

IGPメトリックってどのように使っていますか？

竹中 幹, 藤原 達弥

NTTコミュニケーションズ

m.takenaka@ntt.com, tat.fujiwara@ntt.com

2025/1/24 JANOG55 (みやこめっせ)

登壇者紹介

- 竹中 幹
(たけなか もとき)

🔗 <https://github.com/Motok1>



- 略歴
2021年～ NTTコミュニケーションズ
イノベーションセンター

- 業務内容
 - 検証用 ISP ネットワークの運用・更改
 - 社内へのネットワーク関連の技術支援
 - 最新技術の研究・PoC 開発

- 藤原 達弥
(ふじわら たつや)



- 略歴
2014年～2024年 古河電工
(ルータのソフトウェア開発)
2024年～ NTTコミュニケーションズ
イノベーションセンター

目次

- IGP メトリックの使い方
- 経路要件を満たす IGP メトリックを決める際の課題
- 課題に対する我々のアプローチ
- 我々のネットワークへの適用と動作確認
- まとめと議論ポイント

■IGP メトリックの使い方

- 経路要件を満たす IGP メトリックを決める際の課題

- 課題に対する我々のアプローチ

- 我々のネットワークへの適用と動作確認

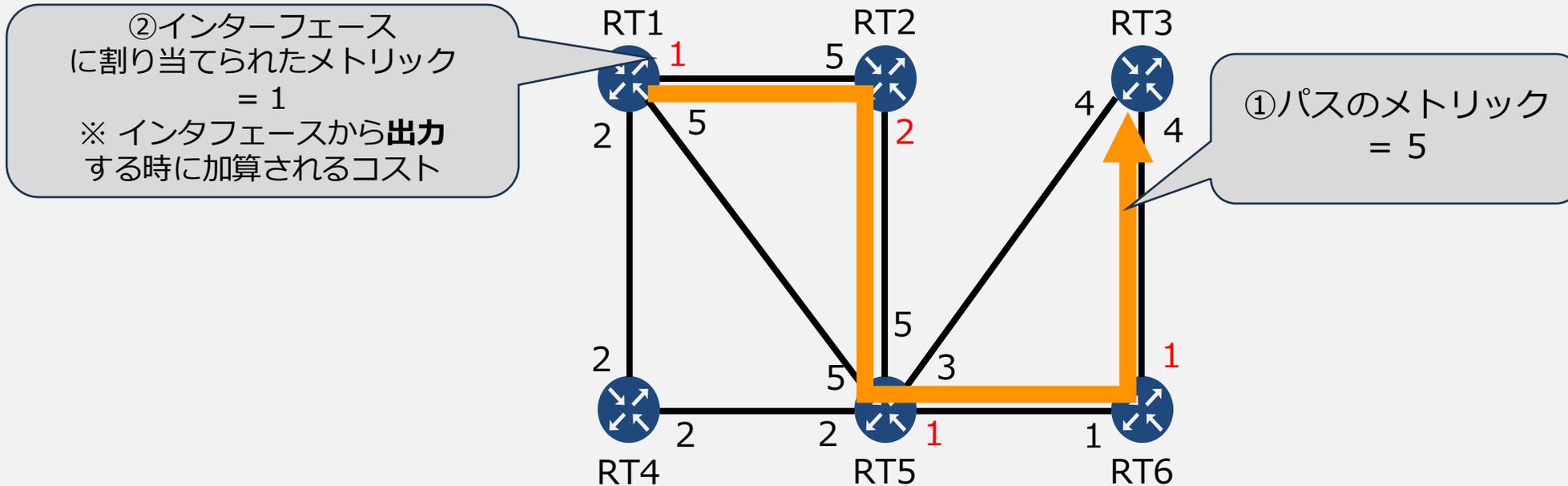
- まとめと議論ポイント

IGP メトリックとは

① IGP で計算されたパスのメトリック

② IGP において、ルータのインターフェースに割り当てられているメトリック

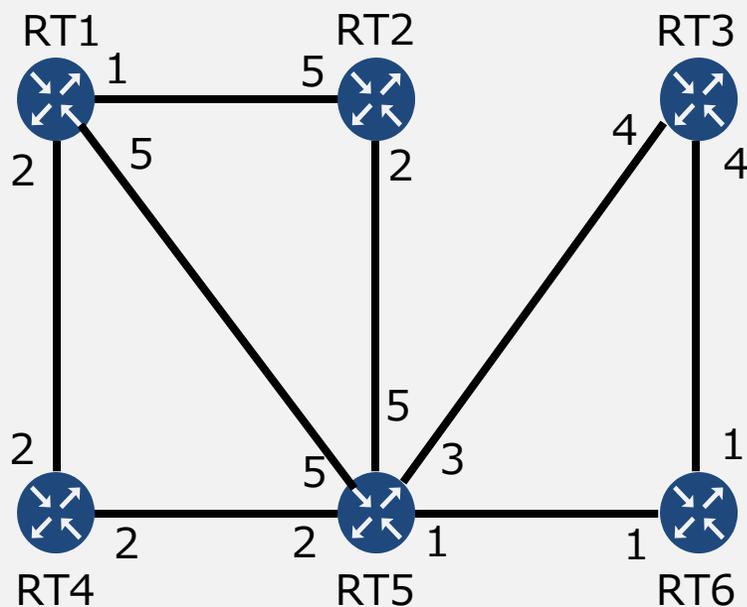
- OSPF の Cost (ref: <https://datatracker.ietf.org/doc/html/rfc2328>)
 - 1 以上 65535 以下の整数値
- IS-IS の default metric (ref: <https://datatracker.ietf.org/doc/html/rfc5305>)
 - 1 以上 16777214 以下の整数値



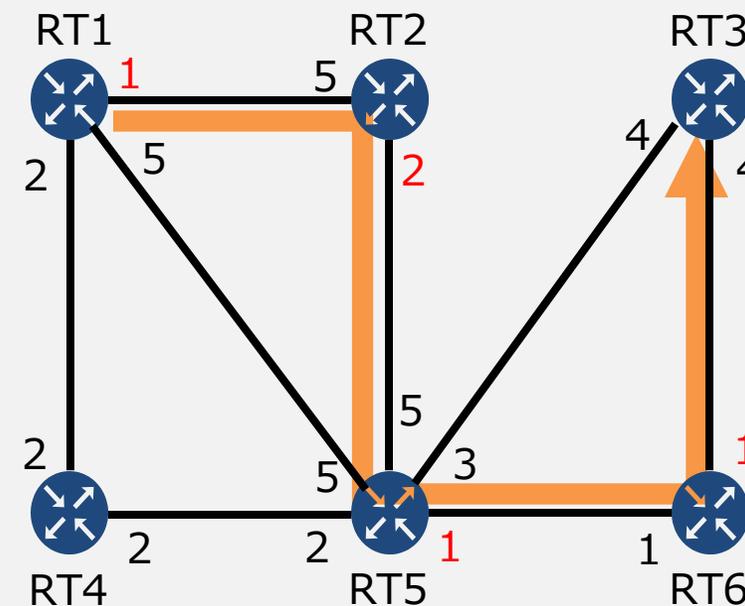
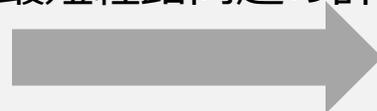
我々は ② をどのように使うかについて話します→

IGP では最短経路問題を解いている

例：次のネットワークで IGP が動作している場合、
このメトリックにおいて RT1 から RT3 の経路はどう決まるか？



各ルータにおいて
ダイクストラ法等で
最短経路問題の計算

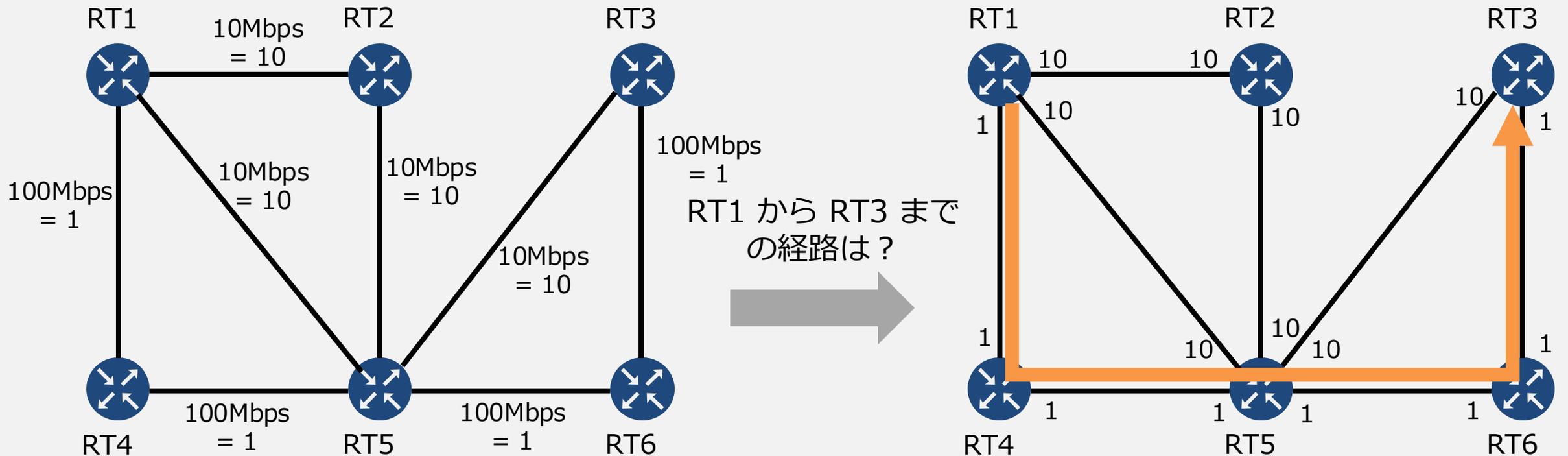


IGP メトリックが決まると経路が決まる(1/2)

