

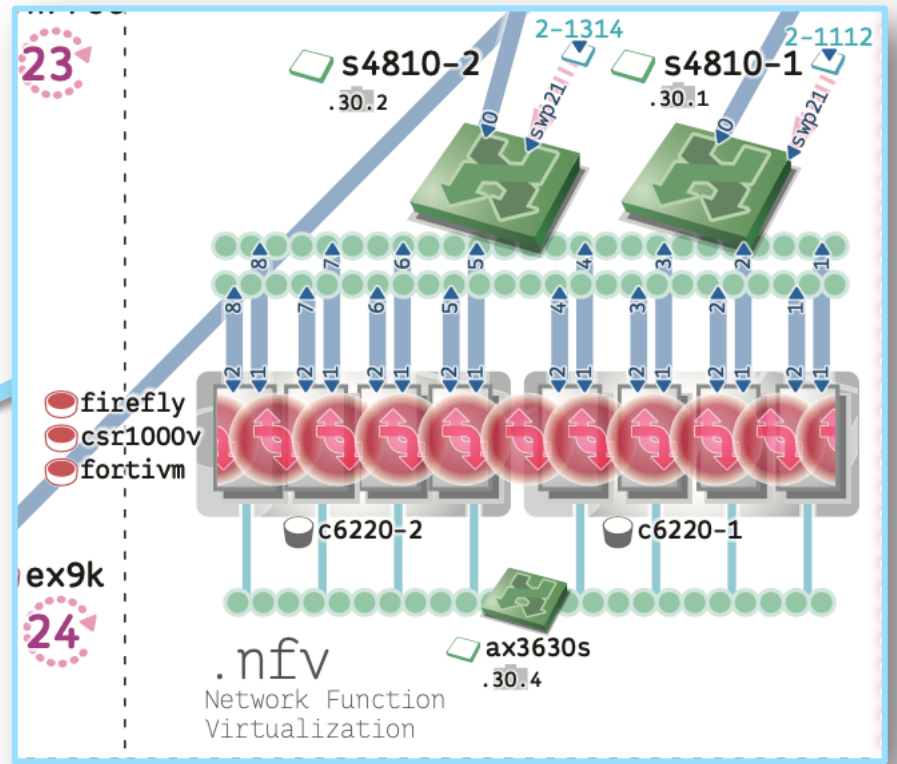
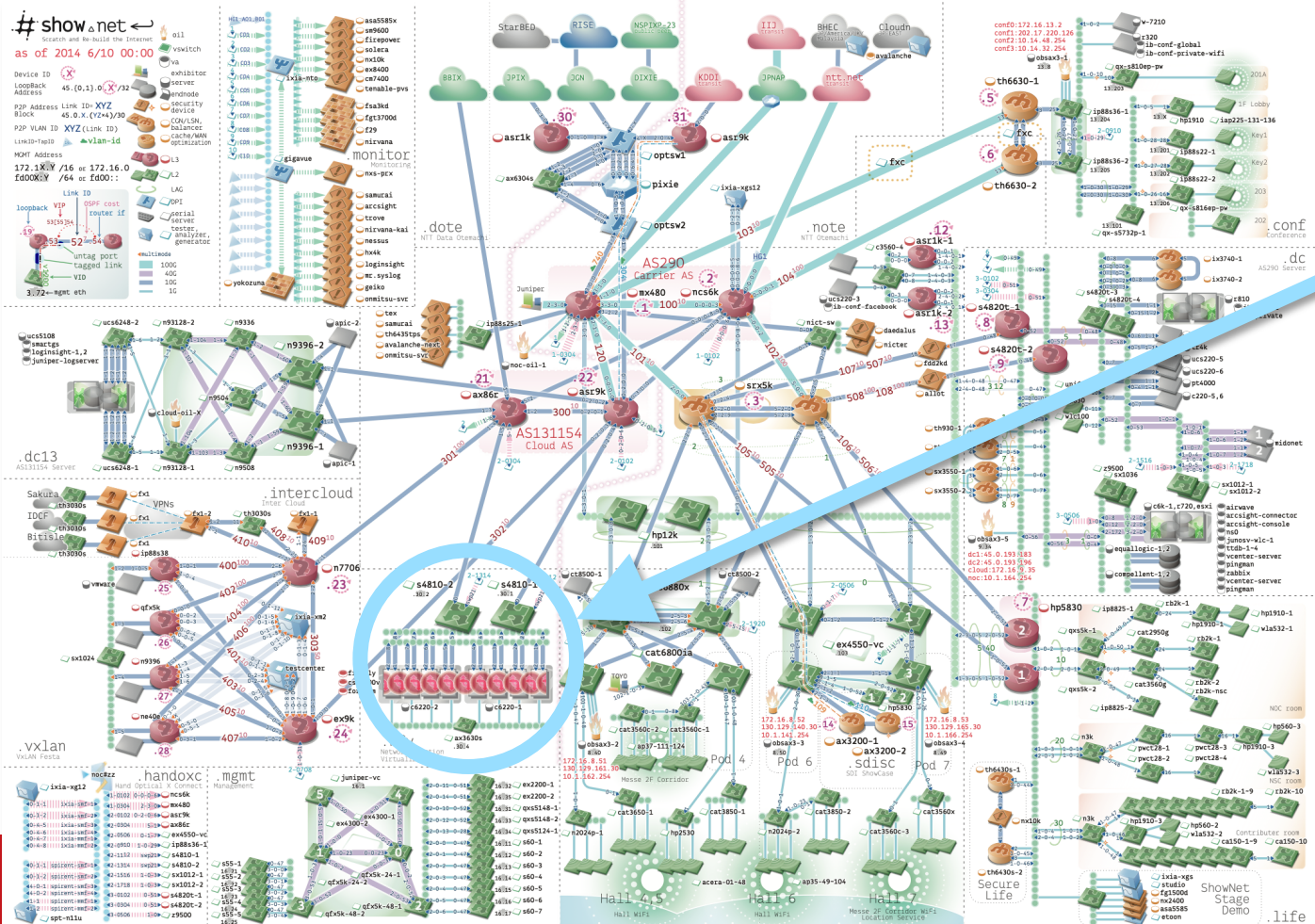
SDN/NFV@ShowNet 2014 ~ 2016

東京大学 情報基盤センター /
Interop Tokyo ShowNet NOCチーム
中村遼

2014年

- @ShowNet: OpenFlowからSDN/NFVへ
 - 2013年まではいかにOpenFlowを使うかが主眼
 - 2014年からはソフトウェア化されたネットワークインフラをいかに作るかにフォーカス
- OpenFlowはSDNを実現するための手法のひとつに
 - OpenFlow以外にもオーバーレイなどが手段として登場
 - VXLANのRFC7348が発行されたのが2014年8月
 - 2014年10月にnvo3からProblem StatementとFramework for Data Center Network VirtualizationのRFCが発行

