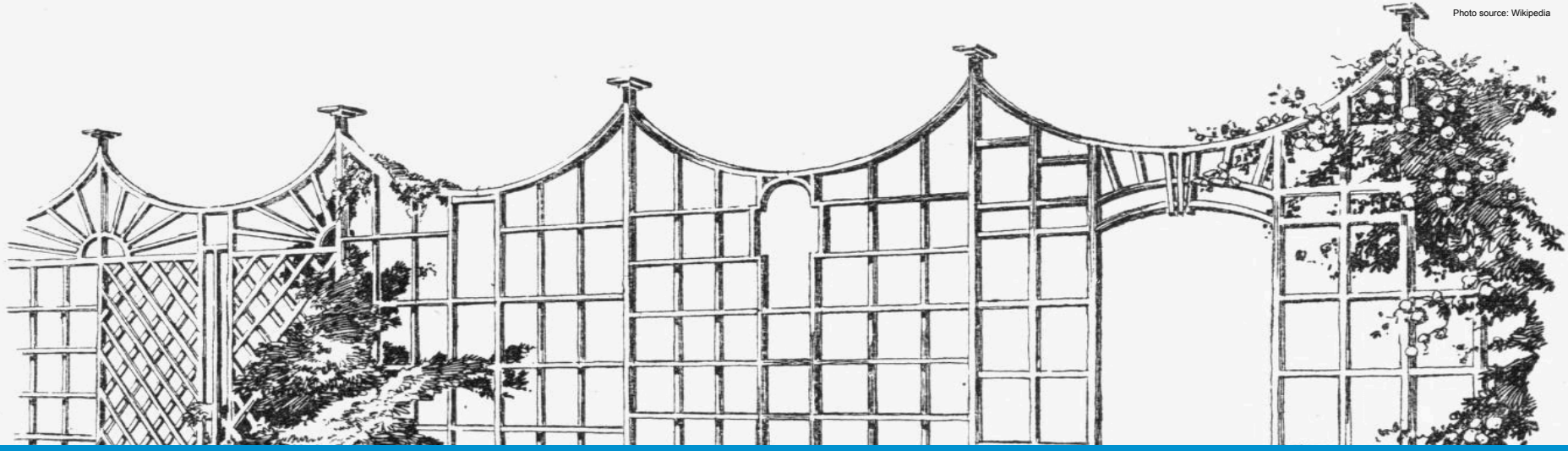


TRELLIS

CORD Network Infrastructure

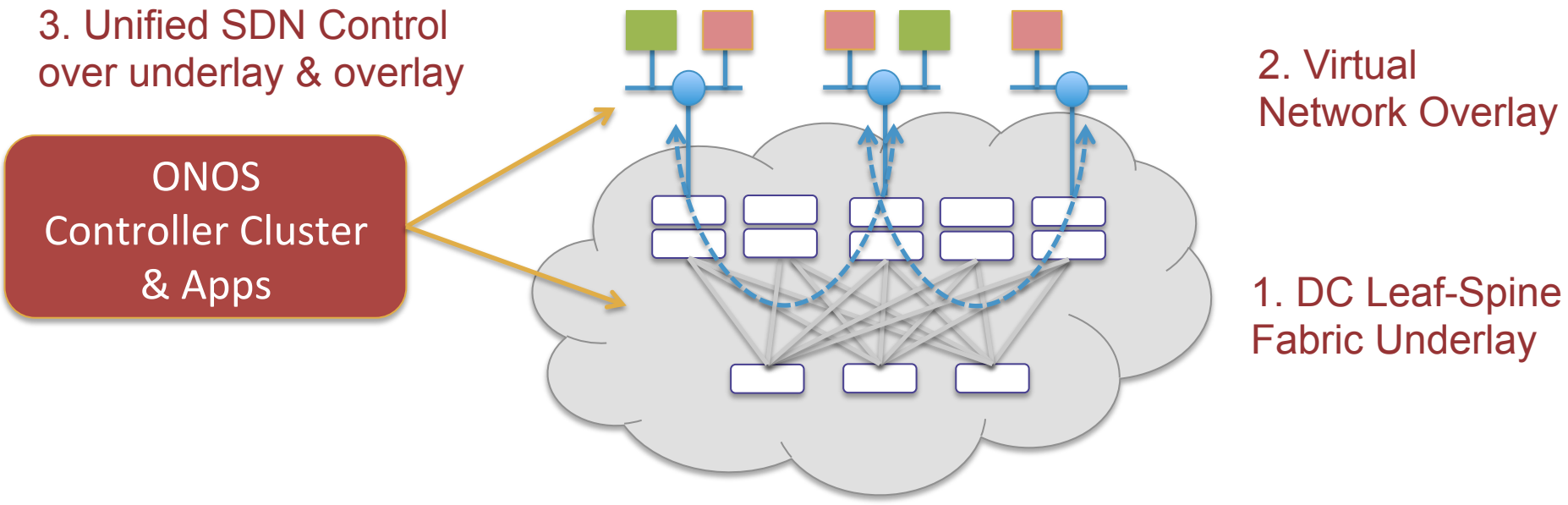
Saurav Das

… with Ali Al-Shabibi, Jonathan Hart, Hyunsun Moon, Charles Chan & Flavio Castro

#OpenCORD

CORD
Central Office Re-architected as a Datacenter

What is Trellis?