■ 各リンクの帯域幅を IGP メトリックに反映する

- OSPF ではデフォルトの IGP メトリックが 100/リンクの帯域(*Mbps*)で決定する

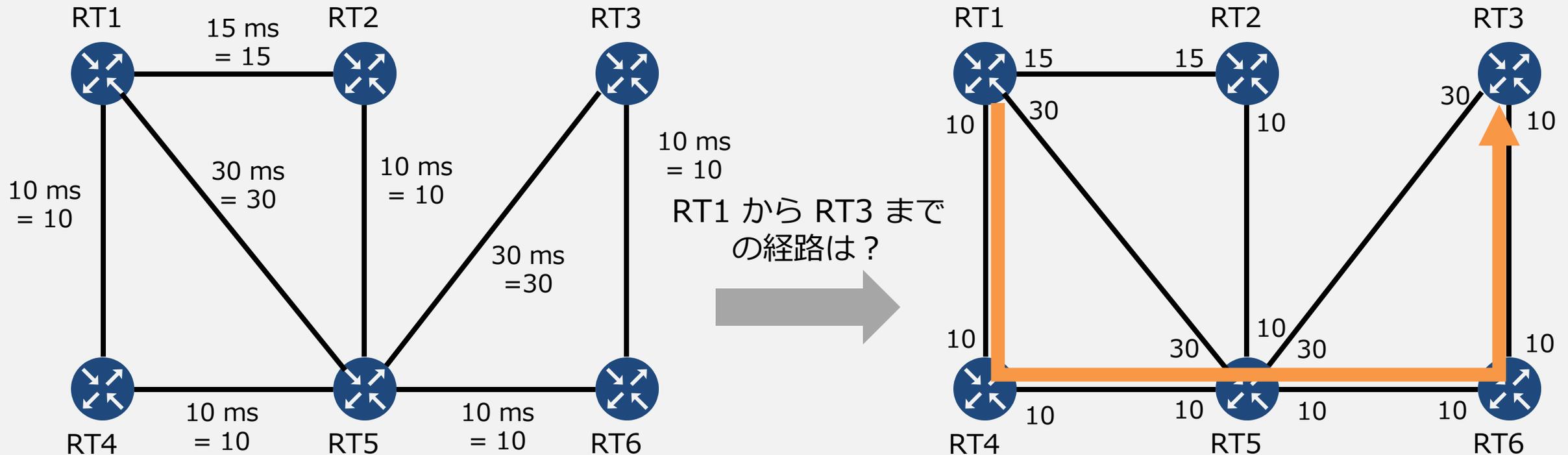
→ 帯域が広い (100Mbps=1) リンクが使用される



IGP メトリックが決まると経路が決まる(2/2)

