

SDN/OpenFlowにおける可視化技術

林 經正



自己紹介

- 日本電信電話株式会社にて、
 - 通信向けLSI
 - ネットワーク装置
 - 仮想ネットワーク
 - IDS/IPS
 - 米国にてベンチャー企業立ち上げ、新規事業立ち上げに従事
- ベンチャー企業にて、
 - 「ノンシフト24時間365日運用」を2年間経験
 - 仮想化システムやクラウドシステムの監視システムの開発
- 2012年に起業後、
 - 大手通信事業会社やベンチャー企業にてSDN/NFV関連のシステム開発、運用等に従事

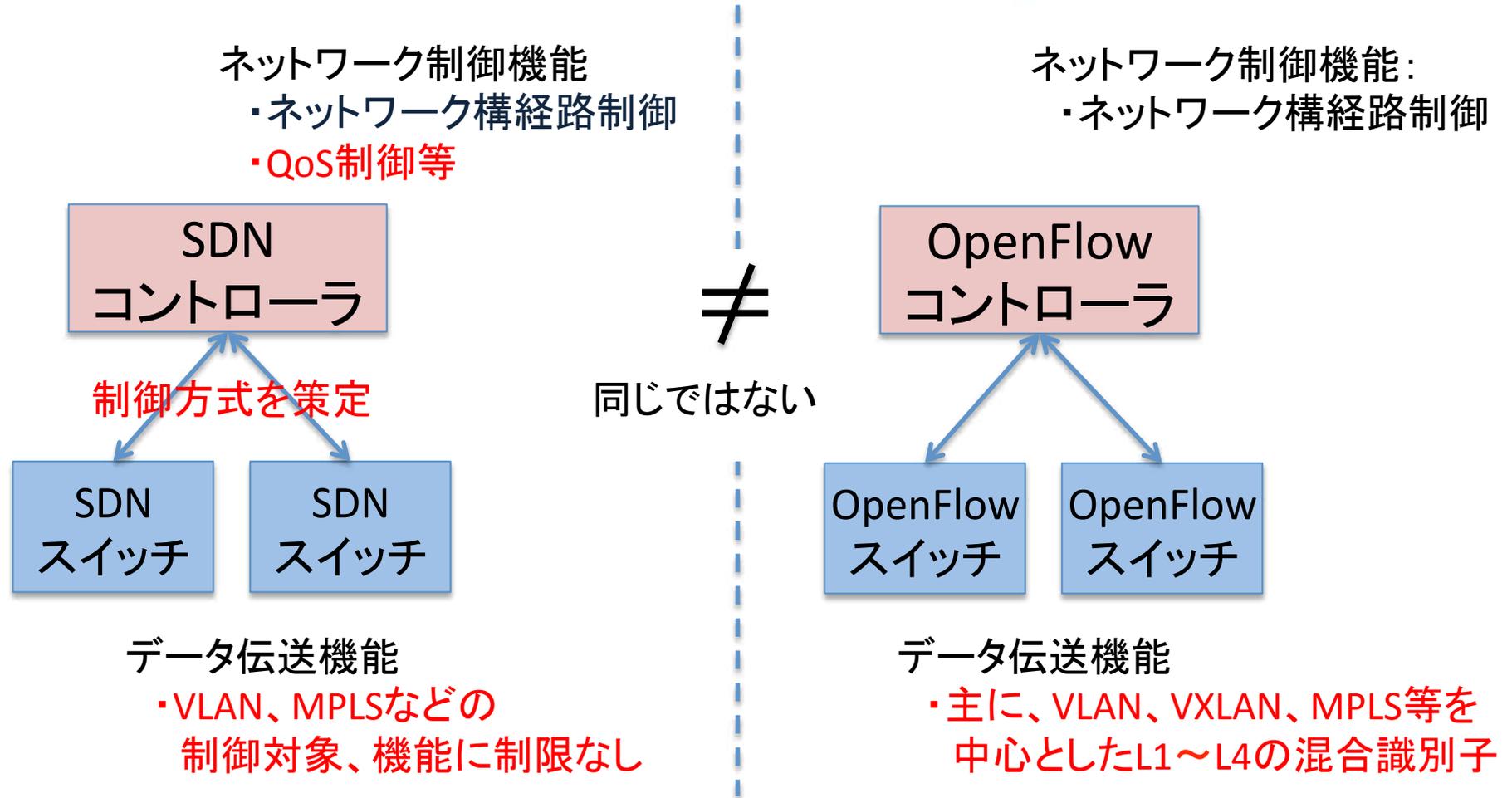
アジェンダ

1. SDNとOpenFlow
2. SDN/OpenFlowが作り出すネットワーク仮想化
3. SDN/OpenFlowにおける可視化技術の現状【結論】
4. SDN/OpenFlowの利用の市場動向
5. 実利用の世界から見る種々の課題
6. 求められる可視化技術
7. 理想と、現実の可視化技術
8. まとめ

1. SDNとOpenFlow

1. SDNとOpenFlow

『SDN』: ネットワーク制御と、データ転送を分離

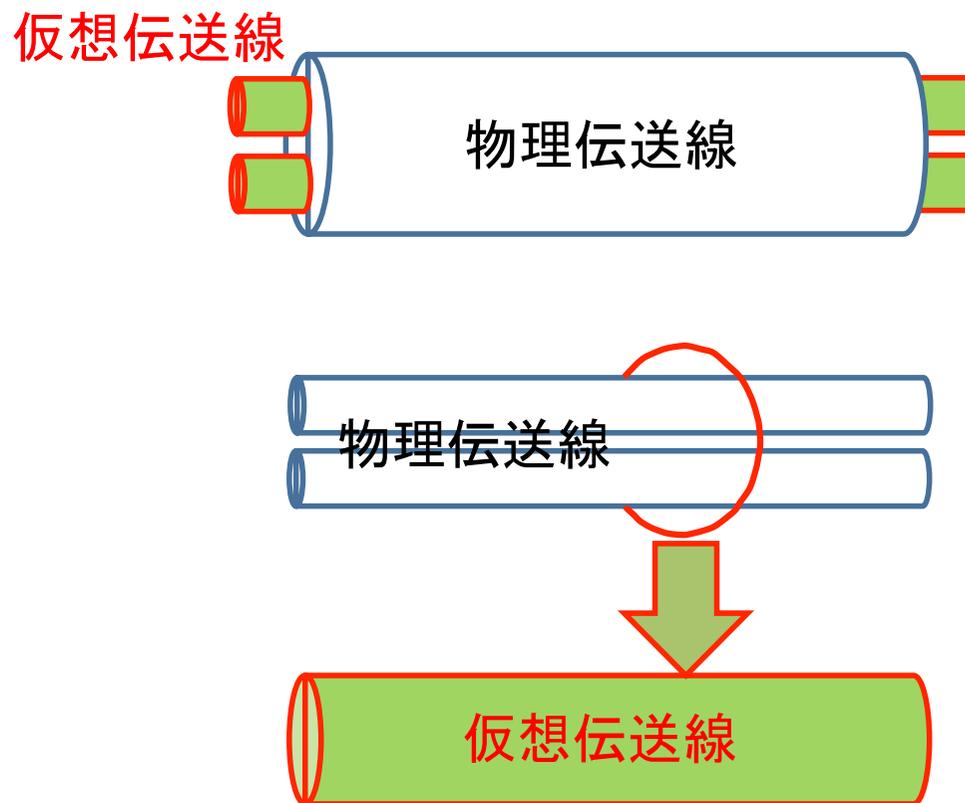
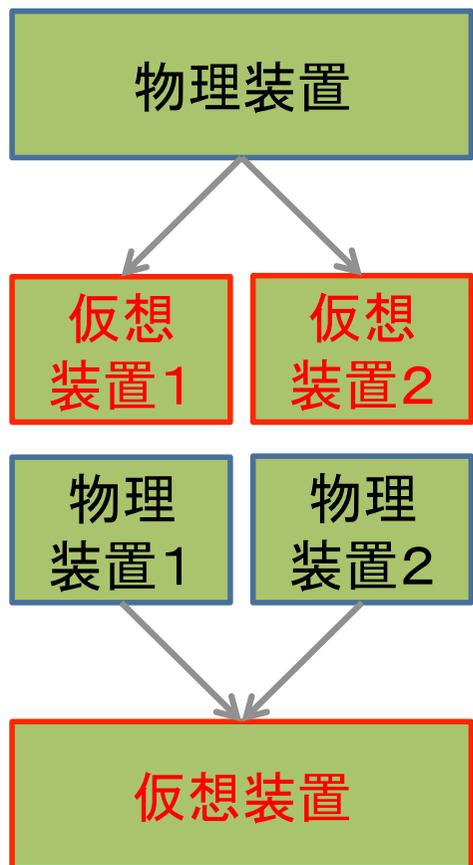