2014年のSDN/NFV@ShowNet



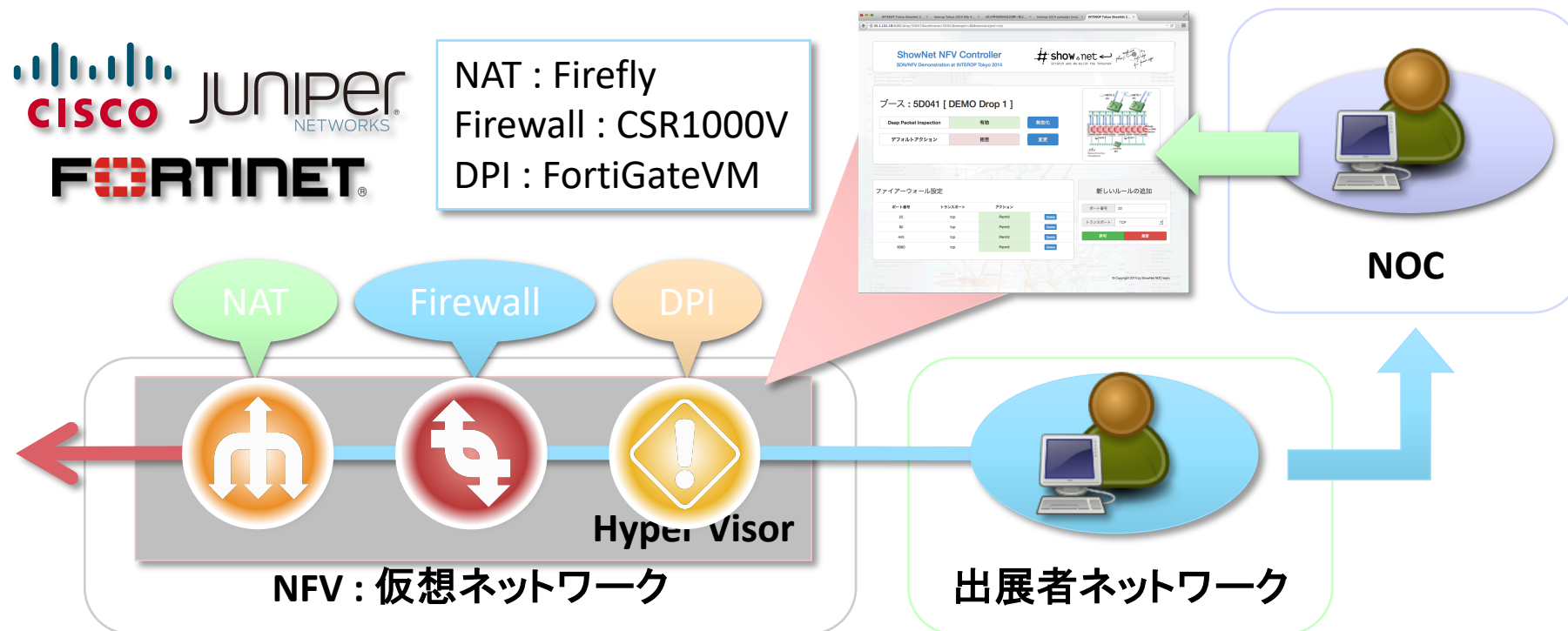
ShowNet 2014における Network Function Virtualizationの概要

- AS131154 : クラウドASにNFVクラウドを構築
 - 出展者ごとに仮想ネットワークを自動的に構築
 - 要望にあわせた柔軟なネットワークの変更
- AS290からユーザのネットワークを接続