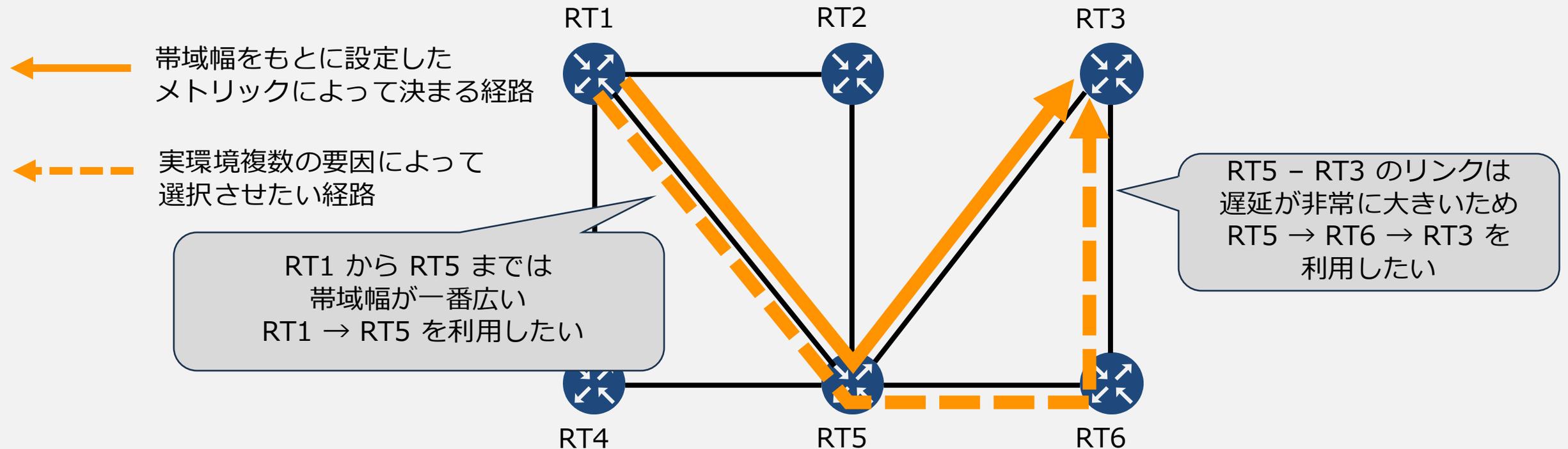
■ 実測の遅延値を IGP メトリックに反映する

→ 到達までの 2 ルータ間の転送遅延が最小の経路が使用される



帯域幅や遅延などから IGP メトリックを決められないとき

- 実運用のネットワークには、さまざまな環境要因をもとに経路を選択したいケースもある

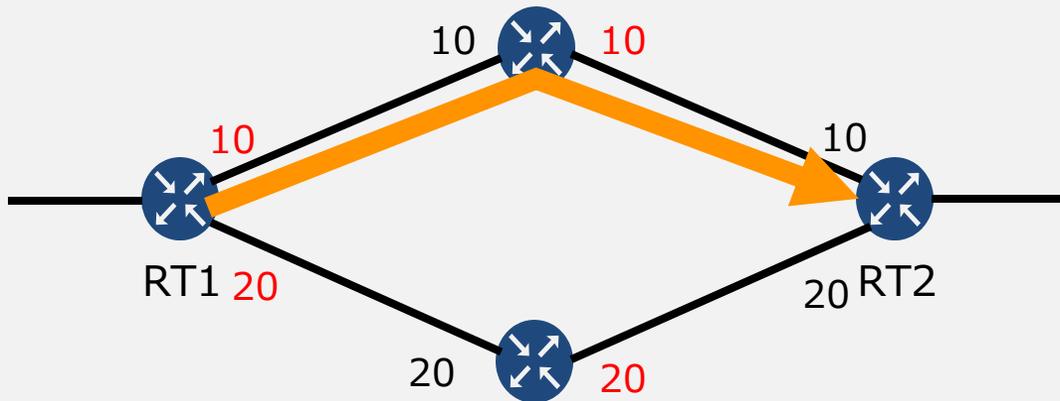


IGP メトリックの他の使い方を考える必要がある

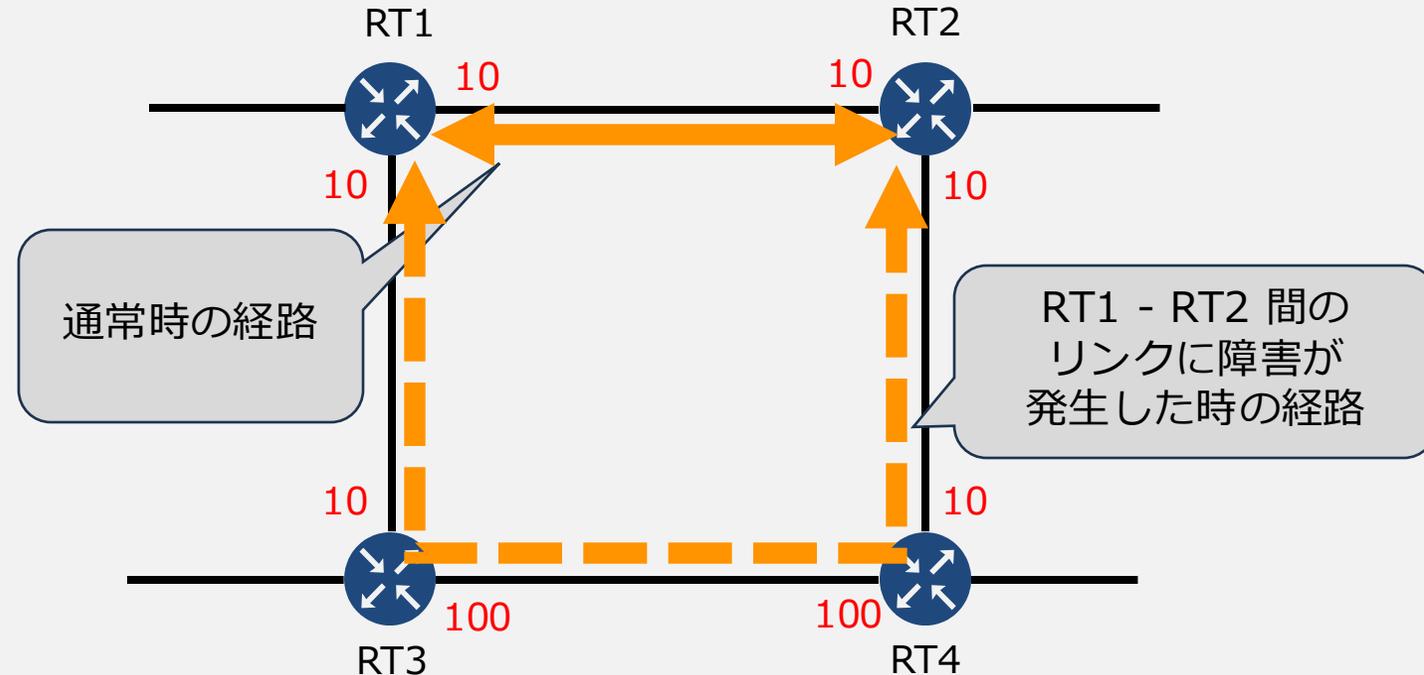
IGP に意図した経路を選択させたい場合はどうすれば良いか？

- ネットワークの一部のリンク構成に着目して
通したいリンクのコストを下げる、または、通したくないリンクの
コストを上げる

RT1 から RT2 までの一部分で
上回りの経路を通したい場合は
上回りのリンクを 10、下回りのリンクを 10 より
大きい値にする



左右のネットワークにトラフィックを流すとき
上段を主系、下段を副系にしたい場合は
主系リンクを 10、副系リンクを 100、渡りを 10 などにする

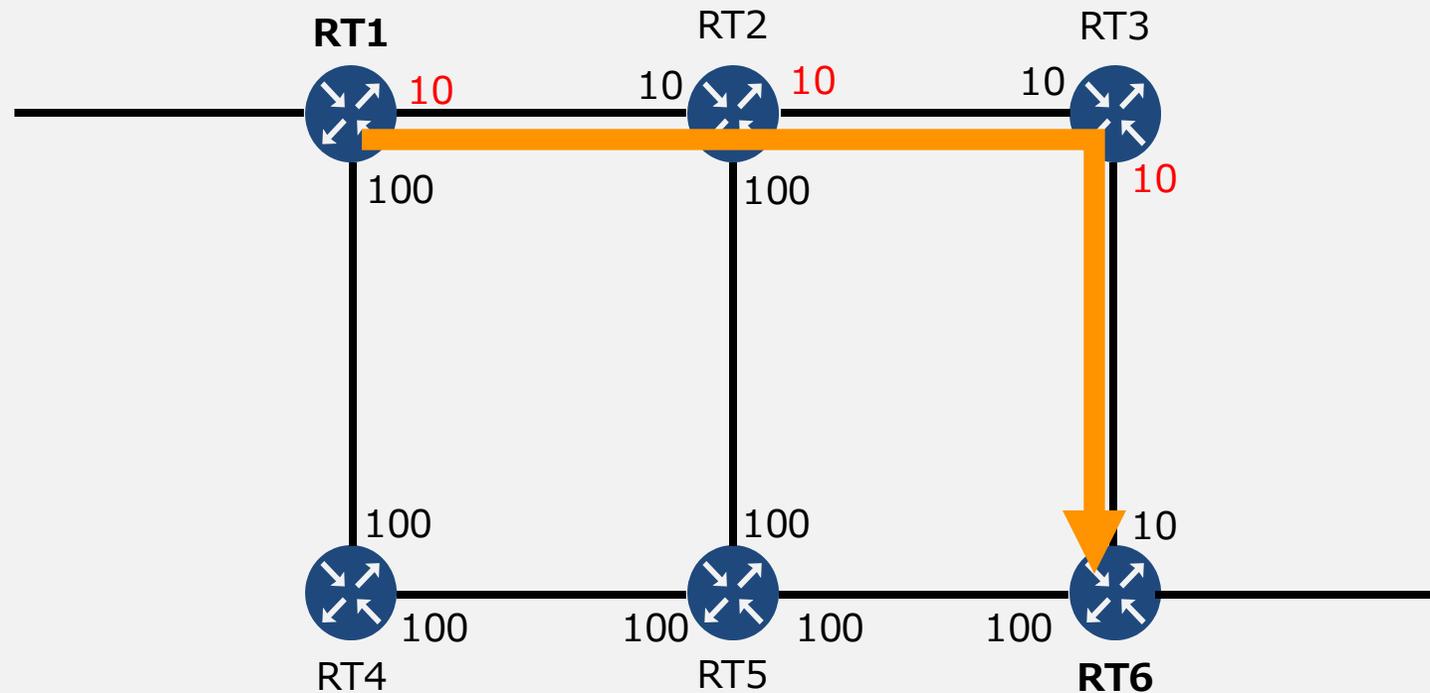


他にも→

IGP に意図した経路を選択させたい場合はどうすれば良いか？

- IGP ドメインにおいて、2 台のルータ間でトラフィックを通したい経路上のリンクコストを下げる

通さないリンクを 100、通すリンクを 100 より十分に小さい値にする



意図した経路を選択させたいオペレータが暗黙的に実施していること

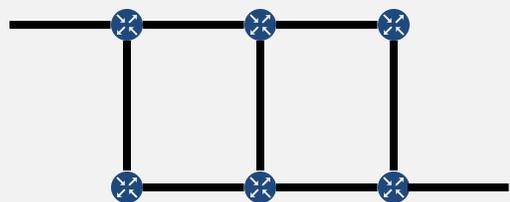
- 最短経路問題の解が期待するリンクや経路を通るようにメトリックを適した値に調整する

実現したい経路制御

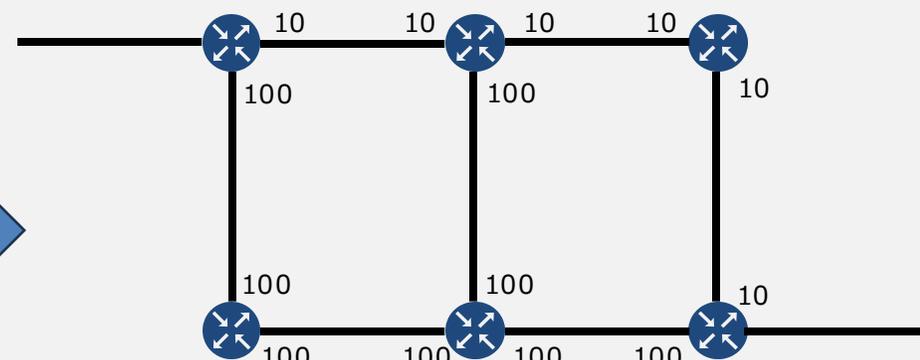
このネットワークのリンク構成の時は
このリンクを優先的に利用したい…

and/or

ルータ A からルータ B まで向かう
トラフィックはこの経路を通したい…



ネットワークトポロジ



実現したい経路制御
を満たす IGP メトリック

■IGP メトリックの使い方

■経路要件を満たす IGP メトリックを決める際の課題

■課題に対する我々のアプローチ

■我々のネットワークへの適用と動作確認

■まとめと議論ポイント

IGP メトリックを決める難しさ

実現したい経路制御

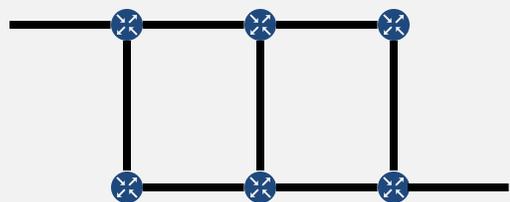
このネットワークのリンク構成の時は
このリンクを優先的に利用したい…

and/or

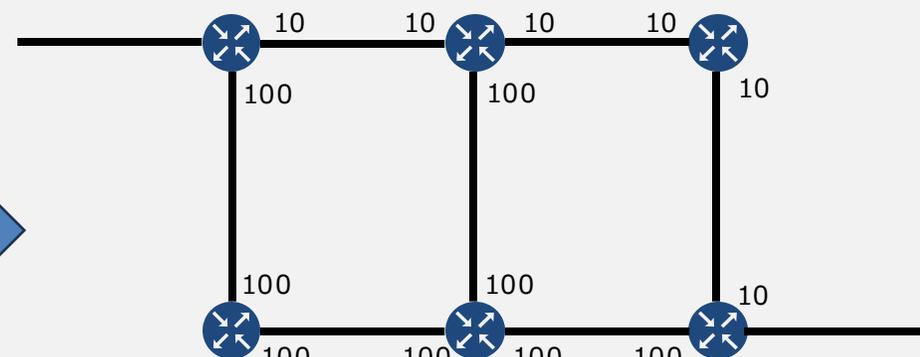
ルータ A からルータ B まで向かう
トラフィックはこの経路を通したい…

要件が増えても
一部分のリンク構成に着目して
IGP メトリックを決めるため比較的容易

ルータ間の経路要件の数が
増えたり要件の内容が複雑
になってくると IGP
メトリックの算出が難しい



ネットワークトポロジ

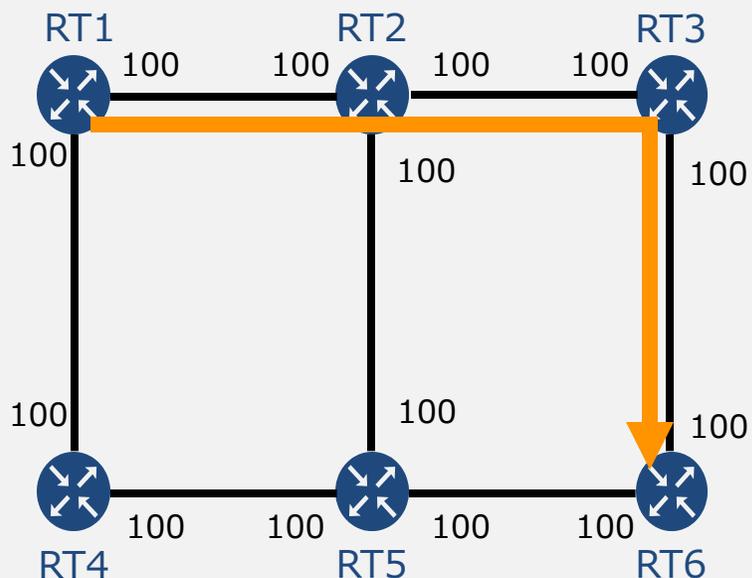


経路要件を満たす
IGP メトリック

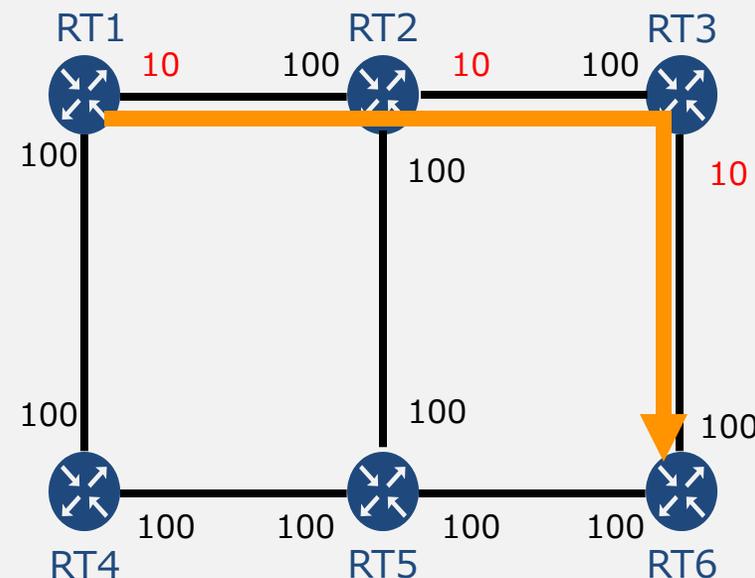
要件の**数**によって IGP メトリックの決定が難しくなる例 (1/3)

