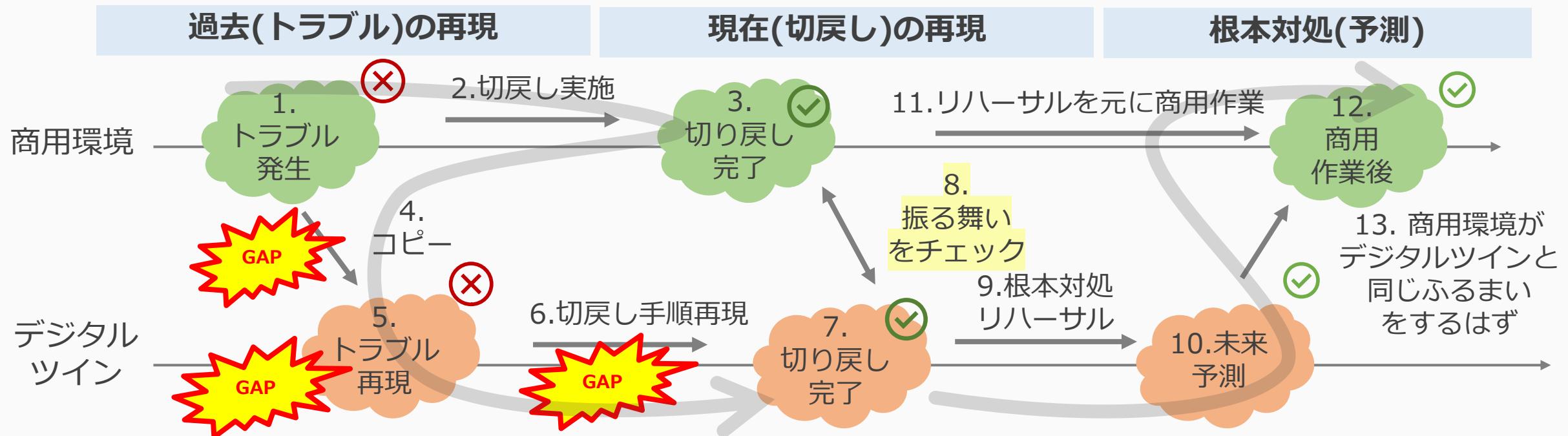


ネットワークのデジタルツイン に求める要件は何ですか？ ～理想的な仮想環境への期待と現状～

過去障害の再現シナリオ

NDT上での信用の積み重ねを検証

各再現フェーズで起こる商用とNDT(Network Digital Twin)のGAPについて、どのようなアプローチをすることで信用を積み重ねられるのかを検証する

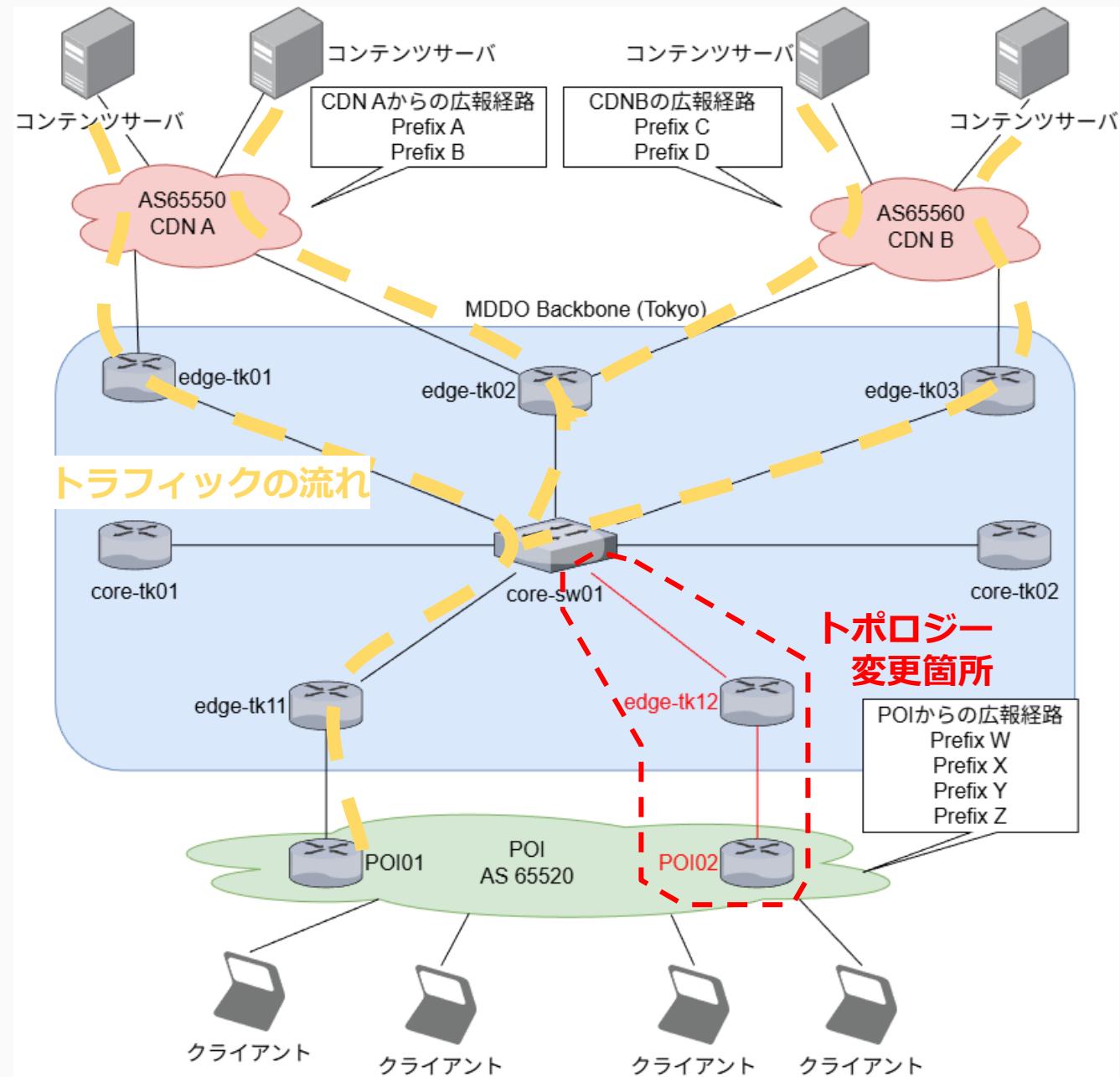


デモ構成

ISPネットワークにおける
POIと接続するルータを
増設するシナリオ

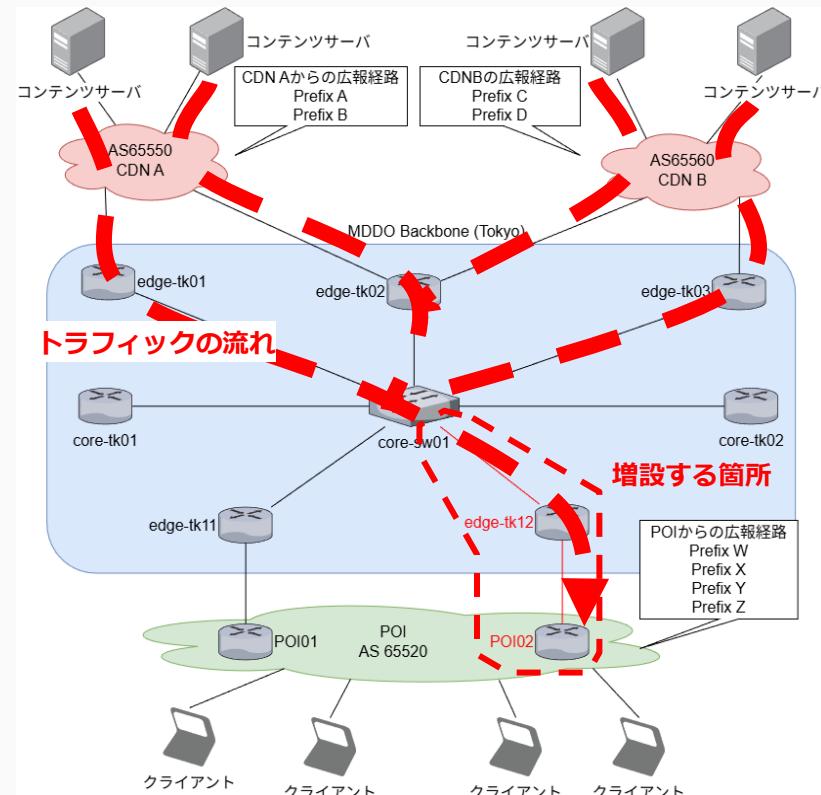
- 増設ルータ(edge-tk12)
はコアスイッチ
(core-sw01)および
POI02と接続
- POIとの接続直後は
増設ルータ経由の
通信は流さない

