: 10000
2026-mm-dd 11:31:34.155	REMOVE	Edge Property	-	28cf9a9b-e1d0-eaa7-a0dd-96d33c6d44b0	[igp_metric]: 16777214
2026-mm-dd 11:31:34.155	ADD	Edge Property	-	28cf9a9b-e1d0-eaa7-a0dd-96d33c6d44b0	[igp_metric]: 2500
2026-mm-dd 11:31:40.450	ADD	Edge	ISIS	0000.0000.0031 → 0000.0000.0051	-
2026-mm-dd 11:31:40.450	ADD	Edge Property	-	fccf9aa2-2e50-e500-1b28-dc87d381d688	[igp_metric]: 2500

新規リンクが追加。
方向によりMax metric相当（16,777,214）や高コストmetric付与。
一部の既存リンクではMax metric相当から10,000や2,500へ変更

例② 11:51 → 11:40 のdiffの一部【11分戻し】

詳細イベント履歴 [2026-mm-dd 11:51:00 ➔ 2026-mm-dd 11:40:00 JST]

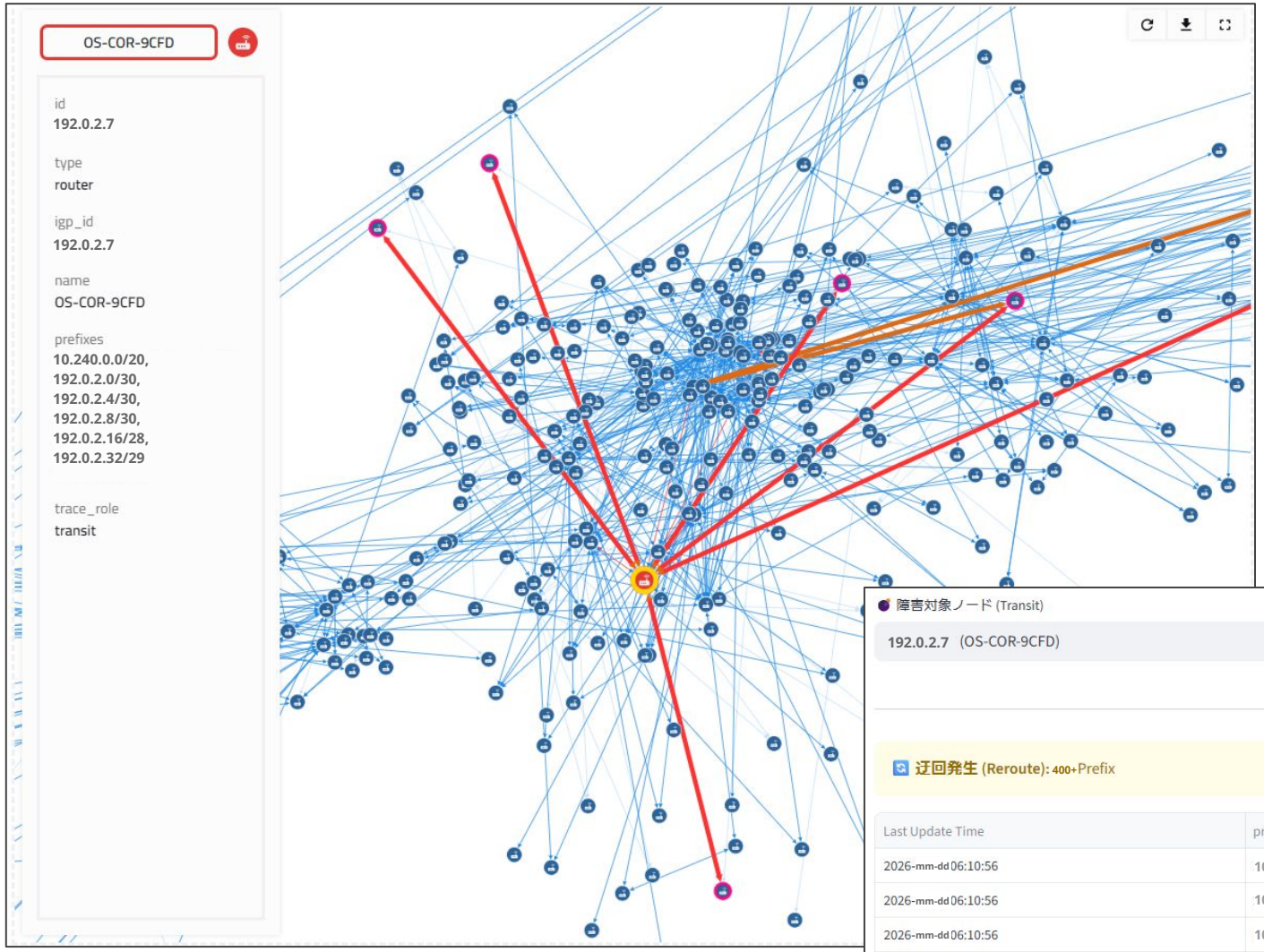
↓ Time (JST)	Action	Type	Protocol	Target	Details
2026-mm-dd 11:45:50.682	ADD	Edge Property	-	28cf9aa1-9edf-2d8c-f0a8-ebac10761baa	[igp_metric]: 10000
2026-mm-dd 11:45:50.682	REMOVE	Edge Property	-	28cf9aa1-9edf-2d8c-f0a8-ebac10761baa	[igp_metric]: 16777214
2026-mm-dd 11:45:50.680	ADD	Edge Property	-	e2cf9aa1-fb8e-9f10-42a4-c05d61d54942	[igp_metric]: 5000000
2026-mm-dd 11:45:50.680	REMOVE	Edge Property	-	e2cf9aa1-fb8e-9f10-42a4-c05d61d54942	[igp_metric]: 16777214
2026-mm-dd 11:45:50.676	ADD	Edge Property	-	82cf9aa2-3779-4a27-c393-83490fc0adb1	[igp_metric]: 2500
2026-mm-dd 11:45:50.676	REMOVE	Edge Property	-	82cf9aa2-3779-4a27-c393-83490fc0adb1	[igp_metric]: 16777214
2026-mm-dd 11:45:50.559	ADD	Node Property	-	0000.0000.0031	[prefixes]: 2001:db8:aaaa::/48
2026-mm-dd 11:45:50.558	ADD	Node Property	-	0000.0000.0031	[prefixes]: 2001:db8:bbbb:1::/64
2026-mm-dd 11:45:50.556	ADD	Node Property	-	0000.0000.0031	[prefixes]: 2001:db8:cccc:1::1/128