- **RT1** から **RT6** は RT1 → RT2 → RT3 → RT6 を通したい

1. 全てのインターフェースの IGP メトリックを仮で 100 と置く



2. 通したいインターフェースの IGP メトリックを一律で (10などに) 落とす

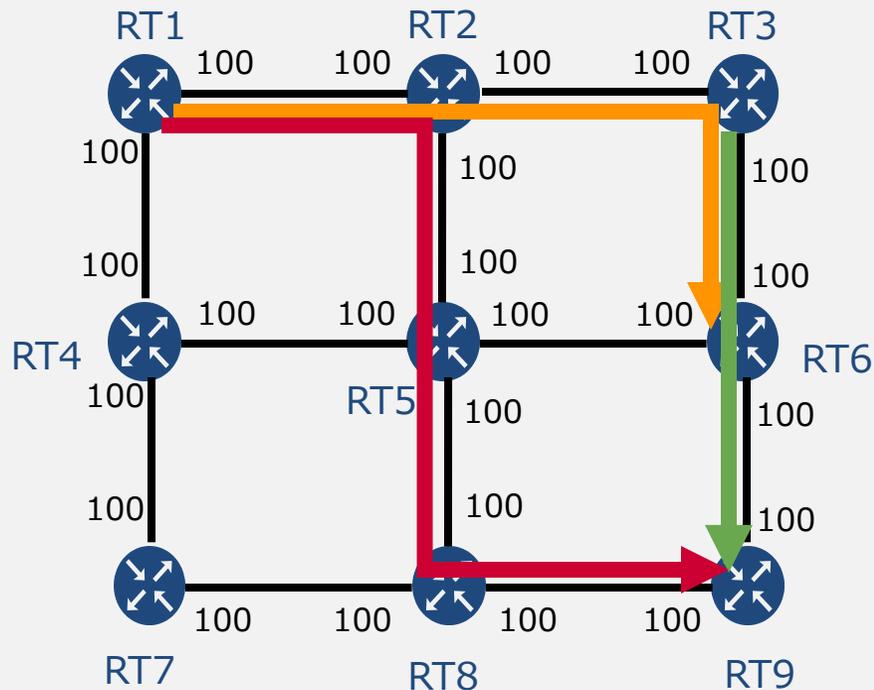


上記の計算ロジックで IGP メトリックを求めることが可能

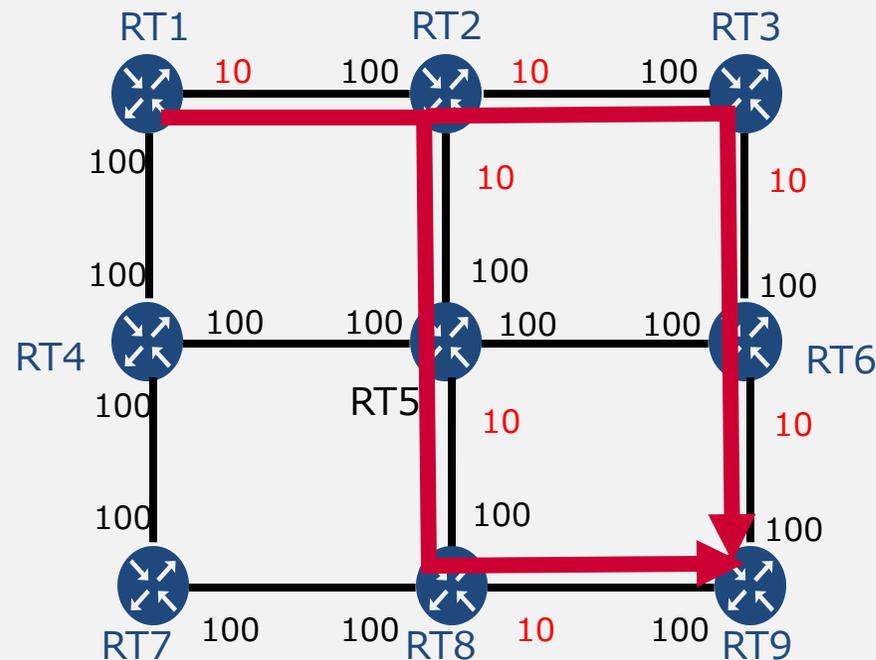
要件の数によって IGP メトリックの決定が難しくなる例 (2/3)

- RT1 から RT6 は RT1 → RT2 → RT3 → RT6 を通したい
- RT1 から RT9 は RT1 → RT2 → RT5 → RT8 → RT9 を通したい
- RT3 から RT9 は RT3 → RT6 → RT9 を通したい

1. 全てのインターフェースの IGP メトリックを仮で 100 と置く



2. 通したいインターフェースの IGP メトリックを一律で (10などに) 落とす



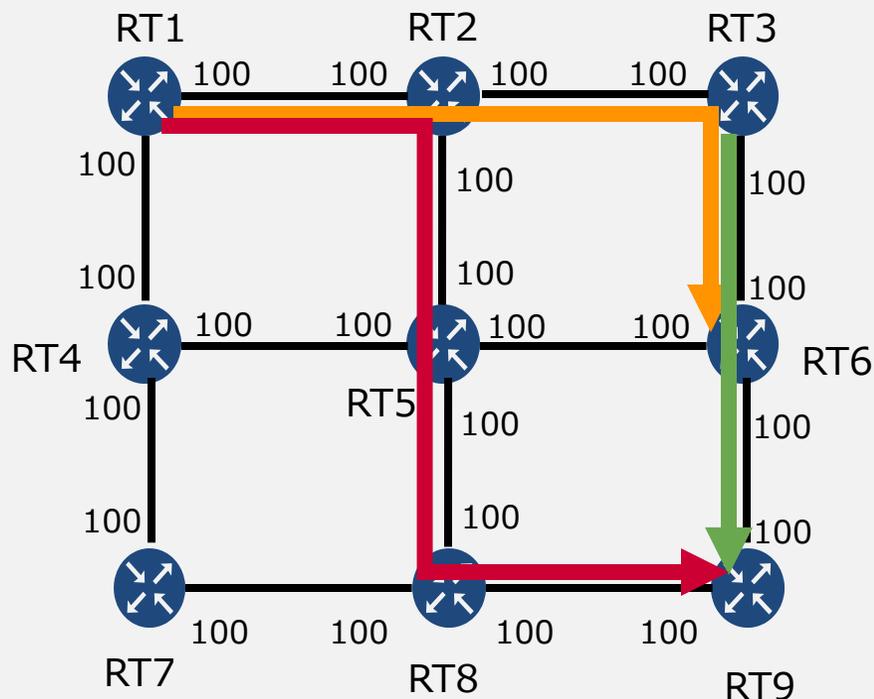
橙経路を満たすために決めた IGP メトリックの影響で赤経路が ECMP になってしまう

単純な計算ロジックで IGP メトリックを求めることができない

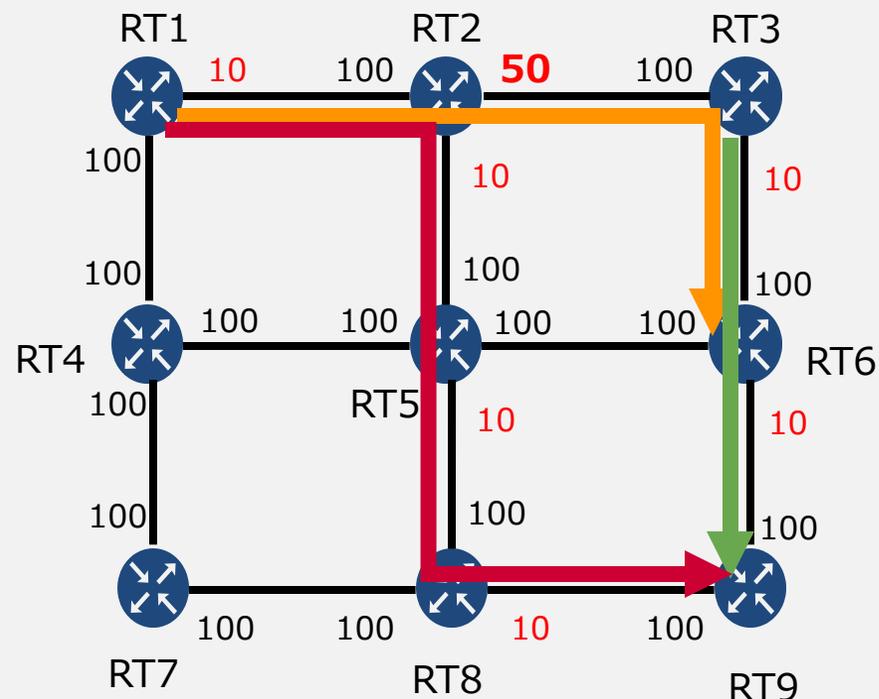
要件の数によって IGP メトリックの決定が難しくなる例 (3/3)

- RT1 から RT6 は RT1 → RT2 → RT3 → RT6 を通したい
- RT1 から RT9 は RT1 → RT2 → RT5 → RT8 → RT9 を通したい
- RT3 から RT9 は RT3 → RT6 → RT9 を通したい

1. 全てのインターフェースの IGP メトリックを仮で 100 と置く



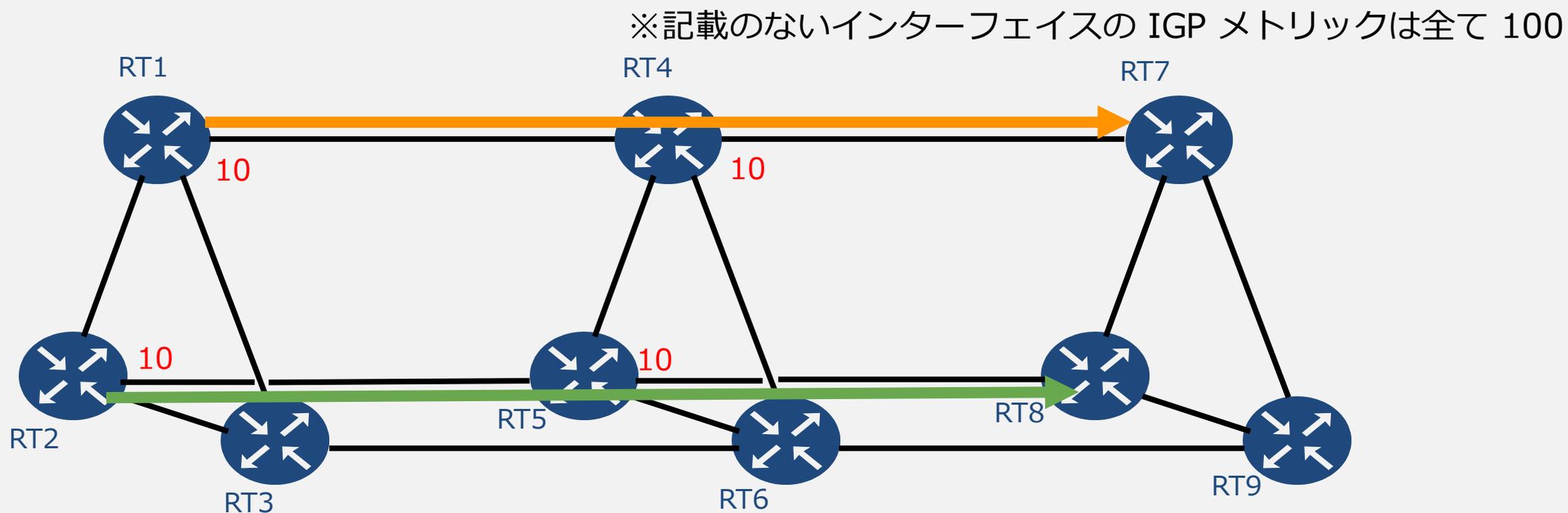
2. 通したいインターフェースの IGP メトリックを全ての経路が要件を満たすように調整しつつ落とす



要件を満たす IGP メトリック
の調整が必要

要件の**内容**によって IGP メトリックの決定が難しくなる例 (1/3)

- RT1 から RT7 は RT1 → RT4 → RT7 を通したい
- RT2 から RT8 は RT2 → RT5 → RT8 を通したい



先ほど紹介した計算ロジックで IGP メトリックを求めることが可能

要件の**内容**によって IGP メトリックの決定が難しくなる例 (2/3)