※コンテナ基盤としては
containerlab,
CNFはjuniper社のcRPD



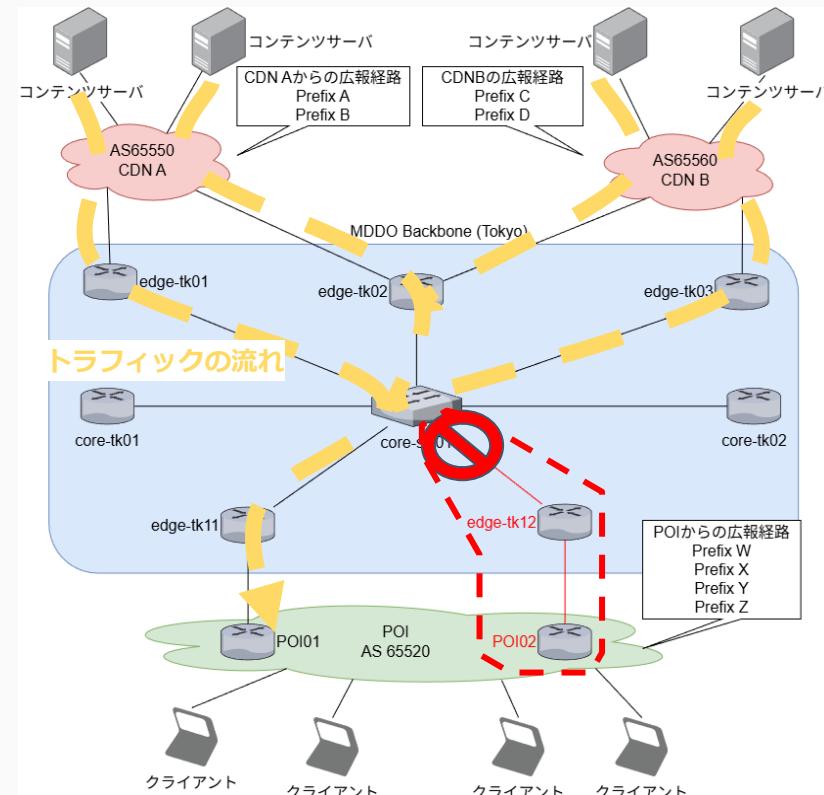
過去障害の再現シナリオ

過去(トラブル)の再現



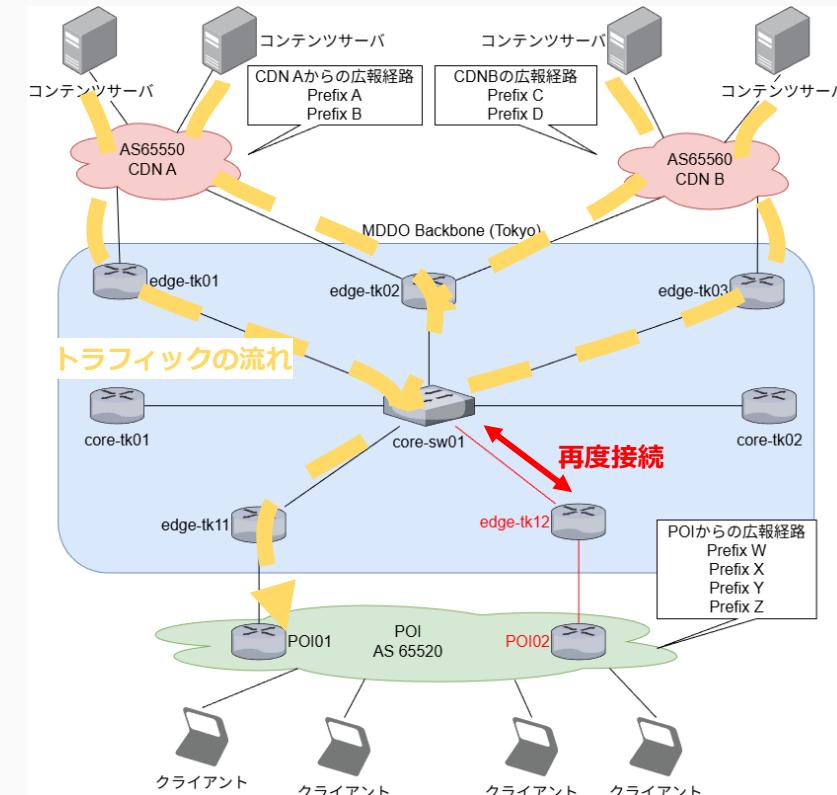
POIとのBGP接続時に
増設ルータ側へ
意図しない
経路切り替えが発生

現在(切戻し)の再現



切戻し作業を行い、
元の通信経路へ復旧

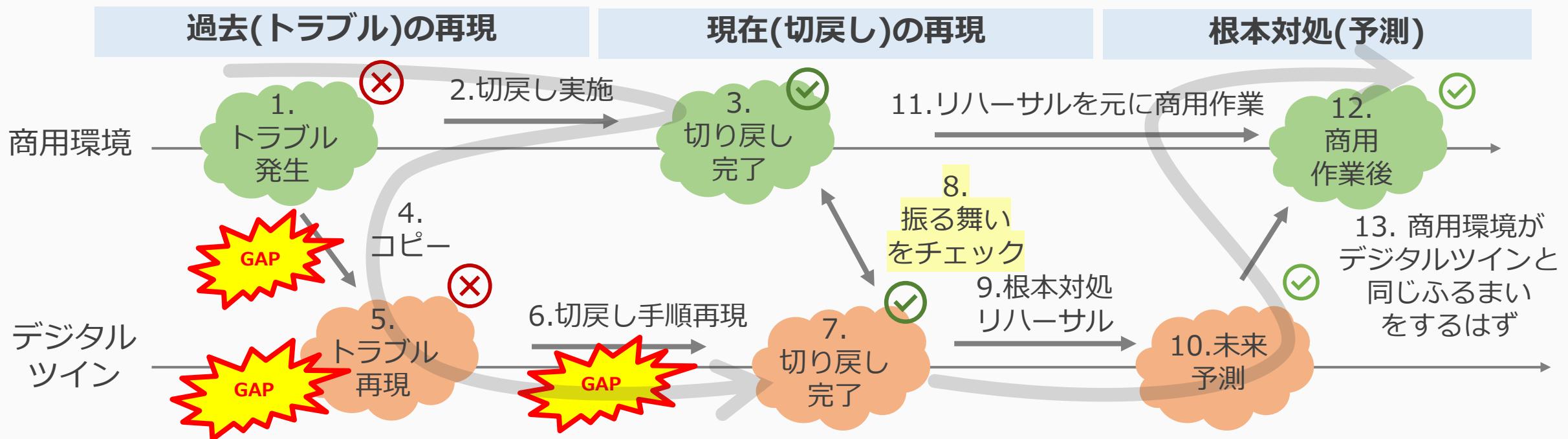
根本対処(予測)