Nodeに対して新規Prefixを追加。
複数リンクのmetricがMax metric相当から
通常 or 高コストのmetricに変更。

ユースケース③: ノード障害時のシミュレーション

SoftBank

ノード障害時に影響を受けるBGP経路の候補を探索



※ このケースは**特定のルートリフレクタから収集した BGP経路の断面情報**をベースに計算。
→ RRのBestpathを基準にNext-hopへの到達性を計算

そのため、各エッジノードの実際のベストパスとの間でシミュレーション結果に差分が生じる場合がある。

- 実際には途切れていない
- 別の迂回ルートを通っている、‥

障害対象ノード (Transit)

192.0.2.7 (OS-COR-9CFD)

迂回発生 (Reroute): 400+Prefix

Last Update Time	prefix	peer_ip	nexthop
2026-mm-dd 06:10:56	10.0.0.0/20	10.fg.aw.1	10.qr.gw.77
2026-mm-dd 06:10:56	10.0.16.8/30	10.fg.aw.1	10.zu.xq.28
2026-mm-dd 06:10:56	10.0.16.40/30	10.fg.aw.1	10.jk.ak.91
2026-mm-dd 06:10:56	10.0.16.48/30	10.fg.aw.1	10.un.ce.78
2026-mm-dd 06:10:56	10.0.16.0/30	10.fg.aw.1	10.un.ce.78

