

SDNの事業展開とWhitebox技術の最前線

林 経正



自己紹介

- 日本電信電話株式会社にて、
 - 通信向けLSI構成法の研究開発、
映像配信系ネットワークサービスの開発、
IDS/IPS中心にネットワークセキュリティの研究開発、
 - 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、
 - Webマーケティングサービスの開発・運用に従事し
 - NTTグループに買収される経験
- 別のベンチャー企業にて、
 - クラウドシステムの監視システムの開発、販売に従事
 - データセンター事業や通信事業会社のBPRコンサルティングに従事
- 2012年2月に起業
 - 自動化、仮想化、セキュリティ技術を軸に技術開発に従事

本日講演のアジェンダ

- ① SDNの実用化の状況
- ② SDN/NFVの実用化を通して、
今後必要となる技術と実務
- ③ SDN/NFVの実用化時の課題
- ④ 今後期待が高まるSD-WAN
- ⑤ Whiteboxの性能問題の評価
- ⑥ 最新Whiteboxの紹介

① SDNの実用化の状況

SDNの実用化の状況

- 2012～2014年
 - ユースケースベースの機能検証
 - フラグシップ的なSDN導入
 - SDNを利用したサービスオペレーションの仕組みと、課題の把握
- 2015～2016年
 - 全オペレーションの自動化
 - プリセールスと見積
 - 設計とプロビジョニング
 - 設定・運用と課金
 - ポストセールス
 - SDN/NFV実用が本格化
 - SDNソリューションのパッケージ化が促進
 - SD-WANの実用開始

SDNの実用化の状況

- 通信キャリア・ビジネス

- VM連動型オーバーレイSDN
- データセンター・ゲートウェイ
- NFVサービスを支えるSDN
- vEPC

- エンタープライズSDN

- 鉄道、大手娯楽サービス、電力などの基幹インフラ・システムへのSDN導入
- 専用線に代わるネットワークサービスとしての「SDNソリューション」のパッケージ化が促進
- 大学キャンパスへのSDN導入が促進

③ SDN/NFVの実用化時の課題

今後必要となる技術と実務を考える

- OpenFlow 導入の課題
- SDN サービス運用課題

今後必要となる技術と実務を考える

- OpenFlow 導入の課題
- SDN サービス運用課題

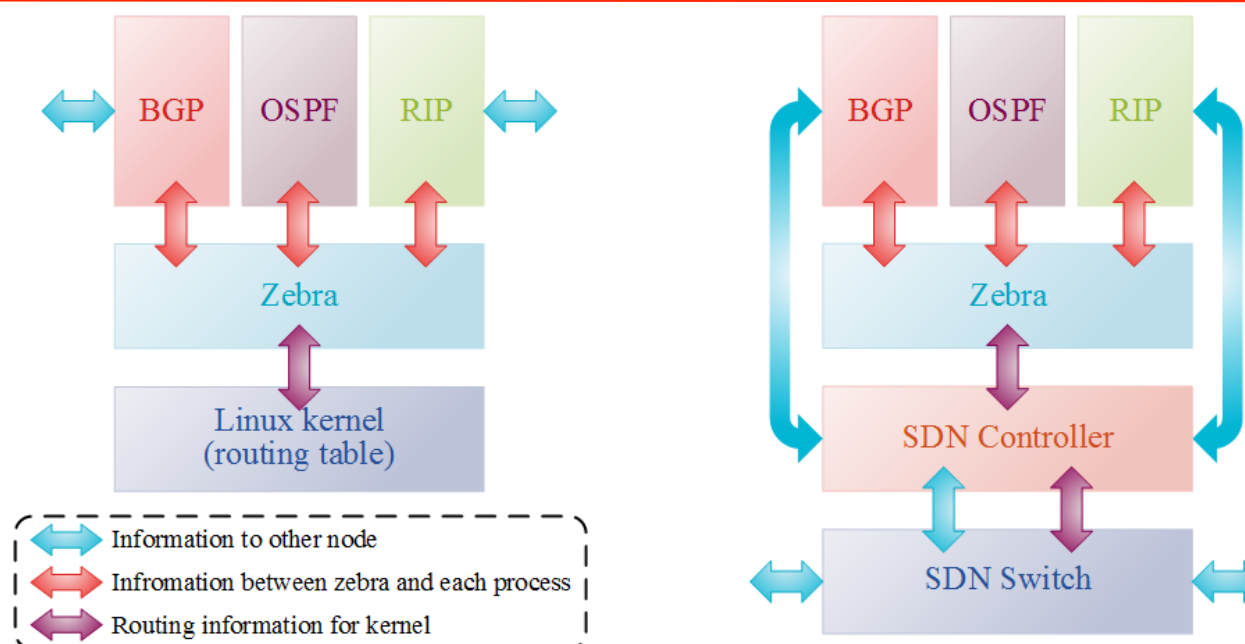
OpenFlow 導入の課題

- 既存ルーティングプロトコルと OpenFlow などの SDN コントローラは勝手に連携されない
- 既存ルーティングプロトコルをハンドリングしてくれる実装モデルの提案
 - OpenFlow スイッチの「AP-GW」化



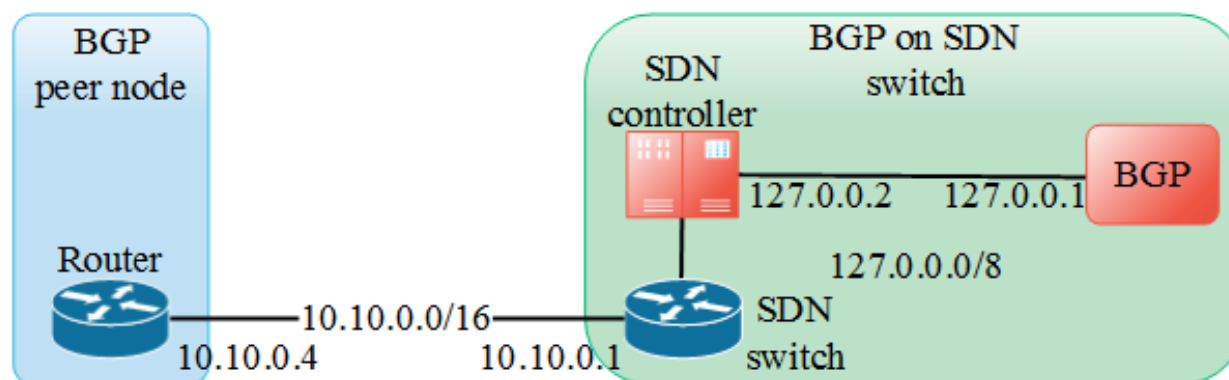
- 既存IP網との連携と、NWリソース管理機能の利用
- 提案モデルのプロトタイプを実装
 - 提案モデルの実現可能性を評価