NDT上で原因特定
根本対処の手順をNDTで実施し、
効果を判断

前提条件

- 「1.トラブル発生」から「3.切り戻し完了」までのフェーズはすでに作業完了済み。
- 4 「コピー」のフェーズからデモを開始する。

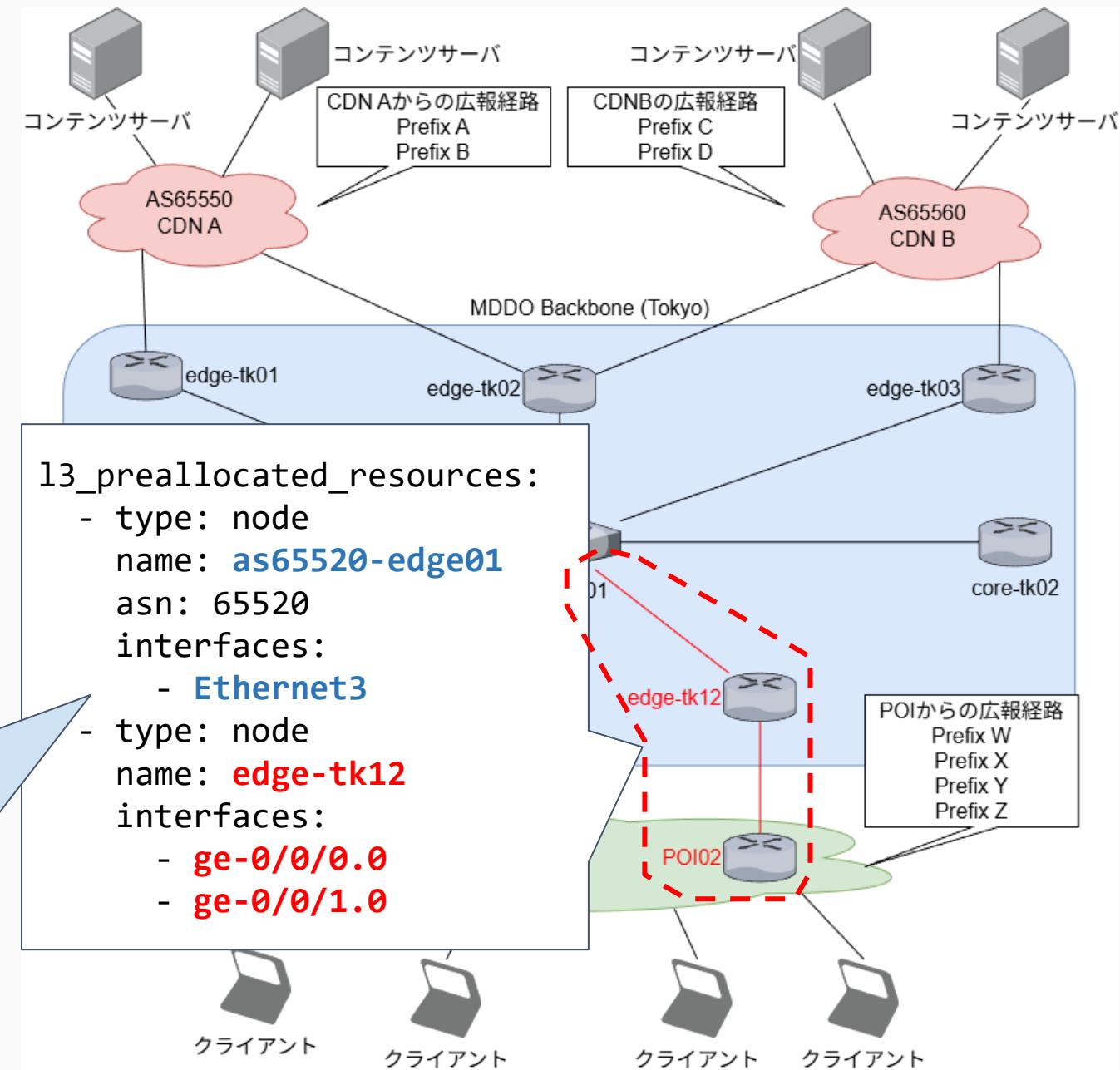


4. 「コピー」 -事前定義-

- コンテナ基盤上では動的なリソース追加はできないため、事前に**増設分のリソース定義**をしてNDTを起動する。

増設先の対向ルータ情報
は本来では不明

NDTの検証構成として
の補完ルールに則り、
対向ルータ名と対向IF名
を指定している

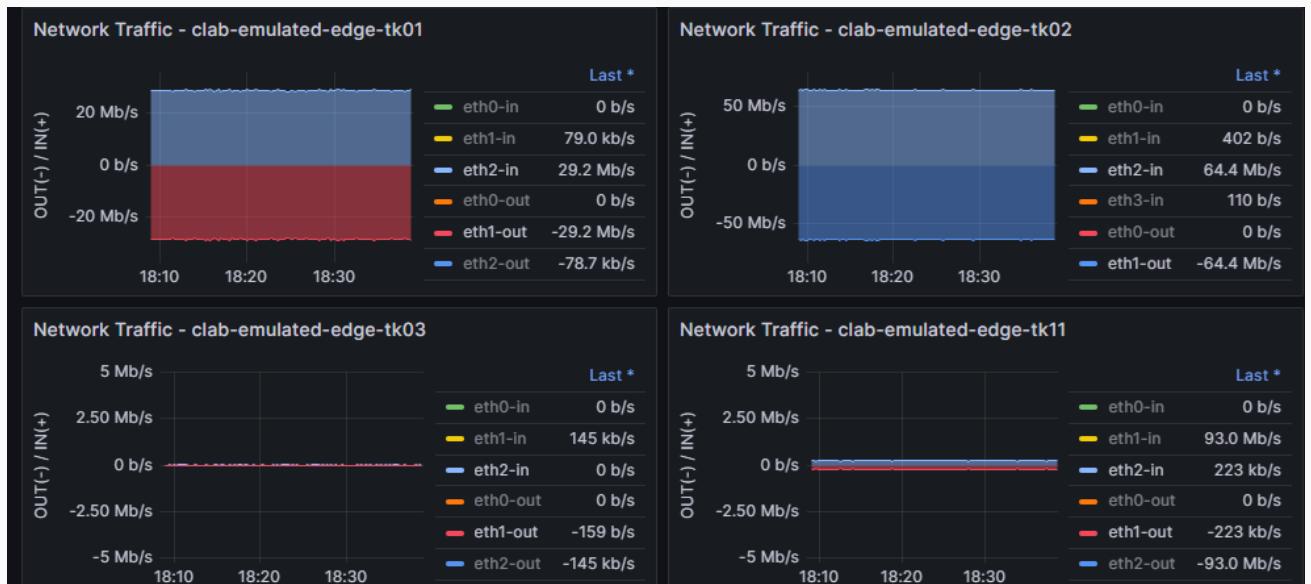


4. 「コピー」 -環境の起動-

- NDT環境の起動

- 過去のトラブル時に影響のあったEndtoEndのトラフィックを流し、トラブル前の通信状態を再現

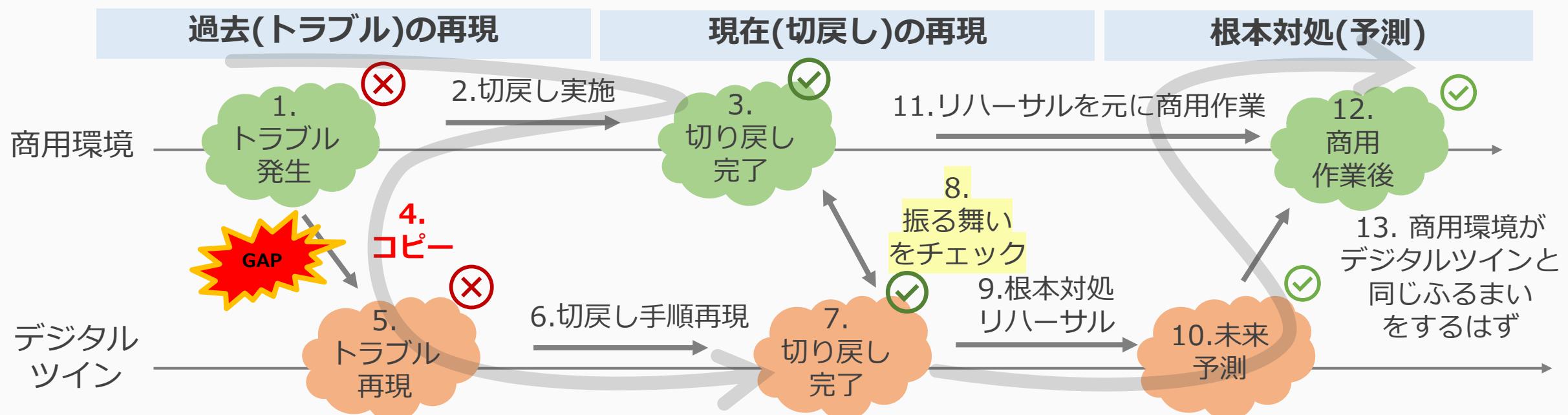
```
mddo@mddo-srv02:~/playground/demo/candidate_model_ops$ bash  
11_manual_steps.sh  
  
{"mddo-bgp": [{"physical": {"network": "mddo-bgp", "snapshot": "original_asis", "label": "original_asis"}, "logical": []}]}  
Network:mddo-bgp uses BGP, expand external-AS network and splice it  
into topology data  
{}  
{  
    "status": 201  
}
```



4. 「コピー」 -増設対象のルータをNDT上へ追加-

- ここでGAPが発生

- ①コンテナ基盤上で回線開通前の状態再現ができない
- ②商用IF名でトポロジー変更ができない

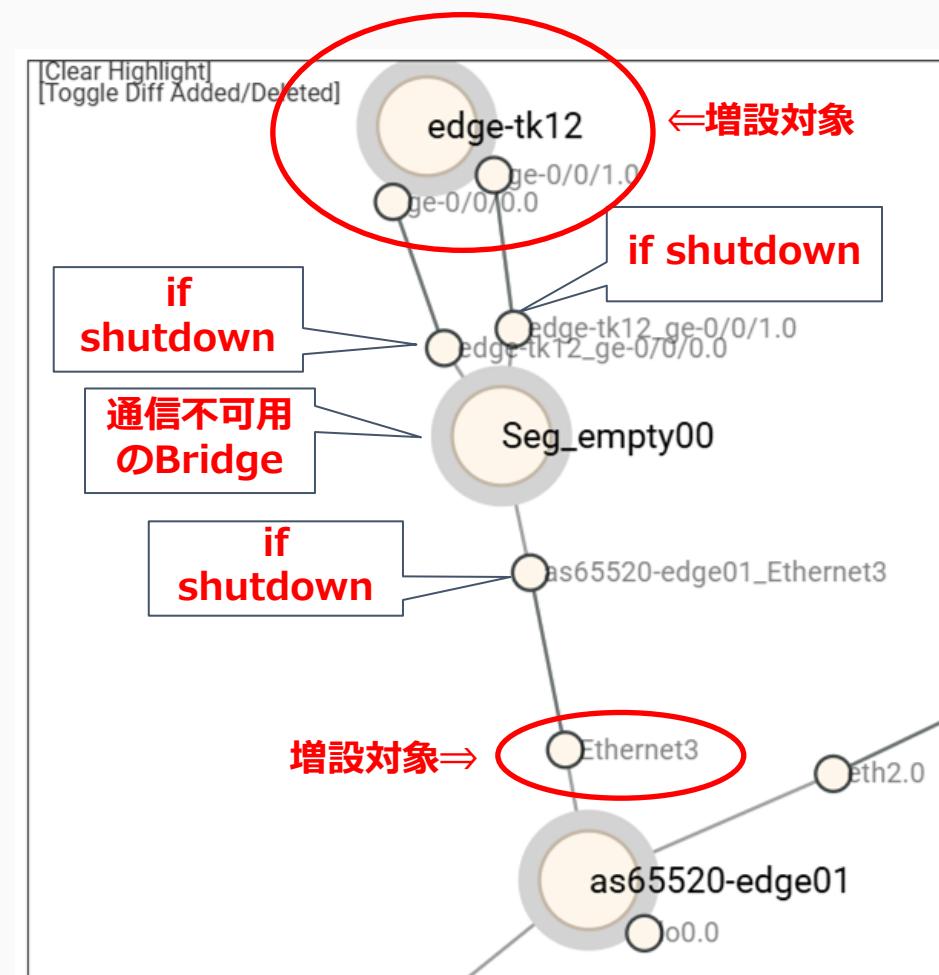


①回線開通前の 状態再現ができない

- 開通前の状態を再現するには
増設するIFを通信できない状態で
保持する仕組みが必要



通信不可用のOVS Bridge
(Seg_empty00)を用意し、これに
繋がるportをIF Shutdown状態にする
仕組みを実装して状態を再現



```
mddo@mddo-srv02:~/playground/demo/candidate_model_ops$ docker exec -it mddo-clab-docker /bin/bash
mddo-srv02:/home/mddo/playground/repos/mddo-worker/clab# ip link | grep "ovs-system state DOWN"
659: br25p0@if658: <BROADCAST,MULTICAST,M-DOWN> mtu 9500 qdisc noqueue master ovs-system state DOWN
673: br25p1@if672: <BROADCAST,MULTICAST,M-DOWN> mtu 9500 qdisc noqueue master ovs-system state DOWN
685: br25p2@if684: <BROADCAST,MULTICAST,M-DOWN> mtu 9500 qdisc noqueue master ovs-system state DOWN
```