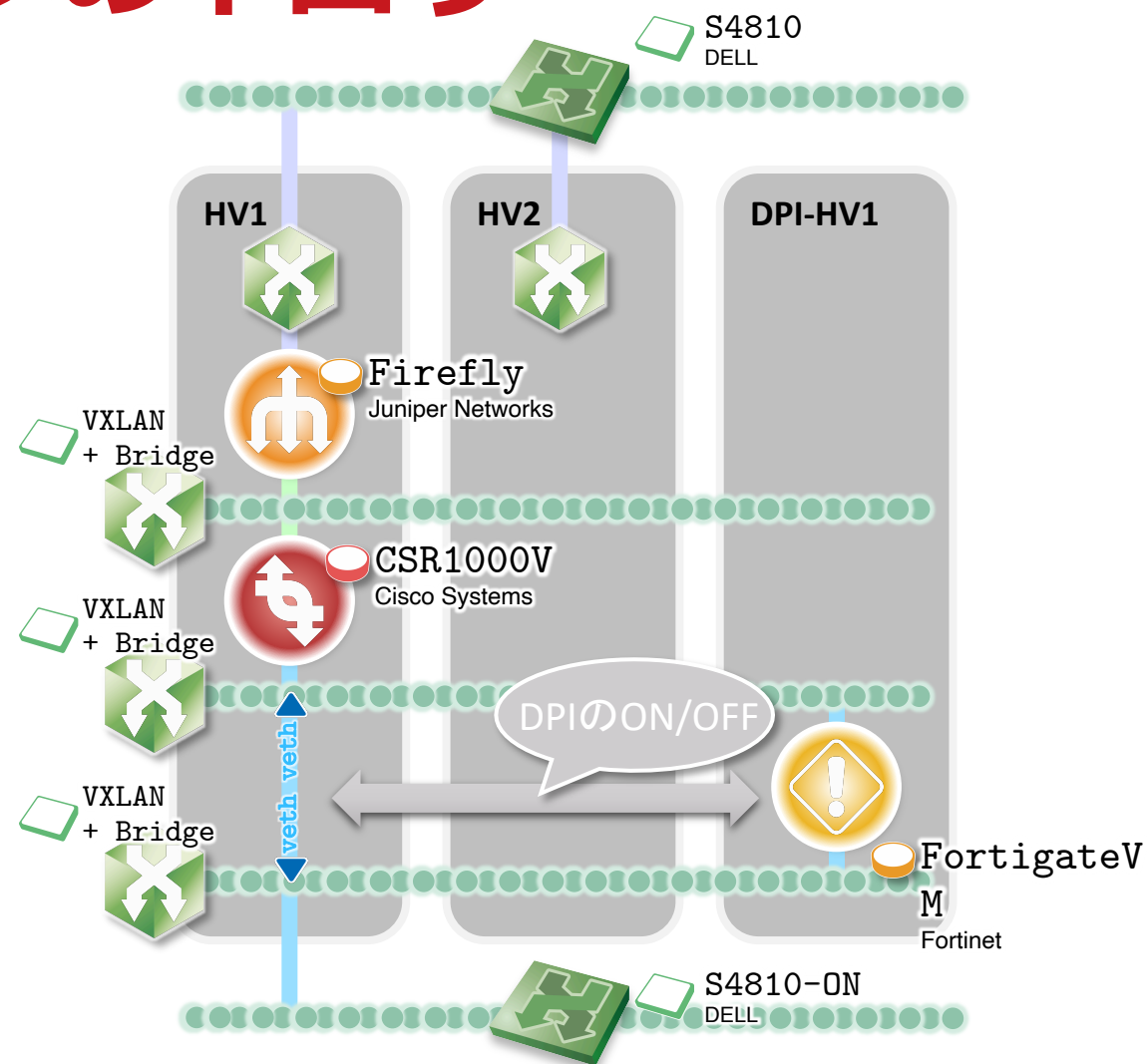
1出展社1仮想ネットワーク

- 1つの仮想ネットワークに3つのVirtual Appliance
- Web画面から各機能を設定可能



ネットワークの下回り

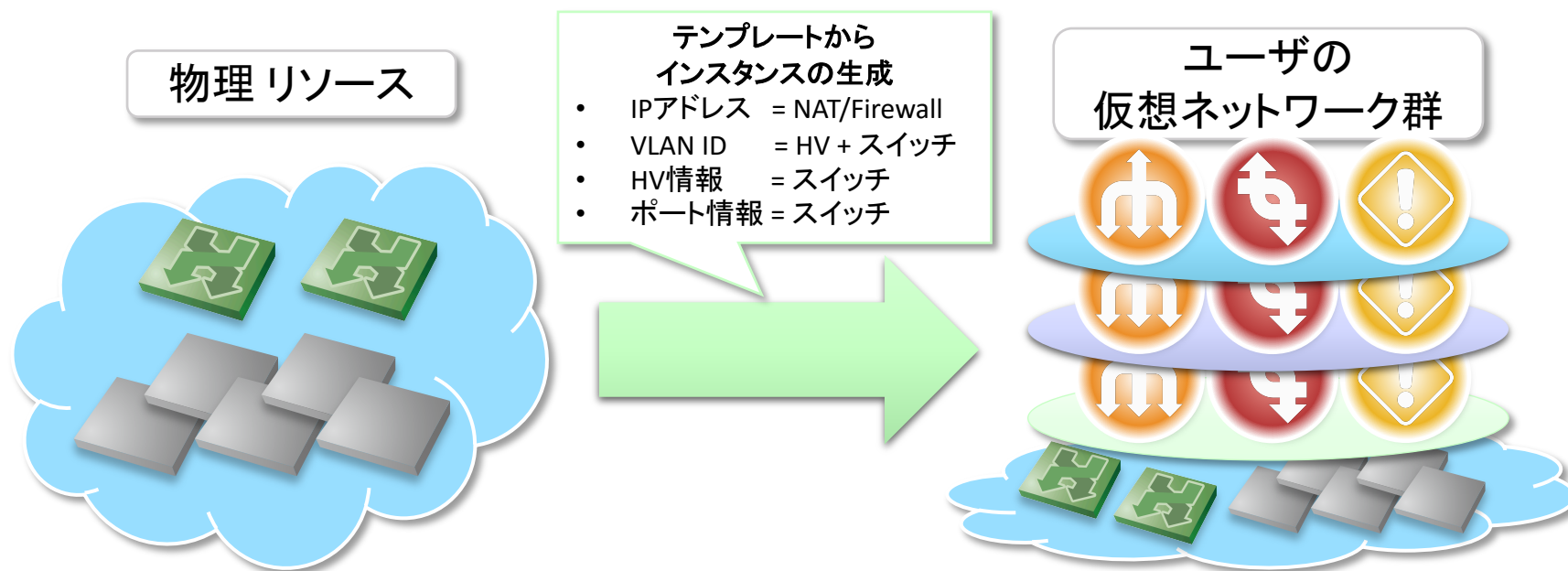
- OpenFlowではなくVXLANベース
 - 同一HV内は通常のbridge
 - HVをまたぐときはVXLAN
 - 機能を使わないときはvethでVAをバイパス



仮想ネットワークのテンプレート化

• ネットワークのテンプレート化と生成

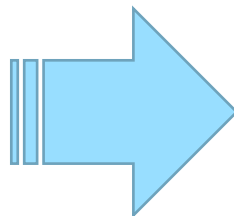
1. 仮想ネットワークのテンプレートを用意
2. 変数化されたユーザごとの情報を埋め込んだconfigをVAごとに作成、流し込んで仮想ネットワークを生成



SDN/NFV@ShowNet 2014を終えて

- ネットワークをソフトウェアでいじるのがだいぶこなれてきた
 - 1仮想ネットワークをゼロからデプロイするのに要する時間は1.5~2分程度
 - サーバ運用やプログラミングの知見がネットワークの設計・構築・運用に直結しはじめる
 - ただし、当時は仮想ネットワークの性能が課題だった
 - Linuxのソフトウェアパケット処理性能がまだつらかった
 - DPDKの最初のtag v.1.2.3r0が2012年9月。2014年はまだVAにDPDKが入るほどではなかった
 - 例えばVPPの最初のtag v.1.0.0が2015年12月, Linux VXLANのRemote Checksum Offloadが2015年1月

2013年は徹夜の連続だったのが...

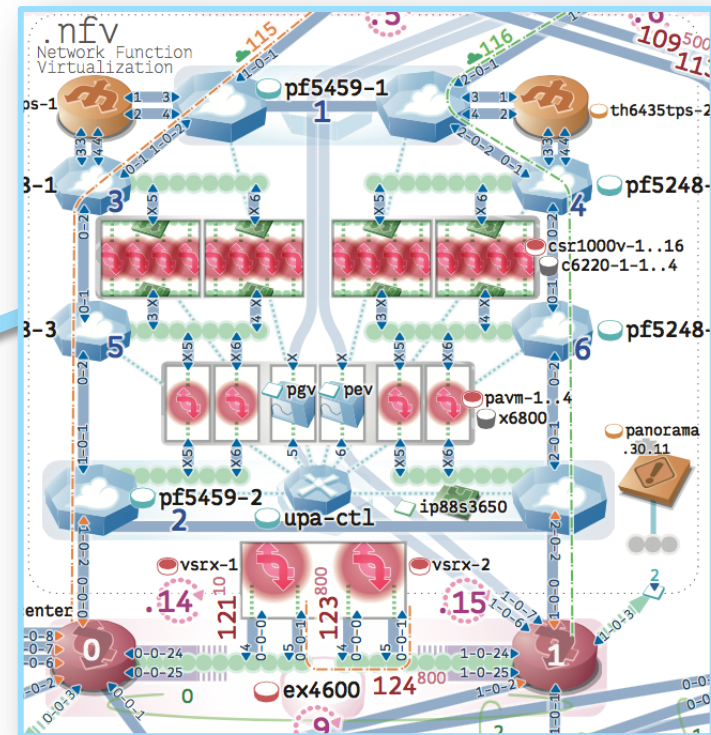
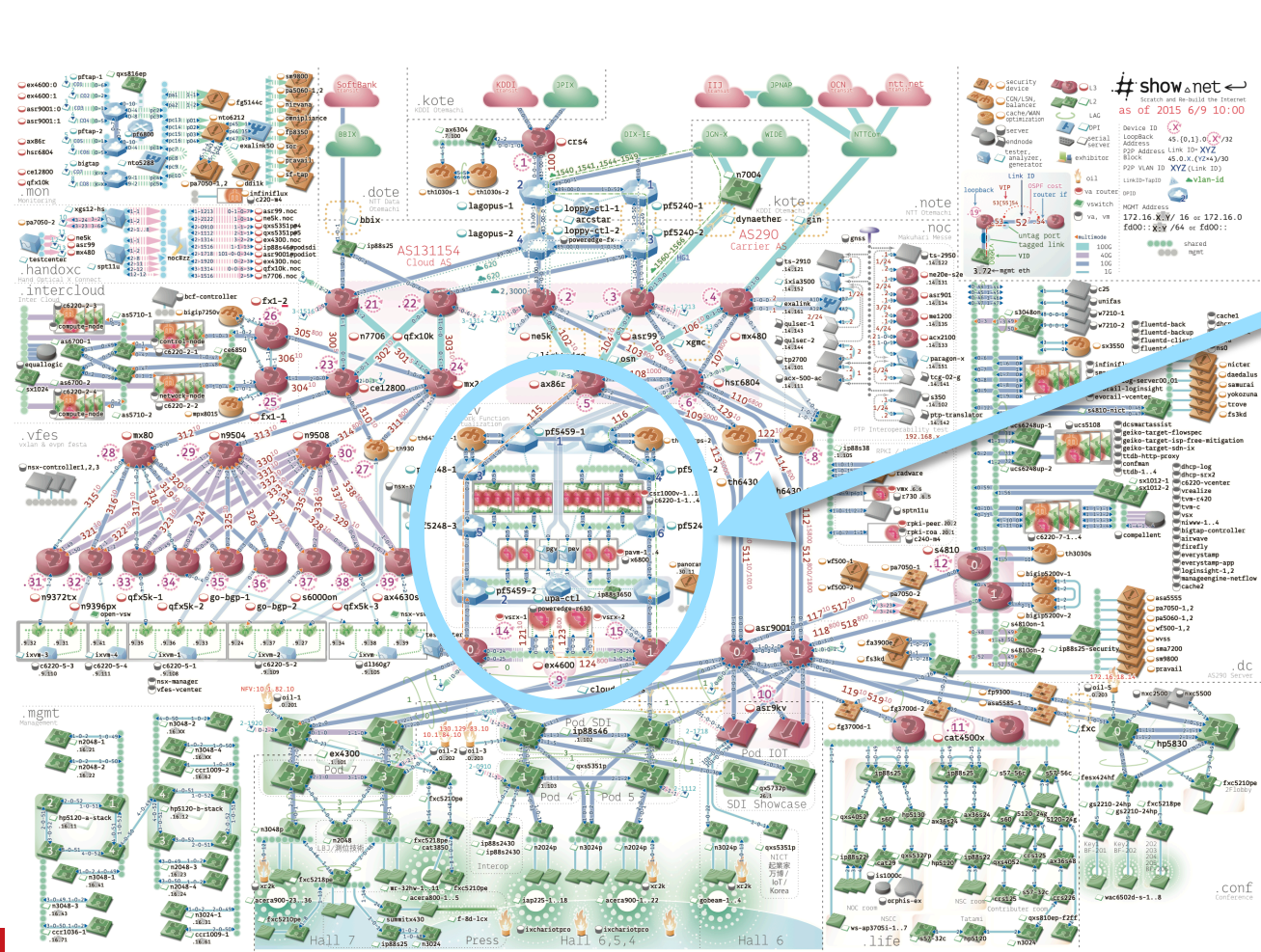