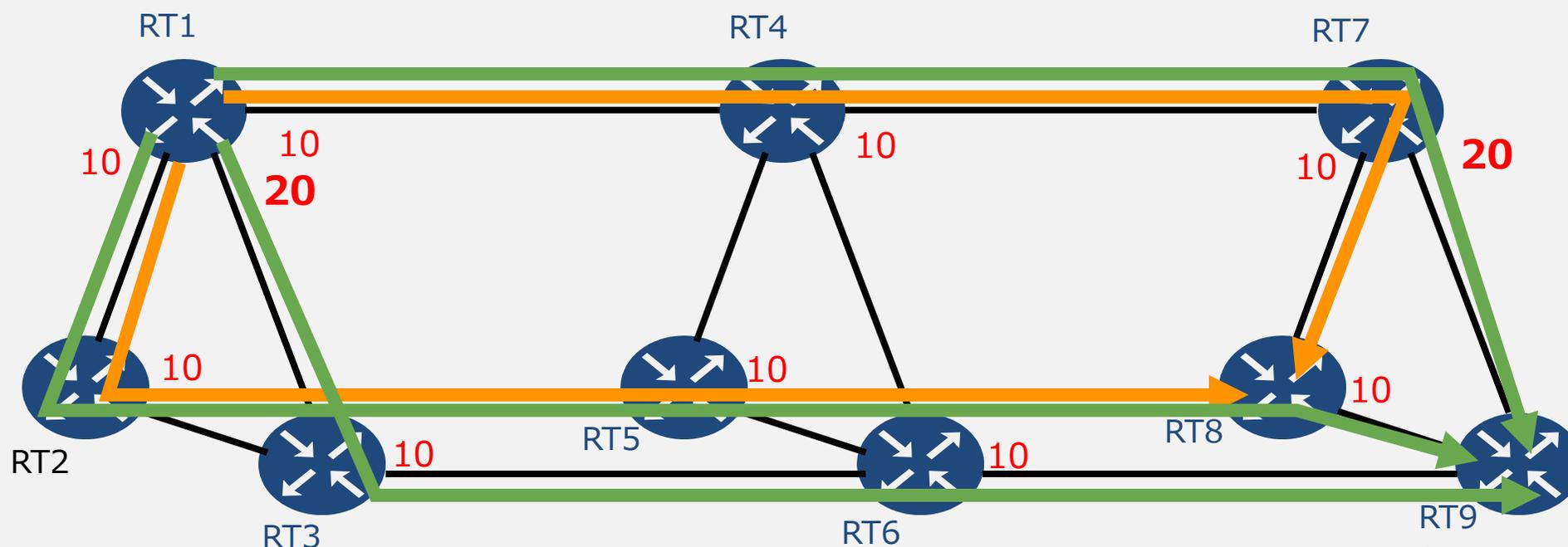
- **RT1** から **RT8** は以下の 2 経路でトラフィックを分散したい

- RT1 → RT4 → RT7 → RT8
- RT1 → RT2 → RT5 → RT8

- **RT1** から **RT9** は以下の 3 経路でトラフィックを分散したい

- RT1 → RT2 → RT5 → RT8 → RT9
- RT1 → RT3 → RT6 → RT9
- RT1 → RT4 → RT7 → RT9

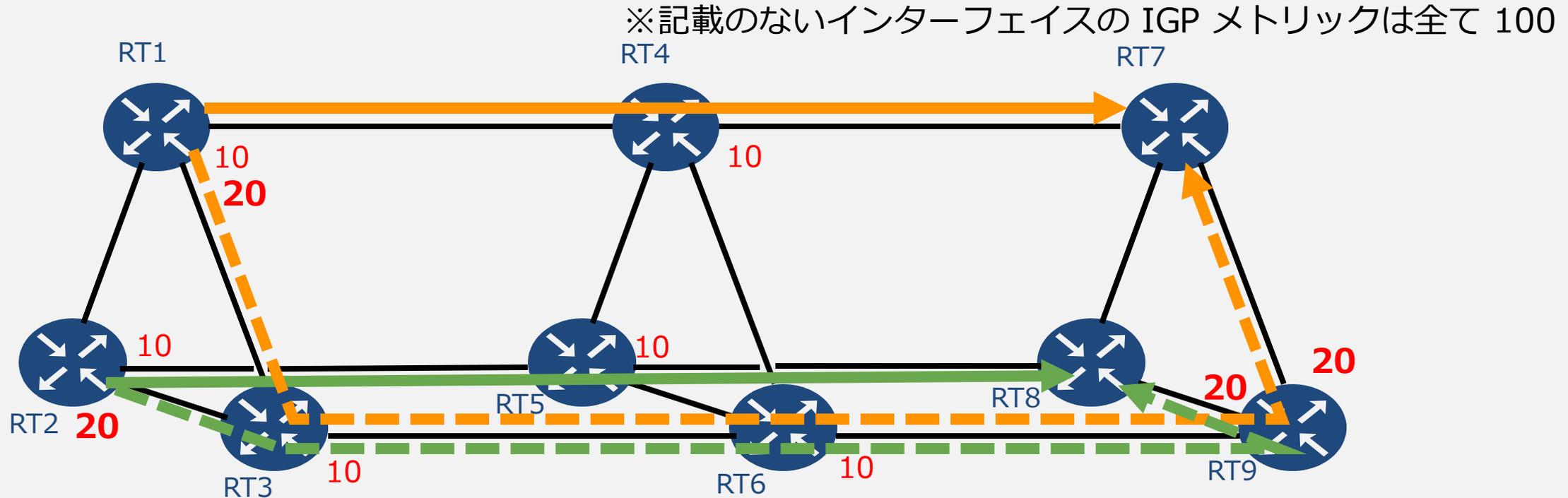
※記載のないインターフェイスの IGP メトリックは全て 100



単純な計算ロジックで IGP メトリックを求めることができない

要件の**内容**によって IGP メトリックの決定が難しくなる例 (3/3)

- **RT1** から **RT7** は RT1 → RT4 → RT7 を通したい、指定経路の障害時には RT1 → RT3 → RT6 → RT9 → RT7 を通したい
- **RT2** から **RT8** は RT2 → RT5 → RT8 を通したい、指定経路の障害時には RT2 → RT3 → RT6 → RT9 → RT8 を通したい



単純な計算ロジックで IGP メトリックを求めることができない

経路要件を満たす IGP メトリック決定の課題

■ ルータ間の経路要件の**数が多い**場合

→ ある経路要件を満たすために設定した IGP メトリックの値が他の経路要件に影響を及ぼす

■ ルータ間の経路要件の**内容が複雑な**場合

→ 1 つの経路要件内の一要素 (分散経路のうちの一経路/主系 など) を満たすために設定した IGP メトリックの値が他の要素 (分散経路の他経路/副系 など)に影響を及ぼす



■ 経路要件を全て満たす IGP メトリックを発見することが困難で時間がかかる

■ そもそも経路要件を全て満たす IGP メトリックが存在するかわからない

■IGP メトリックの使い方

■経路要件を満たす IGP メトリックを決める際の課題

■課題に対する我々のアプローチ

■我々のネットワークへの適用と動作確認

■まとめと議論ポイント

やりたいこと

- 多数の複雑な経路要件を持つネットワークで、経路要件を満たす IGP メトリックを決定したい



これって数理最適化問題にできるのでは？

各経路要件は**制約条件**

経路要件

要件1
要件2
…
要件n

目的関数は IGP メトリック
の合計を最小にする

IGP メトリックは**探索する変数**



ネットワークトポロジ



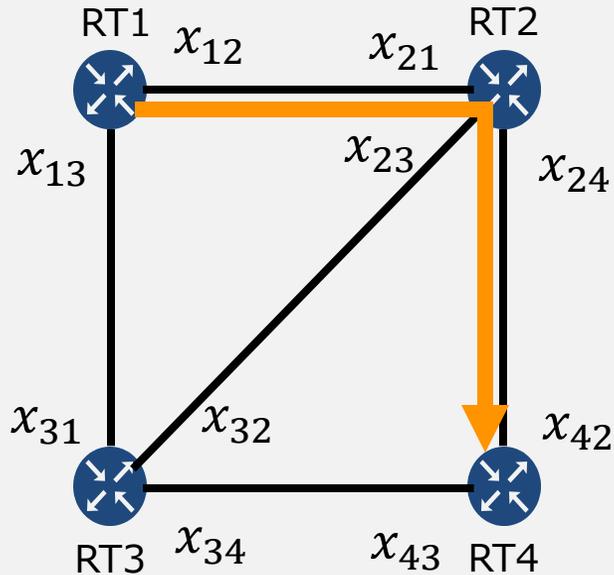
数理最適化問題を解く

経路要件を満たす
IGP メトリック

数理最適化問題の具体例①

■ RT1 から RT4 は RT1 → RT2 → RT4 を通したい

探索する変数は IGP メトリック



制約条件

メトリックの制約
 $1 \leq x_i \leq 65535$

経路の制約

$x_{12} + x_{24} < x_{12} + x_{23} + x_{34}$

$x_{12} + x_{24} < x_{13} + x_{32} + x_{24}$

$x_{12} + x_{24} < x_{13} + x_{34}$

RT1 → RT4 の候補パスの中で、
通したい経路が合計メトリック最小

目的関数

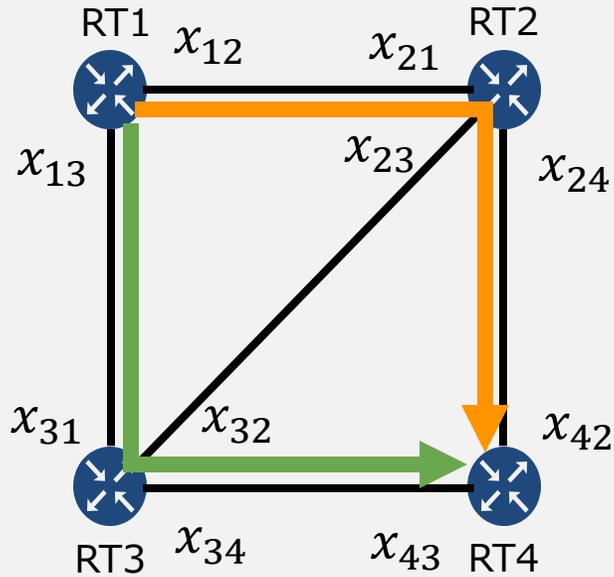
$$\min \sum x_i$$

IGP メトリックの合計を
最小にする

数理最適化問題の具体例②

- RT1 から RT4 は正常時に RT1 → RT2 → RT4 を通し、主系障害時に RT1 → RT3 → RT4 を通したい

探索する変数は IGP メトリック



制約条件

メトリックの制約

$$1 \leq x_i \leq 65535$$

主系の制約

$$x_{12} + x_{24} < x_{12} + x_{23} + x_{34}$$

$$x_{12} + x_{24} < x_{13} + x_{32} + x_{24}$$

$$x_{12} + x_{24} < x_{13} + x_{34}$$

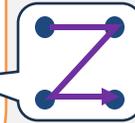
主系障害時の制約

$$x_{13} + x_{34} < x_{12} + x_{23} + x_{34}$$

$$x_{13} + x_{34} < x_{13} + x_{32} + x_{24}$$

目的関数

$$\min \sum x_i$$

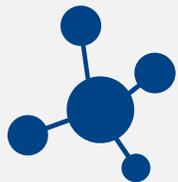


計算ツールを作ってみました

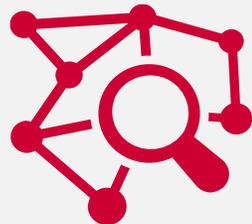
- 数理最適化問題をベースにした IGP メトリックの計算ツールを実装した

経路要件

要件1
要件2
…
要件n



ネットワークポロジ
(IGP メトリックなし)



計算ツール



ネットワークポロジ
(IGP メトリックあり)

