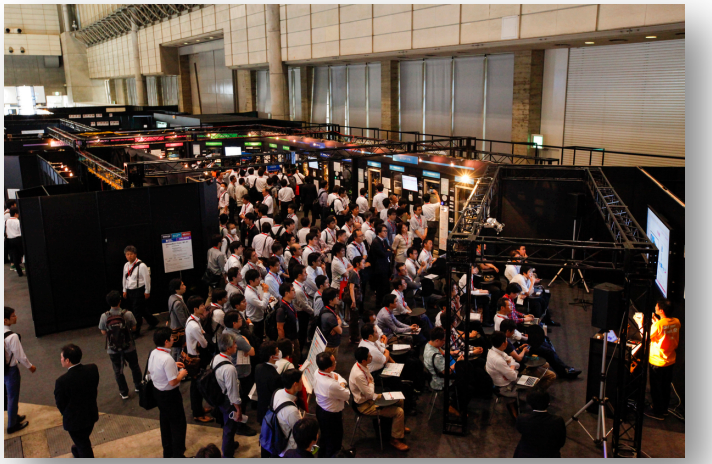
技術の適材適所でつくるSDN/NFV @ Interop Tokyo 2016 ShowNet

Interop Tokyo 2016
ShowNet NOCチーム
中村 遼

Interop Tokyo

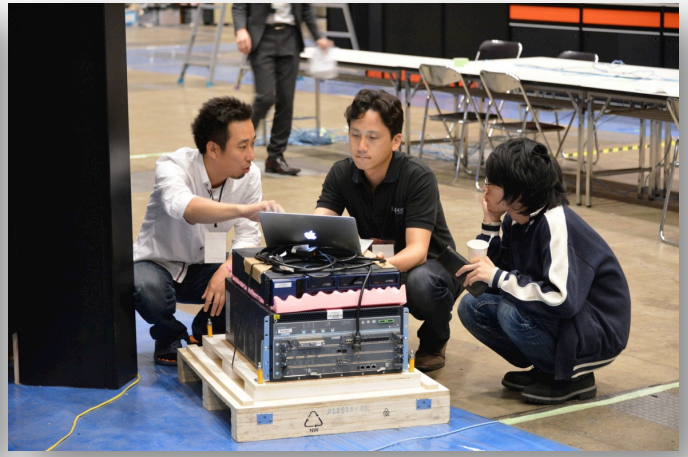
- **世界最大のネットワーク機器と技術の展示会**
  - 2016年は第23回
  - 毎年6月に幕張メッセで開催
  - 来場者数約14万人

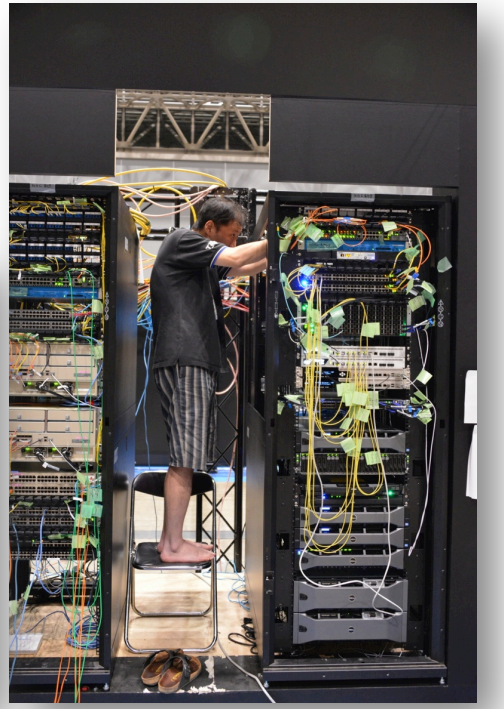


show net

Scratch and Re-build the Internet

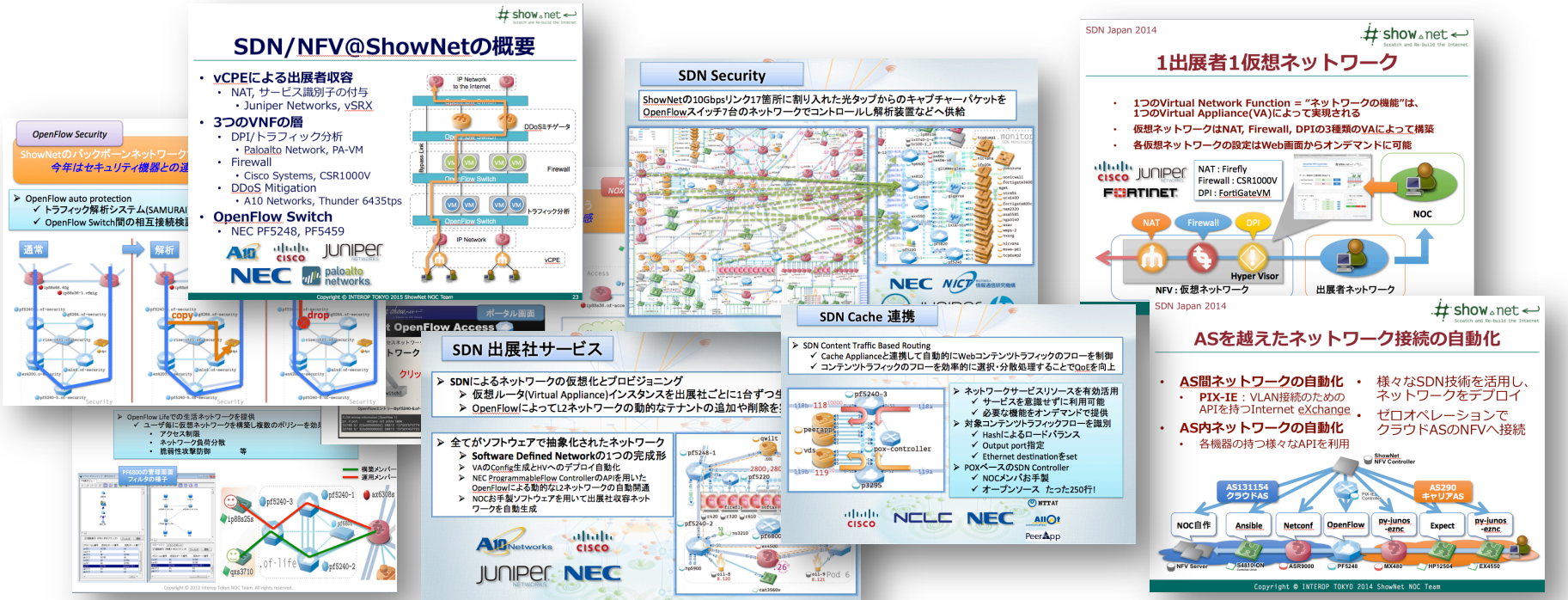
- **“I know it works because I saw it at Interop”**
  - Interopで構築される世界最大の
    デモンストレーションネットワーク
  - 最新の技術で10年先のインターネットを構築する
  - 様々な技術の相互接続生検証の場
  - 出展者や来場者へのネットワーク提供