IP既存網に導入するSDNスイッチの仕組み



- OpenFlow ポートは OpenFlow の機能以外でパケットを送出できない
 - BGP などのデーモンが外部と通信するには OF の機能が必要
 - packet in で処理、veth を用いる
- パケットの転送はフロー情報にもとづく
 - ルーティング情報とフロー情報を変換する機能が必要

SDN-NAPTの実装

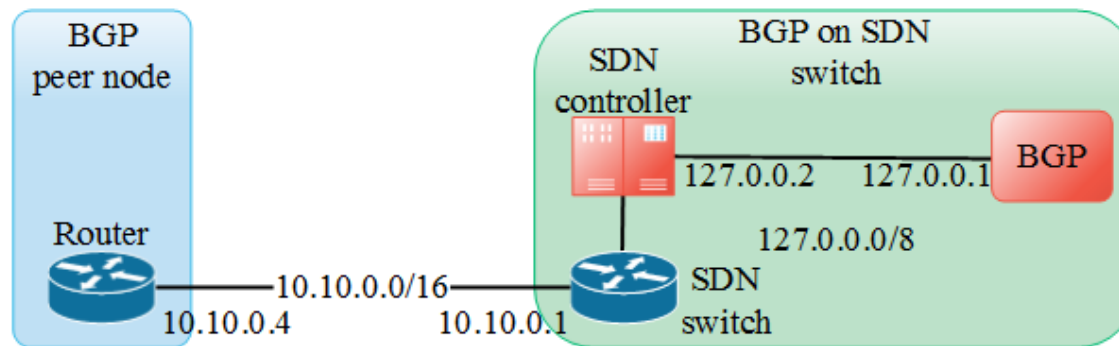


Global		Local (Loopback)	
dst	src	src	dst
10.10.0.4:179	10.10.0.1:37348	127.0.0.2:8179	127.0.0.1:44156

- 内部の通信をRAWソケットで, 外部の通信に OFの機能を用いることでOFポートを用いた通信が可能
 - 通信の等価性を維持するためにRAWソケットを使用
 - BGPに限らず任意のサービスを稼働可能

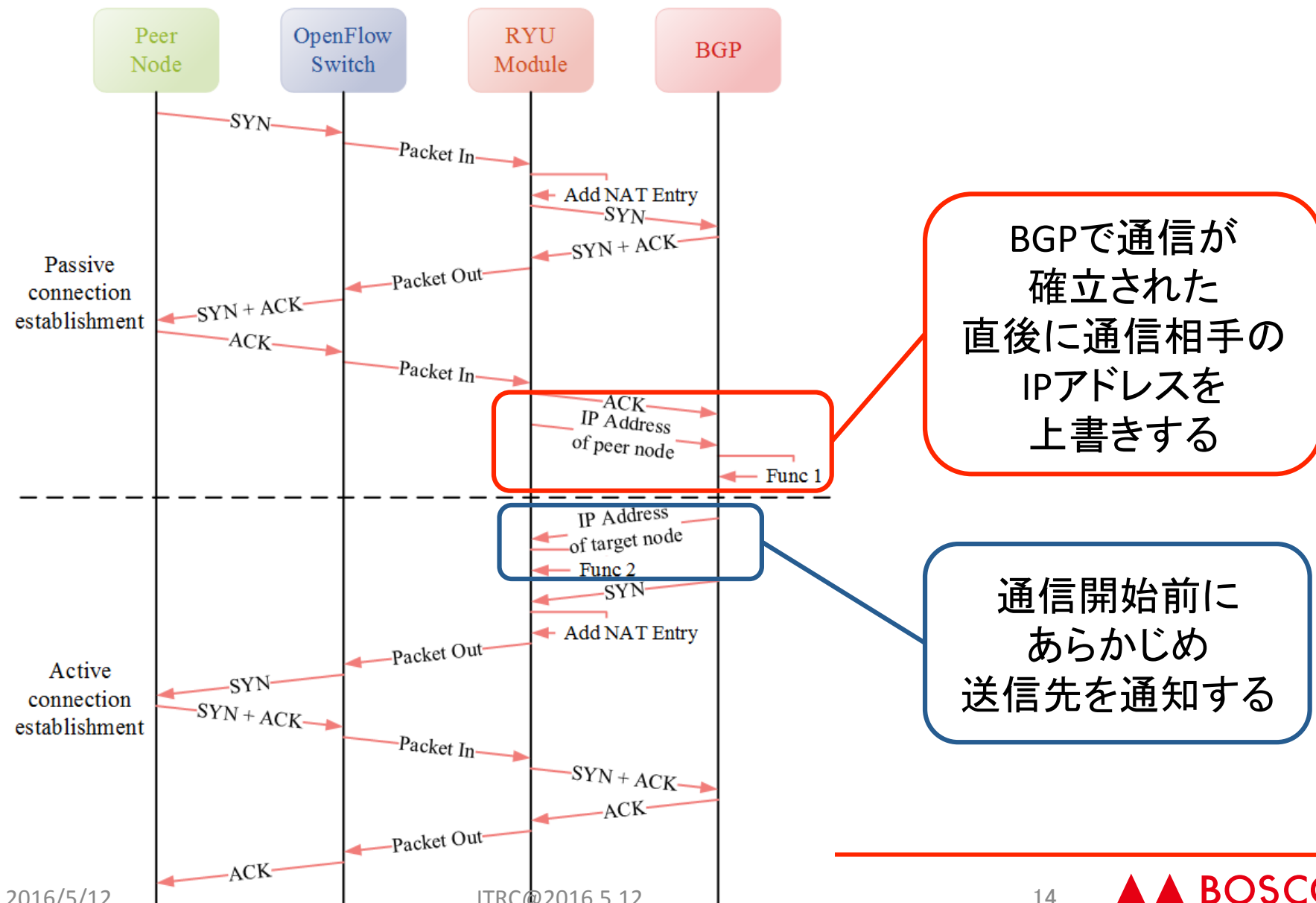
SDN-NAPT実装上の注意点

- ローカルのサービスから見ると、ゲートウェイはローカルアドレスとなる
 - ゲートウェイとして自分自身を指定できない



- BGPはピアノードとしか通信しない
 - BGPから見るとコントローラと直接通信することになるため設定にない接続と判断され、強制的に通信が切断される
- BGPはピア通信である
 - ローカル側から能動的にアクセスする場合、あらかじめコントローラに通信相手を通知する必要がある