■IGP メトリックの使い方

■経路要件を満たす IGP メトリックを決める際の課題

■課題に対する我々のアプローチ

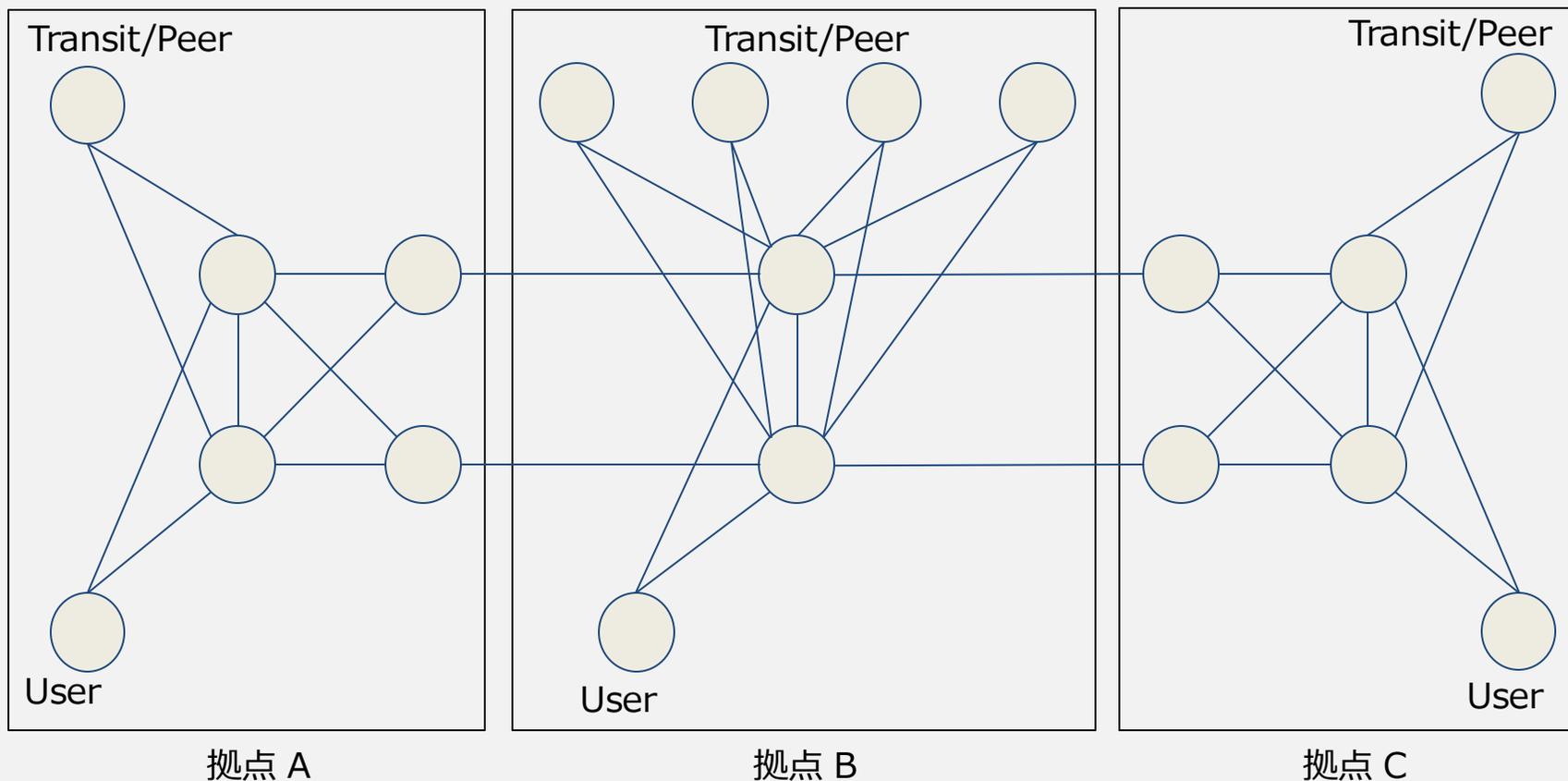
■我々のネットワークへの適用と動作確認

■まとめと議論ポイント

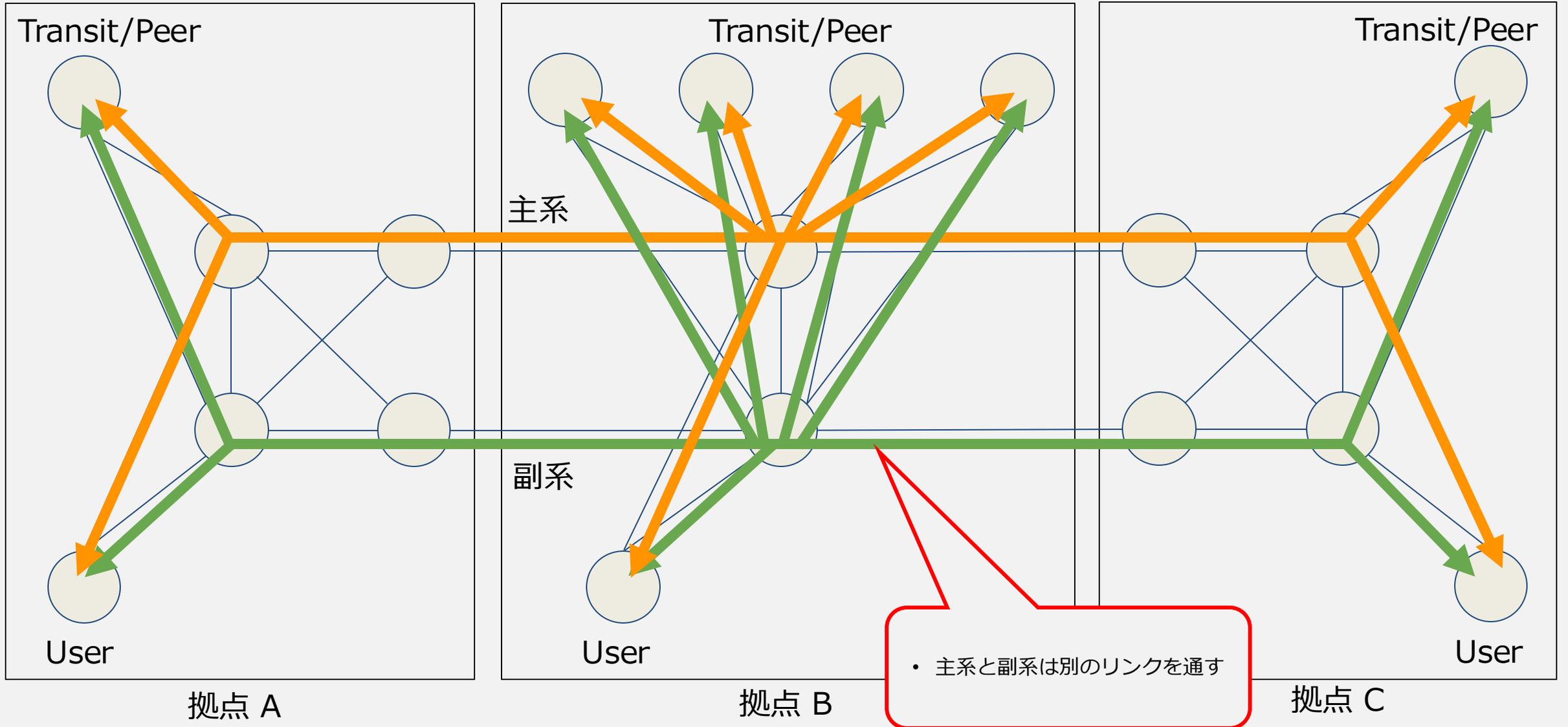
我々の運用ネットワーク

- AS HANABI (AS38639)
- 社内の検証用ネットワーク
- IGP は OSPF を使用している

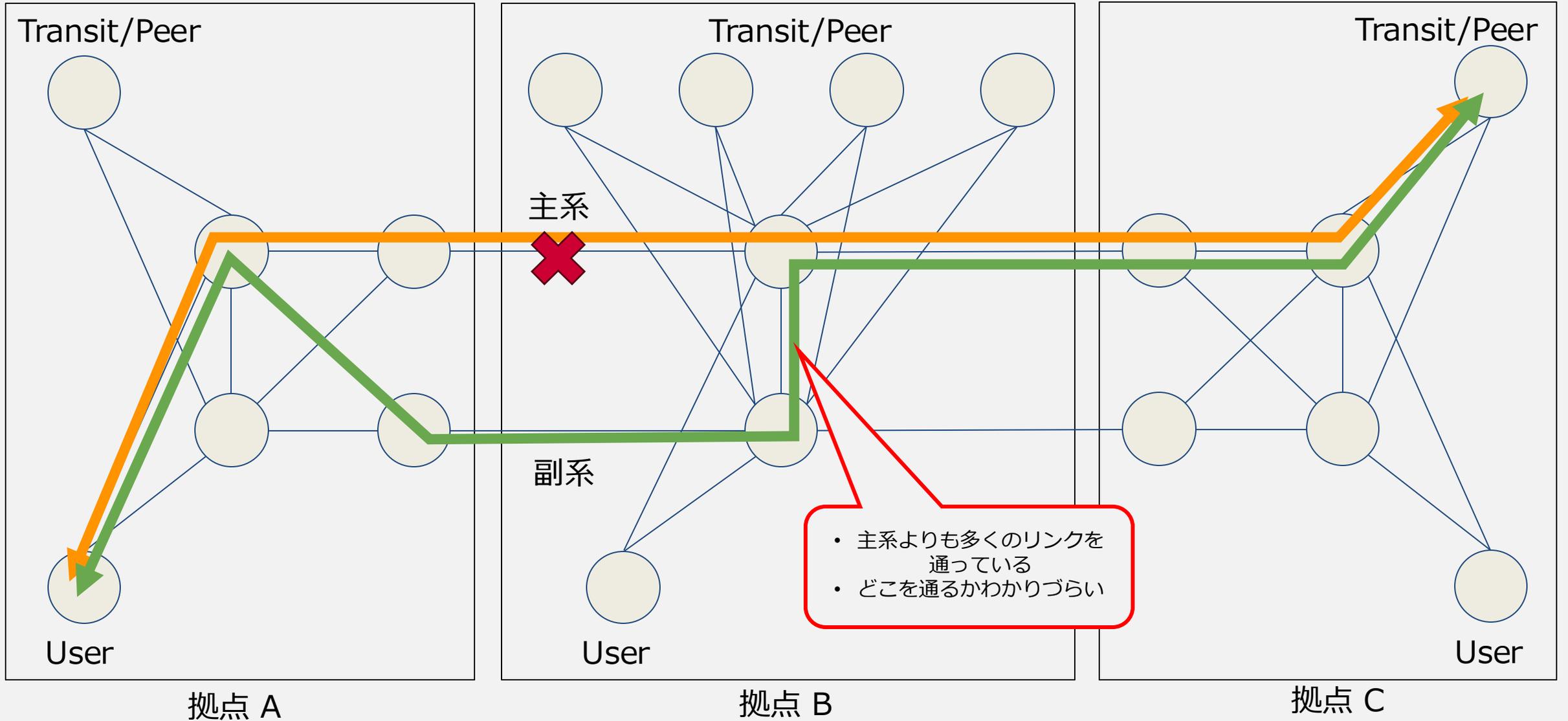