静的情報をベースとしたNDTの例

SoftBank

Configを取り込み、NWトポロジーを描画。作業計画・影響確認・監視導線に利用

主な機能

1	**トポロジー可視化**	NW構成・接続関係・ノード情報の把握
2	**影響シミュレーション**	構成変更時の影響範囲確認、経路・到達性の確認
3	**監視・運用連携**	構成差分の把握/アラート・監視情報との連携

使い分け：設計・構成としてのあるべき姿（Intent） か 実際にいまどう見えているか（State）

ここまでのふりかえり

SoftBank

BGP-LSでカバーできること

- **IGPトポロジーの準リアルタイム/時系列監視:**
  OSPF/ISISレベルのノードやリンク、
  隣接関係、メトリック情報を収集可能。
  さらに時系列化することで、
  任意時刻のトポロジ状態を後から確認できる。
- **IGPトポロジーの差分検出:**
  作業前後、またはIntent実行前後のStateを比較し
  Link追加/削除、metric変更などを検出可能。
  人間では瞬時に判断しにくいトポロジーの変化を
  システム的に把握できる。
- **自動化結果の事後検証:**
  Intentへの制御とトポロジーを突合することで、
  意図した変更が反映されたか、
  想定外の変化が発生していないかを確認できる。
  自動化運用の安全性確認に活用が可能。

BGP-LSのみではカバーできないこと

- **物理IF / Bundleとの厳密な紐づけ**
  BGP-LSで把握できるのはIGP上の論理情報まで。
  物理IFやBundle、LAGの対応関係は把握できない。
  ⇒ **Telemetry、LLDP等のメトリクス、装置Config等の設計情報が必要**
- **NDTでのシミュレーション:**
  この取組み単体では仮想NW環境を構築できない。
  直近で見えている課題として、物理IFの静的情報とL3の動的情報をいかにマッピングするか。
  ⇒ **Config、Inventory情報、Policy等の設計情報が必要**
- **他レイヤ、およびトラフィック状態の監視:**
  物理的なLink Up/Downや光レベル、
  またリソース利用率やIF Traffic、Errorなどは
  BGP-LSだけでは把握ができない。
  ⇒ **SNMP、Syslog、Flow情報等のメトリクスが必要**