2. SDN/OpenFlowが作り出すネットワーク仮想化

2. SDN/OpenFlowが作り出すネットワーク仮想化

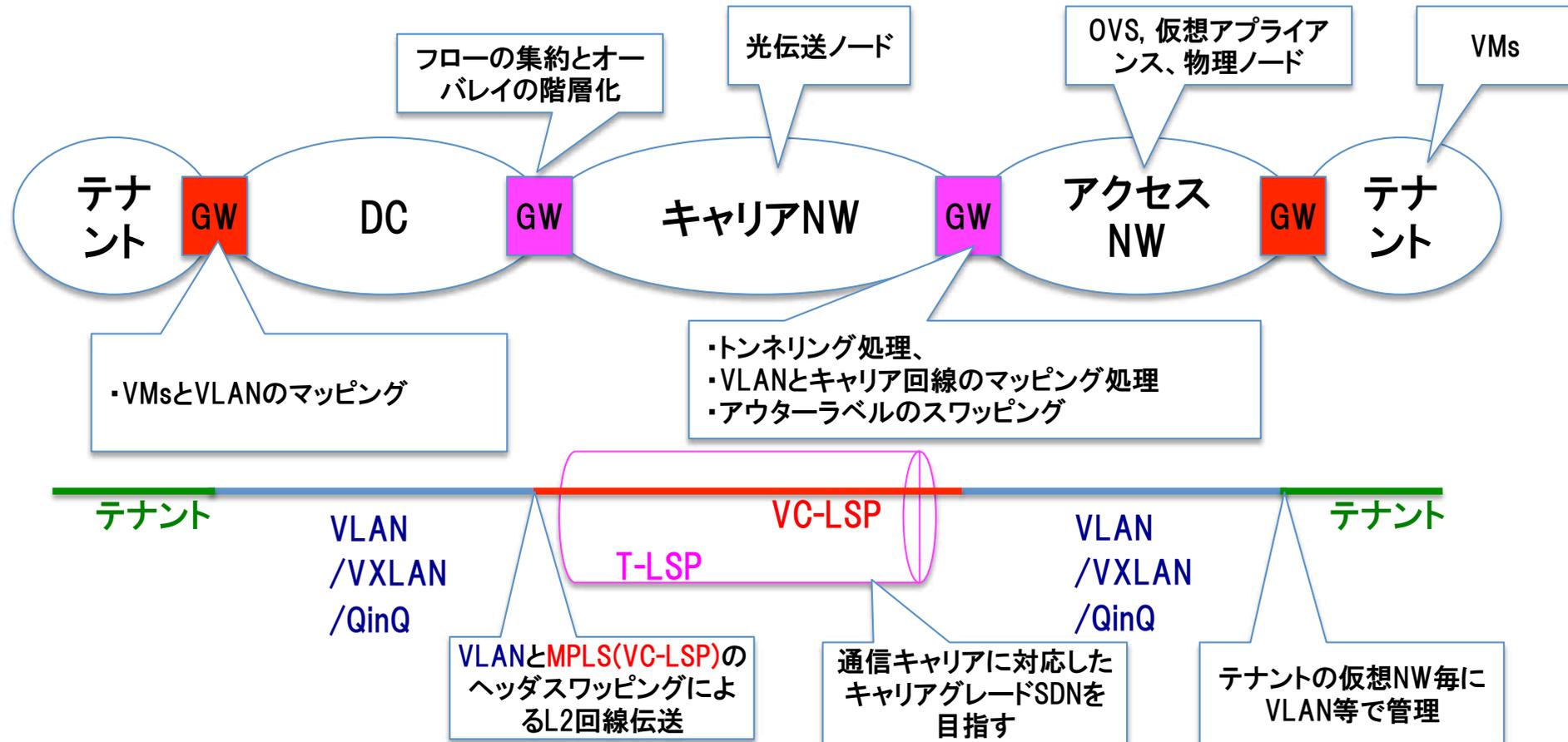
ネットワーク装置の仮想化

ネットワーク伝送路の仮想化



SDN/OpenFlowの特徴: データ伝送に特化

SDNを利用したネットワーク構成例



SDN利用を後押しするもう一つの背景

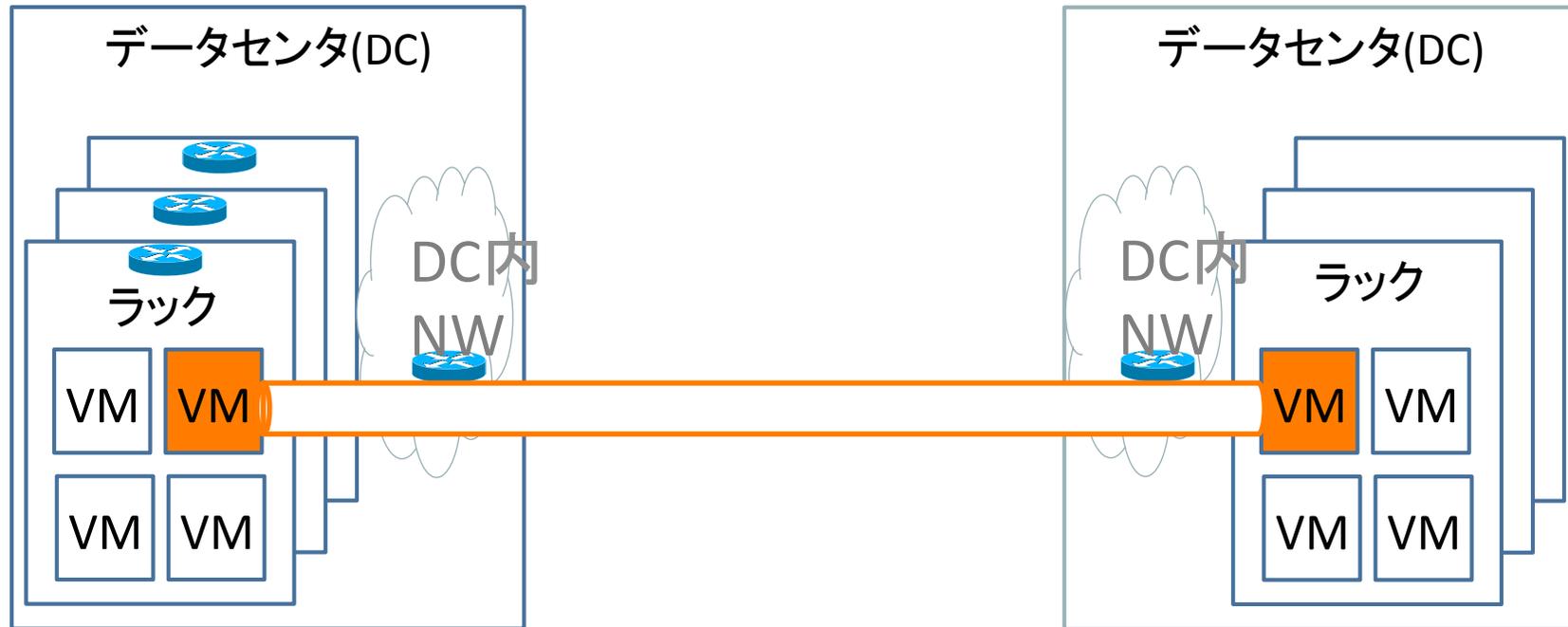
- コンピューテーションノード(CP)やストレージは仮想化、物理資源の共有化が進んできた
- これらは、ソフトウェアで一元管理できる
- CPやストレージを繋ぐ「ネットワーク」の管理・運用は、以前と手作業で行われている
 - 自動化の強い要望
 - ビジネスが要求する早いスピードに訴求

SDN/OpenFlow利用に期待すること

- 「**仮想化**」は、物理リソースの有効活用に有効
→CAPEXの低コスト化
- 「**自動化**」は、人手を介さずサービスを利用するユーザーがその場で開始、変更、廃止が可能
→OPEXの低コスト化