SDN-NAPTにおけるセッション開始シーケンス



ルーティング情報とフロー情報の等価変換

- ルーティング情報とフローエントリの対比について

Kernel routing table	Flow entry
Destination	match : nw_dst
Gateway	actions : setfield
Metric	priority
Iface	actions : output

- 送信先はIPアドレスではなく、転送先のMACアドレスを使用
- priority と Metric は数値の意味が異なる

- 変換例

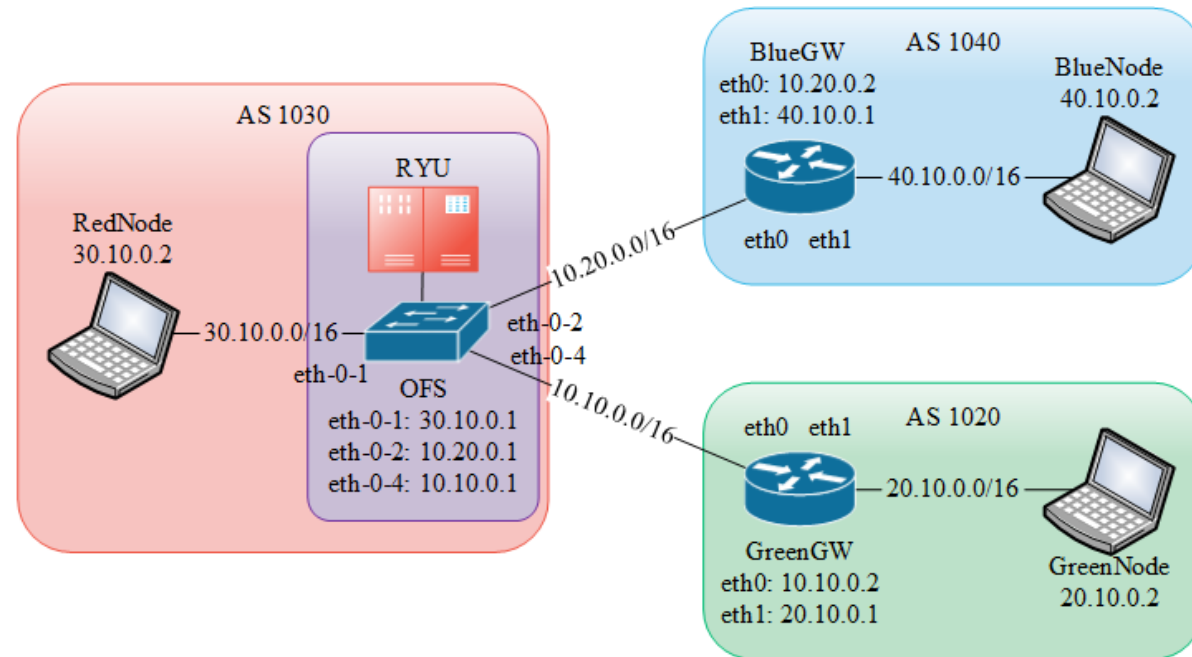
- ルーティング情報

```
# route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
20.10.0.0      10.10.0.2      255.255.0.0    UG      0      0      0 eth1
```

- フローエントリ

```
# ovs-ofctl dump-flows br0 --protocol=OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=2.976s, table=0, n_packets=0, n_bytes=0,priority=1000,
  ip,nw_dst=20.10.0.0/16 actions=set_field:08:00:27:e0:db:bf->eth_dst,
  set_field:08:00:27:bf:22:e9->eth_src,output:2
```

動作検証



- SDNスイッチの動作検証を行う
 - BGPメッセージの正常なやりとりの確認
 - BGPメッセージと連動した各ゲートウェイスイッチのフローおよびルーティング情報の確認
 - ping による疎通テスト

検証結果 (1)

- BGP メッセージの内容について
 - GreenGW, OFS 間の通信

No.	Time	Source	Destination	Protocol	Length	Info
3	0.113349	10.10.0.2	10.10.0.1	TCP	74	59110 > bgp [SYN] Seq=0 Win=1460
4	0.158063	10.10.0.1	10.10.0.2	TCP	74	bgp > 59110 [SYN, ACK] Seq=1460
5	0.158145	10.10.0.2	10.10.0.1	TCP	66	59110 > bgp [ACK] Seq=1 Ack=1460
6	0.158430	10.10.0.2	10.10.0.1	BGP	119	OPEN Message
7	0.226337	10.10.0.1	10.10.0.2	TCP	66	bgp > 59110 [ACK] Seq=429490722
8	1.115588	10.10.0.1	10.10.0.2	BGP	138	OPEN Message, KEEPALIVE Message
9	1.115648	10.10.0.2	10.10.0.1	TCP	66	59110 > bgp [ACK] Seq=54 Ack=1
10	1.116089	10.10.0.2	10.10.0.1	BGP	104	KEEPALIVE Message, KEEPALIVE Message
11	1.192189	10.10.0.1	10.10.0.2	TCP	66	bgp > 59110 [ACK] Seq=1 Ack=92
12	1.228733	10.10.0.1	10.10.0.2	TCP	66	bgp > 59110 [ACK] Seq=1 Ack=92
13	1.229402	10.10.0.1	10.10.0.2	BGP	85	KEEPALIVE Message
14	1.259803	10.10.0.2	10.10.0.1	TCP	66	59110 > bgp [ACK] Seq=92 Ack=1
15	1.360882	10.10.0.1	10.10.0.2	TCP	66	bgp > 59110 [ACK] Seq=20 Ack=1
16	2.130025	10.10.0.2	10.10.0.1	BGP	177	UPDATE Message, UPDATE Message
17	2.193100	10.10.0.1	10.10.0.2	BGP	172	UPDATE Message, UPDATE Message
18	2.193150	10.10.0.2	10.10.0.1	TCP	66	59110 > bgp [ACK] Seq=203 Ack=1