BGP-LSで捉えたトポロジー変化に 各種メトリクスを組み合わせることで より深い活用が可能

各種メトリクス利活用、重畳について

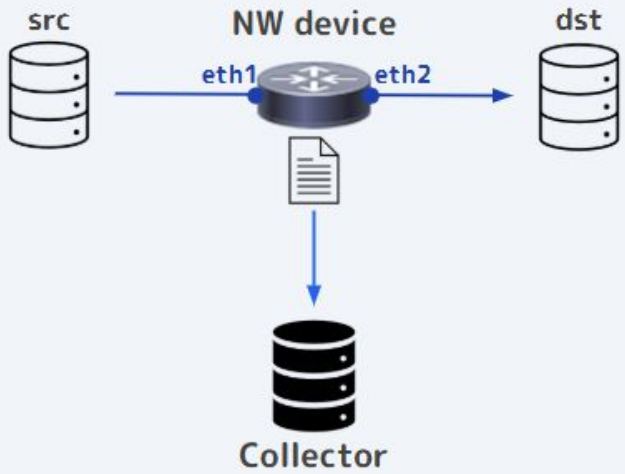
BGP-LS × 各種メトリクスの活用の検討

SoftBank

BGP-LS
NWトポロジーの動的観測

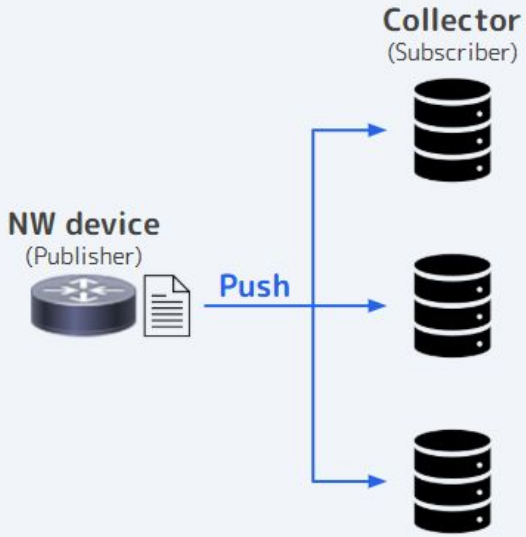
＋

① IPFIX
フロー単位のメトリクス収集



＋

② Streaming Telemetry
ノード/リンク単位のメトリクス収集



＋

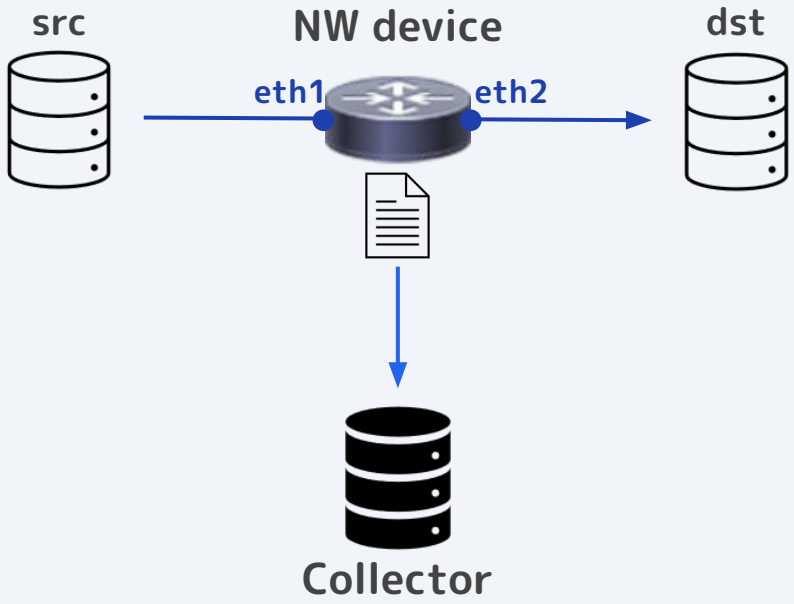
③ TWAMP
E2Eの品質メトリクス収集

動的トポロジーに各種メトリクスを重畳させることで 調査 / 検証 / 異常検知 に寄与

① IPFIX とは

IPFIXとは: 観測点におけるメトリクスをフローごとに収集する仕組み

※IPFIX: IP Flow Information Export (RFC7011-7015)



取得できるデータのイメージ

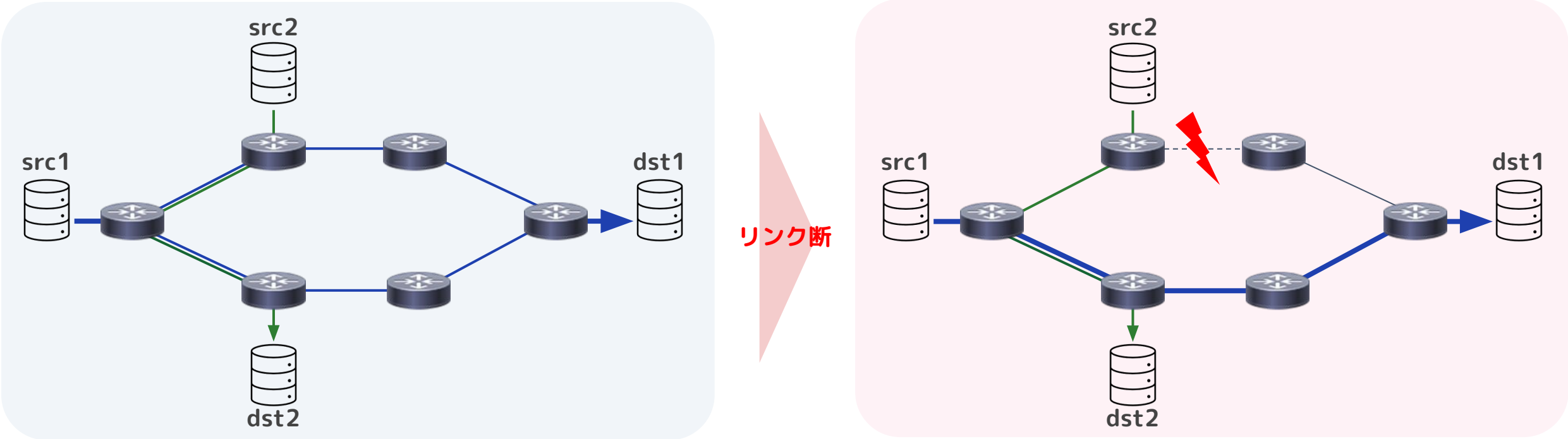
観測点	送信元	宛先	プロトコル	Bytes
eth1	src	dst	TCP	100Mbps
eth2	src	dst	TCP	100Mbps