このネットワークでツールの
有用性を確認してみた



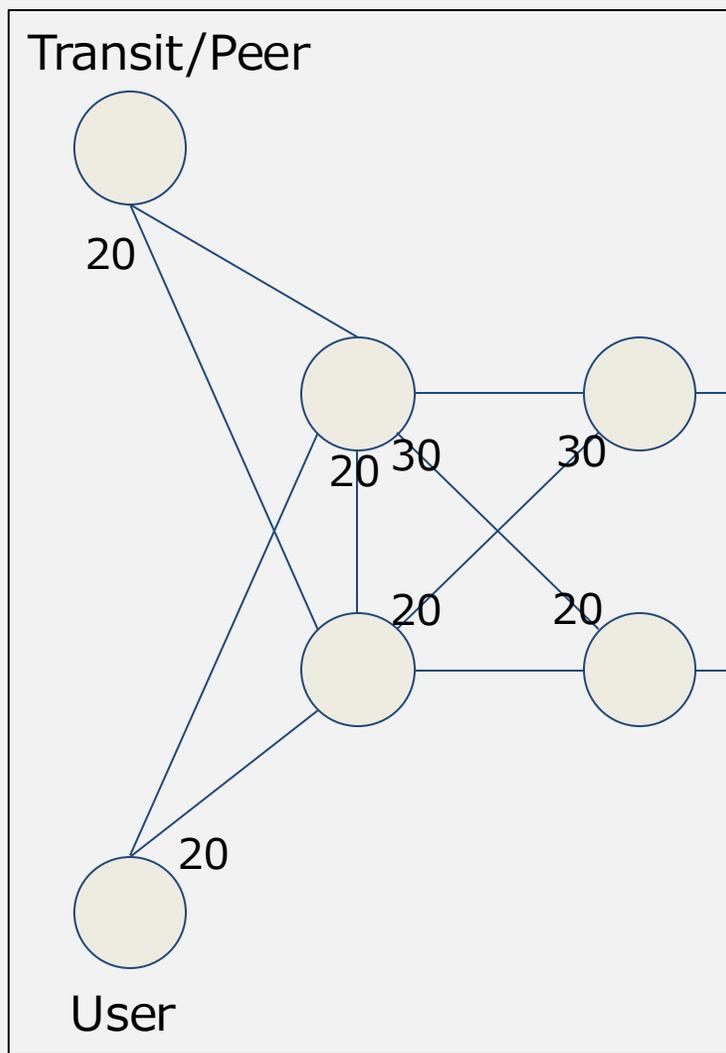
実現させる経路制御ポリシー



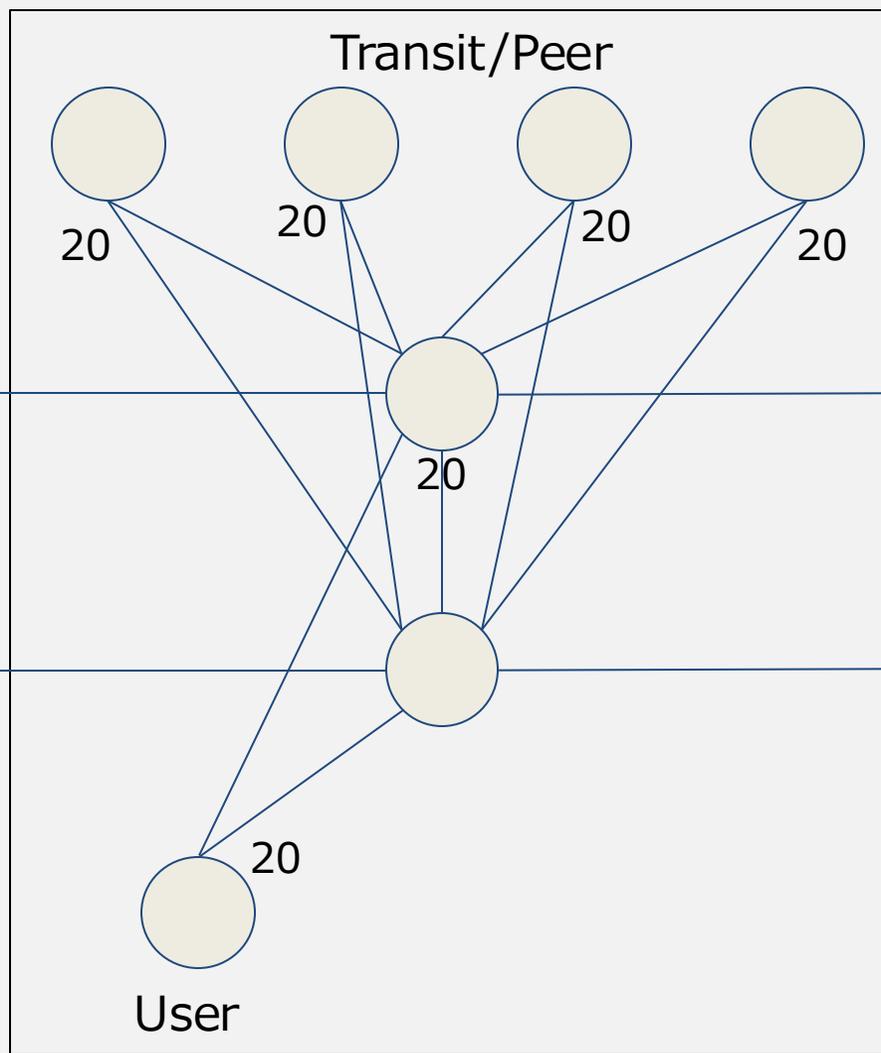
ポリシー適用前の経路



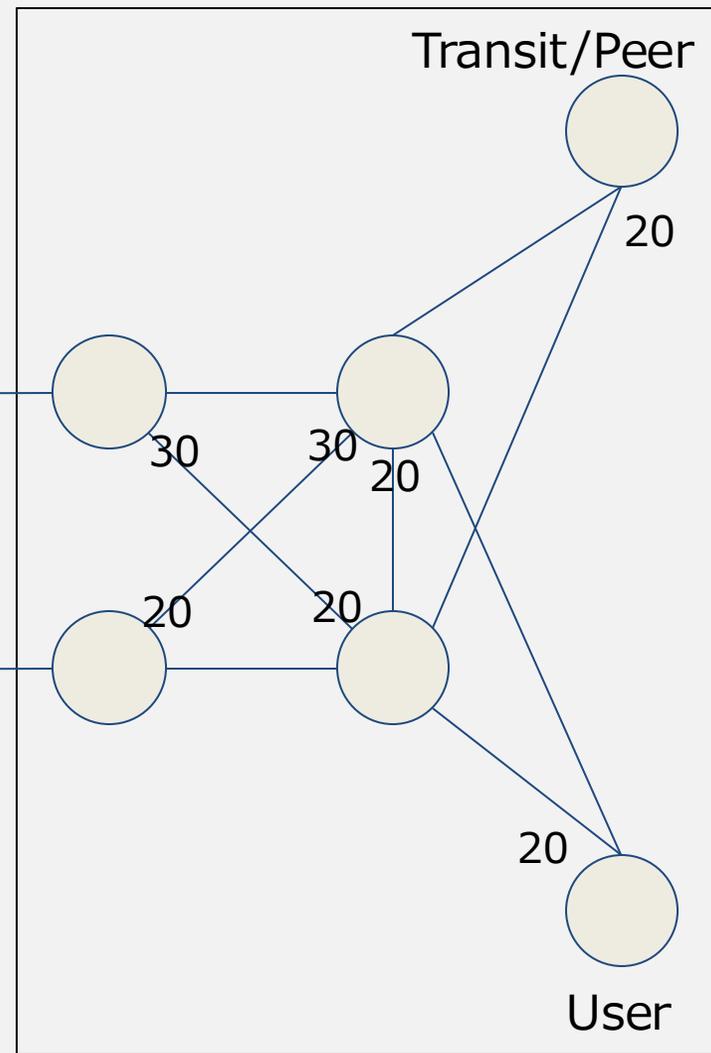
ツールで計算した IGP メトリック



拠点 A



拠点 B

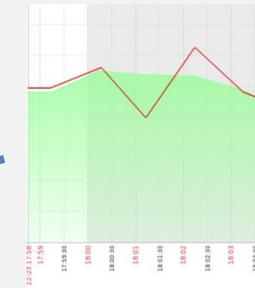
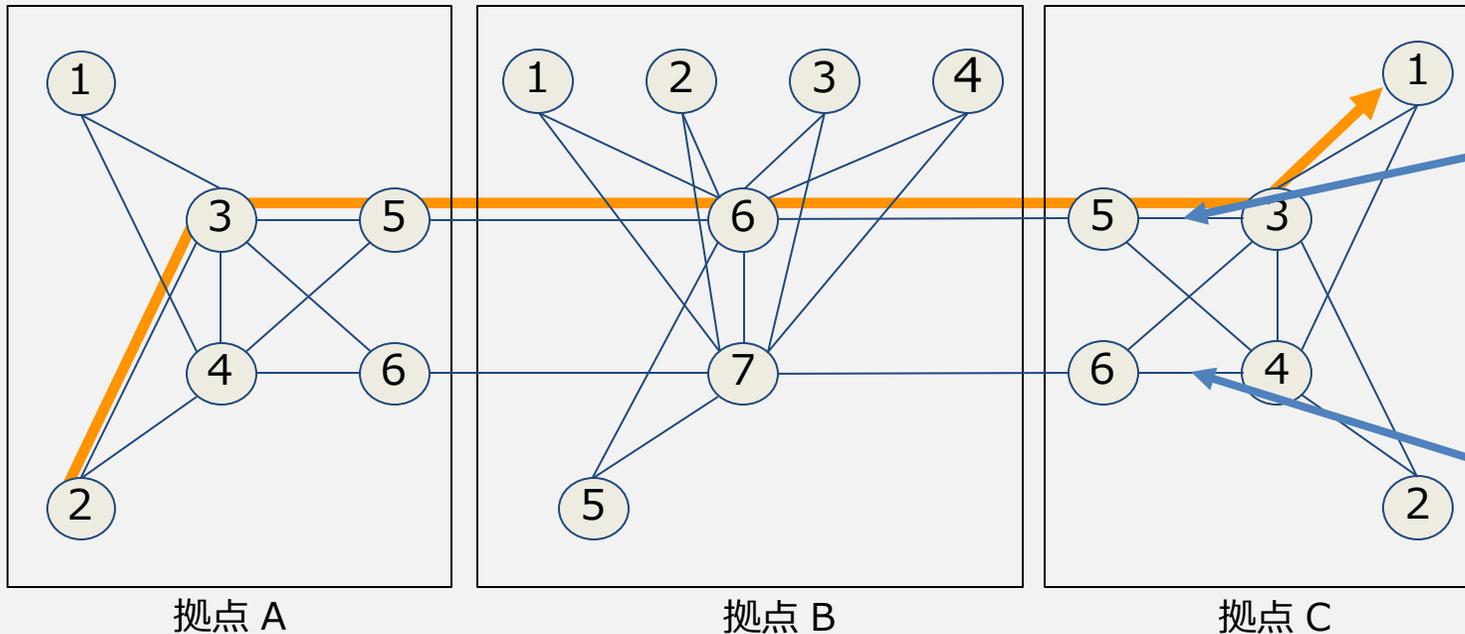