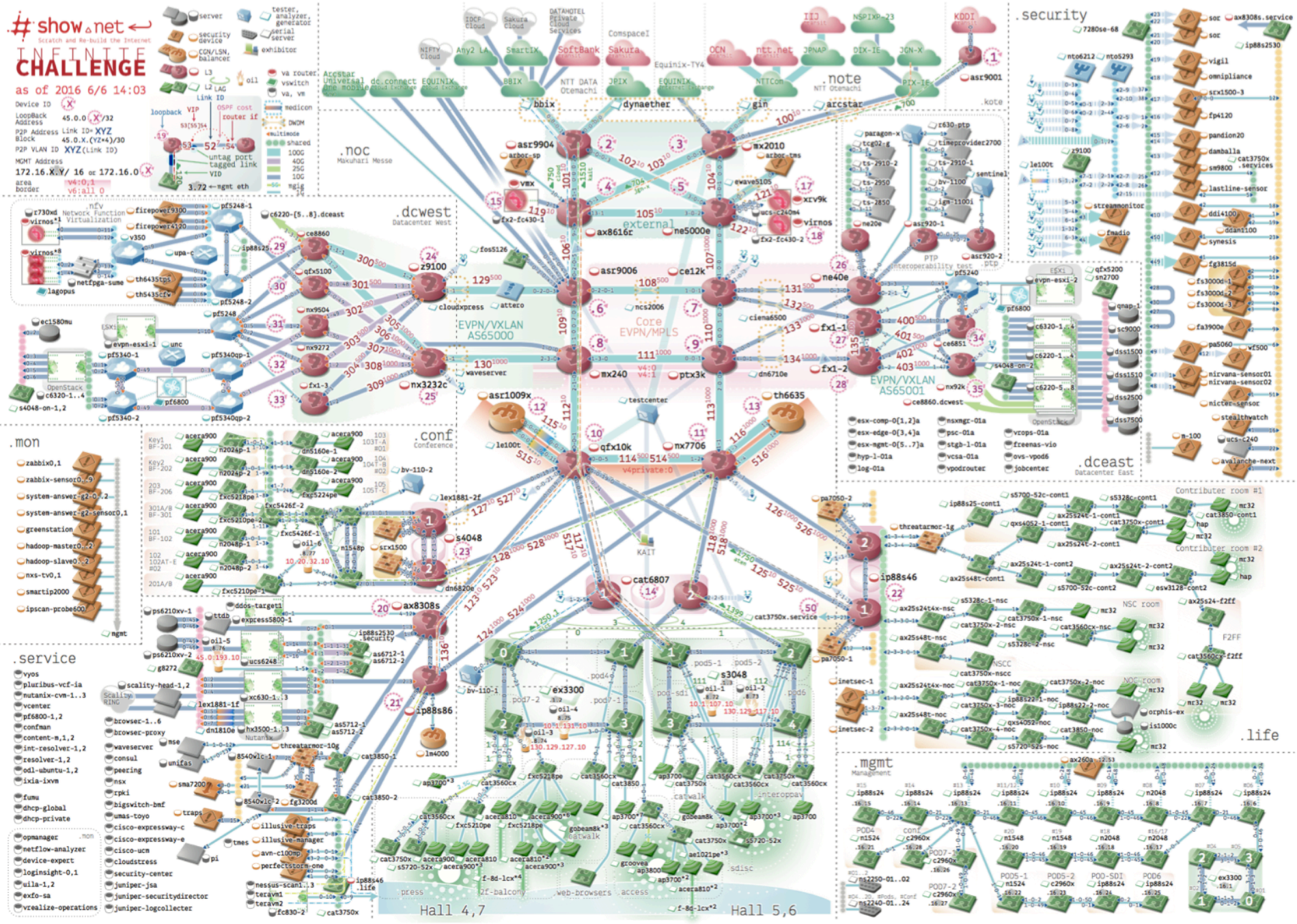


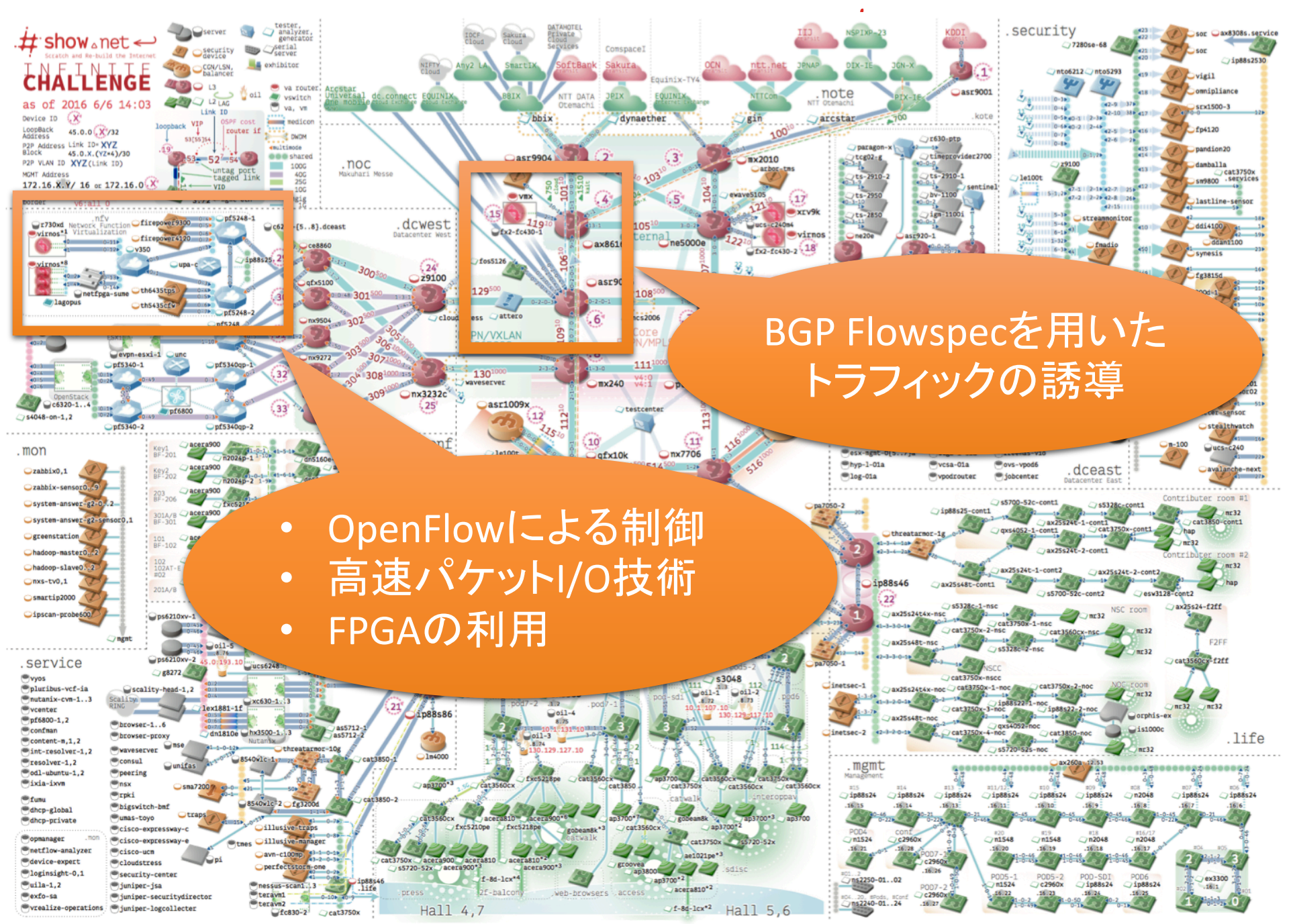


ShowNetにおけるSDN/NFV

- **ShowNetは「生きた」ネットワーク**
  - 出展者や来場者の実トラフィックを転送
  - ShowNetで求められるのは**動くSDN**と**相互接続性**
  - これらを念頭に2012年から様々な取り組みを実施

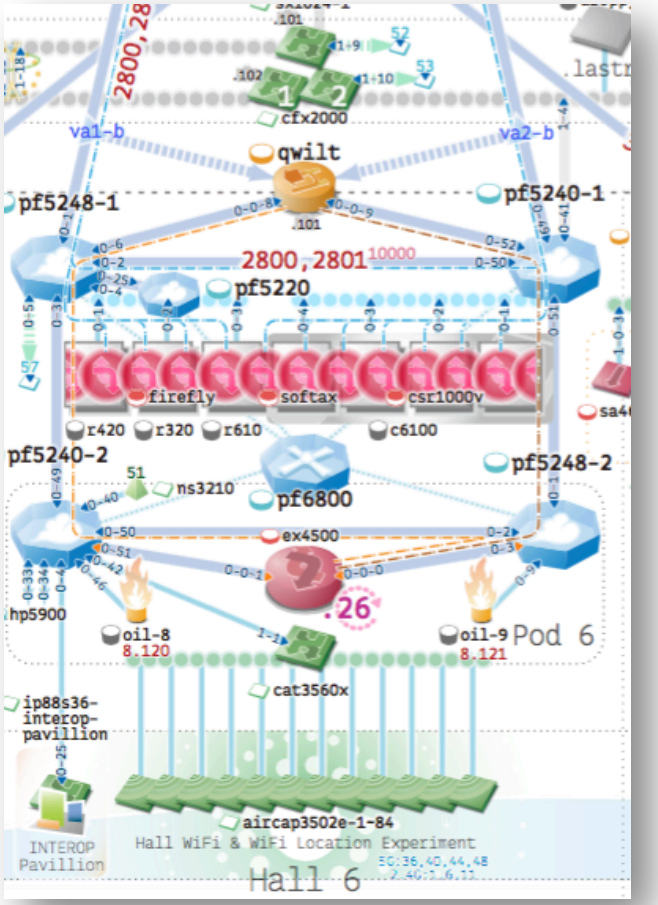






2015年までの苦しみ

- **SDN/NFVなネットワークの「島」化**
  - 特定出展者のネットワークのみをNFVで直収
  - NFV網で障害が置きたときの冗長構成は？
    - 通常のIPバックボーンにFallbackしてほしい
  - NFVの適用と除外は？
    - VLAN取り回してconfigを入れ替えるのは正直しんどい

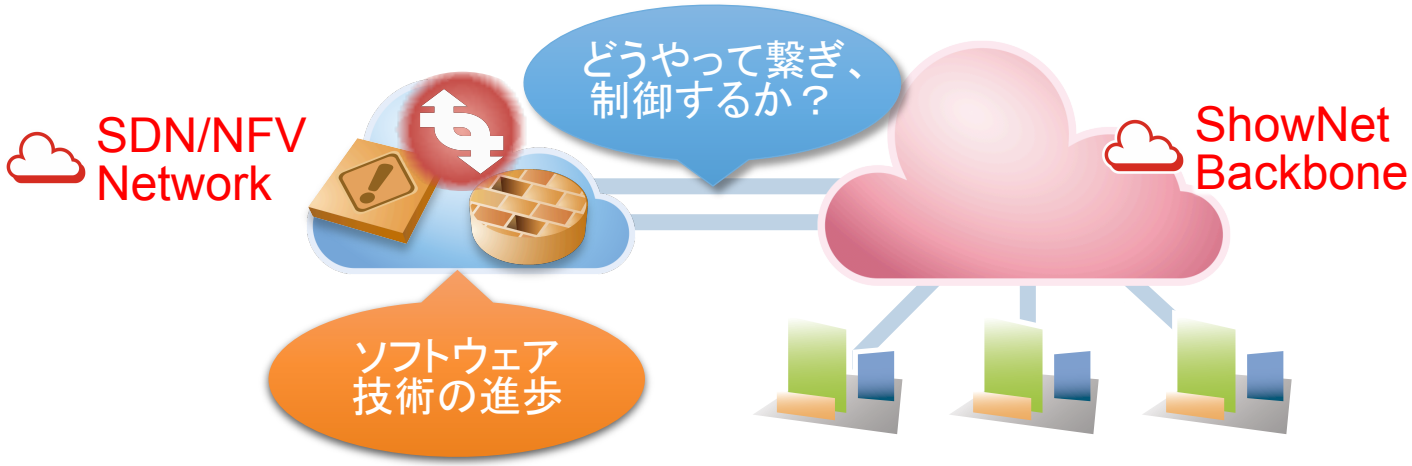


続 2015年までの苦しみ

- **属人化の果てに**
  - コントローラ/オーケストレータをNOC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ブラックボックス化
  - トラシューには通常のRoutingに加えて、OpenFlow, Software Networking, Virtual Machineの知識が必要
  - 物理仮想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の知識まで
    - (ありもののオーケストレータをもってきても同じ問題はおこるのでは？)