②商用IF名でトポロジー 変更ができない

商用IF名でトポロジー変更するには
以下の課題があった。

- 商用のIF名はNDT上ではコンテナルータのIF名(ethX)に置き換わってしまう。
 - IFがつながるBridgeの対向ポート名も商用IF名からは特定できない。

OVS・CNFのIFをマッピングする機能を実装

マッピング例 : 商用(Original)のIF名

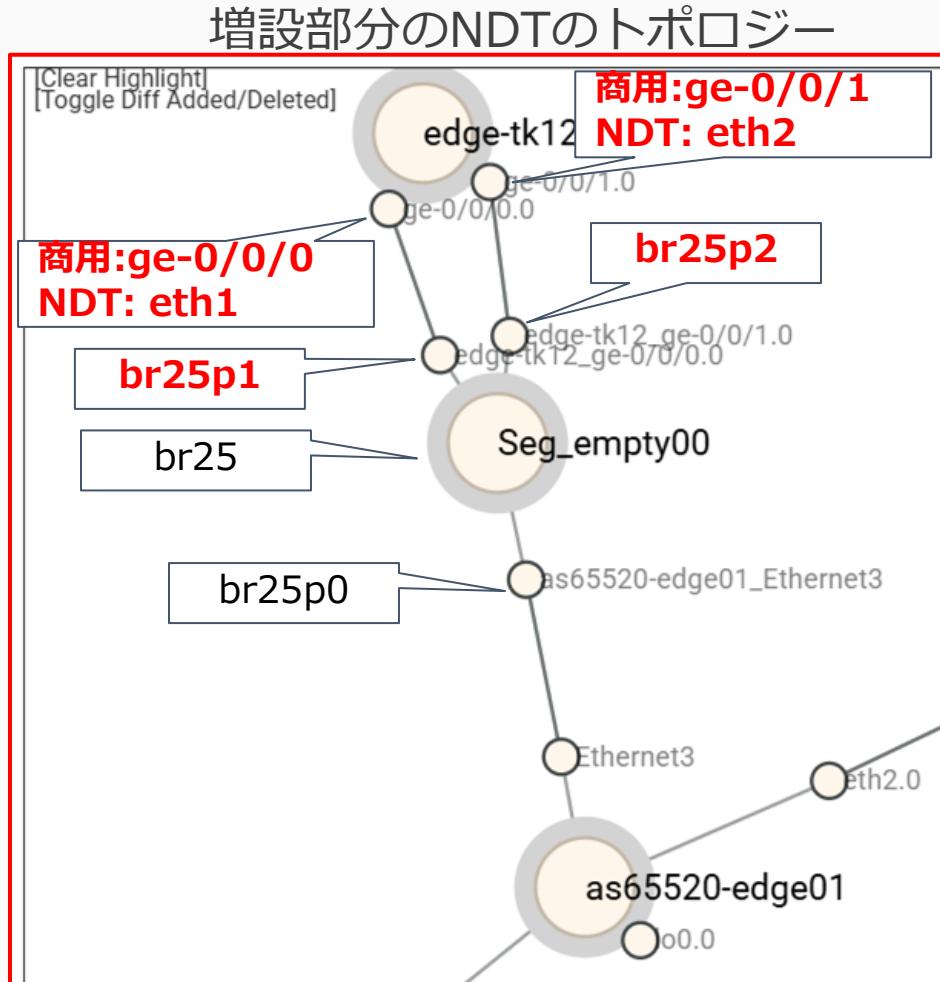
ge-0/0/0

NDT上のコンテナルータに対応するIF名

eth1

NDT上の接続先Bridgeの対向ポート名

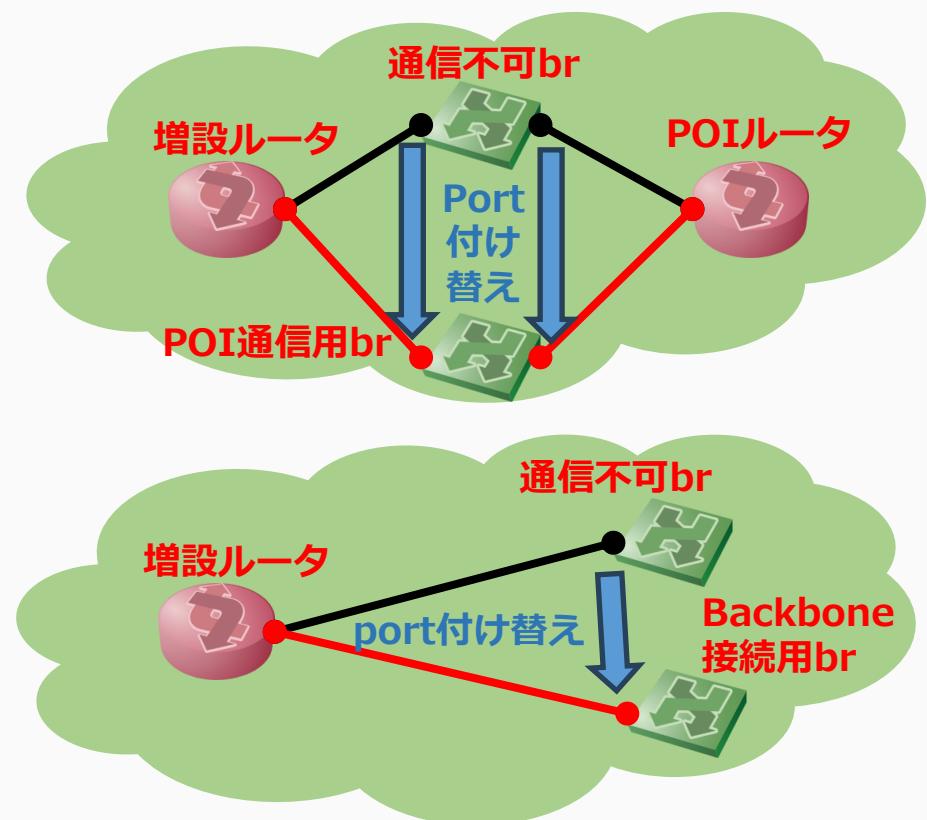
br25p1



②商用IF名でトポロジー 変更ができない

アドホックな変更を実現するために、OVSの付け替えを自動化

- 操作①：POI側との接続
通信不可用のovs-brから
POIとの通信用の新設したovs-brへ
対象のポートを付け替えする
- 操作②：Backbone側との接続
通信不可用のovs-brから他Backbone
ルータと通信できる既設のovs-brへ
対象ポートを付け替えする