比較的余裕が生まれた

2015年

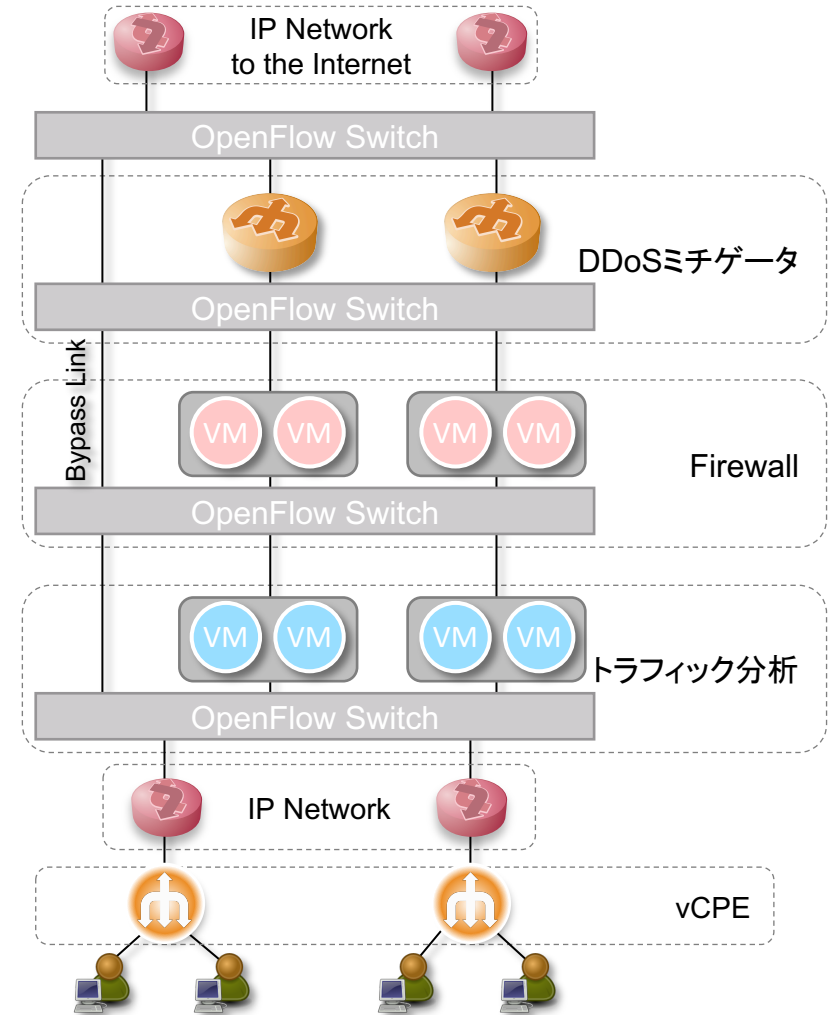
- @ShowNet: 2014年の課題の克服を目指す
 - 1VAの性能はいきなりあがらない
 - ネットワーク全体として性能をスケールアウトさせるには？
- NFV関連の標準化やオープン実装、PoCが進行
 - 2013年、ETSI ISG NFVからドキュメントがpubされ始める
 - 2014年10月、OPNFVが登場
 - 2015年、SFC WGからProblem StatementとArchitectureのRFCが出る

2015年のSDN/NFV@ShowNet



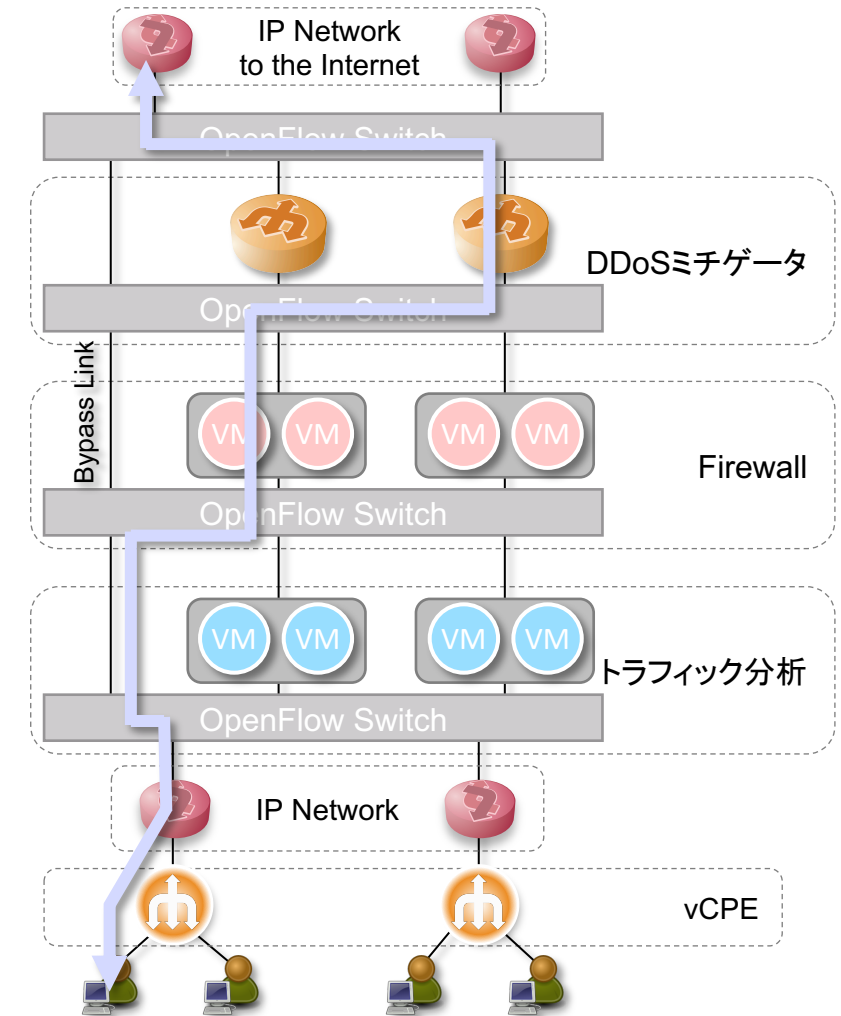
スケールアウトできるNFV/Service Chaining

- ユーザにVAを占有させるのをやめる
 - トラフィックごとに各サービスを適用するかどうかを判断する
- 同一サービスを複数VAで構成
- OpenFlowを使って、複数VAへ全トラフィックをロードバランス
- 様々な高速化手法を使って全体の性能を底上げ