2016年のコンセプト: Service Chaining

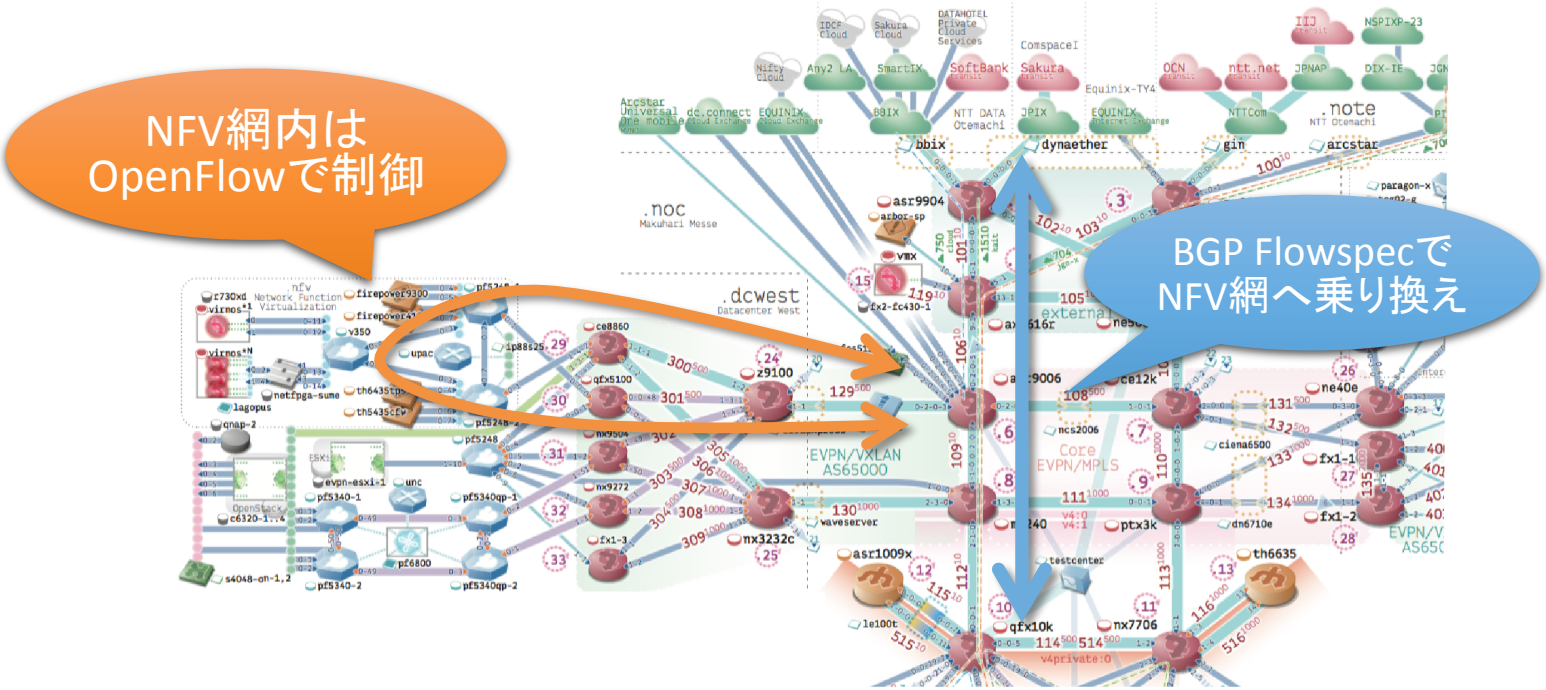
- **バックボーンネットワークの技術とSDNの融合**
  - ネットワーク全体をSDNで構築するのは正直難しい
  - どうやってバックボーンとNFV網を適切に接続するか？
- **そして、Network Functionの進化**
  - 2013年からは性能との戦いでした…
  - 目覚ましいソフトウェアパケット転送の高速化と高度化



バックボーンとNFVの融合

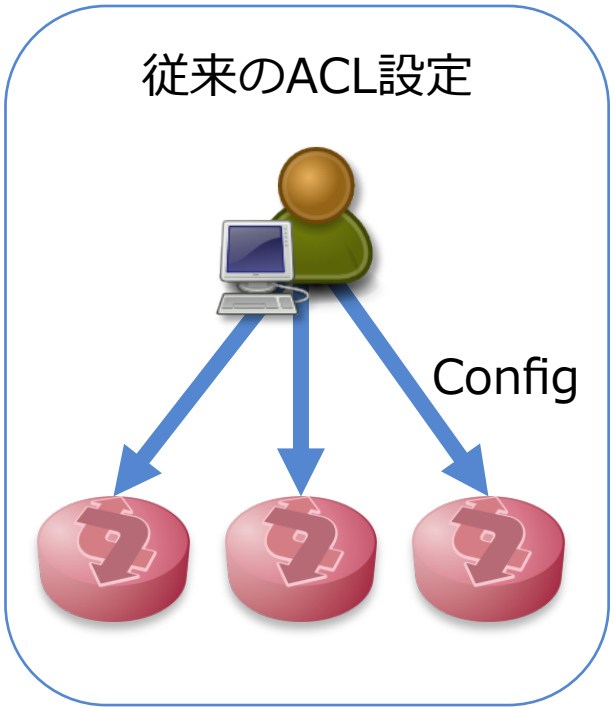
SDN/NFV@Interop Tokyo 2016 ShowNet

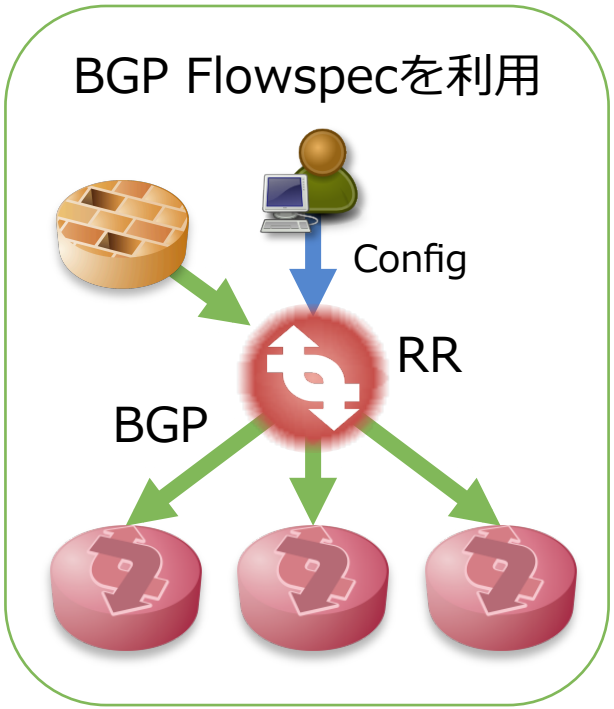
- **BGPで”乗り換え”、OpenFlowで”連結”**
  - バックボーン上で、BGP Flowspecを用いて狙ったユーザのトラフィックを**NFV網へ乗り換える**
  - Data Center内に構築したNFV網では、OpenFlowでよりきめ細かな**Functionの連結**を制御する



BGP Flow Specification

- **フィルタルールをBGPにのせて伝搬する技術**
  - RFC5575 Dissemination of Flow Specification Rules
  - BGPのNLRIを使って細かなパケットのタイプを指定し、適用するアクションを伝搬する



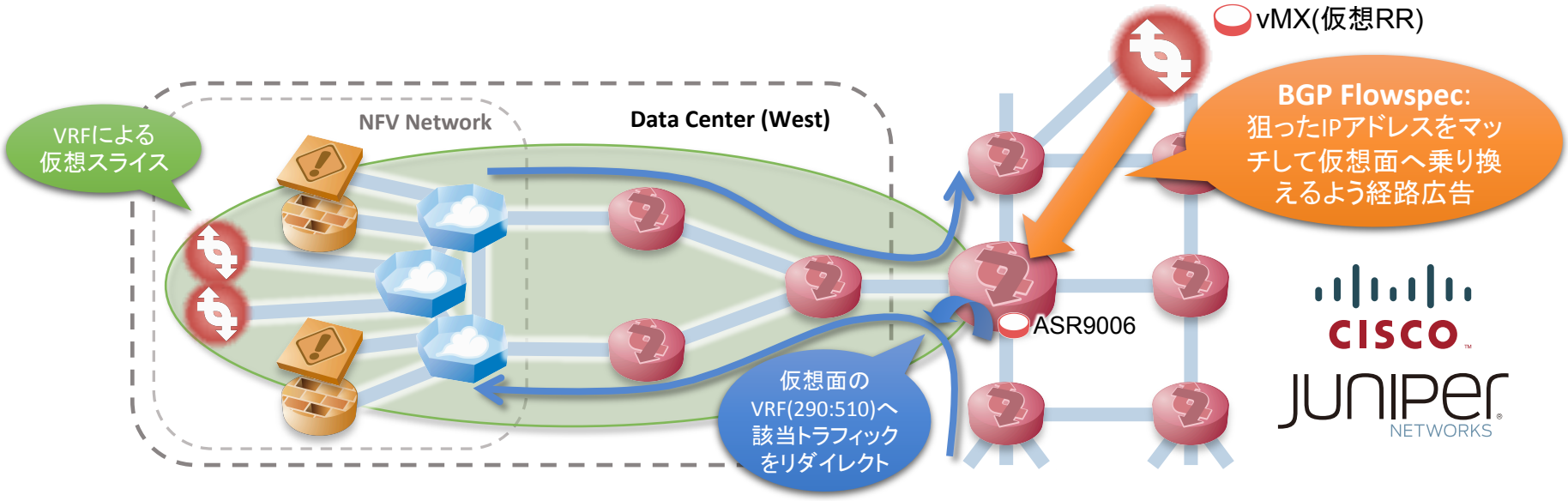


NRLI types
- **Destination Prefix**
- **Source Prefix**
- IP Protocol
- Port
- and so on