- 正しくBGPメッセージがやりとりされている

検証結果 (2)

- フローの登録内容

- GreenGW

```
# route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          10.10.0.1      0.0.0.0         UG    0      0      0 eth0
10.10.0.0        0.0.0.0        255.255.0.0     U      0      0      0 eth0
10.20.0.0        10.10.0.1      255.255.0.0     UG    1      0      0 eth0
20.10.0.0        0.0.0.0        255.255.0.0     U      0      0      0 eth1
30.10.0.0        10.10.0.1      255.255.0.0     UG    0      0      0 eth0
40.10.0.0        10.10.0.1      255.255.0.0     UG    0      0      0 eth0
```

AS1030 (OFS)
の
経路情報

- OFS

```
Switch# ovs-ofctl dump-flows br0 --protocol=OpenFlow13
OFPST_FLOW reply (OF1.3) (xid=0x2):
  cookie=0x0, duration=216.25s, table=0, n_packets=0, n_bytes=0,
  priority=1000,ip,nw_dst=40.10.0.0/16 actions=set_field:00:0a:85:07:0c:34->eth_dst,set_field:00:1e:08:08:93:38->eth_src,output:4
  cookie=0x0, duration=200.05s, table=0, n_packets=0, n_bytes=0,
  priority=1000,ip,nw_dst=20.10.0.0/16 actions=set_field:00:0a:85:07:0c:38->eth_dst,set_field:00:1e:08:08:93:36->eth_src,output:2
  cookie=0x0, duration=218.471s, table=0, n_packets=31, n_bytes=2692,
  priority=10 actions=CONTROLLER:65535
```

AS1020
(Green) の
経路情報

経路情報が正しく交換されている
ping も経路情報にもとづき到達する

今後必要となる技術と実務を考える

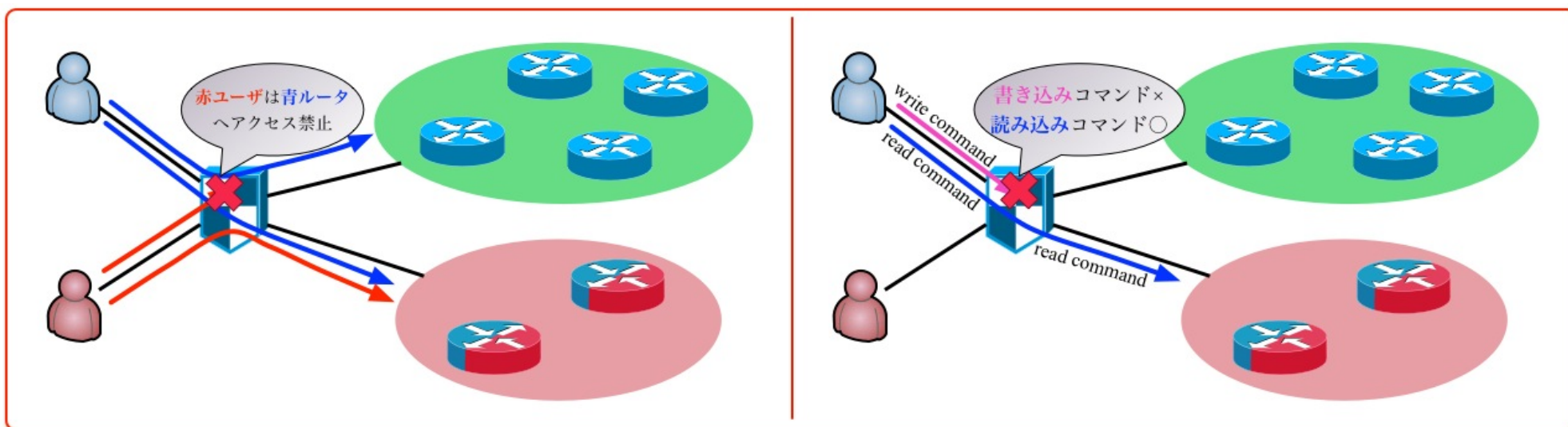
- OpenFlow 導入の課題
- SDN サービス運用課題

SDN サービス運用課題

- SDN導入コストが高い
 - 新しいシステムの導入費用は高くつく
- システム移行コストが高い
- 事業統合や分割、会社の買収によるシステム統合
 - 異なる2つ以上のネットワーク運用ポリシーの違いの吸収
例：一部のenable機能を他のシステム部隊に提供
- 求められるもの
 - ユーザ単位、インフラリソース単位での利用機能の制御
 - あらゆるネットワーク・インフラを一元管理