フロー情報　メトリクス

BGP-LS × IPFIX の活用イメージ

SoftBank

組み合わせることで リンクを流れるトラフィックをフロー単位で動的に把握できる

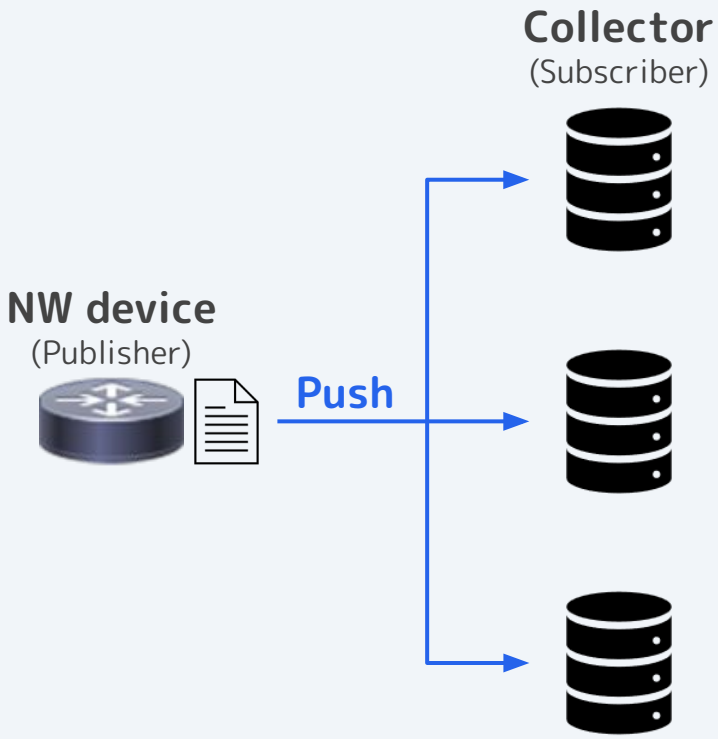


トポロジーの変化とトラフィック量の変化の因果関係を明確化

② Streaming Telemetry とは

Streaming Telemetryとは: Pub/Sub方式でNW機器からメトリクスを収集する仕組み

(RFC8639, 8641, 9232)



取得できるデータの例

インターフェースに関するメトリクス

- トラフィック量 (ingress/egress)
- エラーパケット数 (ingress/egress)