Unique combination of these three components

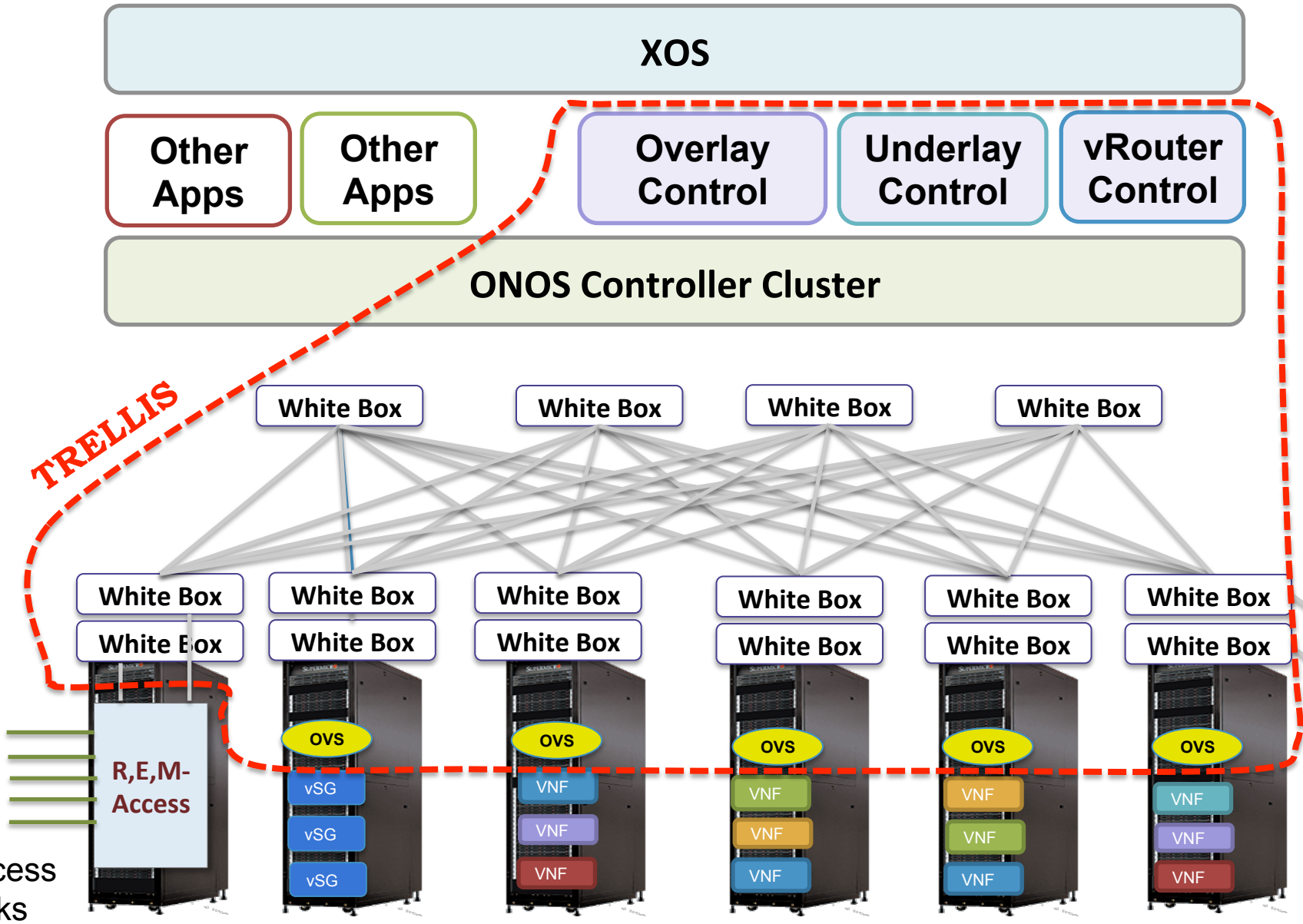
Trellis Benefits

Common control over underlay & overlay networks enable simple, efficient impls of:

- Distributed Virtual Routing for tenant networks
- Optimized delivery of multicast traffic streams
- and many more optimizations & new capabilities to be introduced in the near future

Trellis is the enabling Network Infrastructure for CORD

CORD Architecture



Outline

Underlay Fabric