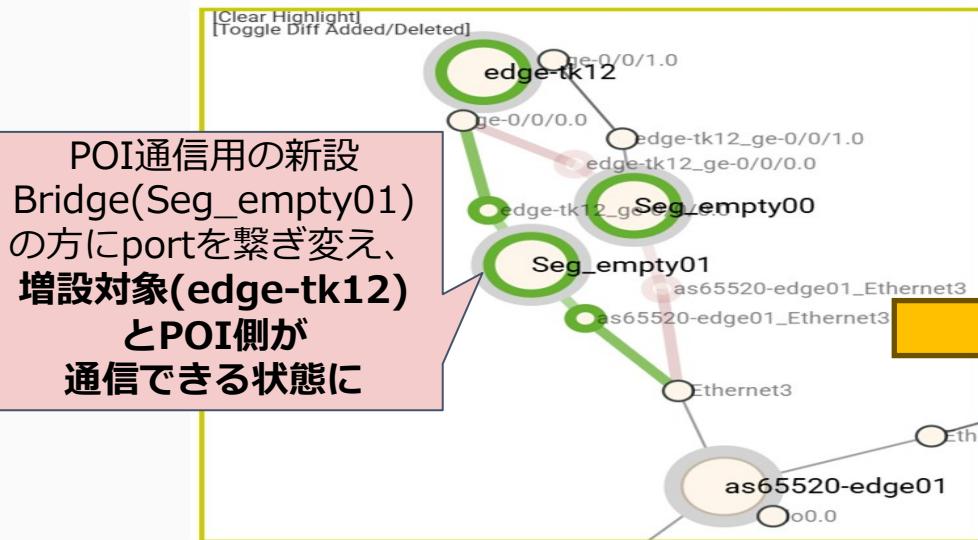


4. 「コピー」 -トポロジーの変更-

商用IF名を使ってアドホックな構成変更を実現

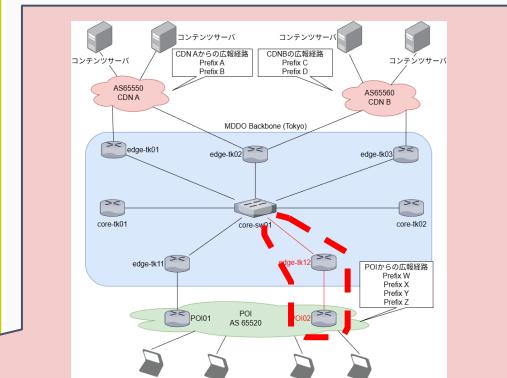
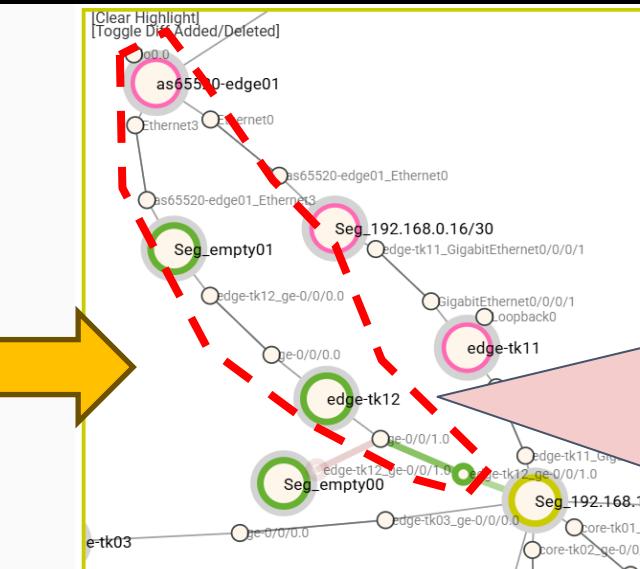
- ①POI側トポロジー変更

```
mddo@mddo-  
srv02:~/playground/demo/candidate_model_ops$ ./topo_frontend.py  
link --src as65520-edge01[Ethernet3] --dst edge-tk12[ge-0/0/0.0]
```



- ②Backbone側トポロジー変更

```
mddo@mddo-  
srv02:~/playground/demo/candidate_model_ops$ ./topo_frontend.py  
link --src edge-tk12[ge-0/0/1.0] --dst core-tk01[ge-0/0/0.0]
```



5. 「トラブル再現」

-BGP接続-

- 商用作業同等の手順をNDT上で実行する



手順の実行後、
BGP接続は確立するが、、、



```
root@edge-tk12# run show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed      History Damp State      Pending
inet.0
          40          37          0          0          0          0          0
Peer          AS      InPkt  OutPkt  OutQ  Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.200.2      65520      51      53      0      0      21:17 Establ
      inet.0: 8/9/9/0
192.168.255.101      65500      57      49      0      0      21:36 Establ
      inet.0: 29/31/31/0
```

設定作業2(切り戻しポイント作成)

```
save 20251204-5.conf
run file list
```

- 切り戻し用ファイルがあることを確認

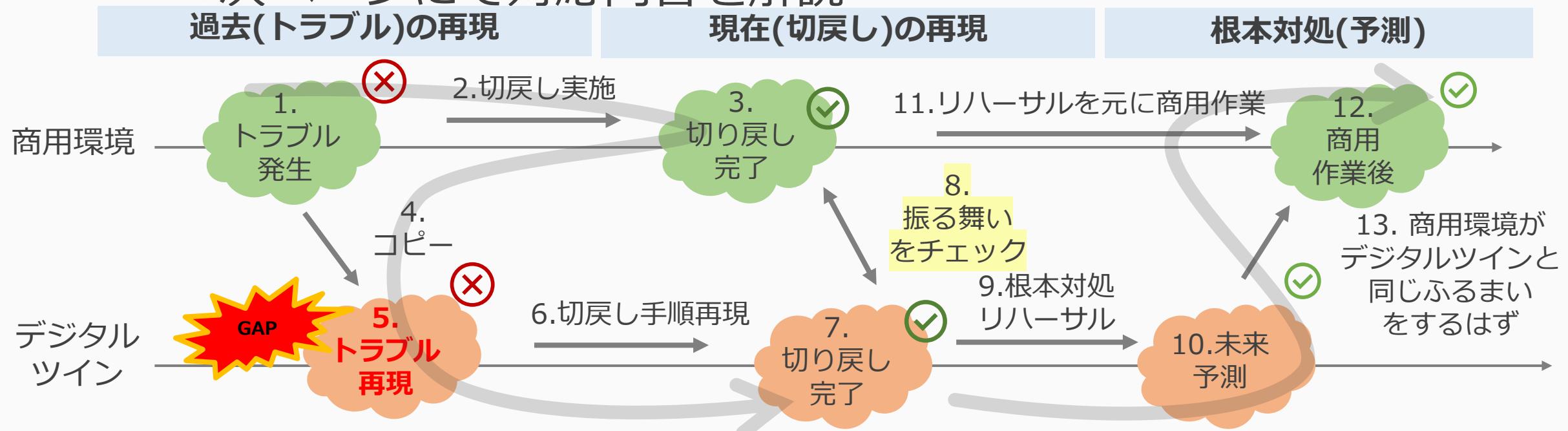
設定変更3(設定投入)

```
set protocols bgp group 192.168.200.2 type external
set protocols bgp group 192.168.200.2 hold-time 90
set protocols bgp group 192.168.200.2 family inet unicast
set protocols bgp group 192.168.200.2 peer-as 65520
set protocols bgp group 192.168.200.2 neighbor 192.168.200.2 local-address 192.168.200.1
set protocols bgp group 192.168.200.2 neighbor 192.168.200.2 import POI-East_in
```

- 入力失敗が出ていないか確認
- 異常なログが出ていないことを確認

5. 「トラブル再現」 GAPの発生

- 商用手順をNDT上で実行するにあたりGAPが発生
 - ①物理とCNFの構成差分
 - ②オペレーション時の読み替え問題
- ⇒次ページにて対応内容を解説



①物理とCNFの構成差分

- 物理とCNFの構成差分による実行できないコマンド
 - シャーシ型・ボックス型等の機種依存コマンド
 - CNFの製品コンセプトでオミットされた機能

⇒この差分はいったんSKIPする。
トラブル再現時に必須な場合は再現可能なCNF/VNFに対象を置き換える。

コンテナ上のlinux OSの機能で
IF down/upなど
代替できる振る舞いもあった

機種依存コマンド

事前確認2(トランシーバーの確認)

show chassis hardware | no-more



シャーシ型機種
依存コマンド
の実行エラー

```
root@edge-tk12> show chassis
^
syntax error, expecting <command>.
root@edge-tk12> show chassishardware
^
```

オミットされたコマンド

```
root@edge-tk12# set interfaces eth1 disable
^
syntax error.
```

```
root@edge-tk12# set interfaces eth1 gigether-
options802.3ad
^
syntax error.
```

②オペレーション時の読み替え問題

- 商用のIF名でオペレーション：
CNFでのIF名で操作することになる。
マッピング機能を元に商用IF名から
NDT上のIF名への変換をした手順を
事前準備する必要が発生。
- トラフィック確認：
振る舞いの再現確認に使う
トラフィック監視画面もマッピング
機能を基に**Original IF名を埋め込み**
する対応が発生。