システムに関するメトリクス

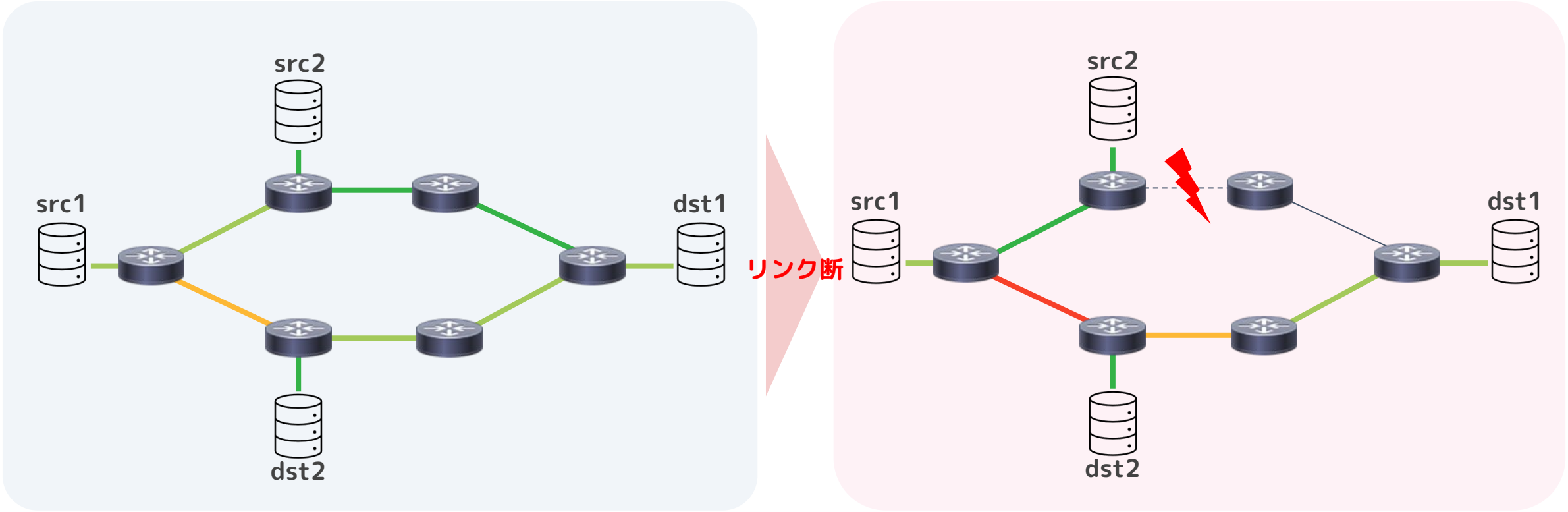
- CPU使用率
- メモリ使用量

BGP-LS × Streaming Telemetry の活用イメージ

SoftBank

組み合わせることで リンクを流れるトラフィックの全量を確認できる

※IPFIXではリンクのトラフィックの全量は確認が困難（L2, C-Planeなどのフロー外トラフィックを観測できない）



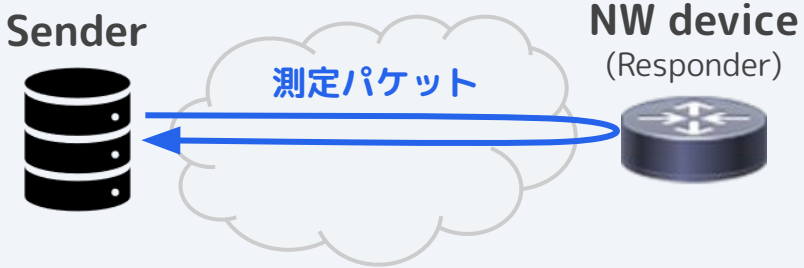
トポロジー変更に伴って実際のリンク利用率がどう変化したかを把握

③ TWAMP とは

SoftBank

TWAMPとは: ネットワークの2点間で測定パケットを流して 双方向の品質を取得するプロトコル

※TWAMP: Two-Way Active Measurement Protocol (RFC5357)