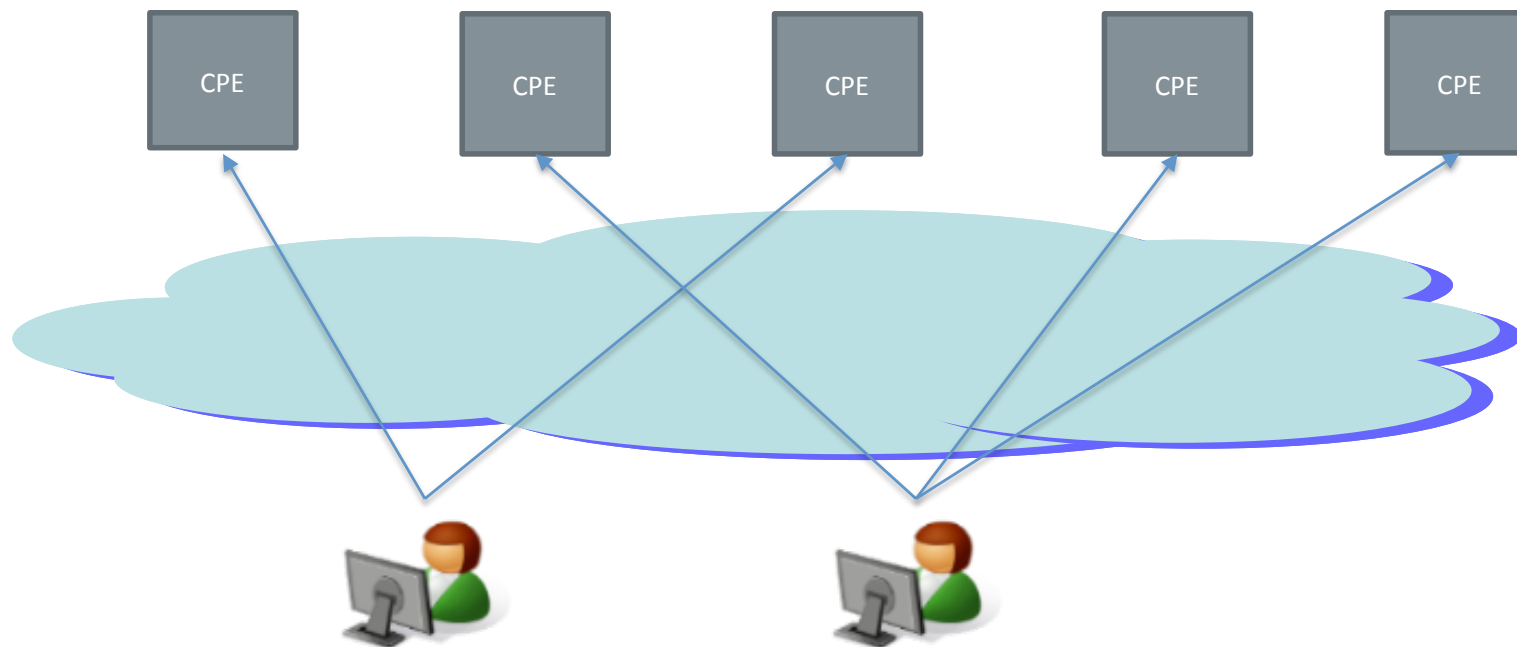
インフラの利用機能の制御例

- インフラ操作コマンドの制御により、SDNを利用したネットワーク変更に伴う運用ポリシーを吸収



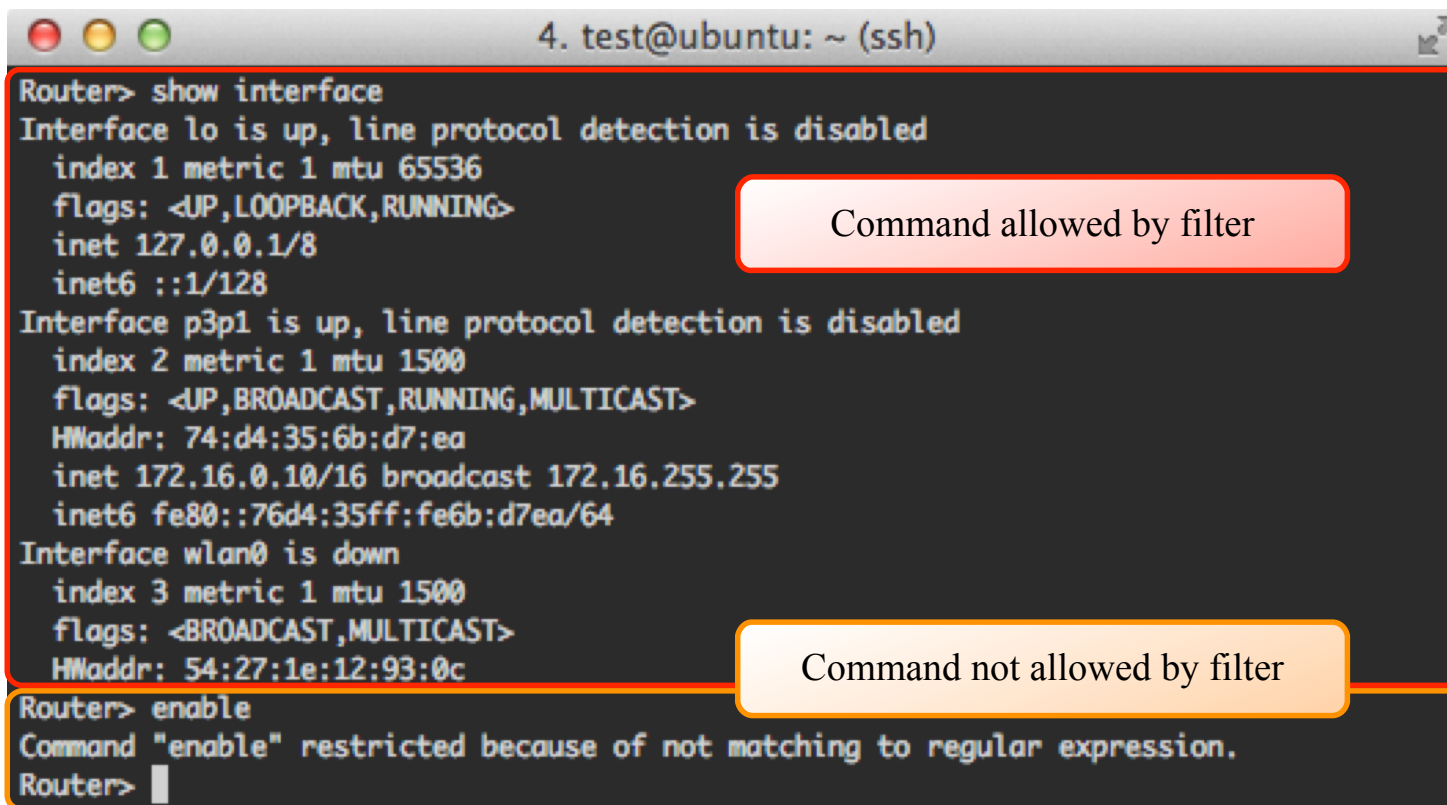
リアルタイムに全リソースの制御変更

- 原因調査のために投入したコマンドのログが記録できること
 - いつ、誰が、何を行ったかを一元管理
- 操作する対象リソースや仕様コマンドを制限



制御例

- 複数運用チーム間で、インフラ情報表示コマンド”show”だけが許されている例



The image shows a terminal window titled "4. test@ubuntu: ~ (ssh)". The terminal displays the output of the "show interface" command, which lists details for three interfaces: lo, p3p1, and wlan0. The output is as follows:

```
Router> show interface
Interface lo is up, line protocol detection is disabled
  index 1 metric 1 mtu 65536
  flags: <UP,LOOPBACK,RUNNING>
  inet 127.0.0.1/8
  inet6 ::1/128
Interface p3p1 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,MULTICAST>
  HWaddr: 74:d4:35:6b:d7:ea
  inet 172.16.0.10/16 broadcast 172.16.255.255
  inet6 fe80::76d4:35ff:fe6b:d7ea/64
Interface wlan0 is down
  index 3 metric 1 mtu 1500
  flags: <BROADCAST,MULTICAST>
  HWaddr: 54:27:1e:12:93:0c
```

Two callouts are present:

- A red callout box labeled "Command allowed by filter" points to the "show interface" command.
- An orange callout box labeled "Command not allowed by filter" points to the "enable" command.