Filtering Actions
- Rate limit
- Drop
- Sampling
- **Redirect**
- Marking

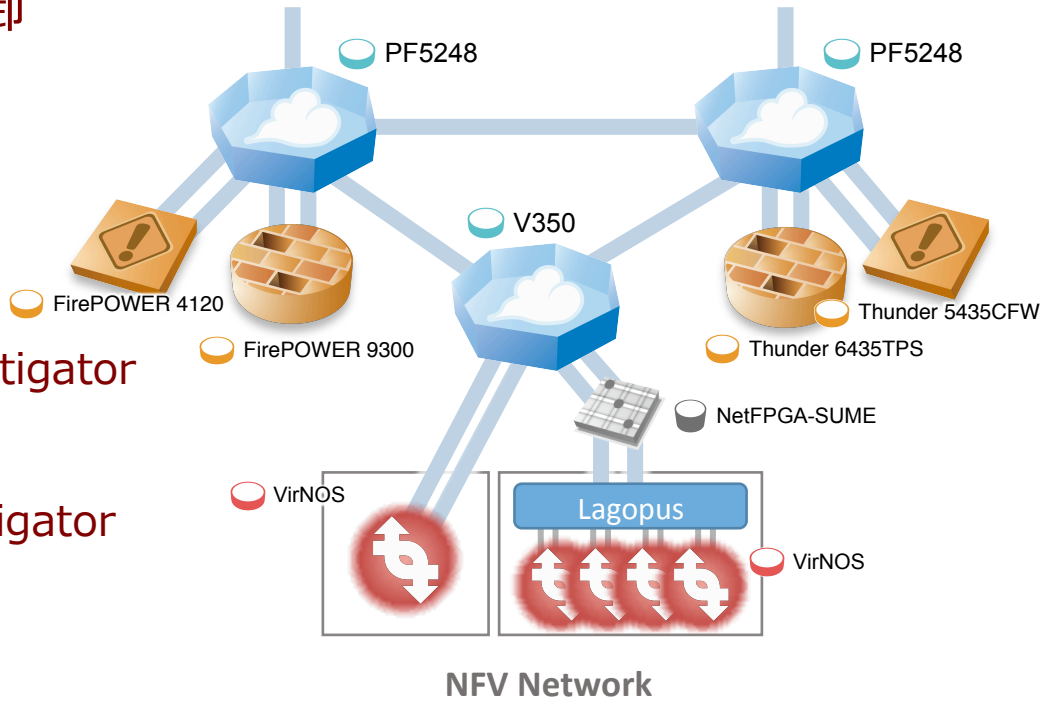
BGP Flowspecで”乗り換え”

- **狙ったトラフィックをNFV網へRedirect**
  - BGP FlowspecのActionのうち、VRF Redirectを利用
  - VRFを使って別仮想面として構築したNFVネットワークに、狙ったトラフィックをリダイレクトする
  - 好きなトラフィックを好きなときにNFVへ
    - RRでのFlowspecの設定を追加/削除するだけ



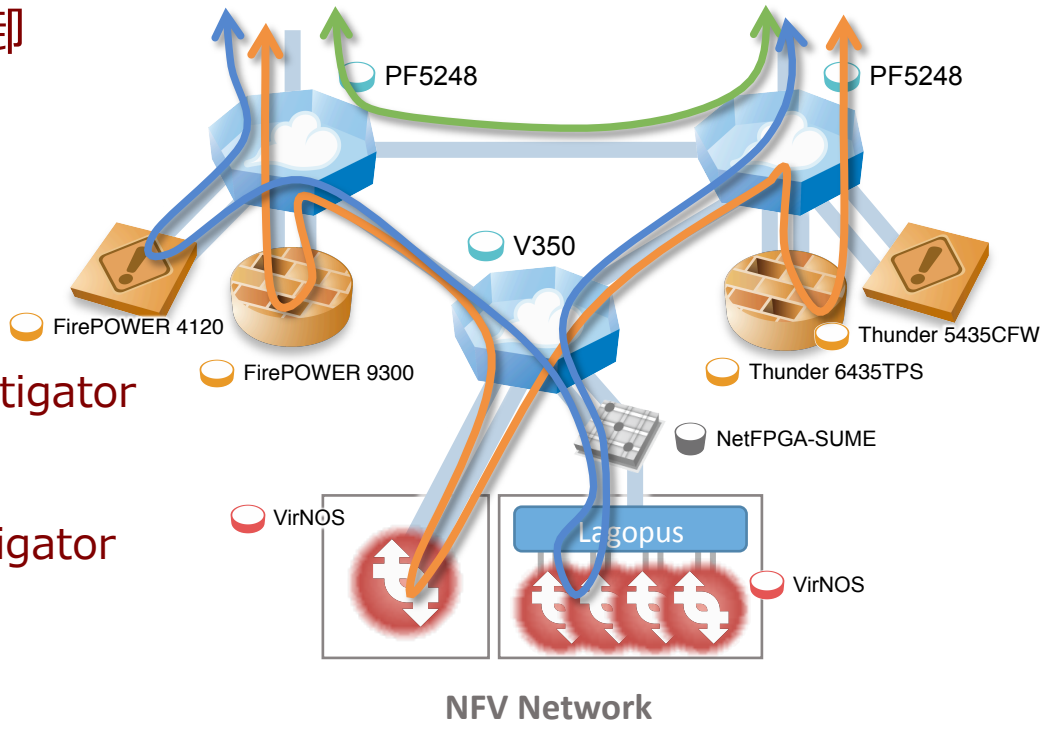
OpenFlowで"連結"

- **DC内ではOpenFlowで細かいチェインを構築**
  - **Flowspec**でネットワークを乗り換え、DC内では**OpenFlow**で制御
  - 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



OpenFlowで"連結"

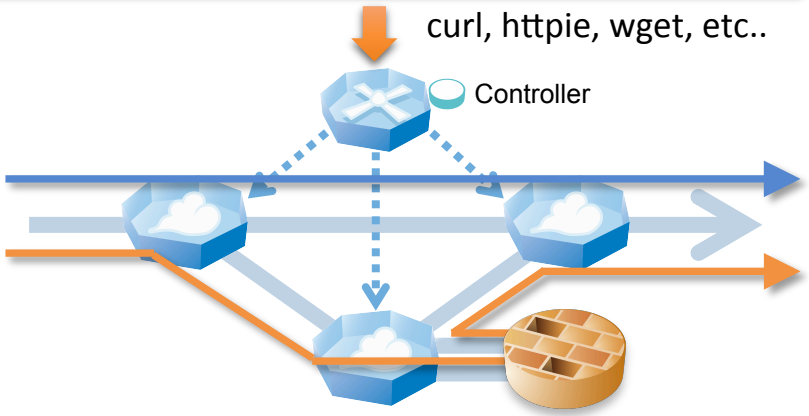
- **DC内ではOpenFlowで細かいチェインを構築**
  - **Flowspec**でネットワークを乗り換え、DC内では**OpenFlow**で制御
  - OpenFlow Switch
    - **NEC**: PF5248, **FXC**: V350
  - Network Functions
    - **A10 Network**
      Thunder 5435CFW: Firewall
      Thunder 6435TPS：DDoS Mitigator
    - **Cisco Systems**
      FirePOWER 9300：Firewall
      FirePOWER 4120：DDoS Mitigator
    - **IP Infusion**
      VirNOS：ACL-based Firewall
      and Lagopus



"操作"のインターフェース

- 慣れ親しんだ/分かり易い手法の利用
  - BGP Flowspecの利用
    - FlowspecはRouting(BGP)の一部
    - 設定、伝搬、FallbackはBGPに従う
  - OpenFlowではREST
    - なるべくシンプルなURLで
    - 慣れ親しんだコマンド群で投入