SDN技術の利用例1

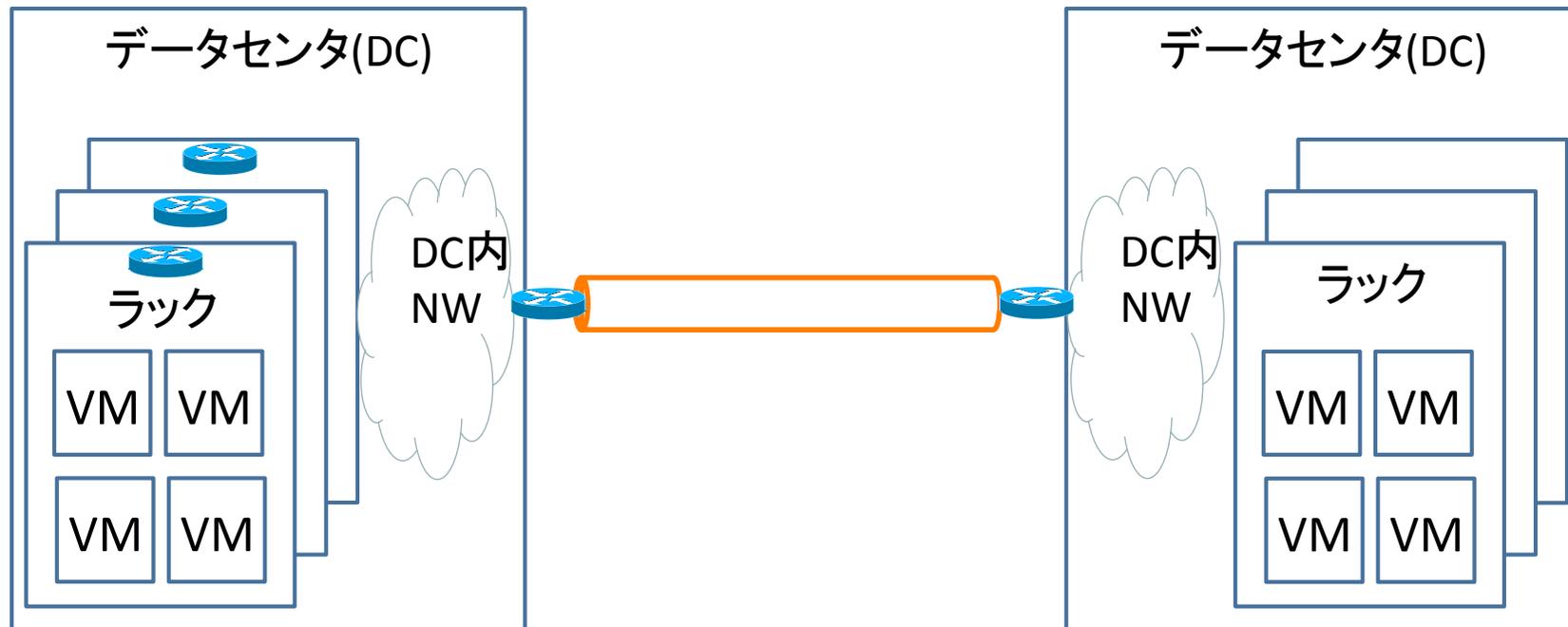
- ソフトウェアソリューションの利用：
 - ハイパーバイザー*上のSDNスイッチ間でオーバーレイ通信
 - TORのSDNスイッチ間でのオーバーレイ通信



*ハイパーバイザー: OpenStackや、CloudStackや、Vmware ESXiなど

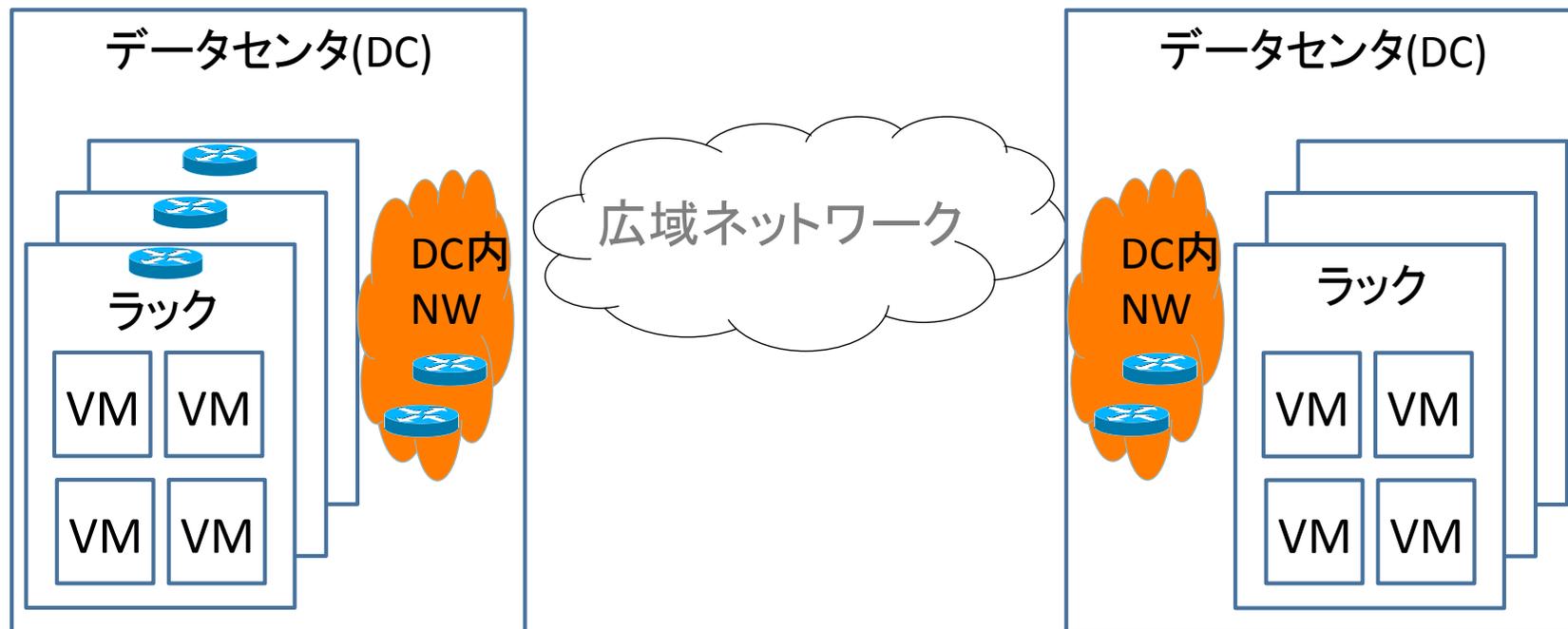
SDN技術の利用例2

- ソフトウェア／ハードウェアソリューションの利用：
DC間のオーバーレイ通信
 - ⇒ 広帯域トラフィック伝送では、ハードウェア
 - ⇒ 3Gbps程度の伝送では、ソフトウェアも利用