取得できるデータの例

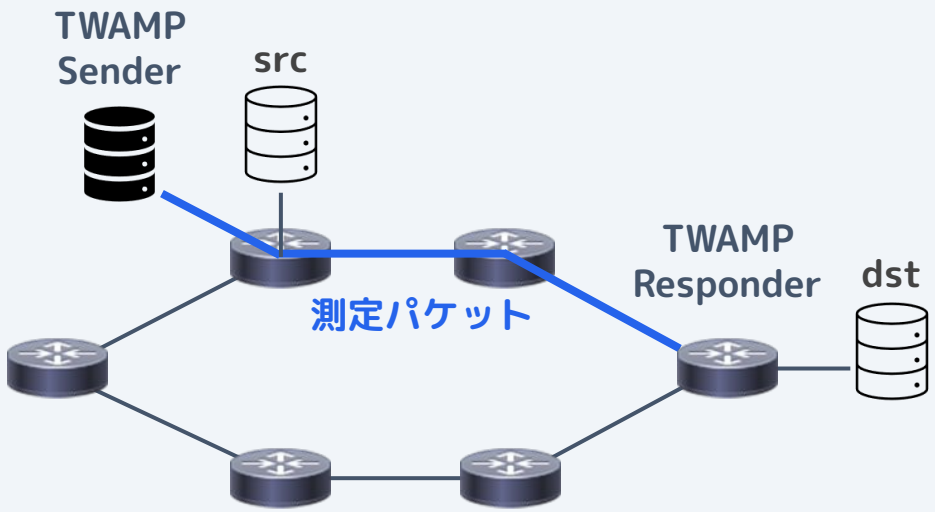
2点間の通信に関するメトリクス

- 遅延 (片道ごと)
- ジッタ (片道ごと)
- パケットロス (往復)