Below the "enable" command, the terminal shows the message: "Command 'enable' restricted because of not matching to regular expression."

SMART Gatewayのデモを御覧ください

【参考】<http://www.bosco-tech.com/smart-gw/>

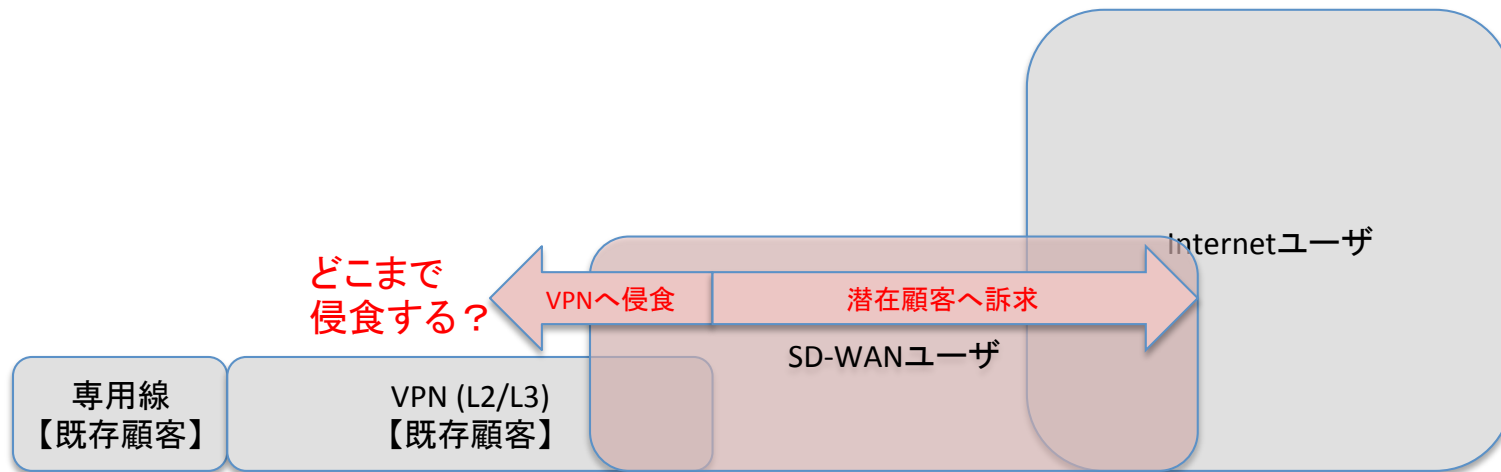
④ 今後期待が高まるSD-WAN

SD-WANの5つの特徴

1. 短時間でネットワークを構成できる
2. どこでも同じセキュリティポリシーで運用でき、柔軟なFailover機能を利用できる
3. アプリケーション志向で、しかもほぼリアルタイムにトラフィックエンジニアリングできる
4. 最適なパスの選択や、ネットワーク上の最適なアプリケーション選択が行える
 - 遅延インパクトの低減やアプリケーションレスポンスの最適化が実現できる
5. Zeroタッチコンフィグが実現できる

SD-WANの脅威と成長領域

- SD-WANの成長領域は以下2つ
 1. 現VPNサービス(一部?)を侵食
 2. これまでVPNを利用していない膨大な数のリトルカスタマー群(潜在顧客)へ訴求