SDN技術の利用例3

- ハードウェアソリューションの利用:
ラック間、DC内VM間の大容量トラフィック通信

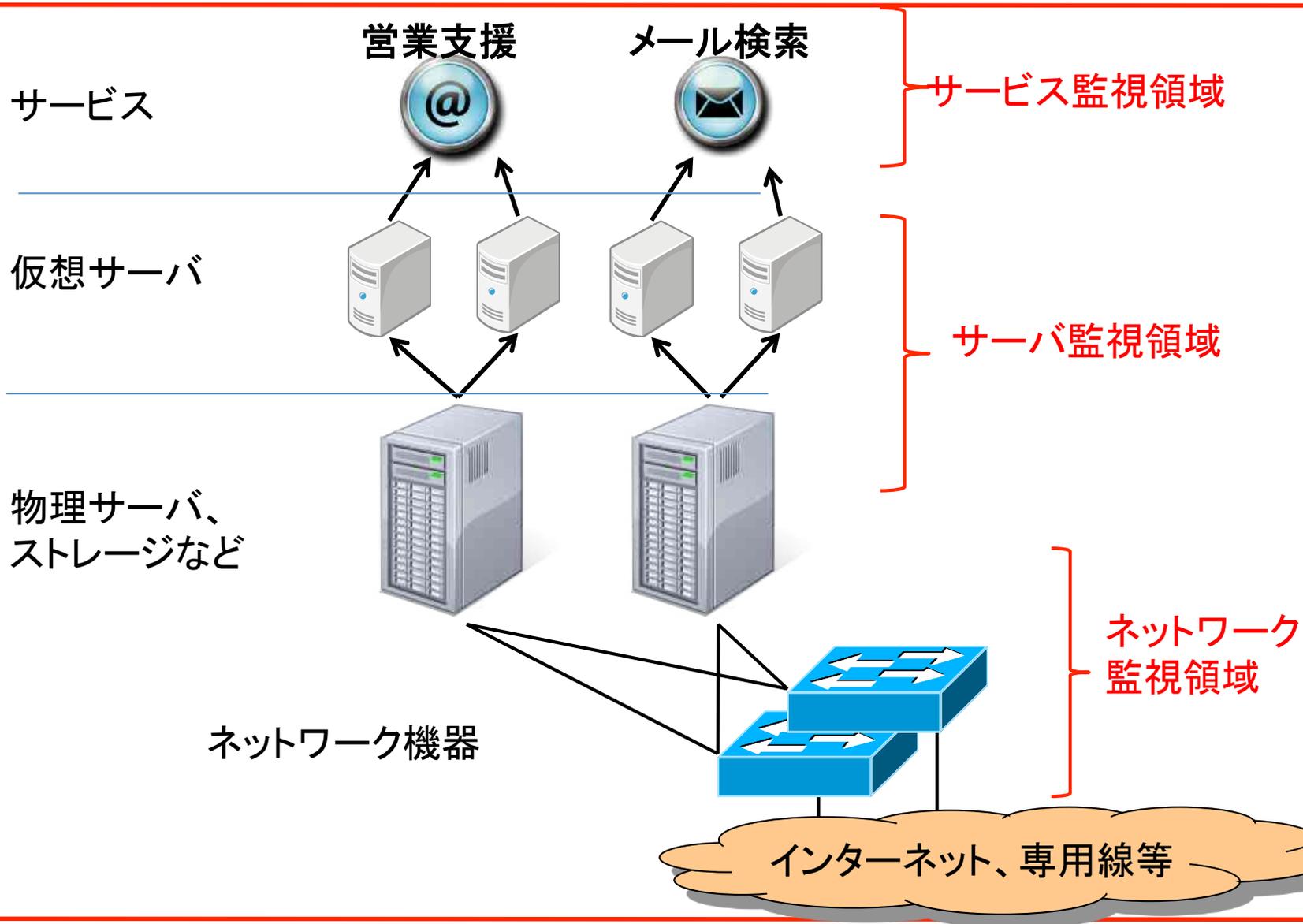


3. SDN/OpenFlowにおける可視化技術の現状

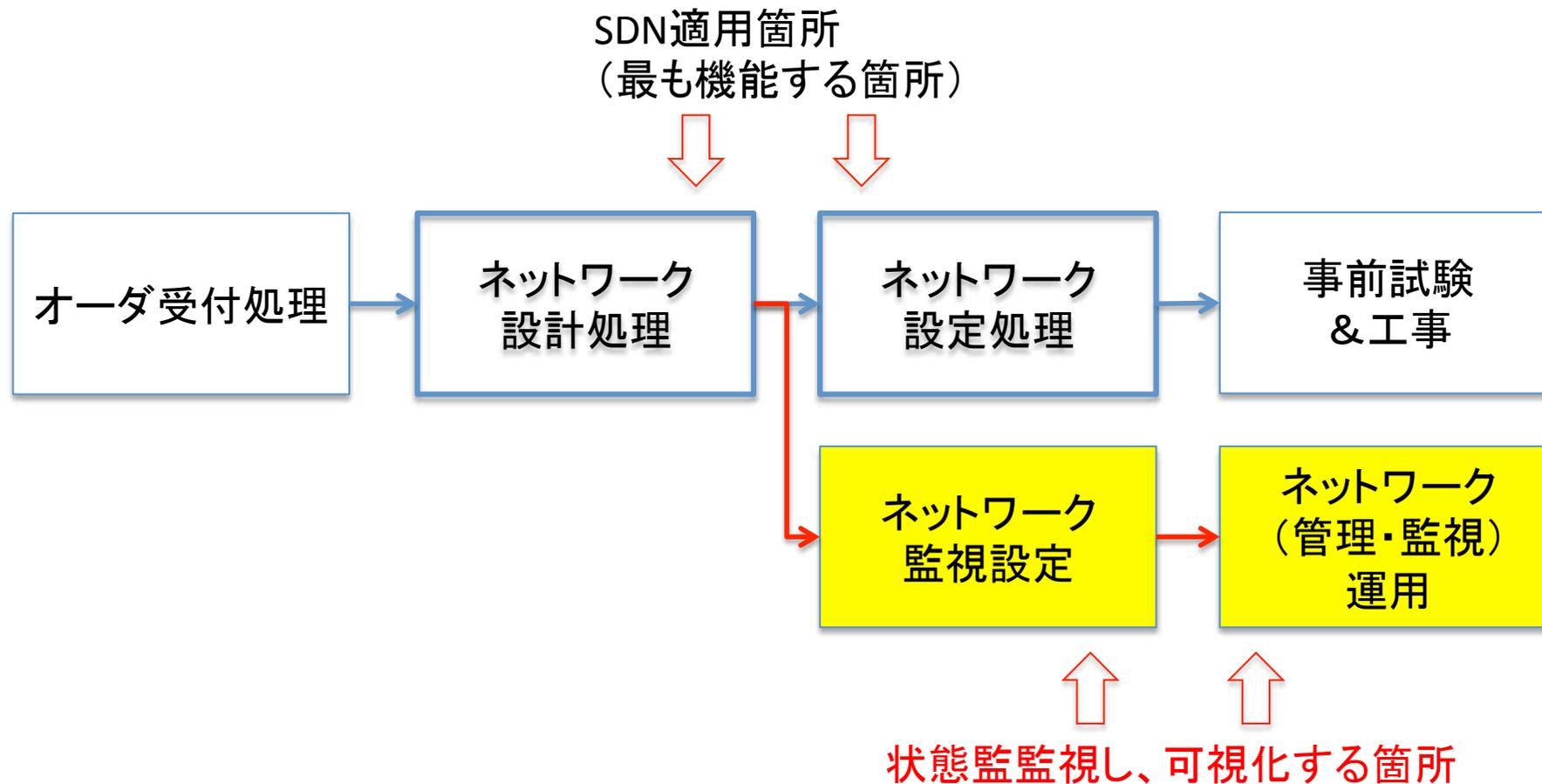
3. SDN/OpenFlowにおける可視化技術の現状

- 定性的な情報表示と、定量的な情報表示を行う程度
 - 【情報取得】
 - SNMPトラップ、GET
 - 管理対象へのtelnet後、CLIコマンド実行による情報取得
 - OpenFlow、OVSDBを利用した状態取得
 - 【可視化技術】
 - 定性情報の表示
 - 定量情報の履歴表示
- 物理と論理のネットワークトポロジー管理をGUI表示
 - 【可視化技術】
 - ネットワークトポロジーの表示
 - 関連する管理対象間のコリレーション情報の表示

管理・監視システムにおける可視化領域



ネットワーク運用プロセスとSDNの適用先



管理・監視対象

監視項目	備考
死活監視	ping等を利用して監視対象を監視
サービス監視	Webであればhttp応答など
性能監視	定量情報監視、統計監視含む
セキュリティ監視	不正ログイン、ファイル改竄監視など
ハードウェア異常監視	正常動作監視など
操作ログ監視	操作ログ、最大利用者数ログ監視など

監視が3つに分かれる理由

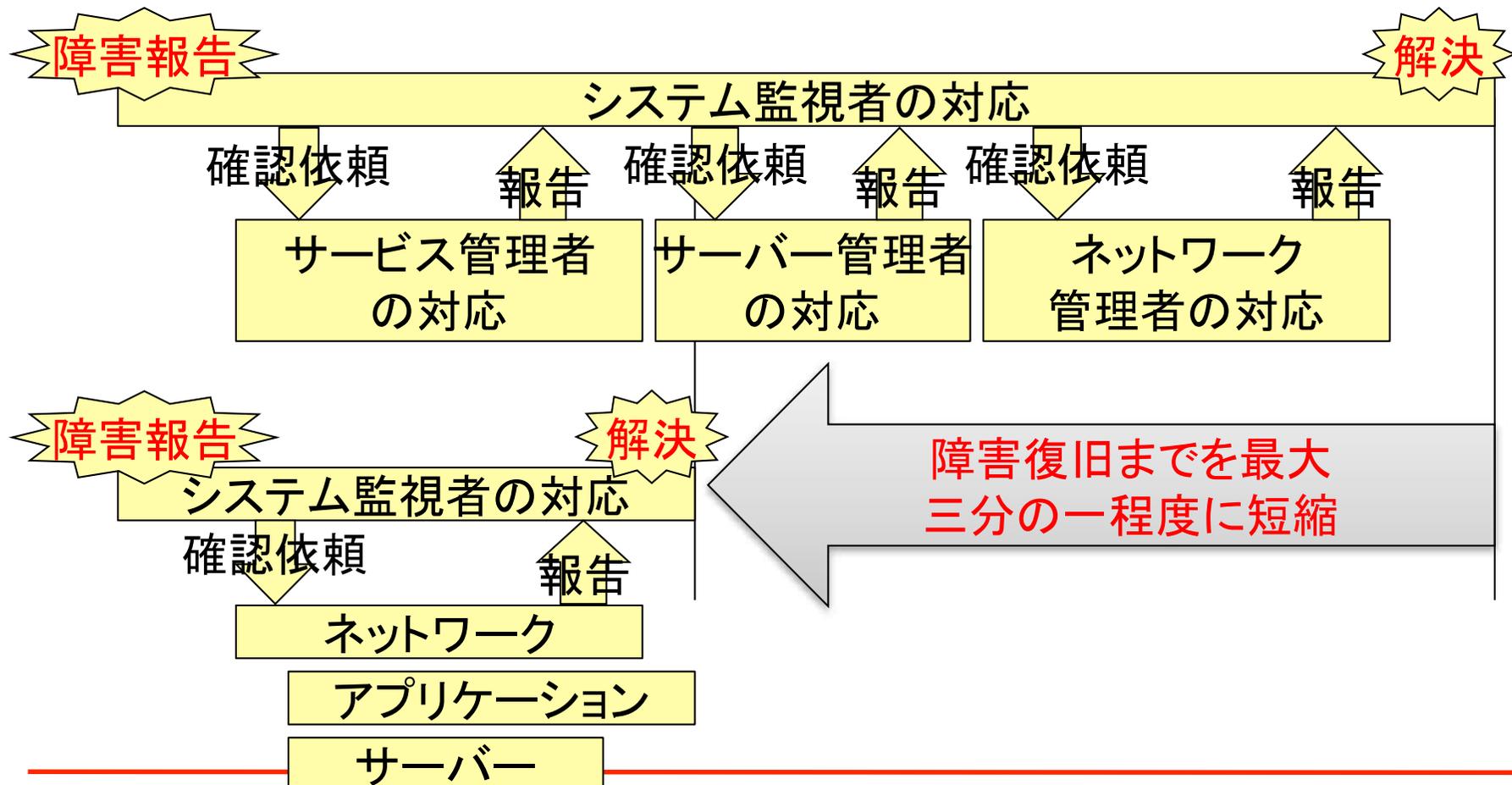
- ・ サービス、サーバ、ネットワークで**利用する技術が異なる**ため
- ・ サービスとサーバで利用する技術は比較的近い