http://172.16.x.x/install/45.0.y.y/24/tps-fp9

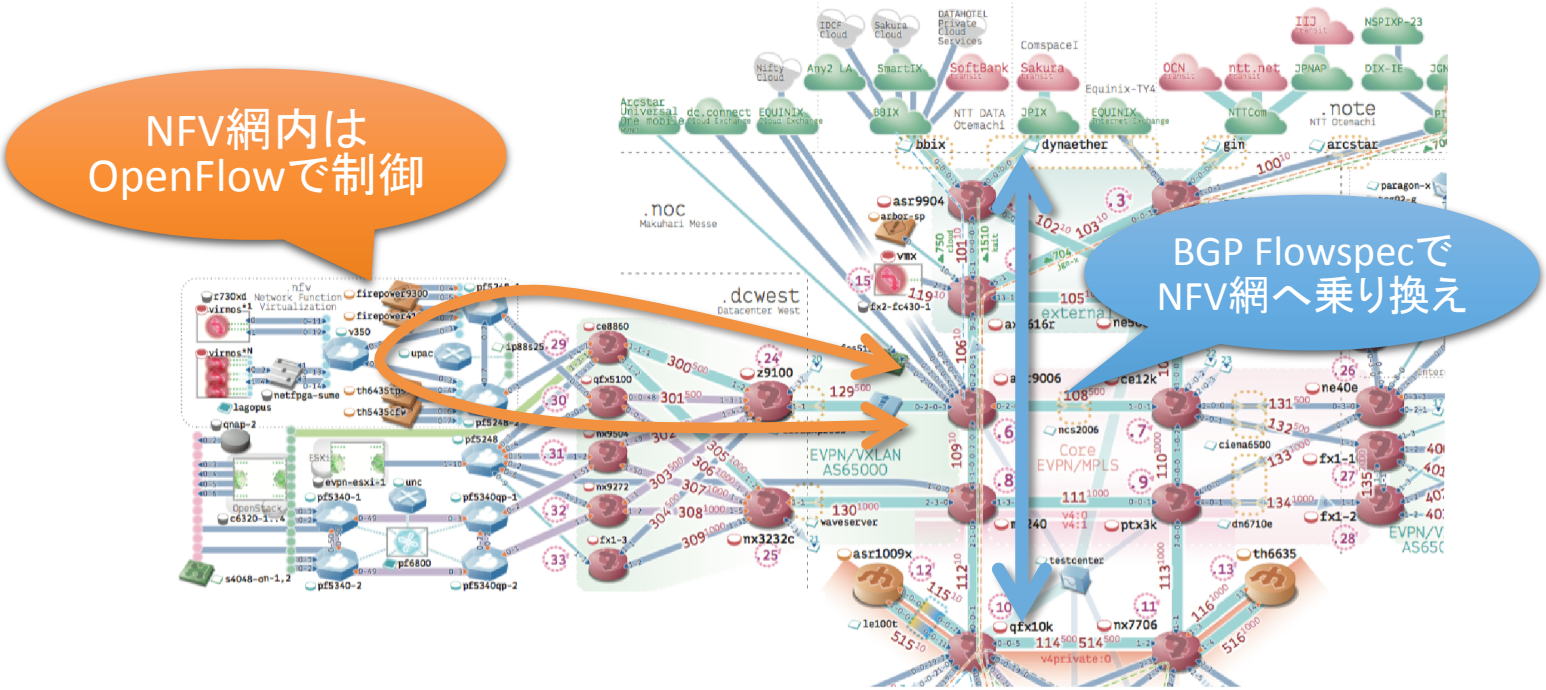


Juniper vMXでのFlowspec設定

```
routing-options {
    flow {
        route nfv-tester-downlink {
            then {
                community flowspec-nfv-downlink;
                routing-instance target:290:510;
            }
            match source 45.0.239.1/24;
        }
        route nfv-tester-uplink {
            then {
                community flowspec-nfv-uplink;
                routing-instance target:290:512;
            }
            match destination 45.0.239.1/24;
        }
        route nfv-4A02-2162-downlink {
            then {
                community flowspec-nfv-downlink;
                routing-instance target:290:510;
            }
            match source 130.129.162.0/27;
        }
        route nfv-4A02-2162-uplink {
            then {
                community flowspec-nfv-uplink;
                routing-instance target:290:512;
            }
            match destination 130.129.162.0/27;
        }
        route nfv-6A18-2091-downlink {
            then {
                community flowspec-nfv-downlink;
                routing-instance target:290:510;
            }
            match source 130.129.255.91/32;
        }
```

トラフィック制御の適材適所

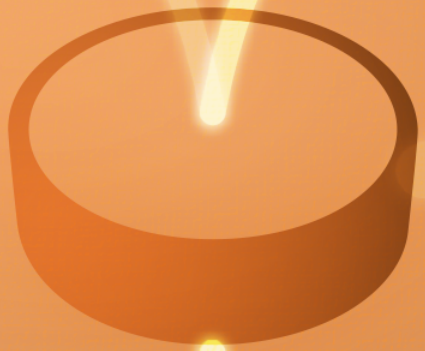
- **BGP Flowspecで乗り換え、OpenFlowで連結**
  - BGPというバックボーンにおけるトラフィック制御を取り入れることで、SDNに特化した製品を使うことなく狙ったトラフィックを柔軟に誘導可能
  - DC内でのみOpenFlowを使ったチェインの制御を行う



やってみてわかったこと

- **IP Routingは偉大**
  - Shortest Path Forwarding
  - 枯れた経路制御技術(e.g., OSPF, BGP)
- **経路制御の適材適所とその接続方法**
  - バックボーンはバックボーンの経路制御を用い、細かいところにはSDN的なアプローチを用いる
  - 接続は、セグメントを繋ぐだけ、とは限らない
    - VRF Redirectはひとつの解になるのでは
  - 任意のトラフィックを任意のタイミングでNFVへ
- **スキルセットに合わせたインターフェースを**
  - 今までと違う操作手法を突然強制するのは難しい
  - ユーザインターフェースの設計も大切

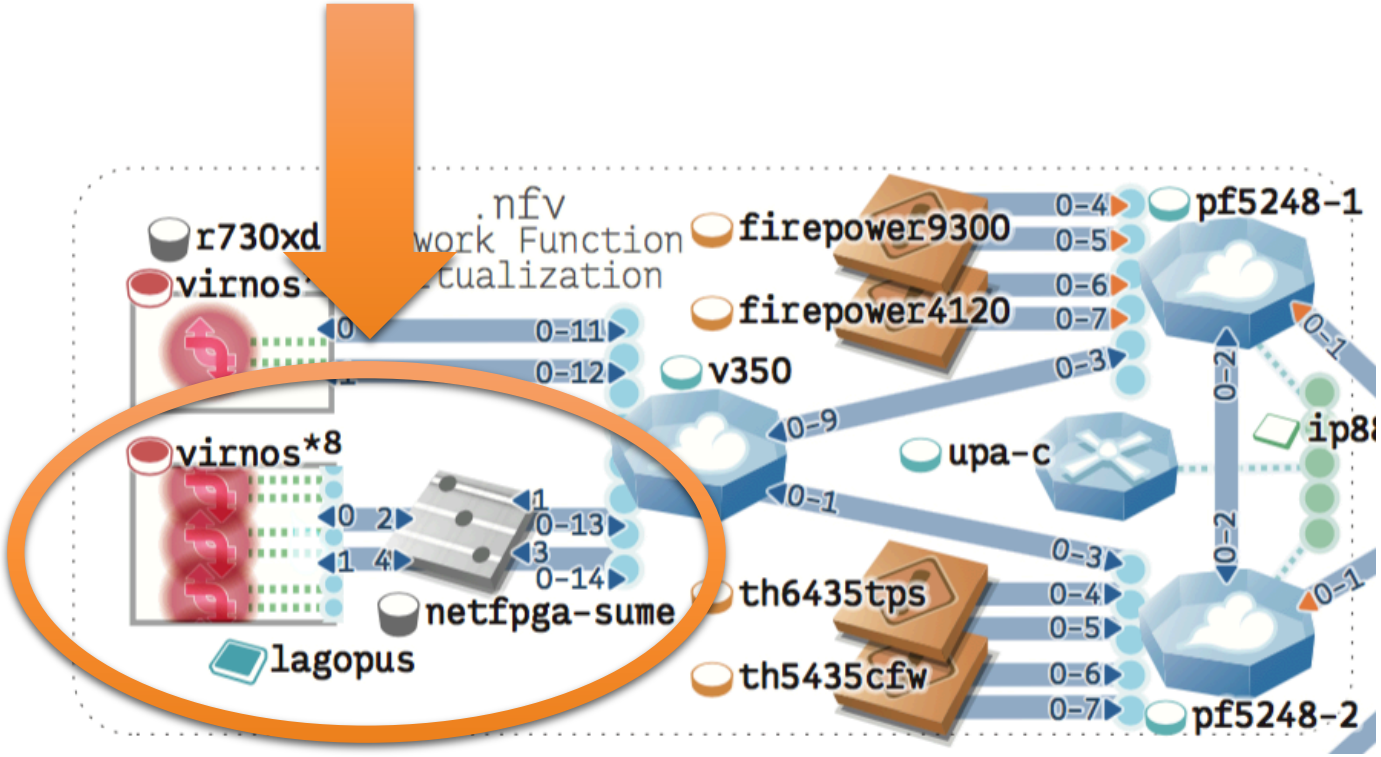
show△net
Scratch and Re-build the Internet