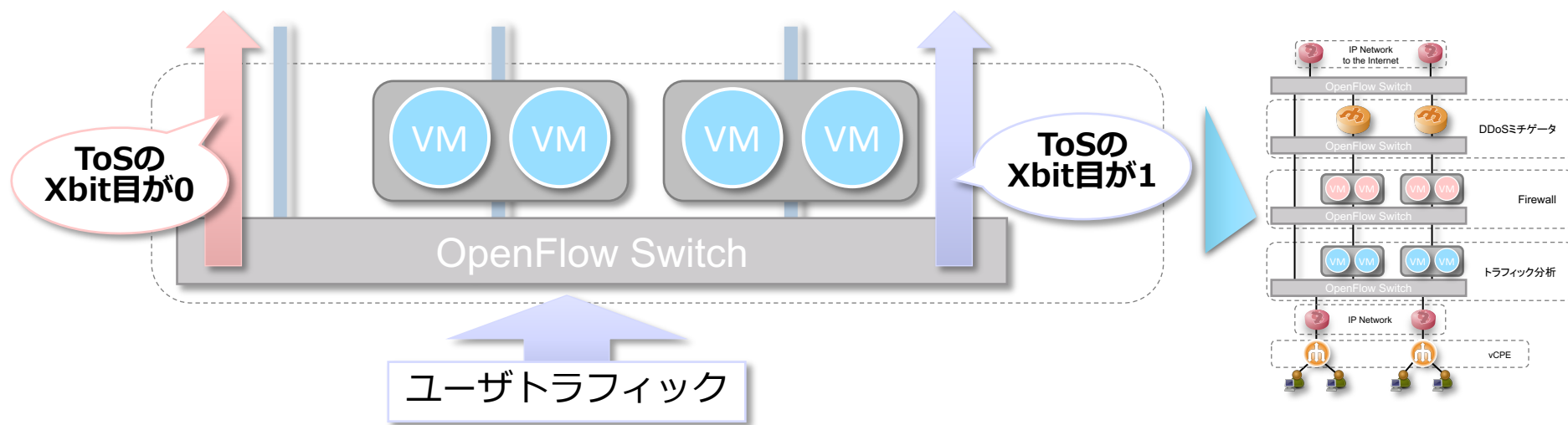
スケールアウトできるNFV/Service Chaining

- VNF Layering Model
 - 複数VMでひとつのVNF層を作り、それを重ねていく
- CPEでパケットに通るべきサービスをマーキング
 - 各サービスをToSのbitとして埋め込む
- OpenFlowでサービスの適用をパケット単位で判断
 - ToSの特定bitが1ならVNFへ、0ならバイパス
- 1つのサービスを構成する複数VMへトラフィックを分散
 - 送信元アドレスをハッシュして転送するVMを決定
 - OpenFlowで決定したVMへOutput



OpenFlowによるパスの切り替え

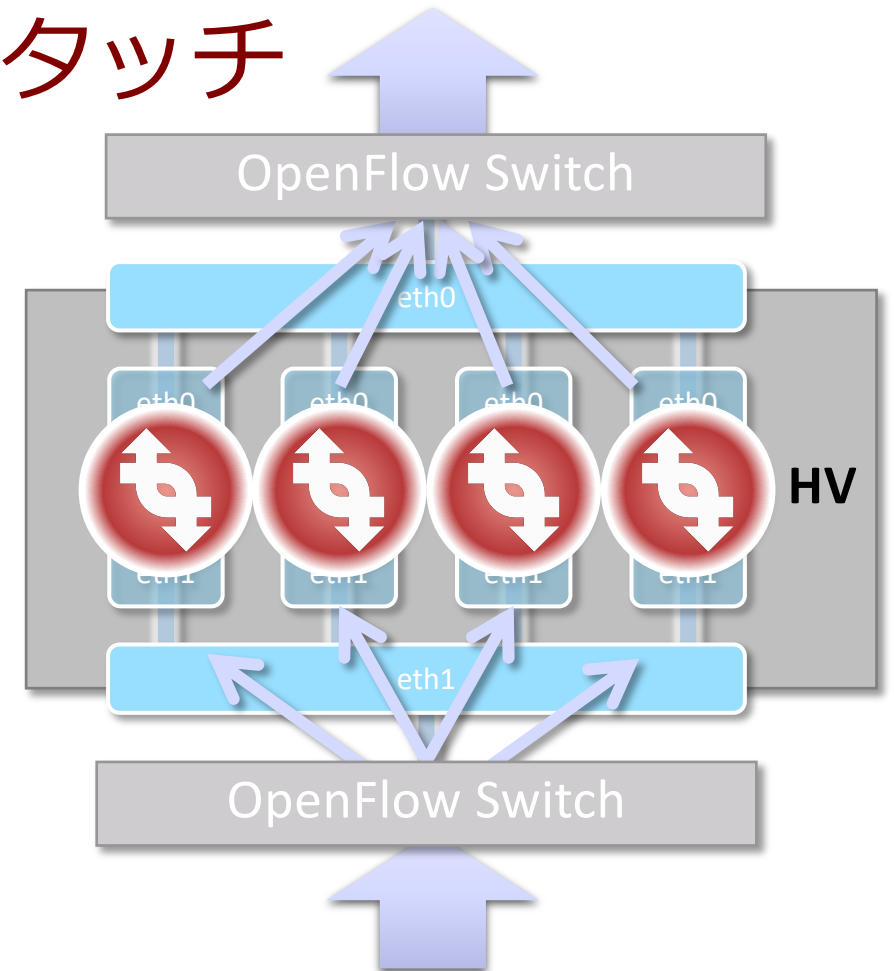
- 転送時にパケットをVNFに通すかを判断する
 - ToSと送信元アドレスをマッチ
 - ユーザ(アドレス)とそのサービス(ToS)をフローとして、フローごとにVNFを通すか制御
 - 各VNFの層は全く同じ手順で動作
 - この後のShowNetでちょいちょい出てくるToSの<s>悪用</s>はこれがおそらく最初？



高速化手法の利用: SR-IOV

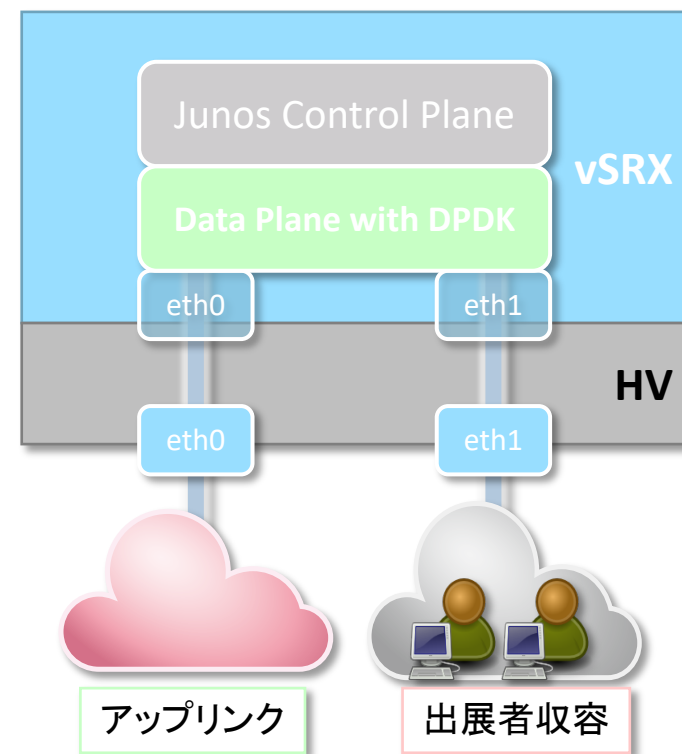
- SR-IOVの仮想NICを各VAにアタッチ

- 最もオーバーヘッドが小さい
- VFごとにMACが異なるので、OpenFlowスイッチでdst MACを書き換えることで同一HV上のVAへトラフィックを分散できる
- Cisco CSR1000Vで利用