拠点 C

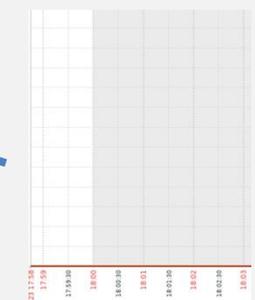
ポリシーを AS HANABI に適用した結果 (正常時)

A-2 から C-1 への traceroute 結果

```
XXX@A-2> traceroute X.X.X.1
traceroute to C-1 (X.X.X.1), 30 hops max, 52 byte packets
 1 A-3 (X.X.X.37) 0.615 ms 0.421 ms 0.413 ms
 2 A-5 (X.X.X.182) 0.425 ms 0.422 ms 0.413 ms
 3 B-6 (X.X.X.161) 1.186 ms 1.072 ms 1.238 ms
 4 C-5 (X.X.X.114) 2.071 ms 1.951 ms 1.964 ms
 5 C-3 (X.X.X.142) 2.620 ms 2.025 ms 1.806 ms
 6 C-1 (X.X.X.1) 1.234 ms 1.164 ms 1.161 ms
```



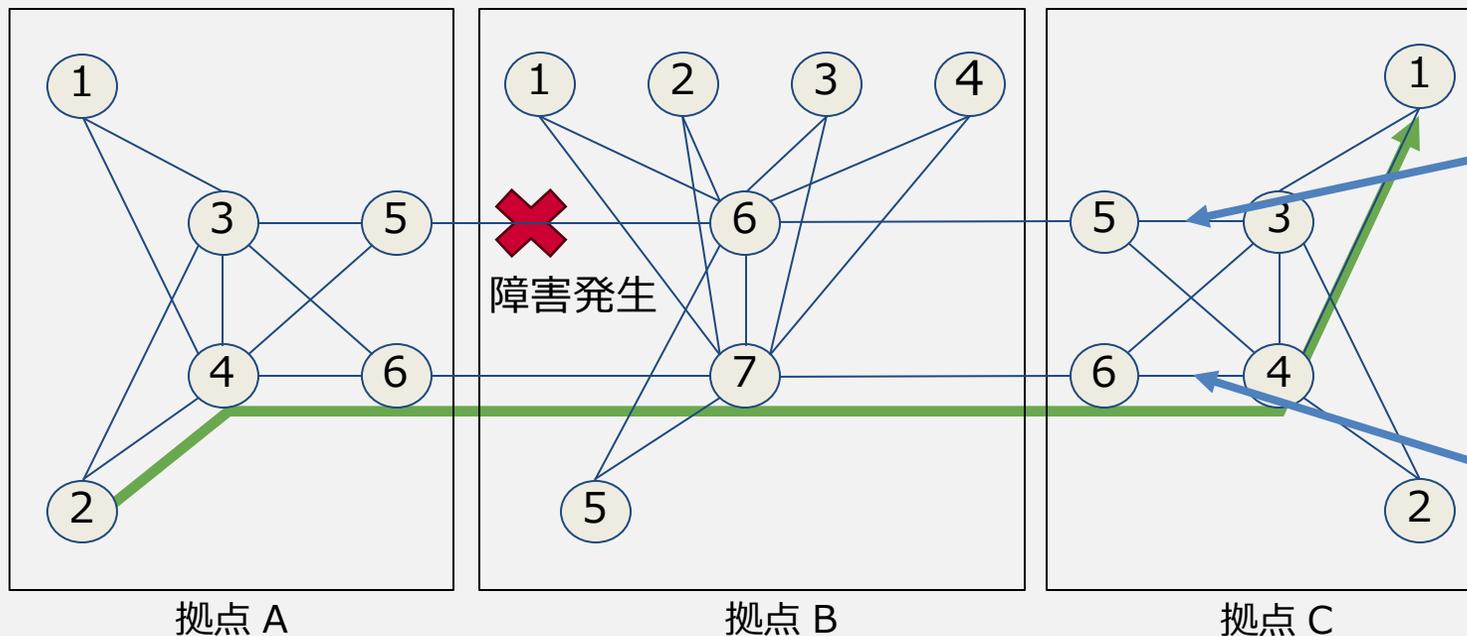
トラフィック量



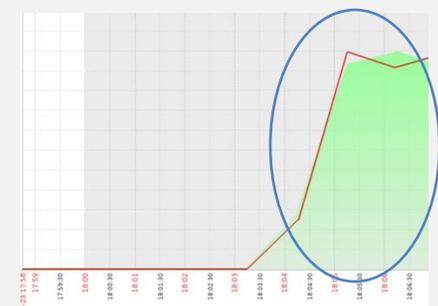
ポリシーを AS HANABI に適用した結果 (模擬障害時)

A-2 から C-1 への traceroute 結果

```
XXX@A-2> traceroute X.X.X.1
traceroute to C-1 (X.X.X.1), 30 hops max, 52 byte packets
 1 A-4 (X.X.X.69) 0.532 ms 0.421 ms 0.405 ms
 2 A-6 (X.X.X.186) 0.436 ms 0.417 ms 0.417 ms
 3 B-7 (X.X.X.85) 2.336 ms 2.239 ms 2.186 ms
 4 C-6 (X.X.X.118) 3.368 ms 3.106 ms 3.084 ms
 5 C-4 (X.X.X.154) 3.163 ms 3.164 ms 3.012 ms
 6 C-1 (X.X.X.1) 2.551 ms 2.458 ms 2.514 ms
```



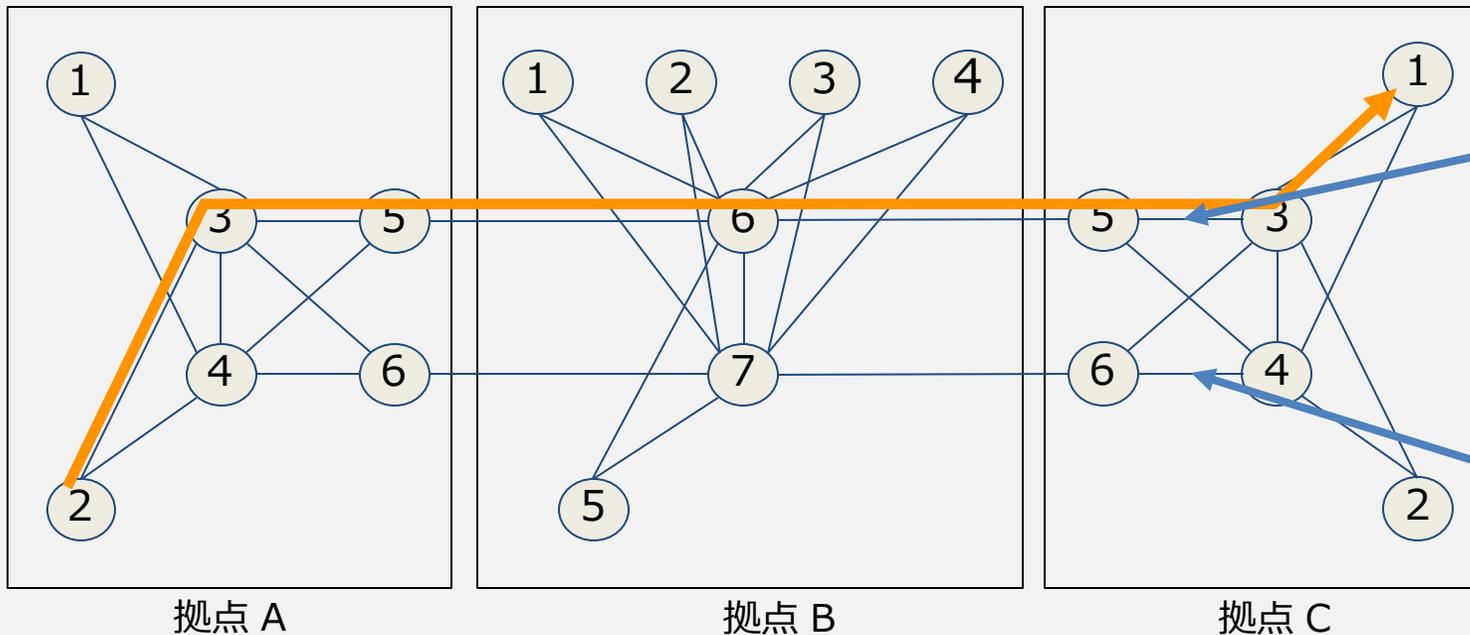
正常時と通過するリンクが
全て変わった



ポリシーを AS HANABI に適用した結果 (復旧時)

A-2 から C-1 への traceroute 結果

```
XXX@A-2> traceroute X.X.X.1
traceroute to C-1 (X.X.X.1), 30 hops max, 52 byte packets
 1 A-3 (X.X.X.37) 0.926 ms 0.463 ms 0.406 ms
 2 A-5 (X.X.X.182) 0.449 ms 0.445 ms 0.436 ms
 3 B-6 (X.X.X.161) 1.340 ms 1.084 ms 9.470 ms
 4 C-5 (X.X.X.114) 2.326 ms 1.930 ms 1.787 ms
 5 C-3 (X.X.X.142) 1.814 ms 1.900 ms 1.839 ms
 6 C-1 (X.X.X.1) 1.231 ms 1.187 ms 1.187 ms
```



通過するリンクが
元に戻った



トラフィック量



■IGP メトリックの使い方

■経路要件を満たす IGP メトリックを決める際の課題

■課題に対する我々のアプローチ

■我々のネットワークへの適用と動作確認

■まとめと議論ポイント

まとめ

- IGP メトリックを調整することで経路制御ができる
- 経路要件が複雑になると IGP メトリックを手計算で求めることが難しい
- 数理最適化問題を利用するアプローチでこの課題を解決した
- 我々のアプローチを AS HANABI に適用して、実現したい経路制御が可能であることを確認した

議論ポイント

■ IGP メトリックの使い方

- IGP メトリックをどのように決めていますか
 - 手動（オペレータの経験則、絶対的なネットワーク運用ルール など）
 - どんなポリシーやルールをもとに決めていますか
 - 自動
 - どんな入力をもとに自動計算していますか
 - 自動化にあたって、諦めた制約や自動化しきれていない部分はありますか

■ ネットワーク運用における経路制御のユースケースとその実現方法

- 他のネットワークでは聞いたことのないユースケースや技術で経路制御をやっている(た)例はありませんか

■ 数理最適化問題のネットワークでの活用事例

- 現在の活用例や適用できそうな事例はありますか