統合監視

 <p>サービス</p>	<ul style="list-style-type: none">・ プロセス起動・制御技術・ データ処理・管理技術・ サーバ上でのオペレーション技術・ ログ監視技術
 <p>サーバ</p>	<ul style="list-style-type: none">・ サーバ起動・制御技術・ HDD・メモリ管理技術・ サーバ上でのオペレーション技術・ ログ監視技術
 <p>ネットワーク</p>	<ul style="list-style-type: none">・ ネットワーク設計技術・ ネットワークプロトコル技術・ OpS・専用機器でのオペレーション技術・ SNMP/TL1監視技術

統合監視が成功した時の例

- システム把握や、インシデント発生原因究明を短時間に可能
- 統合化されたシステムでは、各監視の個別対応では難しい



一般的な定量情報の可視化

- Zabbixを利用した定量情報の履歴表示



一般的な定性情報の可視化

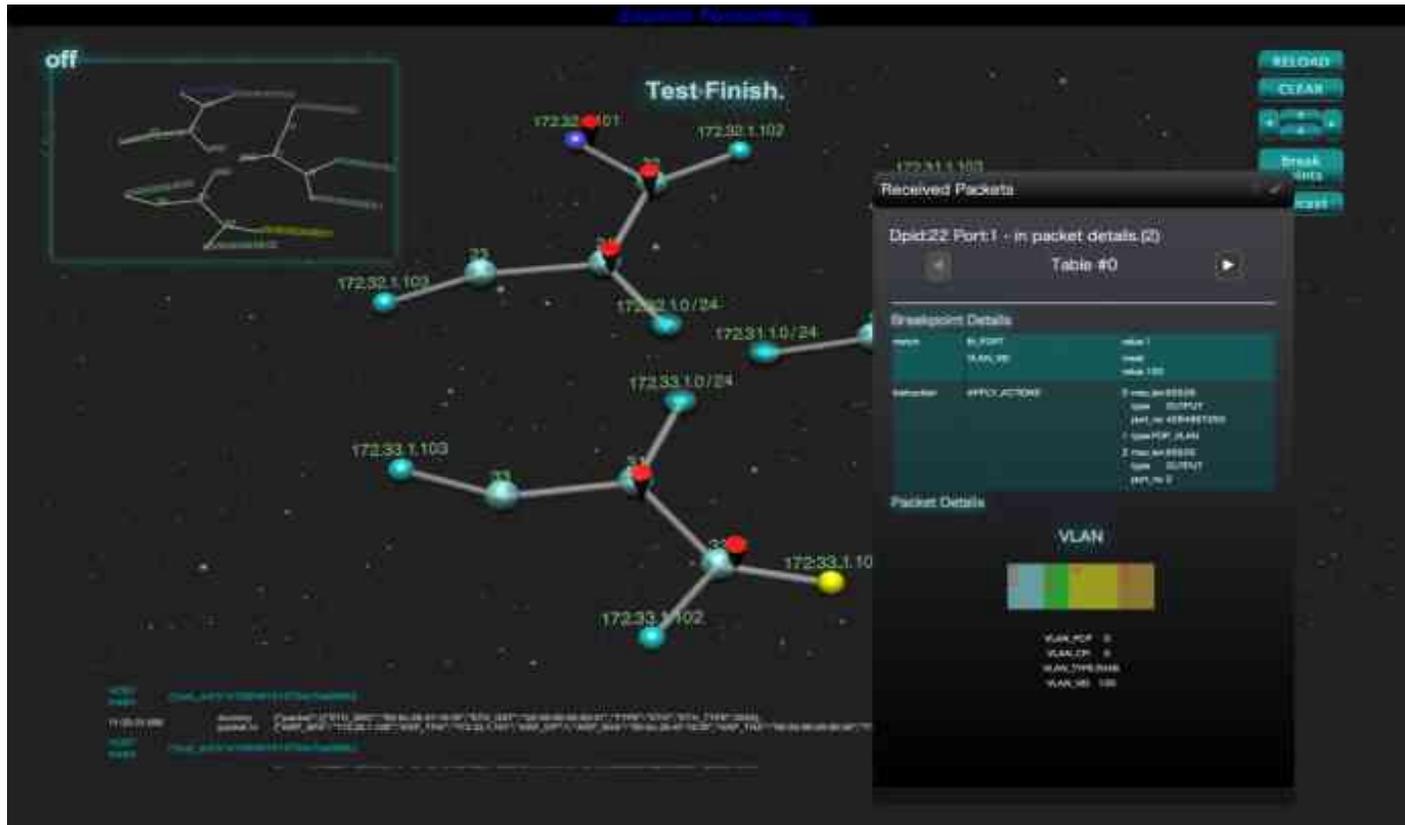
- Zabbixを利用した定性情報の履歴表示

The screenshot displays the Zabbix web interface with the following sections:

- Zabbixサーバの状態**: A table showing server parameters such as Zabbix service status (はい), number of hosts (46), items (7), triggers (0), and users (2).
- システムステータス**: A table showing the status of host groups, including Discovered hosts and OmniSphere, with columns for fatal, severe, and minor problems, warnings, and information.
- ホストステータス**: A table showing the status of individual hosts, including Discovered hosts and OmniSphere, with columns for no problems, problems, and total count.
- 最新20件の障害**: A table showing the most recent 20 problems, with columns for host, problem type, latest change, elapsed time, comments, and actions.

仮想ネットワークポロジの可視化

- グラフィックに凝ったネットワークポロジ管理の例



出典元: APNOMS2014 (NTTコミュニケーションズ)