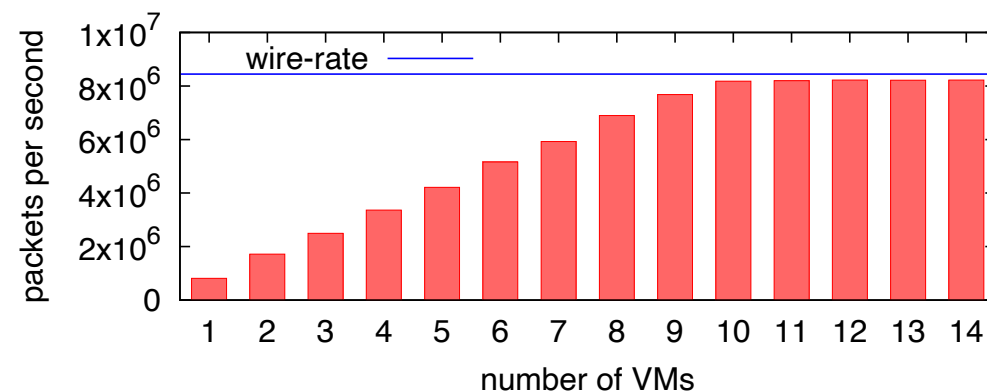
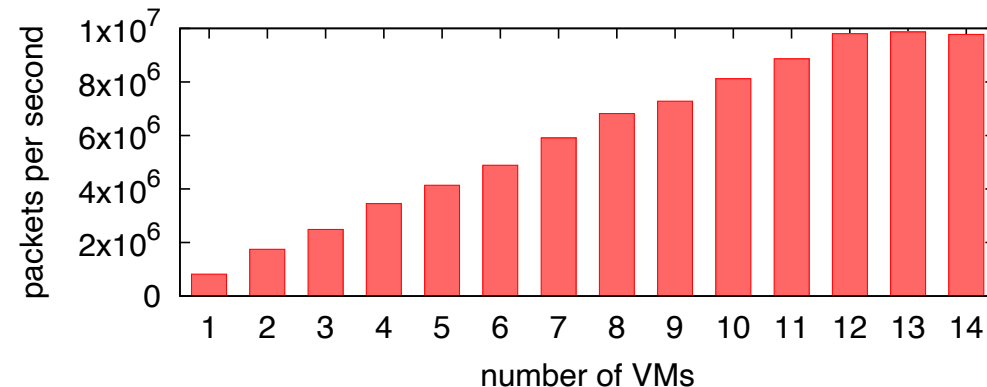
高速化手法の利用: DPDK

- この年初めてDPDK対応のVAが登場
 - Juniper Networks, vSRX
 - 出展社セグメントを収容し、サービスメニューに応じたToSの書き換えを実施



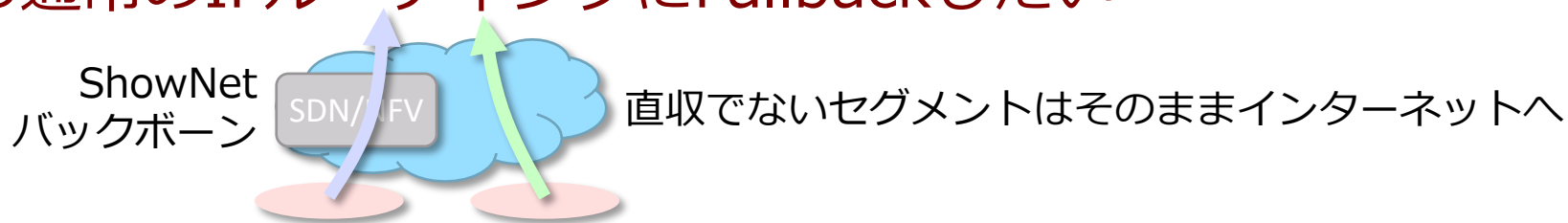
スケールアウトはうまくいった

- Interop後に実験を実施
 - VMを足せば足すだけ全体の性能が向上 (スケールアウト)
 - 汎用サーバ1台でもVMを10台あげれば128-byteのパケットでwire-rate
 - ShowNetの話を含む詳細は論文に
 - Ryo Nakamura, Kazuya Okada, Shuichi Saito, Hiroyuki Tanahashi and Yuji Sekiya, "FlowFall: A Service Chaining Architecture with Commodity Technologies", 2nd IEEE International workshop on CoolSDN 2015, November 2015



SDN/NFV@ShowNet 2015を終えて

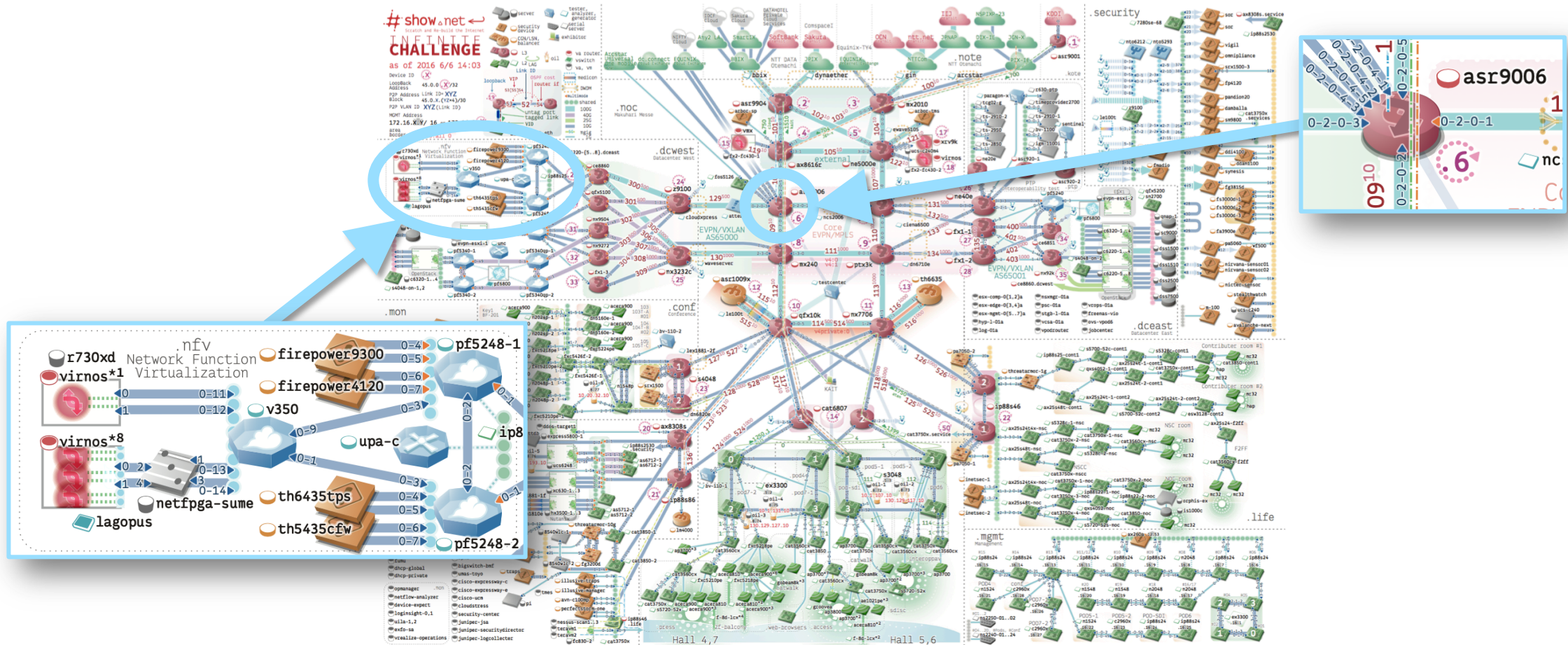
- スケールアウトの大事さ
 - SDN/NFVインフラ全体での性能を向上できる設計
 - サーバと違ってネットワークはただ足せばいいってものではない
 - 個別の部分の性能向上: SR-IOVやDPDKの利用
- 一方で、どうやってユーザトラフィックを引き込むかが課題
 - 2013年からずっと、特定のユーザセグメントのみをSDN/NFVに直収
 - 逆に言えば、直収以外のセグメントには適用できなかった
 - SDN/NFVが壊れたら通常のIPルーティングにFallbackしたい