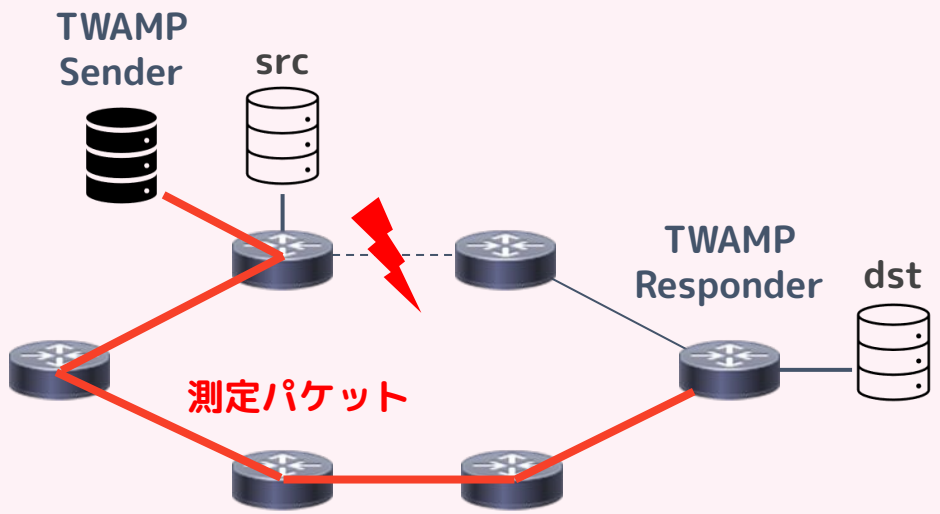
BGP-LS × TWAMP の活用イメージ

SoftBank

組み合わせることで 経路変化がE2Eの通信の品質に与える影響を推定できる



リンク断

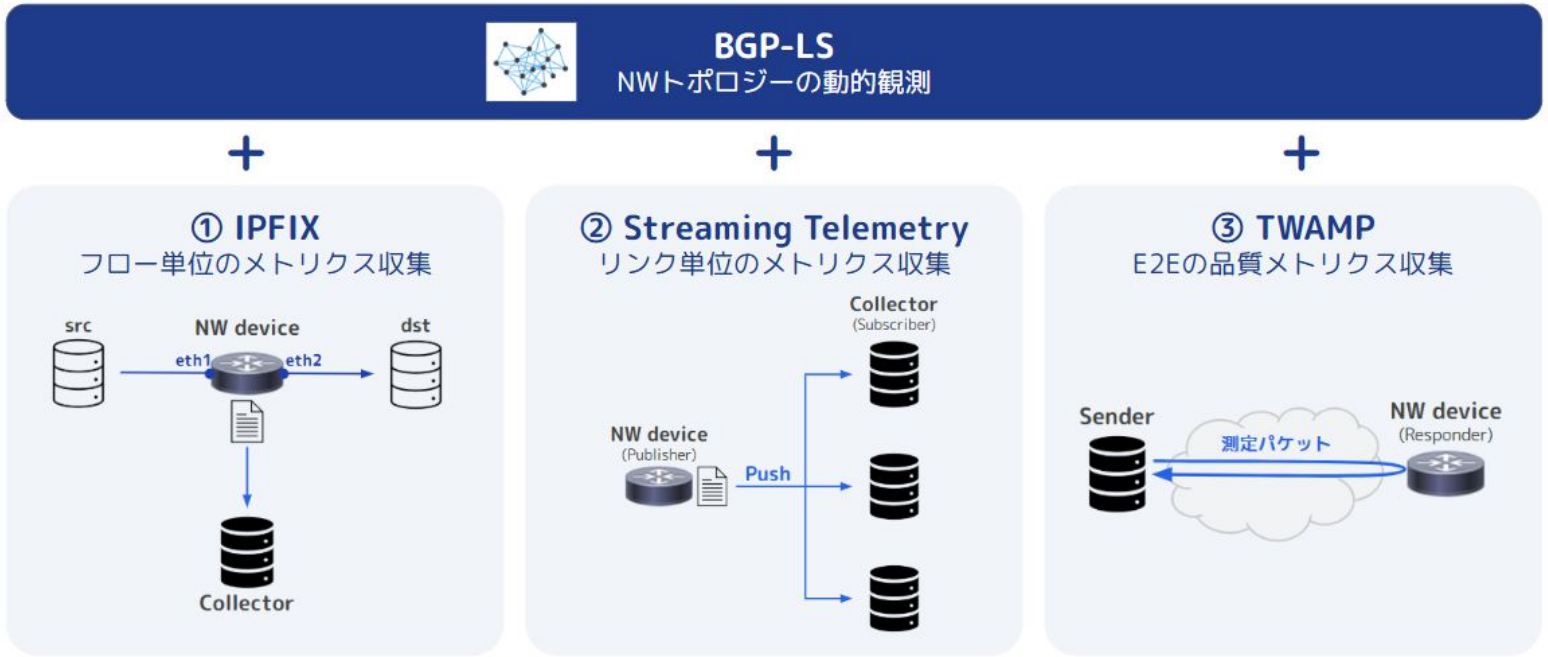


経路変化によって 品質がどう変化したかを確認

BGP-LS × 各種メトリクス のまとめ

SoftBank

動的トポロジーに各種メトリクスを重畳させることで 調査 / 検証 / 異常検知 に寄与



活用に向けたさらなる検討

- NDTの拡充への寄与: 要件定義~実装までの検討
- AIによる予測: 過去のトポロジーとメトリクスを用いた将来のデータの予測
- クローズドループへの貢献: リアルタイムのRCA / アクション実行可否の判断

全体のクローズ

SoftBank

まとめ

- NWトポロジー把握において、静的な情報（= Config, Intent）に加え
  実際のL3レベルでの変動（= State）を動的観測する必要性を検討。
- BGP-LS+BMPを用いて、ルーティングプロトコルの状態変化や
  トポロジー変動を継続的、かつ時間軸で追えることを例示。
- BGP-LSによる動的トポロジーに各種メトリクスを重ね合わせることで
  調査や検証、異常検知に寄与するアプローチを提示。

議論ポイント（その他も大歓迎です）

- **データの保持期間**や**サンプリング粒度**、**フラッピング発生時の処理**をどう設計すべきか。
- **観測されたStateと構成上のIntentが一致しない場合**、どちらを正として是正すべきか。
- BGP-LS+BMPで把握できるのは、L3の制御プレーン由来の状態に限定される。
  - 実運用において、**L3トポロジー観測のみ**で何ができるか。
  - **L1/L2や設備台帳など他レイヤ・他系統の情報**と組み合わせることで何ができるか。
- **メトリクスの重畳**にはどういったアプローチ/ユースケースが考えられるか。

End of Slide