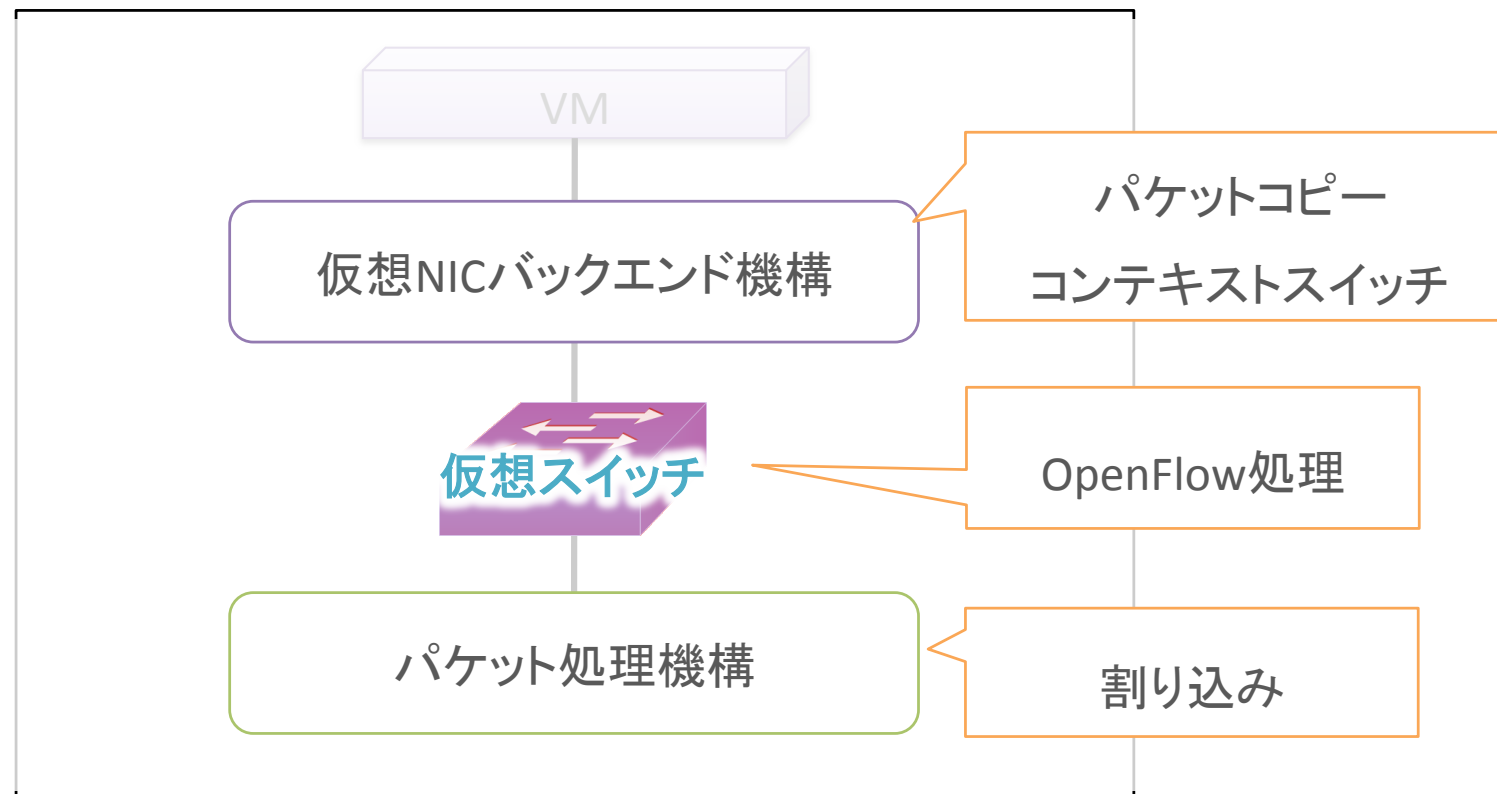


BOSCO TechnologiesにおけるSD-WANの取組

- 【取組1】
広域NWにおいてSD-WANの5つ特徴の基本機能を実現し、検証する
 - 数万台のエッジノードを想定した機能評価を5月13日より順次開始
- 【取組2】
SD-WANエッジにロックインされない仕組みを検討
 - Whiteboxを利用できるSD-WANコントローラのシステム構成
 - 仮想リソースのコントロールプレーンの制御方式
 - 仮想リソースのマネジメントプレーンの制御方式
- 【取組3】
SD-WANの特徴をManagement Planeから保管するツール開発
 - アプリケーション志向インフラ制御システムの構成法
→製品化「SMART Gateway」

⑤ Whiteboxの性能問題の評価

Whiteboxシステムモデルの性能問題を把握することが重要

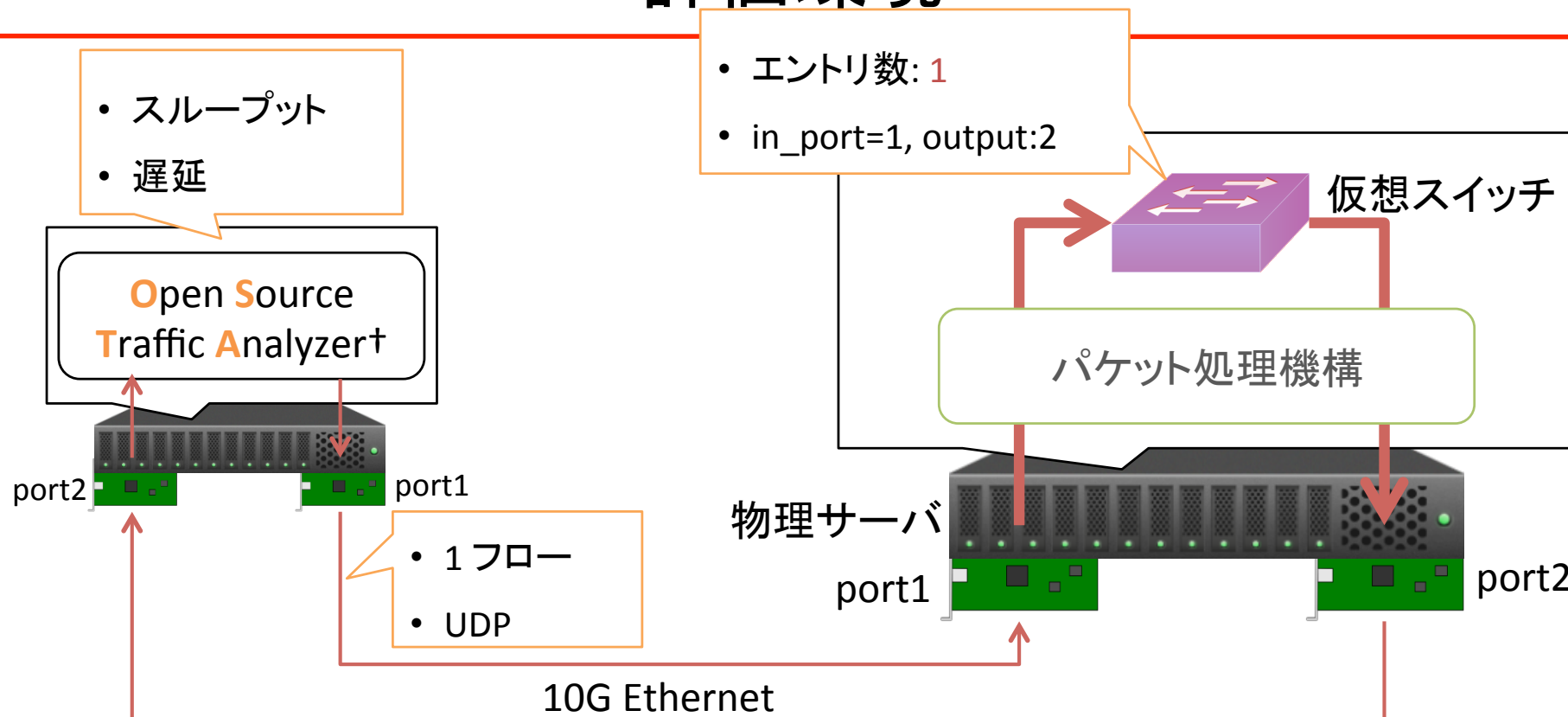


新たなパケット処理機構(仮想スイッチ)/仮想NICバックエンド機構が提案

パケット処理機構/仮想NICバックエンド機構

- パケット処理機構 (仮想スイッチ)
 - NAPI: Linux New API (Open vSwitch)
 - netmap (VALE)
 - Intel DPDK (OVS with DPDK, Lagopus)
- 仮想NICバックエンド機構
 - vhost-net
 - vhost-user