2016年

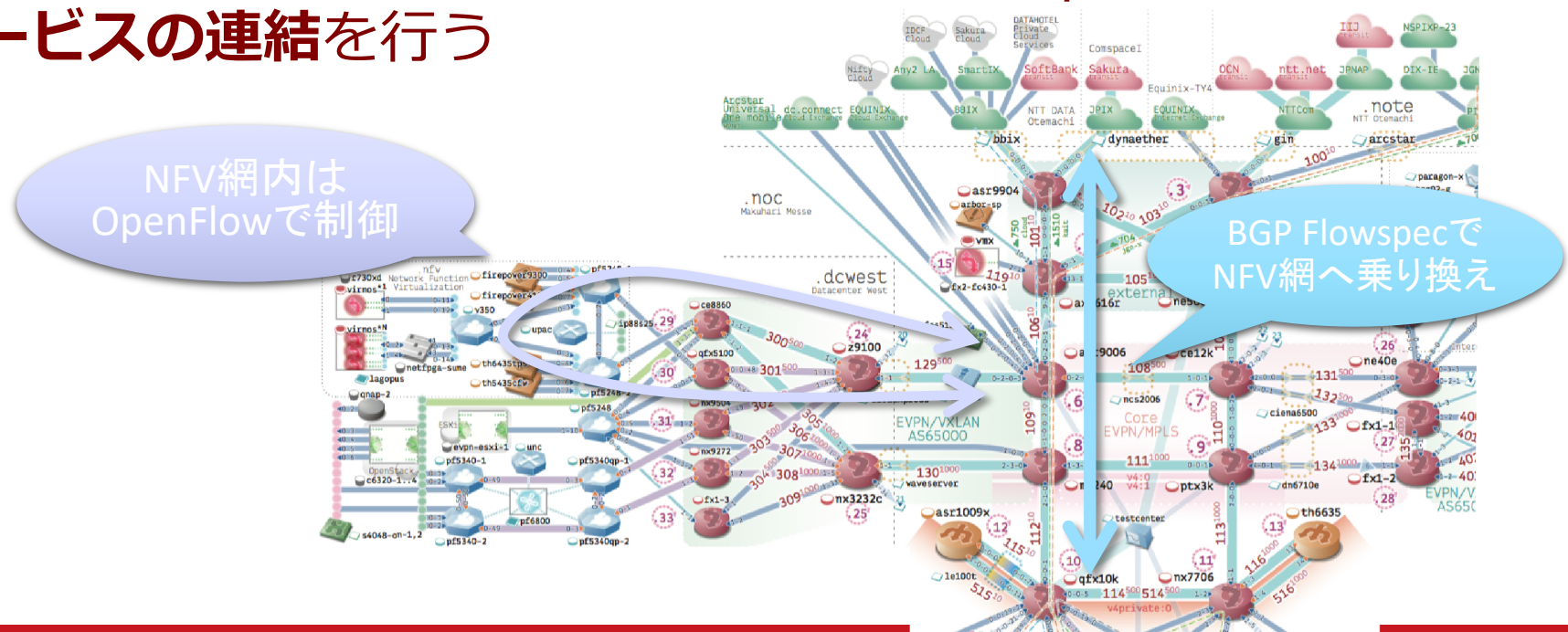
- @ShowNet: 2015年の課題の克服
 - SDN/NFVとIPバックボーンをいかに組み合わせるか
 - 任意のトラフィックへのサービス適用とFallback
- プログラマビリティの深化
 - P4: Programming Protocol-Independent Packet Processors
 - 最初の論文が出たのが2014年7月
 - behavioral-model (bmv2)の1.0.0が2016年6月
 - 既存のネットワーク機器もそれっぽく制御したい
 - <https://github.com/openconfig/public> のinitial commitが2015年9月
 - YANG 1.1, RFC7950が2016年8月

2016年のSDN/NFV@ShowNet 2016



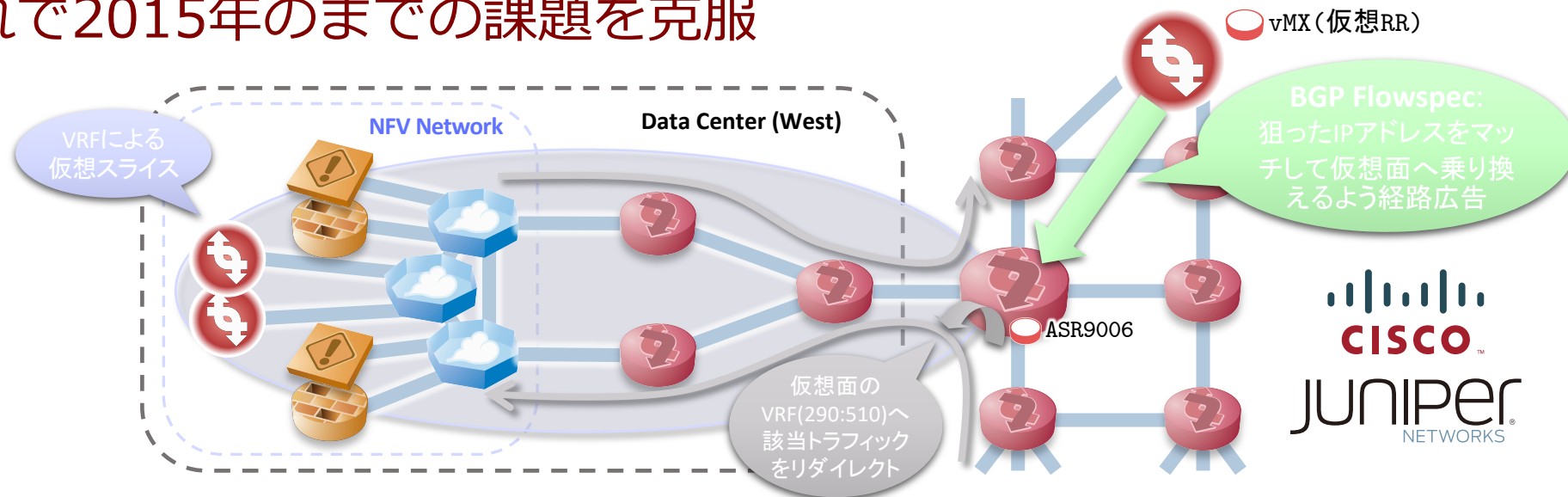
SDN/NFV@ShowNet 2016

- BGPで"乗り換え"、 OpenFlowで"連結"
 - BGP Flowspecを用いて狙ったトラフィックをNFV網へ乗り換える
 - Data Center部分に構築したNFV網では、 OpenFlowでよりきめ細かなサービスの連結を行う