論理/物理ネットワークトポロジーの可視化

- 設計した仮想ネットワークリソースと、事前に用意した物理ネットワークの構成図の可視化を一括管理

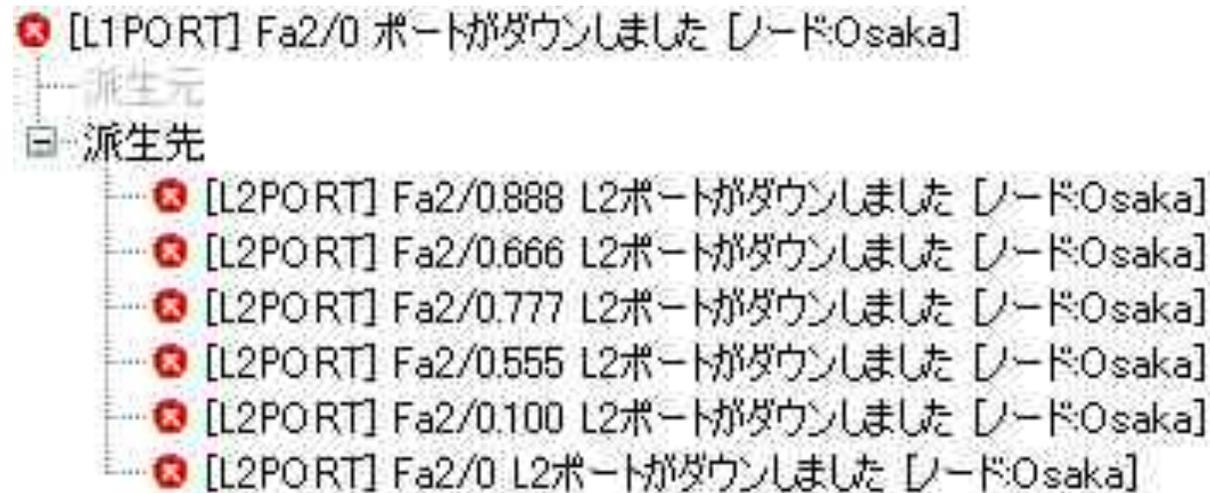
設計した情報を選択し、トポロジー上で確認

設	種類	状態	RSVP-LSP SET名	RSVP-LSP(ロ)	始点ノード	終点ノード	経路	T
<input type="checkbox"/>	RSVP-L	稼働中	LSP_13_05	LSP_13_05_01	R1[10.10.10.11]	R3[10.10.10.13]	10.0.13.3	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_13_06	LSP_13_06_02	R1[10.10.10.11]	R3[10.10.10.13]	10.0.11.2 10.0.12.3	
<input checked="" type="checkbox"/>	RSVP-L	稼働中	LSP_13_06	LSP_13_06_01	R1[10.10.10.11]	R3[10.10.10.13]	10.0.13.3	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_13_06	LSP_13_06_02	R1[10.10.10.11]	R3[10.10.10.13]	10.0.11.2 10.0.12.3	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_13_06	LSP_13_06_03	R1[10.10.10.11]	R3[10.10.10.13]	10.0.13.3	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_21_01	LSP_21_01_01	R2[10.10.10.12]	R1[10.10.10.11]	10.0.12.3 10.0.13.1	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_23_20	LSP_23_20_01	R2[10.10.10.12]	R3[10.10.10.13]	10.0.12.3	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_23_20	LSP_23_20_02	R2[10.10.10.12]	R3[10.10.10.13]	10.0.12.3	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_31_01	LSP_31_01_01	R3[10.10.10.13]	R1[10.10.10.11]	10.0.12.2 10.0.11.1	
<input type="checkbox"/>	RSVP-L	稼働中	LSP_32_01	LSP_32_01_01	R3[10.10.10.13]	R2[10.10.10.12]	10.0.13.1 10.0.11.2	
<input type="checkbox"/>	RSVP-L	新規設計	TEST-TORITA06-R	TEST-TORITA06-FR1	[10.10.10.11]	R3[10.10.10.13]	10.0.13.3	
<input type="checkbox"/>	RSVP-L	新規設計	TEST-TORITA08-R	TEST-TORITA08-FR1	[10.10.10.11]	R3[10.10.10.13]	10.0.13.3	
<input type="checkbox"/>	RSVP-L	削除予定	TT-LSR05	TT-PATH05	R3[10.10.10.13]	R1[10.10.10.11]	10.0.13.1	
<input type="checkbox"/>	RSVP-L	削除予定	Test_NICT_01	Test_NICT_01	R1[10.10.10.11]	R2[10.10.10.12]	10.0.11.2	
<input type="checkbox"/>	RSVP-L	削除済	km-set1	km-lsp	R1[10.10.10.11]	R3[10.10.10.13]	10.0.13.3	
<input type="checkbox"/>	RSVP-L	削除予定	km-set2	km-set	R3[10.10.10.13]	R1[10.10.10.11]	10.0.12.2 10.0.11.1	

出典元: ICM研究会2012年11月 (Cloud Scope)

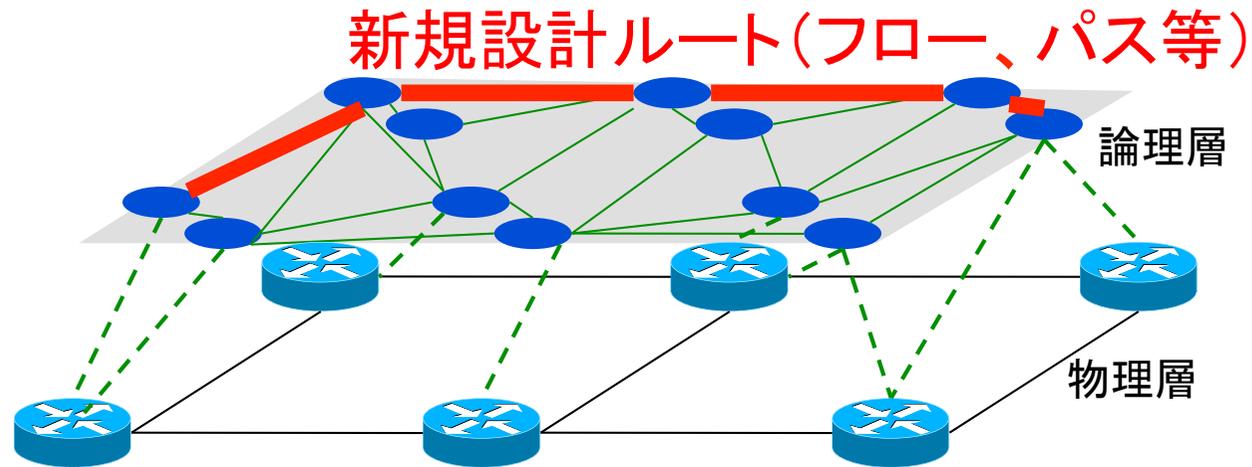
論理/物理関連情報を可視化

- 故障発生時のトラップ情報より、論物関連情報を一括可視化
→ コリレーション情報管理技術を利用



出典元: ICM研究会2012年11月 (Cloud Scope)

論理/物理関連情報を可視化のイメージ



4. SDN/OpenFlowの利用の市場動向

4. SDN/OpenFlowの利用の市場動向

- 2014年時点では主な導入事例はないが、2014年度中には大型法人案件でのSDN適用が促進
 - SDNを利用したソリューションビジネスが大きく展開
- 一部のサービスプロバイダにおいて、CloudStack、OpenStackの導入が試験的に始まった程度
 - 実際は、Opexが上がりコスト高

2014年度以降の市場動向

- エンタープライズ向けSDNサービスにおいても、大手通信事業会社が実ビジネス化
 - SDN/OpenFlowパッケージ化が実現していく
- 今後は、実NWサービスにおいて求められる仮想化機能をNFV (Network Function Virtualization)として提供していく
 - NFV機能として提供すると嬉しいものをREST IFパッケージとして提供

5. 実利用の世界から見る種々の課題

5. 実利用の世界から見る種々の課題

- 共通課題
 - 既存NWビジネス、サービス・プロバイダビジネスの収益増を想定したサービス開発を検討する人材不足
 - 「この技術が課題」という技術思考の課題意識優先度は低い
- 通信キャリアの課題
 - 国外収益を増やすための事業開発
 - 既存システムと、NFVシステムシステム統合に携われる人材不足
 - NFVソリューションビジネスを展開する人材不足
 - デイメンジョンデータ(NTTグループ)では、1,000人を増強する
- DC事業者
 - Opex削減と、収益増を期待できるサービス開発

技術的課題1/4

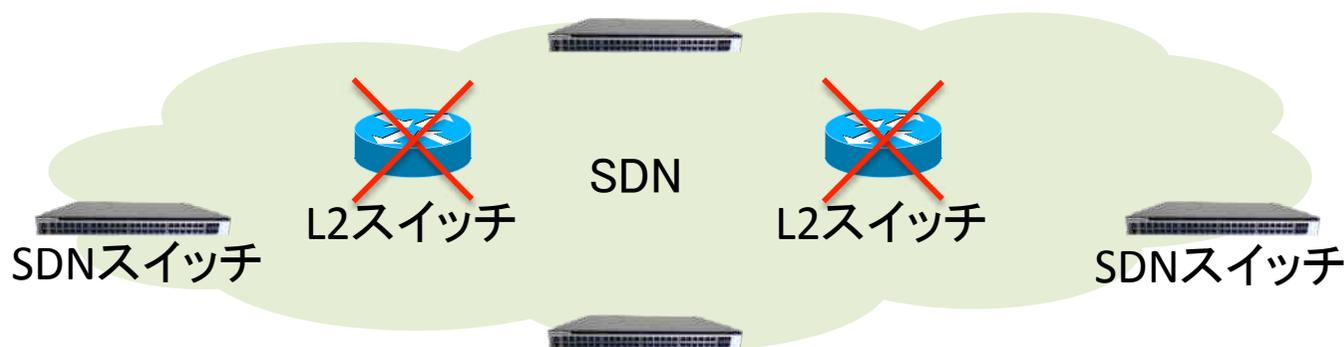
- 既存のルータ・スイッチが提供している機能を併用できない
- OpenFlow用のMIBが定義されていない、実装されていない
- 登録できるフロー数が少なすぎる
 - Broadcom系LSIを利用したスイッチでは、最大2,000エントリという例あり
→VLAN数よりも少ないフロー数
- 意に反して高いコスト
 - OFS(SDNスイッチ):100～400万円を超えるものが多い
 - OFC(SDNコントローラ):一声、1,000万円
 - OPEX:SDN運用に増員
- まだまだ不安なソフトウェア処理
 - ハードウェアのネットワーク装置で実現してきた機能や、性能を置き換えられない
 - 特にユーザデータ転送処理(Dプレーン処理)

技術的課題2/4

- 課題となるフロー更新処理
 - 2,000フロー/秒の更新処理を実現しないOFSは実運用では厳しい
 - グループ機能を利用して、複数のフローを一括変更するなどの手段も組み合わせる必要あり
- BGPなどのネットワーク管理機能を自前で実現する必要あり
 - SDNコントローラにそのような機能はない
 - 開発コスト、運用コストは上る可能性が大きい
 - 大手ベンダのホールインワン・ソリューションを利用すると、オペレーションシステム含めて全てをロックインされる
- OpenFlow仕様の全てを、ハードウェアで実現するのは難しい
 - 3段、4段、それ以上のマルチテーブルは必要か？
 - フロー単位のMETER機能は必要か？
- SDNコントローラの集中制御による性能ボトルネック
 - 分散処理をどのように実現するか？