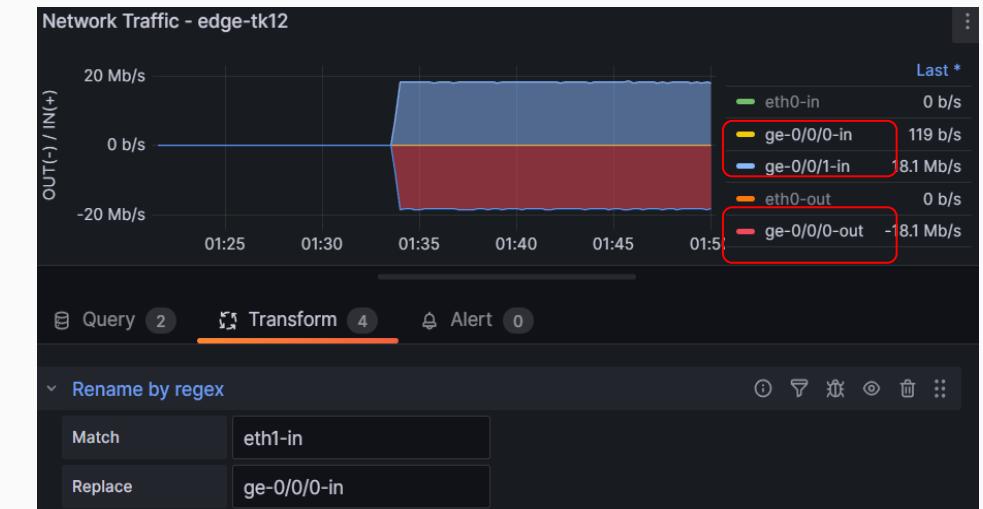
読み替えした手順

事後確認3(状態確認)

```
show interfaces terse | match "eth1|eth2"
```

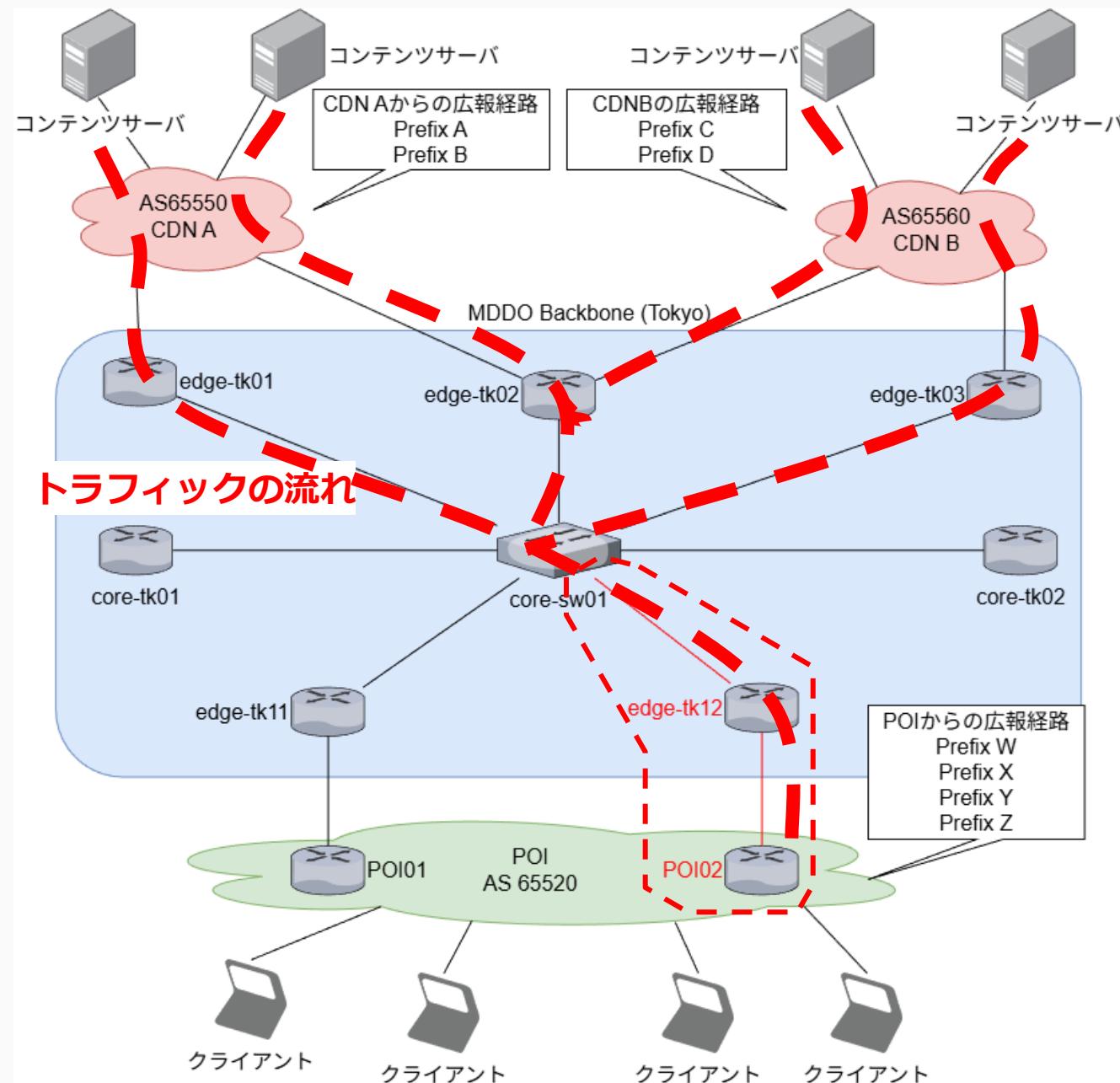
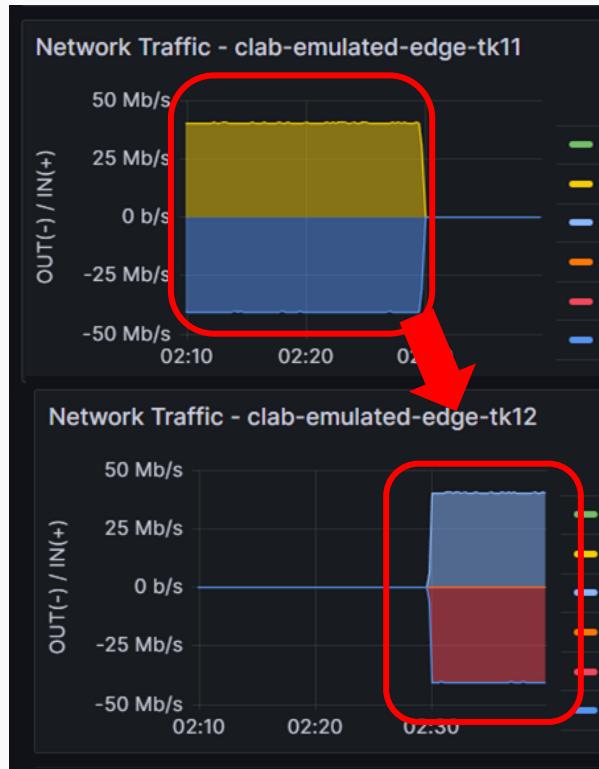
eth1 (ge-0/0/0.0) が up していること

eth2 (ge-0/0/1.0) が up していること



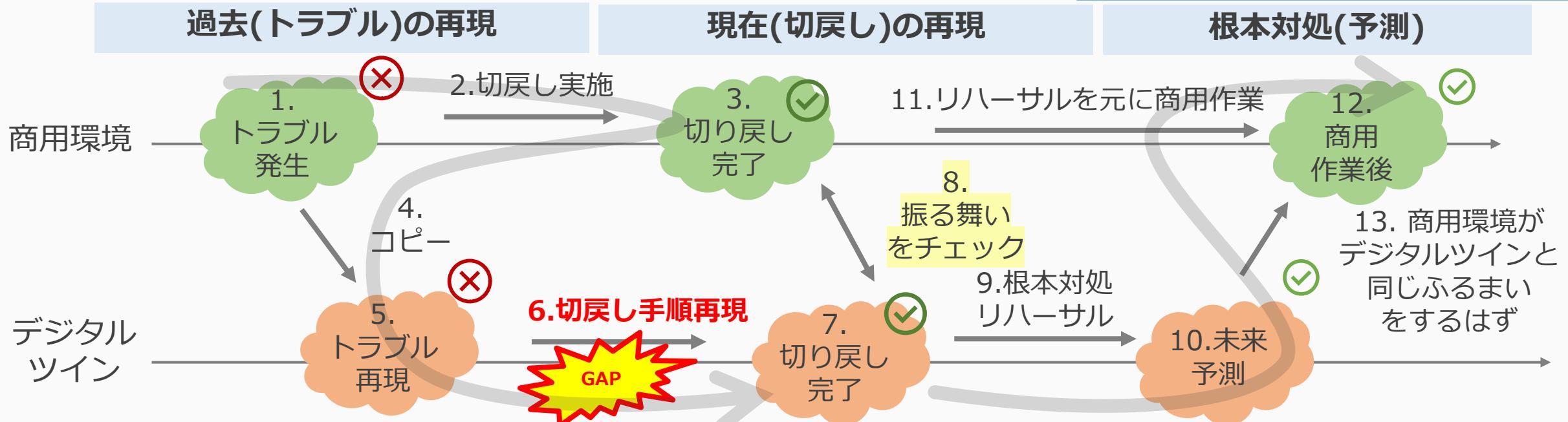
5. 「トラブル再現」

- edge-tk11⇒edge-tk12へ
通信の切替わりが発生



6.切り戻し手順再現

- 商用実績の切り戻し作業をNDT上で再現。
⇒発生したGAPの対処(オペレーション時の読み替えの問題と同じ対応を実施)



切り戻し(eBGP)

```
run file list
load override 20251204-5.conf
```

「load complete」と表示されるか確認

commitの実施

show | compare

scriptで投入した設定が切り戻していること

commit check
commit

7.切り戻し完了

- 商用実績の切り戻し作業をベースとしたNDT変換手順で切り戻し作業は実績通り正常に完了

①POIとのBGP接続前の切り戻しポイントのコンフィグを正常にロード。

```
root@edge-tk12# load override 20251204-5.conf
load complete

[edit]
```

②BGP開通のコンフィグ分がエラーなく巻き戻る内容であることを確認して、ロールバック実行

```
root@edge-tk12# show | compare
[edit protocols bgp]
- group 192.168.200.2 {
-     type external;
-     hold-time 90;
-     family inet {
-         unicast;
-     }
-     peer-as 65520;
-     neighbor 192.168.200.2 {
-         local-address 192.168.200.1;
-         import POI-East_in;
-     }
- }
```