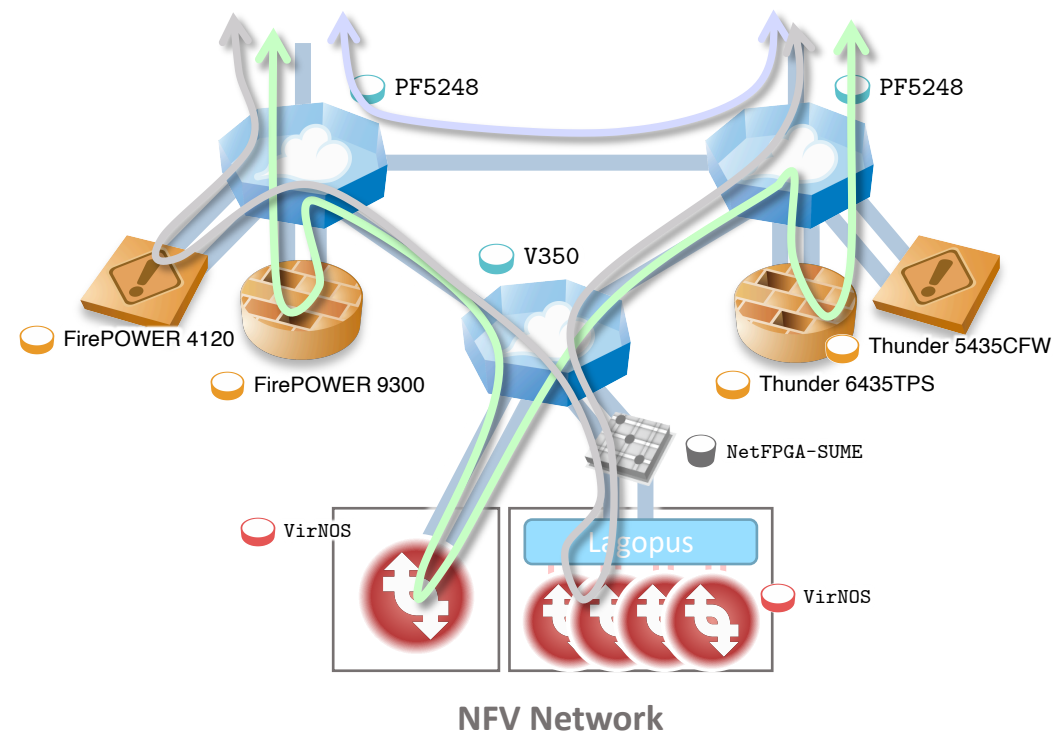
BGP Flowspecで"乗り換え"

- 狙ったトラフィックをNFV網へリダイレクト
 - VRFを使って仮想面として構築したNFVネットワークに、BGP FlowspecのVRF Redirectを使って、狙ったトラフィックだけを乗せ換える
 - BGP Flowspec自体はShowNet 2015でも相互接続検証を実施
 - これで2015年のまでの課題を克服



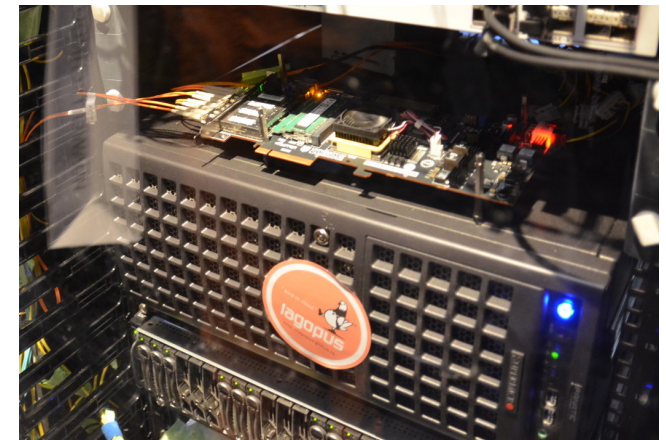
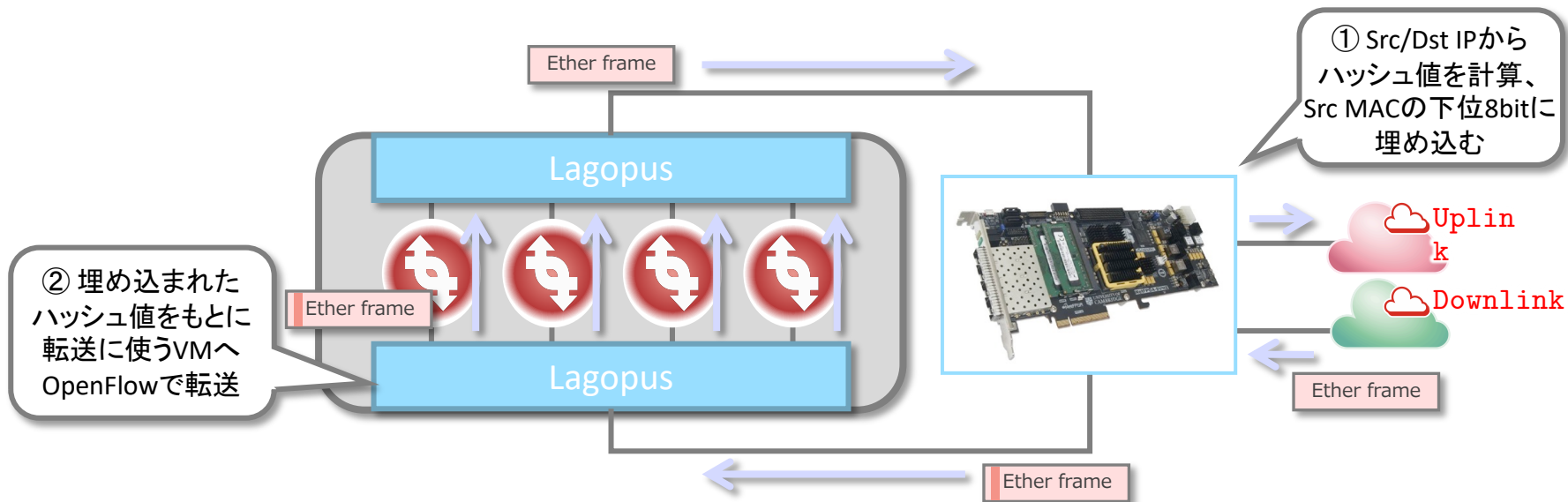
OpenFlowで"連結"

- 2014とは違ってもうこれくらいは簡単
 - OpenFlow Switch
 - NEC: PF5248, FXC: V350
 - Network Functions
 - A10 Network
 - Thunder 5435CFW: Firewall
 - Thunder 6435TPS : DDoS Mitigator
 - Cisco Systems
 - FirePOWER 9300 : Firewall
 - FirePOWER 4120 : DDoS Mitigator
 - IP Infusion
 - VirNOS : ACL-based Firewall and Lagopus



プログラマブルなハードウェアの利用

- FPGAを使ったデモンストレーションを実施
 - OpenFlowはマッチとアクション、計算はできない
 - FPGAでパケットのハッシュ値を計算し、トラフィックの分散先を決定する



むき出しのNetFPGA-SUMEと、LagopusとVirNOSの動作するx86サーバ

SDN/NFV@ShowNet 2016を終えて

- IP Routingは偉大
 - めっちゃ枯れてる
- 技術は機材適所、うまく接続する設計が大事
 - この年と言えばBGP FlowspecとOpenFlowの役割分担
- ソフトウェアによるパケット処理の進化
 - 10Gbpsは余裕で出せるようになった
 - 一方で、真剣に使うにはプログラミングだけでなく、コンピュータアーキテクチャへの理解も必要に

2014 ~ 2016の3年間

- ネットワークにおけるプログラマビリティ
 - OpenFlowから始まってSDNはより広い領域へ
 - データプレーンだけでなく、既存のネットワーク機器をプログラマブルに触るためのAPI
 - NETCONF, RESTCONF, YANGやOpenConfig, Ansible, etc
 - ネットワーク屋さんもプログラミングをよく知る必要が...
- ソフトウェアによるパケット処理の進化
 - ネットワークアプライアンスの仮想化
 - 最初はただOSがVMとして動くだけ
 - それがSR-IOV、DPDK、vhost-userなどでどんどん高速化
 - ネットワーク屋さんもコンピュータをよく知る必要が...
 - CPU pinningやNUMA環境でのVM配置などをはじめ、様々なチューニングポイント

属人化の果てに... 2014~2016の苦しみ

- コントローラ/オーケストレータを個人で実装
 - 実装した人にしか細かい挙動がわからない
 - チーム全体から見ると完全にブラックボックス
 - 幕張メッセからホテルに帰れない...
- トラブルシューティングの難易度が高い
 - 通常のRoutingにOpenFlowやSDN、さらにサーバの知識が必要
 - 物理と仮想が相まって構成も複雑化する一方
 - さらにはプログラミングも
 - 例えばコントローラのAPIを作ろう/触ろうとすると大抵RESTだったので、ちょっとしたWebアプリの知識も必要