Network Functionの進化

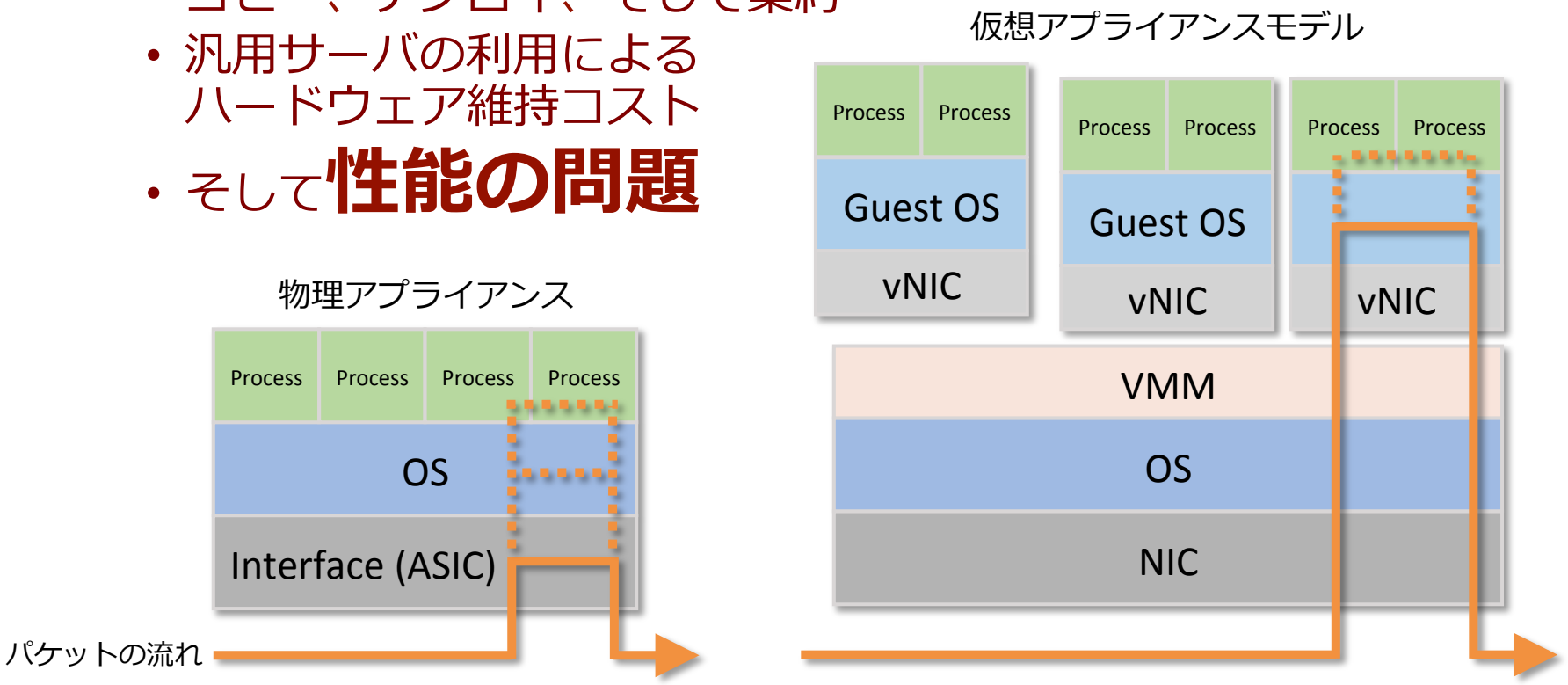
Network Function

- 実際にサービスを担う、Network Function
  - うって変わって細かい話です



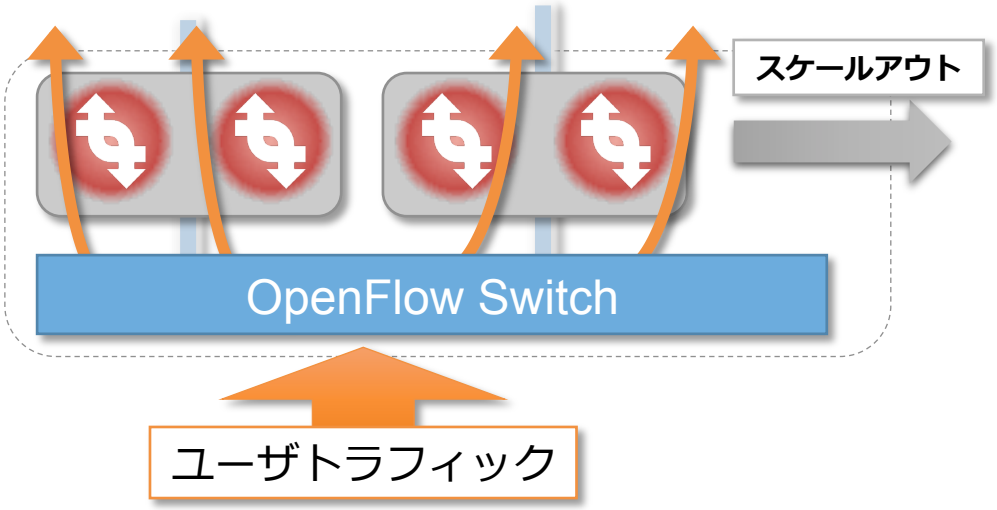
仮想アプライアンス

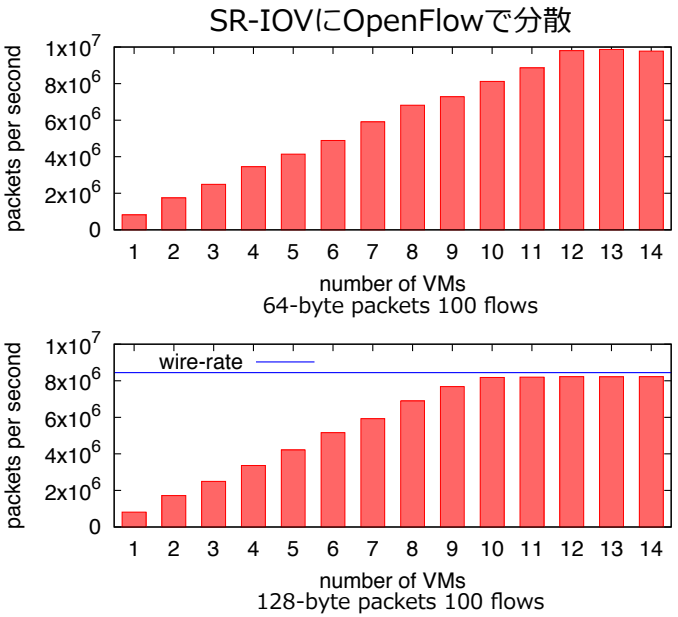
- **仮想マシンとして動作するMiddlebox**
  - Network Function Virtualizationの要
  - ソフトウェア特有の利点と欠点
    - コピー、デプロイ、そして集約
    - 汎用サーバの利用による ハードウェア維持コスト
    - そして**性能の問題**



ShowNetでの作戦: 複数同一VMによるスケールアウト

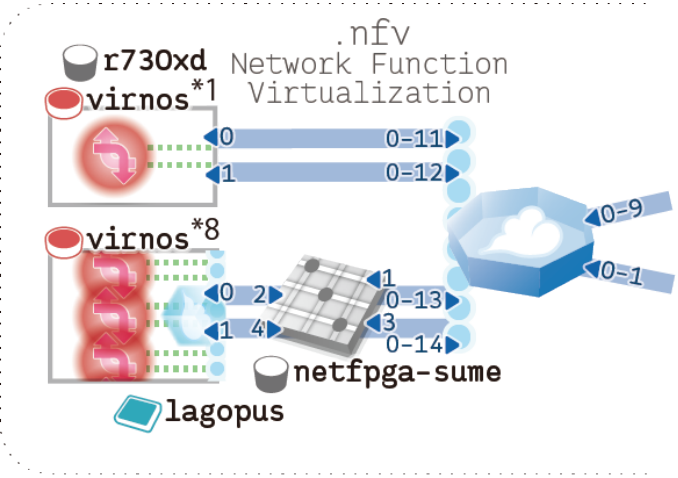
- **複数の仮想アプライアンスを並列に並べる**
  - 1台のVMの性能では不十分
  - 複数の同一のVMを横に並べ、スケールアウトを目指す
  - OpenFlowなどを用いてトラフィックを分散
  - 詳細はSDN Japan 2015での発表資料を御覧ください
    - **ハードウェアの支援**(SR-IOV)を有効活用
    - 一方で**柔軟性が欠如**



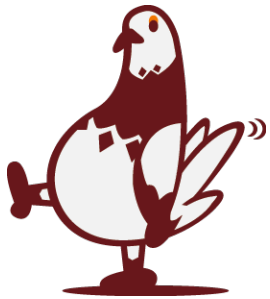


ソフトウェアパケット転送技術の進化

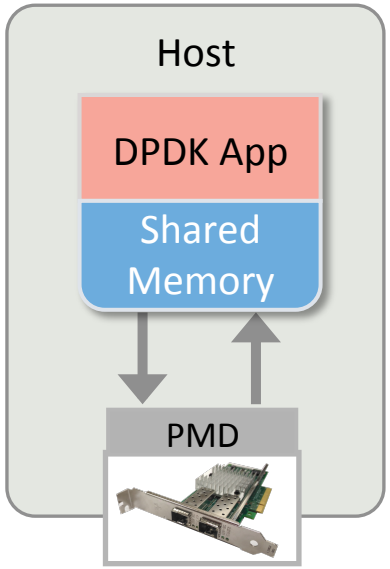
- **高速パケットI/O技術を用いたソフトウェア**
  - Intel DPDKに代表される、Poll Mode Driver、パケットのバッチ処理、そしてカーネルバイパス
  - DPDKを使った高速な仮想スイッチと仮想ルータの登場
    - **VirNOS**: 仮想ルータ
    - **Lagopus**: 仮想OpenFlowスイッチ



ipinfusion™



DPDKアプリの例

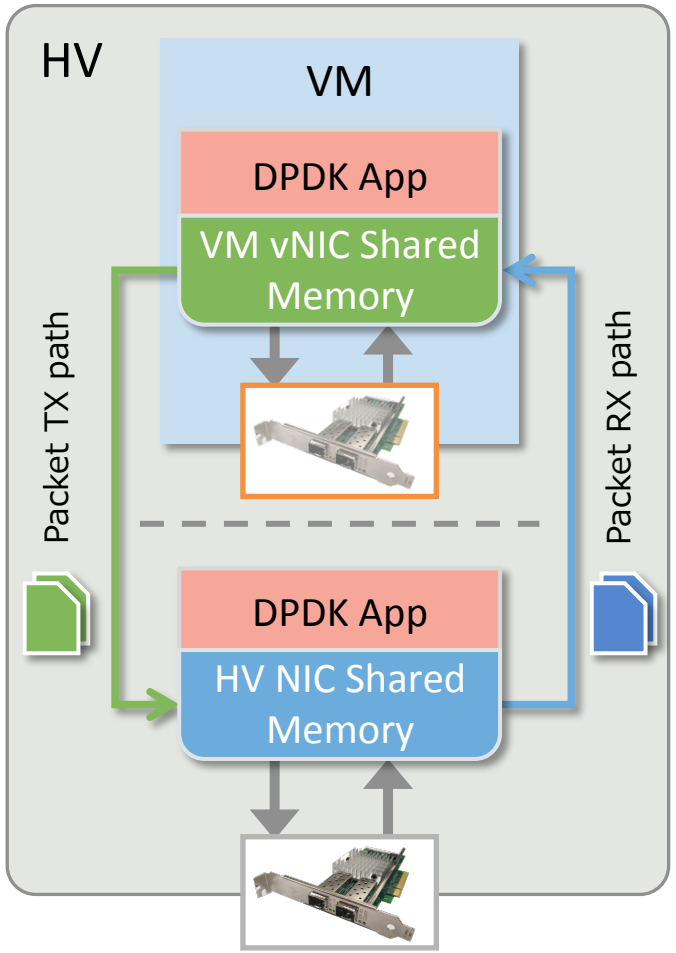


vhost-userとDPDKによる高速通信

- **vhostのユーザランドバックエンド**
  - DPDKアプリはユーザランドで動作するアプリケーション
  - NICを専用ドライバ(PMD)でつかみカーネルをバイパス
  - vhostはHV/VM間のメモリ共有
  - vhost-userはHV側のメモリ空間をカーネルをスキップしてユーザランドアプリに共有