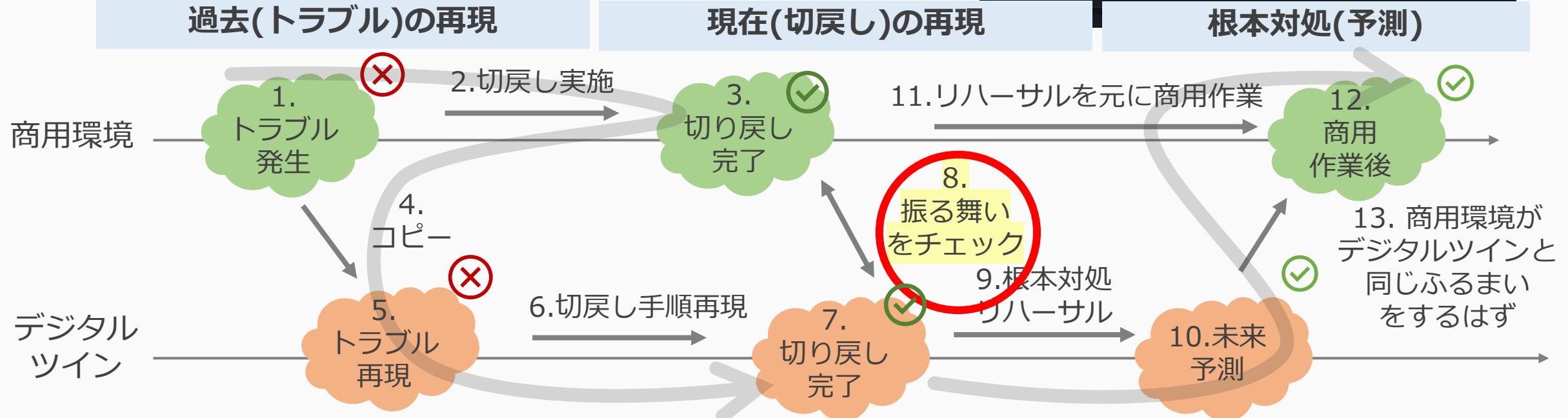
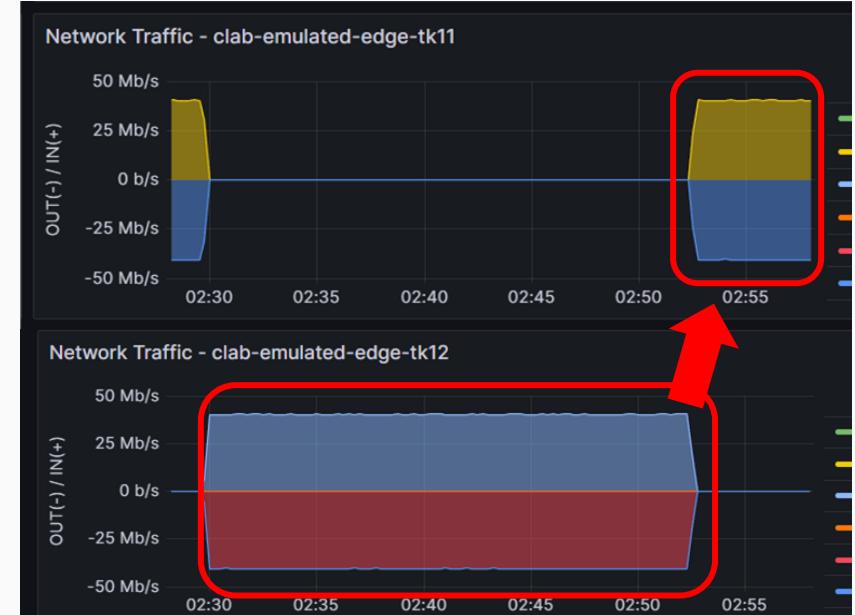
```
[edit]
root@edge-tk12# commit check
```

```
configuration check succeeds
```

```
[edit]
root@edge-tk12# commit
commit complete
```

8.振る舞いをチェック

Grafanaにて切り戻し後の動作を確認
⇒トラフィックが作業前の経路に戻り、
商用(切り戻し)と同じ状態となった



10.未来予測①

- 再度トラブル状態を再現してNDT上で状態を確認して、想定外の経路の優先度になっている箇所を特定。

そこから**誤設定**を検出し、商用向けの修正手順を用意

```
root@core-tk01> show route inet.0: 49 destinations, 94 routes (49 active, 0
holddown, 0 hidden) + = Active Route, - = Last Active, * = Both
<省略>

10.100.0.0/24      *[BGP/170] 00:02:16, MED 100, localpref 100, from
192.168.254.12

192.168.255.11      AS path: 65520 I, validation-state: unverified
> to 192.168.1.12 via eth1
[BGP/170] 23:34:30, MED 100, localpref 100, from
192.168.255.11      AS path: 65520 I, validation-state: unverified
> to 192.168.1.11 via eth1
```



想定では赤字の経路が優先されてほしいが、
青字の経路が優先されている。
⇒設定ミスの箇所を特定

設定変更3(設定投入)

```
set protocols bgp family inet unicast
set protocols bgp group 192.168.255.101 type internal
set protocols bgp group 192.168.255.101 hold-time 90
set protocols bgp group 192.168.255.101 family inet unicast
set protocols bgp group 192.168.255.101 peer-as 65500
set protocols bgp group 192.168.255.101 local-as 65500
set protocols bgp group 192.168.255.101 neighbor 192.168.255.101 local-address 192.168.254.12
set protocols bgp group 192.168.255.101 neighbor 192.168.255.101 export logp-export
```

10. 未来予測②

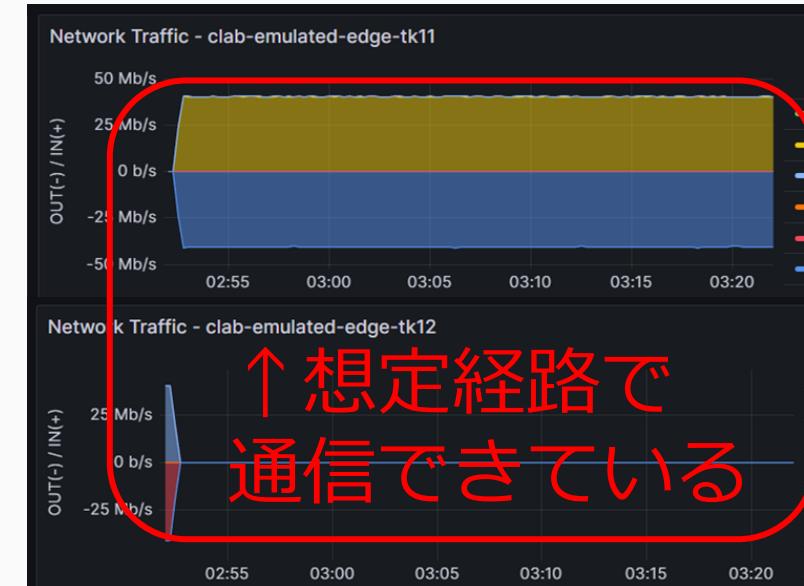
- NDTで修正手順をリハーサルし、増設後の構成でBGP接続しても経路変更が起こらないことがわかった。
⇒手順の妥当性を確認できた

修正手順の実施

```
[edit]
root@edge-tk12# delete interfaces lo0 unit 0 family inet
address 192.168.254.12/32
set interfaces lo0 unit 0 family inet address
192.168.255.12/32
delete routing-options router-id 192.168.254.12
set routing-options router-id 192.168.255.12
```

新設ルータはBGP接続している
(現在から進んだ状態)

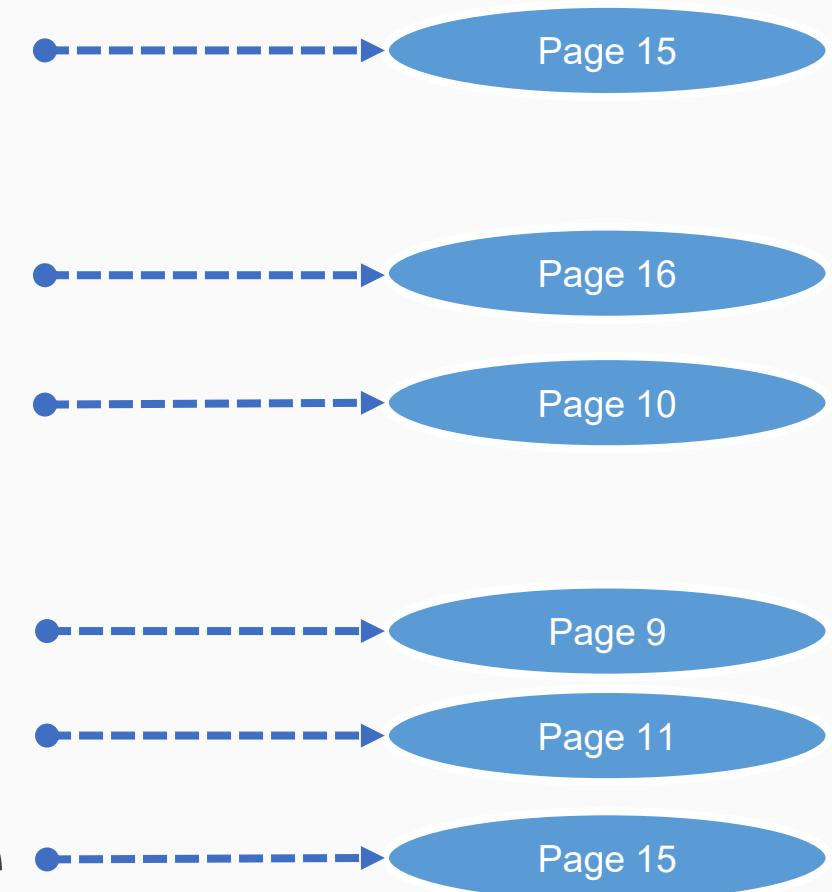
```
root@edge-tk12# run show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths  Suppressed  History
Damp State    Pending
inet.0
0          0
Peer          AS  InPkt  OutPkt  OutQ
Flaps  Last Up/Dwn State |#Active/Received/Accepted/Damped...
192.168.200.2 65520 4 6 0
0          1 Establ
inet.0: 8/9/9/0
192.168.255.101 65500 10 3 0
0          17 Establ
inet.0: 29/38/38/0
```