技術的課題3/4

- 既存のスイッチを併用で出来ない
 - Etherヘッダ、IPヘッダを、SDN内で特殊な使い方をするため
 - 5～30万円の廉価L2スイッチは使えないため、高価なSDNスイッチを買い続けることに。。。。



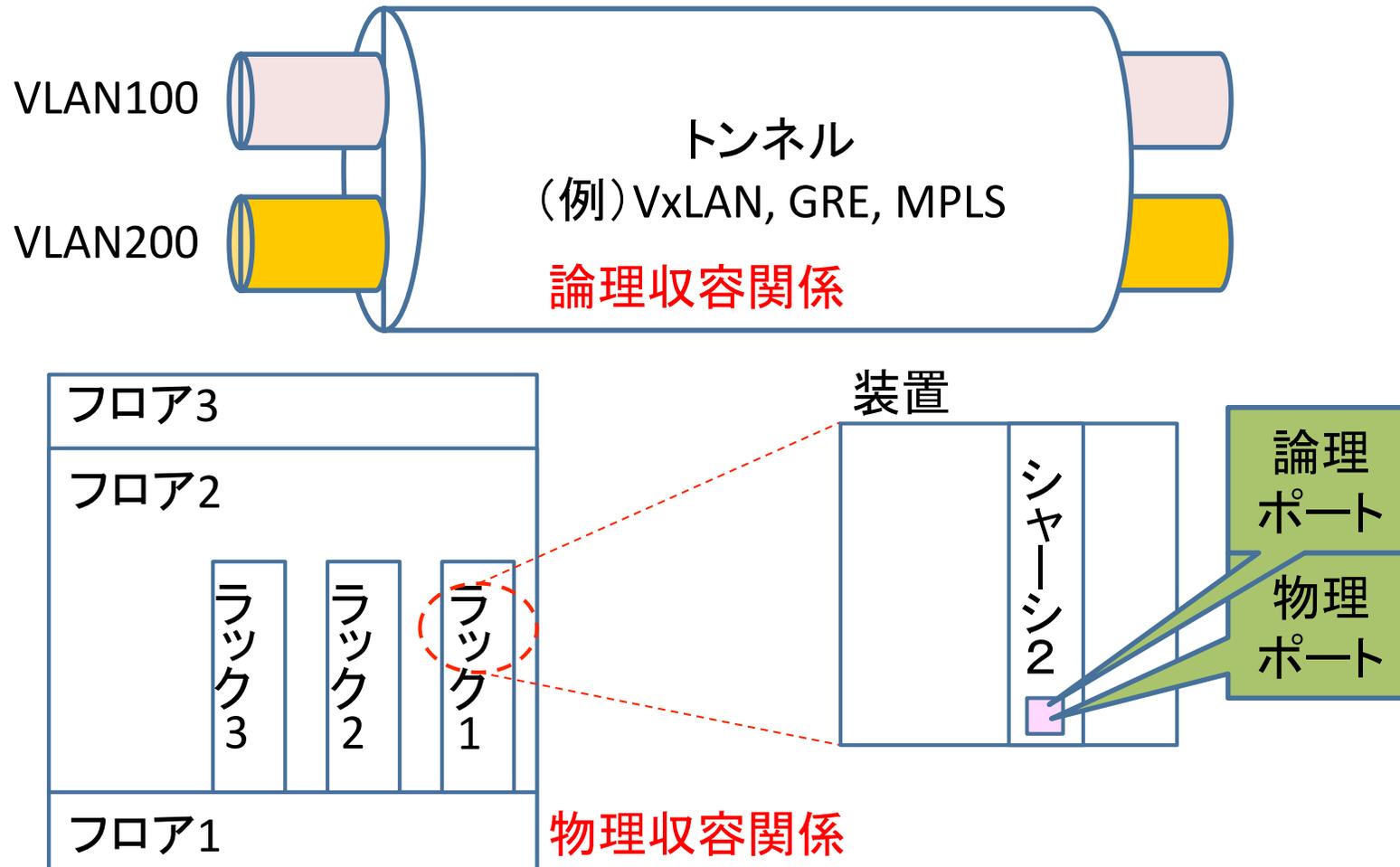
技術的課題4/4

- SDN含むネットワーク管理の自動化、サービス提供の自動化が実現後の課題は、**ストレージの管理**
- 数百GB～数TBのストレージは、簡単に移動できない
- DBミラーリングは長距離(200Km以上)では難しい

6. 求められる可視化技術

6. 求められる可視化技術

- 物理収容、論理収容情報を管理、監視できる事



リソース状態の見える化例

- 状態、障害の管理・監視に分けて整理する
- 定性、定量の管理・監視に分けて整理する

		管理・監視対象						備考	
		コンピュータ ノード	VM	VNI	Software Switch	Tunnel			
						VxLAN/OVS	STT/OVS		
管理・監視項目	状態	定性管理	CN名	VM名	IF名	SW名	ID管理	ID管理	
			up/down	up/down	up/down	up/down	TEP情報 (s,d-port 情報)	TEP情報 (s,d-port 情報)	
					IP address	attached VNI名	VLAN/Tunnel	VLAN/Tunnel	
					VLAN ID				
		定量管理	インスタンス数	CPU利用率	データ送信量	VNI数	対地Tunnel数	対地Tunnel数	
				メモリ利用率	データ受信量		データ送信量	データ送信量	
					Drop量(送信)		データ受信量	データ受信量	
					Drop量(受信)		Drop量(送信)	Drop量(送信)	
							Drop量(受信)	Drop量(受信)	
	障害	定性	down	down	down	down	down	監視では、通知機能が必須 (SNMP trap)	
							Tunnel/VNI/IF		Tunnel/VNI/IF
						Backup Tunnel	Backup Tunnel		
	定量								

可視化例1:フロー生成時のコマンド実行状態

- Flow, Group および Meter を追加するコマンド生成スクリプトを作成した。
 - flowGenerator.py
 - groupGenerator.py
 - meterGenerator.py