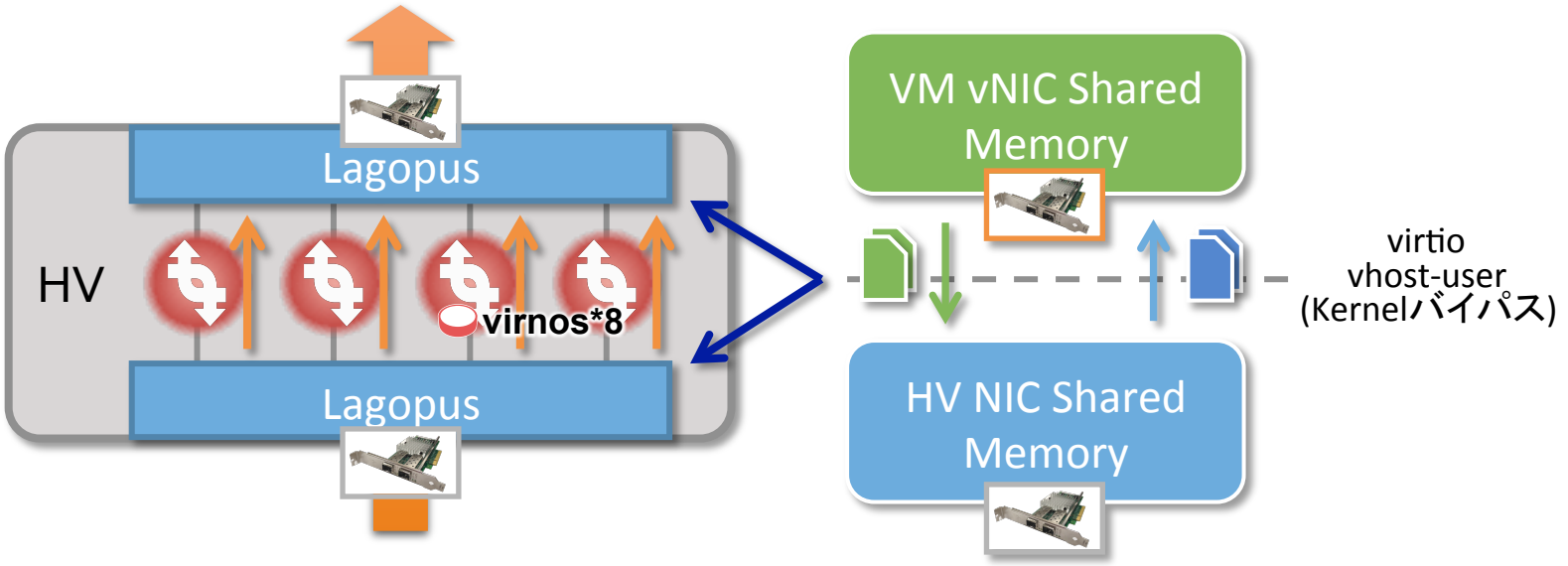
HVのShared Memory空間とVMのShared Memory空間で直接パケットをやり取りする

vhost-userを使ったHV/VMデータパス



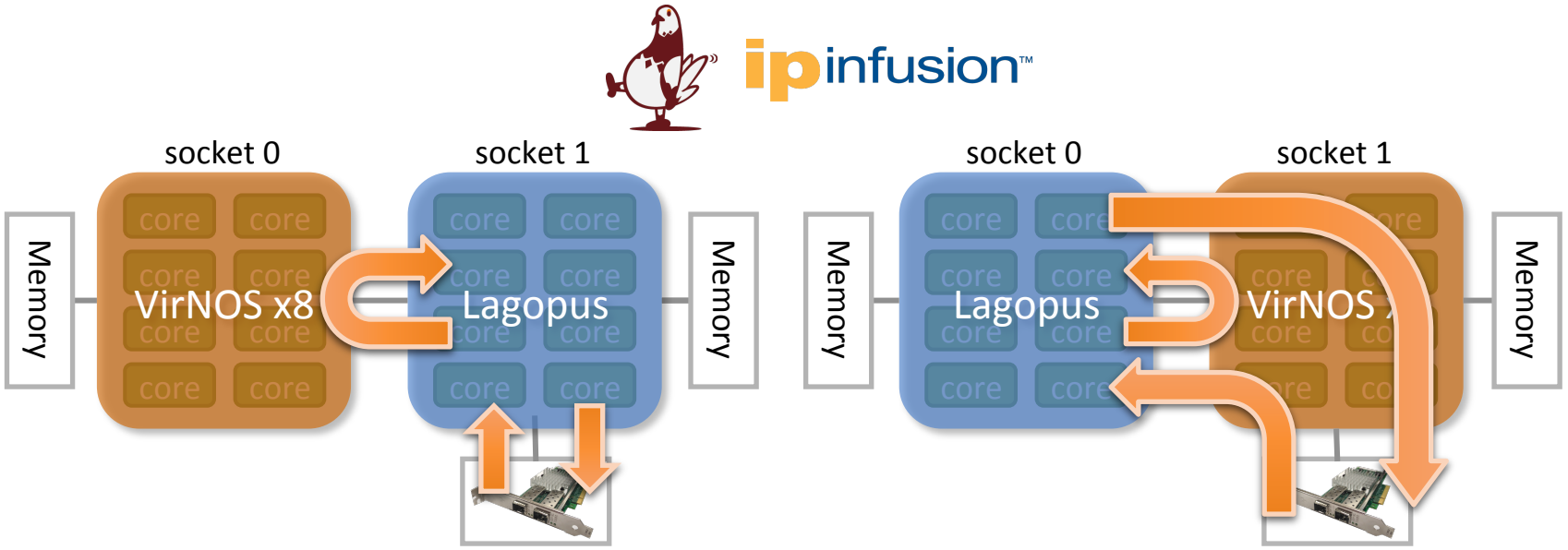
LagopusスイッチとVirNOS

- **HV上の複数のVirNOSへ、Lagopusで転送**
  - DPDKなVMと仮想スイッチ間で**vhost-user**を利用
  - HV側の仮想スイッチとVM側のドライバがshared memoryを介して直接パケットのやり取りを行う
  - →現状最速(?)のHV/VM間通信
  - →VM+SWスイッチで10Gbpsは出せる



ソフトウェアパケット転送における考慮点

- **コンピュータアーキテクチャへの理解が必要**
  - すでにマルチコア、マルチプロセッサは当たり前
  - プロセスやワーカーのCPUへの割り付け方
  - NICが接続されたCPUとVMが動作するCPU
  - 割り込みやロック、そしてパケットの通るパス



OpenFlowについて再考

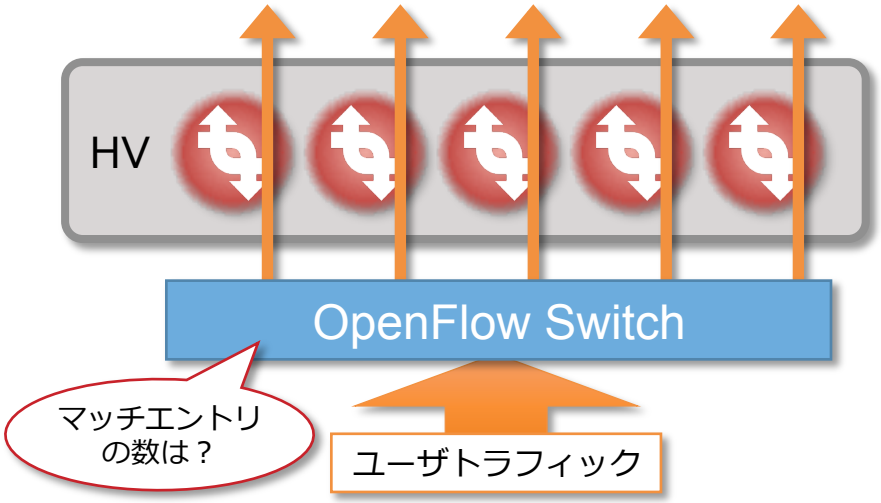
- **OpenFlowはマッチとアクションの組み合わせ**
  - マッチするヘッダフィールドの増加
    - VXLAN, NVGRE, STTなどトンネルプロトコルの増加
  - 制御の粒度とエントリ数のトレードオフ
    - 5タプルを全てマッチすると32+32+16+16+1 (TCP/UDP) = 2の97乗 ものエントリを消費する可能性がある

Version	Date	Header Fields
OF 1.0	Dec 2009	12 fields (Ethernet, TCP/IPv4)
OF 1.1	Feb 2011	15 fields (MPLS, inter-table metadata)
OF 1.2	Dec 2011	36 fields (ARP, ICMP, IPv6, etc.)
OF 1.3	Jun 2012	40 fields
OF 1.4	Oct 2013	41 fields

Table 1: Fields recognized by the OpenFlow standard

From “P4: Programming Protocol-Independent Packet Processors”, P. Bosshart et.al, ACM SIGCOMM CCR, July 2014

OpenFlowを用いた分散@ShowNet



プログラマブルなハードウェアの利用

- **さらなる未来: OpenFlowの次は？**
  - OpenFlowは**マッチ**と**アクション**、**計算**はできない
  - ハードウェアの挙動さえプログラミングしたい！
- **NetFPGA-SUMEを使ったデモンストレーション**
  1. パケットのハッシュ値の計算をFPGAで行う
  2. その計算結果を何らかのフィールドに埋め込む
  3. OpenFlowはその結果だけを見て転送先を決定する

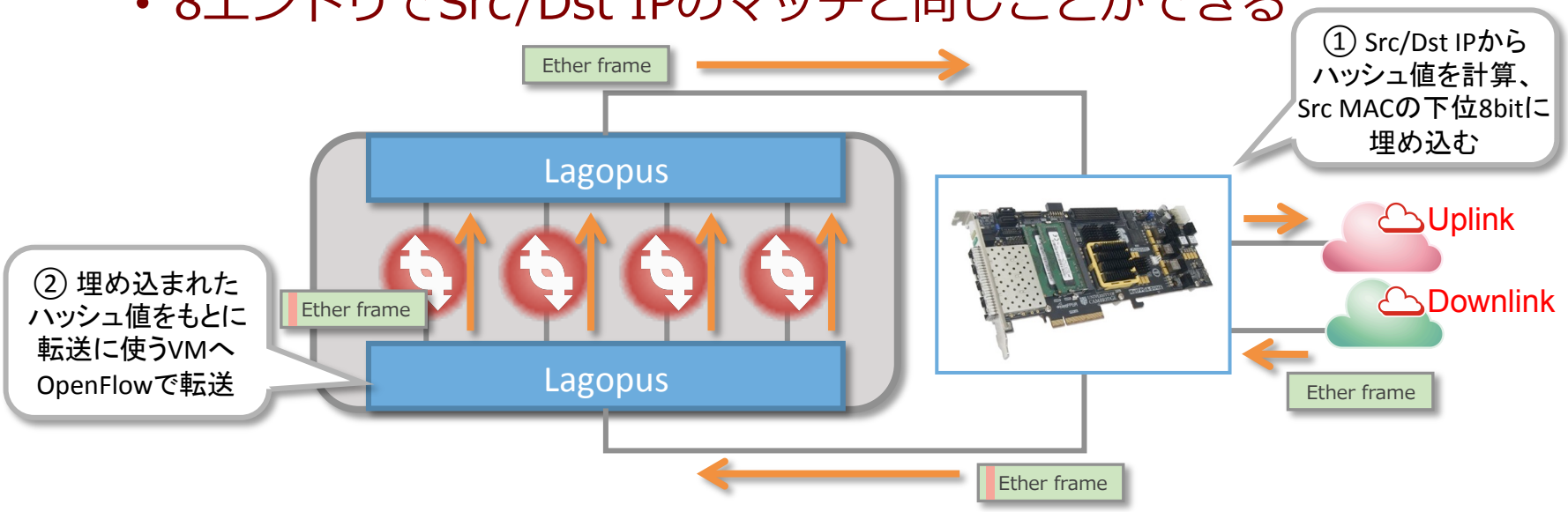
むき出しのNetFPGA-SUMEと、
LagopusとVirNOSの動作するx86サーバ