考察

直面したGAPのまとめ

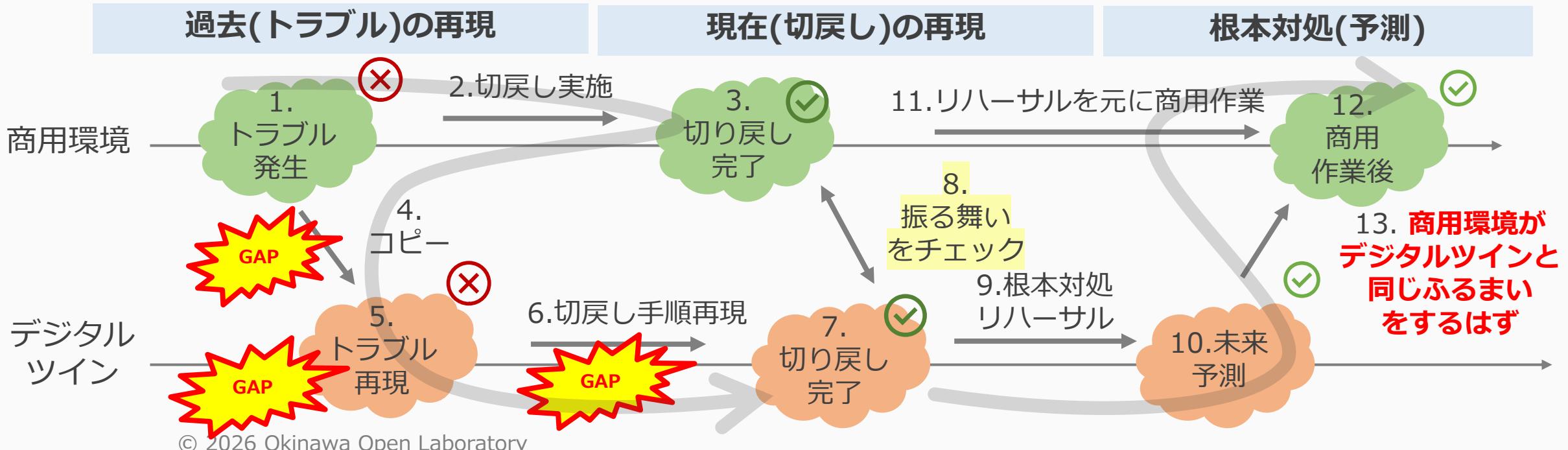
- ・ノードのH/W構成依存コマンド実行
- ・物理トポロジ操作の読みかえ
 - ・操作対象ノードのインターフェース名
 - ・L3トポロジ操作への読みかえ
- ・仮想環境固有のノード状態操作
 - ・Ad hocなリソース追加ができない
 - ・Ad hocなトポロジ変更ができない
 - ・インターフェースshutdownができない



考察

過去(トラブル)、現在(切り戻し)の各再現フェーズにて、NDT (Network Digital Twin)が**商用と同様に振る舞うことが確認できた**。これによって、この過去と現在の間の**GAPが存在**しても NDT上の振る舞いが**信用できる**とわかった。

⇒延長線上であればNDT上での**未来の予測も信用できる**



[おまけ]物理とCNFの構成差分の解消も進んでいる

- containerlab: コンテナルータ上のIF名でトポロジーの定義が可能に

Topology definition

Using Linux interfaces Using interface aliases

Using the `ethx` interface naming convention, the topology would look like this:

```
links:
- endpoints: ["srl:e1-1", "vEOS:eth1"]
- endpoints: ["vSRX:eth3", "vEOS:eth2"]
- endpoints: ["CSR1000v:eth4", "vSRX:eth6"]
- endpoints: ["vEOS:eth3", "CSR1000v:eth2"]
```

IFの読み替えが不要に

Using Linux interfaces Using interface aliases

Using aliased interface names, the topology definition becomes much more straightforward:

```
links:
- endpoints: ["srl:ethernet-1/1", "vEOS:Ethernet1/1"]
- endpoints: ["vSRX:ge-0/0/2", "vEOS:Ethernet1/2"]
- endpoints: ["CSR1000v:Gi5", "vSRX:ge-0/0/5"]
- endpoints: ["vEOS:Ethernet1/3", "CSR1000v:Gi3"]
```

(旧来の)Containerlab上のトポロジー定義
vSRX#1: **eth1** <==> vEOS#1: **eth1**

vSRX#1
ge-0/0/1

vEOS#1
Ethernet1/1

vethXX

vethYY

仮想bridge

最新のContainerlab上のトポロジー定義
vSRX#1: **ge-0/0/1** <==> vEOS#1: **Ethernet1/1**

vSRX#1
ge-0/0/1

vEOS#1
Ethernet1/1

vethXX

vethYY

仮想bridge

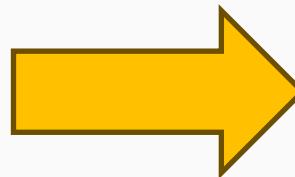
[おまけ]物理とCNFの構成差分の解消も進んでいる

- Nokia社のCNFのSR-SIM : 物理機器と同様のコマンド体系を実現

```
13_preallocated_resources:
  - type: node
    name: as65520-edge01
    asn: 65520
    interfaces:
      - Ethernet3
  - type: node
    name: edge-tk12
    interfaces:
      - 1/1/c12/1
      - 1/1/c21/1
    emulated_params:
      license: ./sros_license.txt
      image: localhost/nokia/srsim:25.7.R1
      kind: nokia_srsim
      type: SR-2s
      components:
        - slot: A
  - slot: 1
    type: xcm-2s
    sfm: sfm-2s
    env:
      NOKIA_SROS_CARD: xcm-2s
      NOKIA_SROS_MDA_1: s36-100gb-qsfp28
```

増設リソースにSR-SIMを指定して利用できる

よりGAPの少ない操作が可能に



```
A:admin@edge-tk12-a# show card 1 memory-pools
=====
Card 1 Memory Pools
=====
Name          Max Allowed  Current Size  Max So Far      In Use
IOM          No limit      2,159,706,112  2,161,803,264  2,144,510,160
<省略>

A:admin@edge-tk12-a# show card a memory-pools
=====
Card a Memory Pools
=====
Name          Max Allowed  Current Size  Max So Far      In Use
BFD          No limit      9,444,656    9,444,656    8,318,528
BGP          No limit      6,291,504    6,291,504    5,015,536

A:admin@edge-tk12-a# show port
=====
Ports on Slot 1
=====
Port          Admin Link Port  Cfg  Oper  LAG/ Port Port Port  C/QS/S/XFP/
Id           State  State   MTU  MTU   Bndl Mode Encp Type  MDIMDX
1/1/c1      Down    Down      -     -     -     -     -     conn  100GBASE-LR4*
<省略>
1/1/c12     Up      Link Up  8704 8704  - netw null  conn  100GBASE-LR4*
1/1/c12/1   Up      Yes      Up   8704 8704  - netw null  cgige
<省略>
1/1/c21     Up      Link Up  8704 8704  - netw null  conn  100GBASE-LR4*
1/1/c21/1   Up      Yes      Up   8704 8704  - netw null  cgige
```

[おまけ]物理とCNFの構成差分の解消も進んでいる

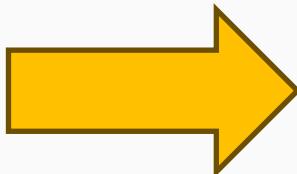
- Juniper社のcJunos Evolved : 物理機器と同様のコマンド体系を実現

```
13_preallocated_resources:
```

```
- type: node
  name: as65520-edge01
  asn: 65520
  interfaces:
    - Ethernet3
- type: node
  name: edge-tk12
  interfaces:
    - et-0/0/1
    - et-0/0/2
emulated_params:
  kind: juniper_cjunosevolved
  image: cjunosevolved:25.2R1.8-EVO
  env:
    CPTX_COSIM: "BT"
```

増設リソースに
cJunos Evolvedを
利用できる

よりGAPの少ない
操作が可能に



```
root@HOSTNAME> show chassis hardware
```

Hardware inventory:

Item	Version	Part number	Serial number	Description
Chassis			JN9989820AJD	JNP1001-36MR [PTX10001-36MR]
Routing Engine 0	BUILTIN	BUILTIN		RE-JNP1001-36MR
CB 0	BUILTIN	BUILTIN		Control Board
FPC 0	BUILTIN	BUILTIN		FPC-JNP1001-36MR
PIC 0	BUILTIN	BUILTIN		8X400GE-MR + 4X100GE-MR
Xcvr 0	REV 01	740-058732	1DJQA042004	QSFP-100GBASE-LR4
Xcvr 1	REV 01	740-058732	1DJQA042004	QSFP-100GBASE-LR4
Xcvr 2	REV 01	740-058732	1DJQA042004	QSFP-100GBASE-LR4
Xcvr 3	REV 01	740-058732	1DJQA042004	QSFP-100GBASE-LR4

```
root@HOSTNAME> show interfaces terse
```

Interface	Admin	Link	Proto	Local	Remote
et-0/0/0	up	up			
et-0/0/0.16386	up	up		multiservice	
pfh-0/0/0	up	up			
pfh-0/0/0.16383	up	up		inet	
et-0/0/1	up	up			
et-0/0/1.0	up	up		inet 192.168.200.1/30	
				multiservice	
et-0/0/2	up	up			
et-0/0/2.0	up	up		inet 192.168.1.12/24	
				multiservice	

```
root@HOSTNAME> show chassis routing-engine
```

Routing Engine status:

Slot 0:	
Current state	Master
Election priority	Master (default)
Temperature	0 degrees C / 32 degrees F
DRAM	1353 MB (6292 MB installed)
Memory utilization	78 percent

5 sec CPU utilization:

User	2 percent
Background	0 percent
Kernel	2 percent
Interrupt	0 percent
Idle	96 percent

1 min CPU utilization:

User	2 percent
Background	0 percent
Kernel	3 percent
Interrupt	0 percent