- オプション入力によるコマンドが短縮される
- 大量の Group などの一斉登録が容易になる

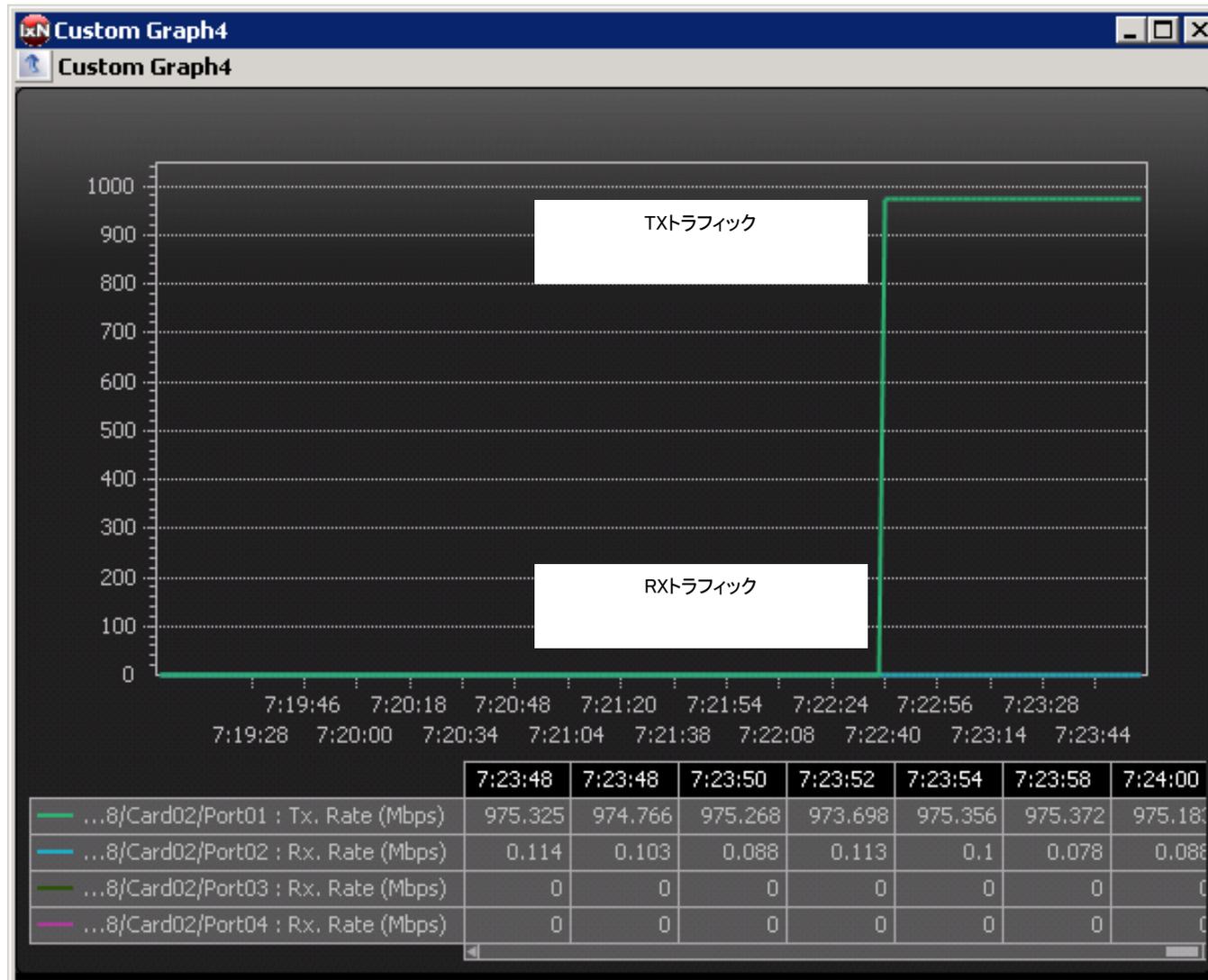
```
hiroki@archleno:~/program/bosco/branches
hiroki@archleno:~/program/bosco/branches 76x23
[hiroki@archleno branches]$ ./meterGenerator.py -h
Usage: meterGenerator.py [options]

Options:
  -h, --help            show this help message and exit
  -c COMMAND, --command=COMMAND
                        command for adding flow (add, mod, del)
  -x, --execute         execute generated command
  -i METER_ID, --meter_id=METER_ID
                        Option to set ID of meter

Band options:
  -R RATE, --rate=RATE
                        Set the limitation rate of meter.
  -B BURST, --burst=BURST
                        Configure the burst of traffic (default = 32000).
  -S, --statistics     Set flag to enable statistics.
[hiroki@archleno branches]$ ./meterGenerator.py -i 1 -R 500000 -S
curl -d '{"dpid":"128983237617","command":"add","meter_id":1,"flags":["kbps","burst","stats"],"bands":[{"type":"drop","rate":500000,"burst_size":32000}]' http://192.168.100.144:8080/of13/meter mod
[hiroki@archleno branches]$ ./meterGenerator.py -i 1 -R 500000 -S -x
[hiroki@archleno branches]$

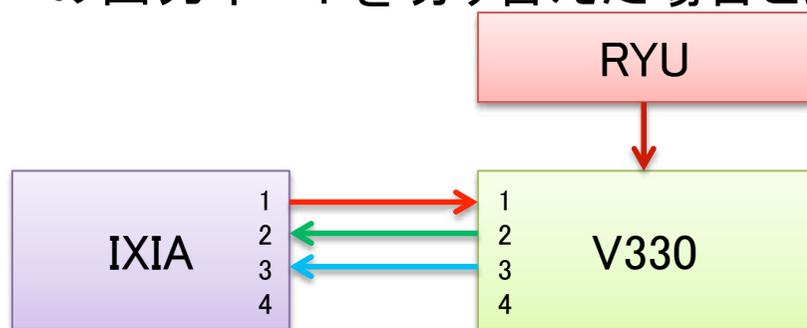
hiroki@archleno:~/program/bosco/branches 76x4
Switch# ovs-ofctl dump-meters br0 -O openflow13
OFPST_METER_CONFIG reply (OF1.3) (xid=0x2):
  meter_id=1,flags=KBPS-BURST-STATS,band_type=DROP,rate=500000,burst=32000
Switch#
```

可視化例2:帯域制御の状態管理



可視化例3: IndirectによるGroup切替え状態

- Indirect の機能により Action Group (AG) を切り替える試験を行った (Flow の出力ポートを切り替えた場合と比較)。



1. IXIA から VLAN 1 ~60 のパケットを送信
2. Group A を経由し, 2番ポートから出力
3. Indirect の機能より AG を RYU から切り替える。
4. Group B を経由し, 3番ポートから出力

Group & Flow	詳細
Flow (2560)	VLAN によって識別する. AG は Group I
Group I	初期の AG は Group A
Group A	2番ポートから出力
Group B	3番ポートから出力

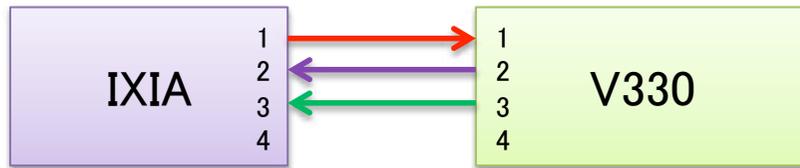
Flow の出力切り替え
では生じなかった
パケットロスが発生

Stat Name	Port Name	de	Line Speed	Link State	Frames Tx.	Valid Frames Rx.
1	192.168.100.96/Card01/Port01	Ethernet - 001	Full 1000 Mbps	Link Up	0	0
2	192.168.100.96/Card01/Port02	Ethernet - 002	Full 1000 Mbps	Link Up	5,527,054	0
3	192.168.100.96/Card01/Port03	Ethernet - 003	Full 1000 Mbps	Link Up	0	2,377,846
4	192.168.100.96/Card01/Port04	Ethernet - 004	Full 1000 Mbps	Link Up	0	3,149,194

計: 5,527,040

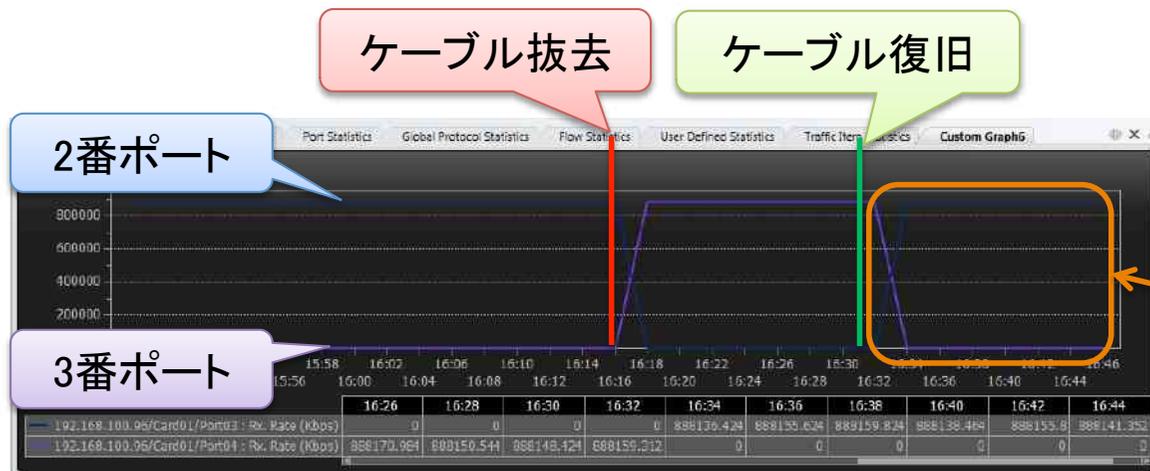
可視化例4: Fast Failoverにおけるリンク復帰状態

- Fast Failover の挙動を物理ケーブル抜去, 復旧することでリンクダウン時, 復旧時の挙動を確認する.



Flow	詳細
Flow	Failover として2番ポートをプライマリ, 3番ポートをサブとして登録する.

- IXIA からパケットを送信する.
- 2番ポートから出力されることを確認する.
- 2番ポートのケーブルを抜去する.
- 3番ポートから出力されることを確認する.
- 2番ポートのケーブルを復旧する.
- 2番または3番のどちらから出力されるかを確認する.



- 復旧による接続切り替えのメリットがない.
- 復旧切り替え時にパケットロスが発生している.

7. 理想と、現実の可視化技術

理想の可視化技術

- 各仮想リソース(アプライアンス、インスタンス)から、直接定量的、定性的情報を収集
- 可能であれば、MIBなどの管理情報として取得

現実の可視化技術

- 物理リソースに関連する情報は、
既存の監視・管理ツールにて可視化
- 仮想リソースに関連する情報は、
 1. 仮想管理ツールから限定的な情報を取得し、
可視化
 2. エージェントを仮想インスタンスにインストール
出来る場合には、エージェントから取得
→ログの文字列から定性、定量的情報を可視化

8. まとめ

まとめ

- SDN/OpenFlowに代表されるネットワーク仮想化の監視・管理対象と、可視化技術
- 仮想リソースに関連する情報は、
 1. 仮想管理管理ツールから限定的な情報を取得し、可視化
 2. エージェントを仮想インスタンスにインストール出来る場合には、エージェントから取得
→ログの文字列から定性、定量的情報を可視化
- 今後、SDN/NFVのサービスパッケージ化が進み、可視化技術の利用方法も変化