DIGILENT® A National Instruments Company

NetFPGA

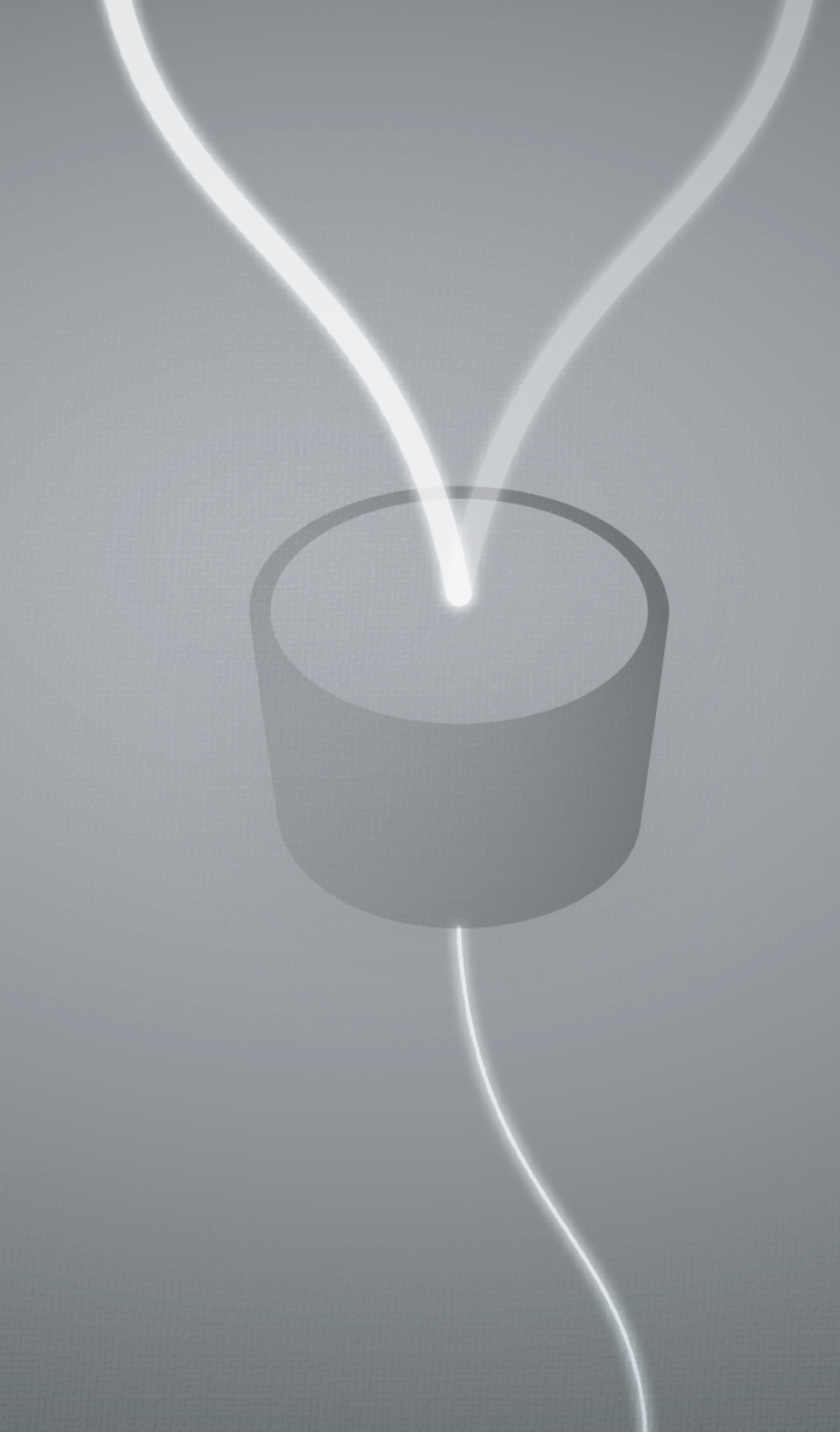
FPGAを使ったデモンストレーション

- **パケットはFPGA→Lagopus→VirNOSと通る**
  1. NetFPGAでパケットの送信元と宛先IPアドレスからハッシュを計算し、送信元MACの下位8bitに埋め込む
  2. Lagopusは送信元MACアドレスをマッチしてトラフィックを複数のVirNOSへ分散
     - 実際にはマスクを使って下位3bitを使用 (8台のVirNOS)
     - 8エントリでSrc/Dst IPのマッチと同じことができる



やってみてわかったこと

- **ソフトウェアによるパケット処理の進化**
  - 進歩の目覚ましい分野
  - 10Gbpsは出せる。そして100Gへ...
  - ただし、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に立ち返る必要がある
- **ハードウェアにおけるプログラマビリティ**
  - OpenFlowによるControl/Data planeの分離
  - ハードウェアの柔軟性は今後さらに重要になるのでは
    - P4: Programming Protocol-Independent Packet Processorに期待
- **対象と問題に合わせて適切な技術/手法の選択を**
  - ボトルネックはどこなのか
  - なにを、どうやって、高速化する技術なのか
  - スケールアウトなのか、スケールアップなのか

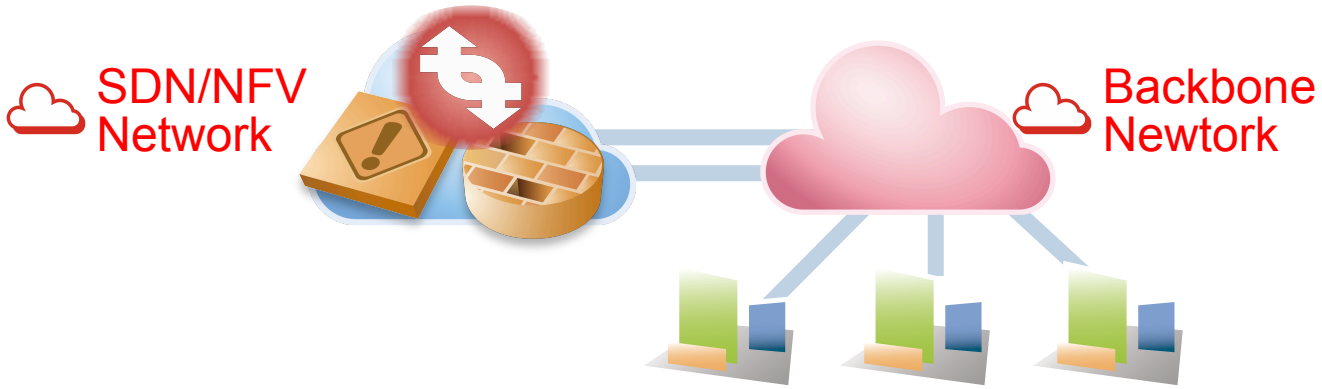


show△net
Scratch and Re-build the Internet

まとめ

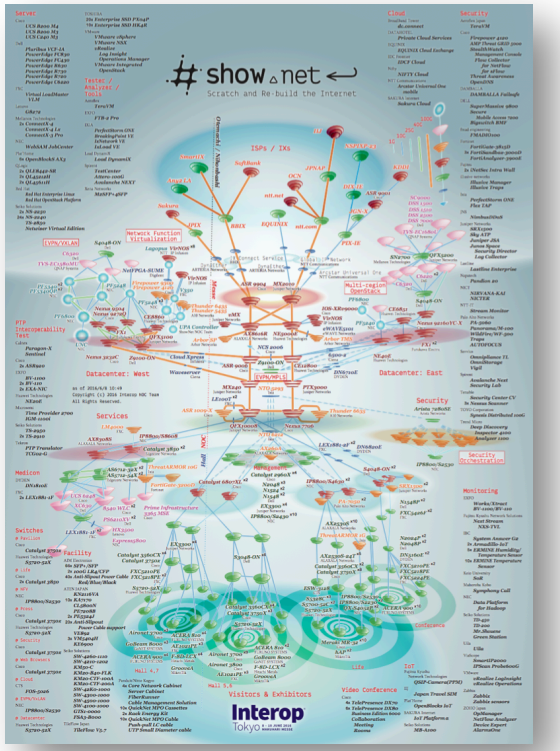
SDN/NFV@Interop Tokyo 2016 ShowNet

- **技術の適材適所で作るSDN/NFV**
  - バックボーンにはバックボーンの技術を
  - より細かい制御にはSDNな技術を
- **Network Functionの進化**
  - ソフトウェアによるパケット処理の高速化
  - コンピュータアキーテクチャへの理解
  - よりプログラマブルなハードウェアへの期待



Interop Tokyo 2016 ShowNet

- **未来のネットワークの1つのカタチ**
  - 10年先のインターネットをつくる
  - その1つのモデルとしての、SDNとNFVの利用
- **Live Network**
  - ShowNetは生きたネットワーク
  - 実際に動くSDNとNFV
- **相互接続性**
  - ひとつひとつは個別の技術
  - オープンな技術の上に成り立つ組み合わせの自由度
  - そして検証とフィードバック




INFINITE
CHALLENGE
show△net