評価環境



物理サーバの性能

CPU	Intel Core i7 3.4GHz (4core, HT:off)
Memory	16GB
NIC	Intel X540-T2 10Gb Ethernet

ソフトウェアバージョン

OS	Fedora 22 (4.0.4-301-x86 64)
OVS	v2.4.0
netmap	v11.1
Lagopus	v0.2.3
DPDK	v2.1.0

評価対象/評価内容

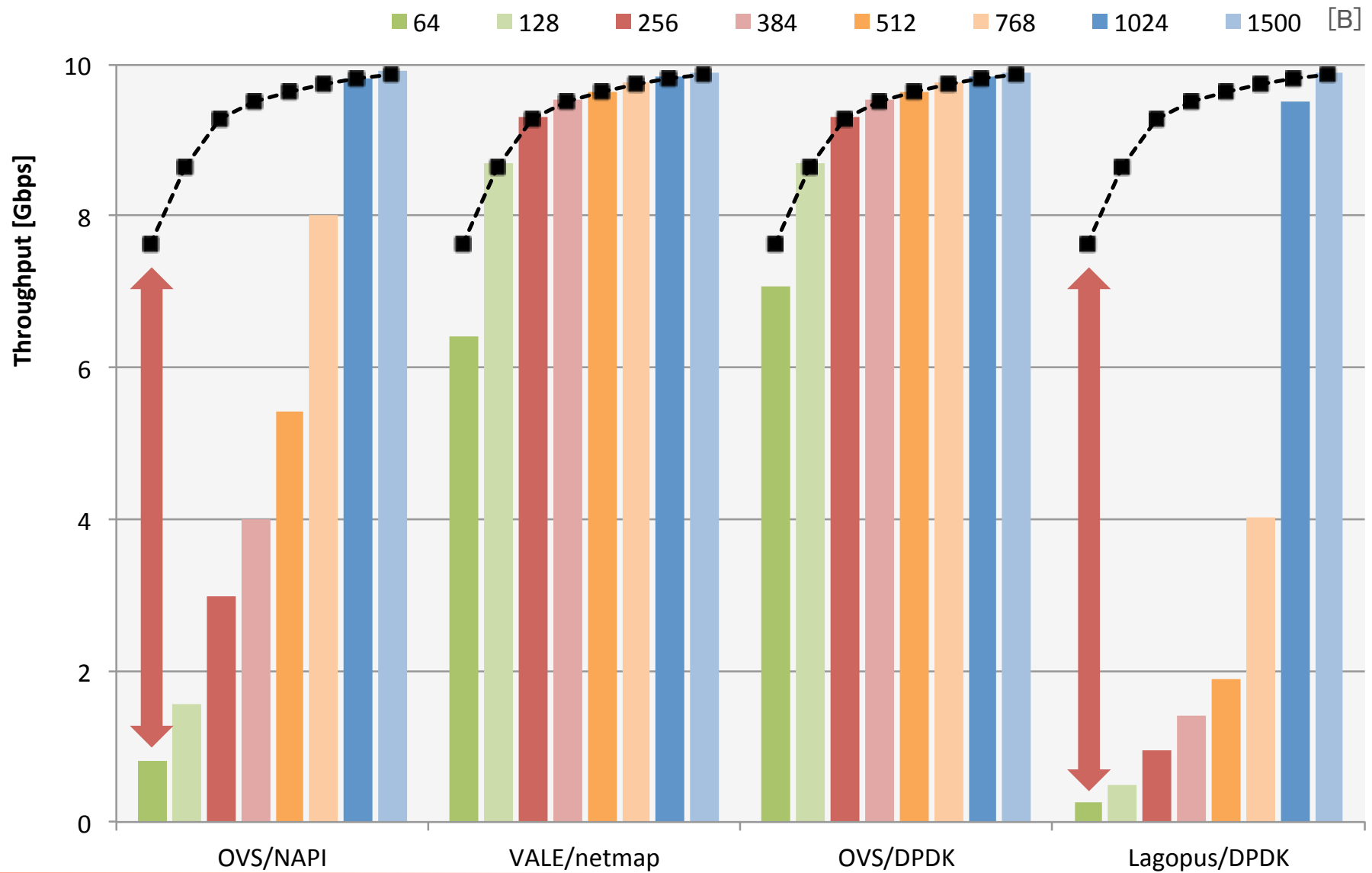
- 評価対象

	OVS/NAPI	VALE/netmap	OVS/DPDK	Lagopus/DPDK
仮想スイッチ (動作空間)	OVS (Kernel)	VALE (User)	OVS (User)	Lagopus (User)
パケット処理機構	NAPI	netmap	DPDK	DPDK

- 評価内容

- パケット転送速度
- パケット転送遅延

評価結果: パケット転送速度



1パケットに必要なCPUサイクル数†

- CPU Hz (3.4G) ÷ pps から概算

OVS/NAPI	VALE/netmap	OVS/DPDK	Lagopus/DPDK
2,294	288	262	7,389

OVS/NAPIに焦点を当てて要因を考察

† Assessing Soft- and Hardware Bottlenecks in PC-based Packet Forwarding Systems,
Paul Emmerich et. al, Technische Universität München, ICN 2015

NFV導入検討するにあたり、 Linuxにおけるパケット処理機構の性能評価

- 結論

- Linux上でパケット処理機構を評価し、性能要因を考察
- 転送速度を重要視する場合はnetmap, Intel DPDK
 - New APIではmemcpyやlockなどに処理に大半のサイクル数を消費
- 遅延ジッタはNew API, netmap, Intel DPDKすべてにおいて抑えられることが確認
 - 厳密に抑えたい場合はIntel DPDKが適切

- 今後の課題

- 多数のフローが混在する環境下での評価
- VM上でDPDKなどのパケット処理機構を用いた際の評価

⑥最新Whiteboxの紹介

最新Whitebox

- LANNER FX-7551



最新Whitebox

- LANNER FX-7551



最新Whitebox

- LANNER FX-7525



最新Whitebox

- LANNER FX-7525