- Bare-metal + Open-source = White-Box
- L2/L3 Leaf-Spine with SDN Control

Virtual Network Overlay

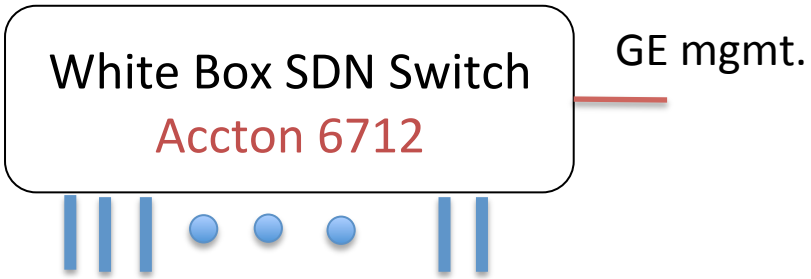
- OVS and VXLAN with SDN Control
- Service chaining

vRouter

- Distributed Virtual Routing
- Multicast handling

Underlay Fabric: Open Hardware & Software Stacks

Spine Switch



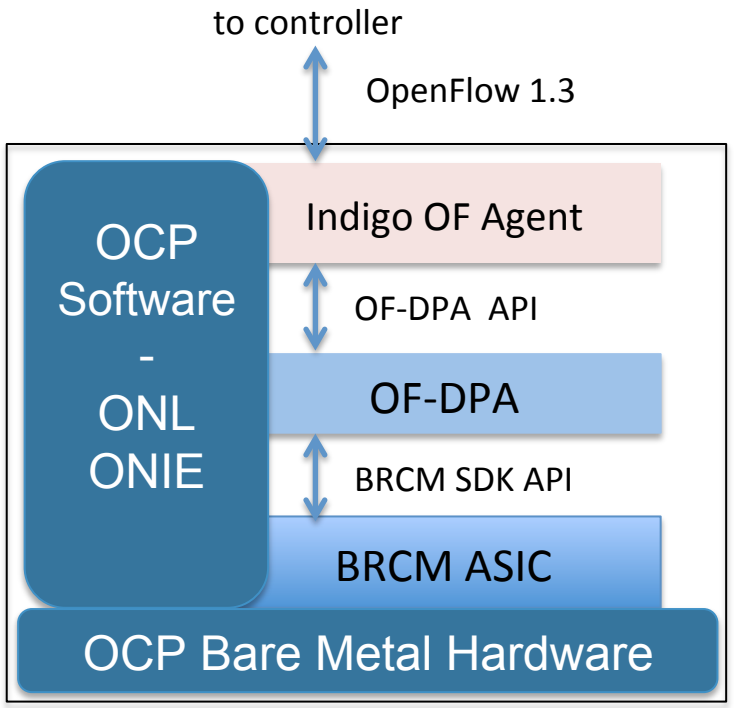
32 x 40G ports downlink to leaf switches
40G QSFP+/DAC

Leaf Switch

8 x 40G ports uplink to different spine switches
ECMP across all uplink ports

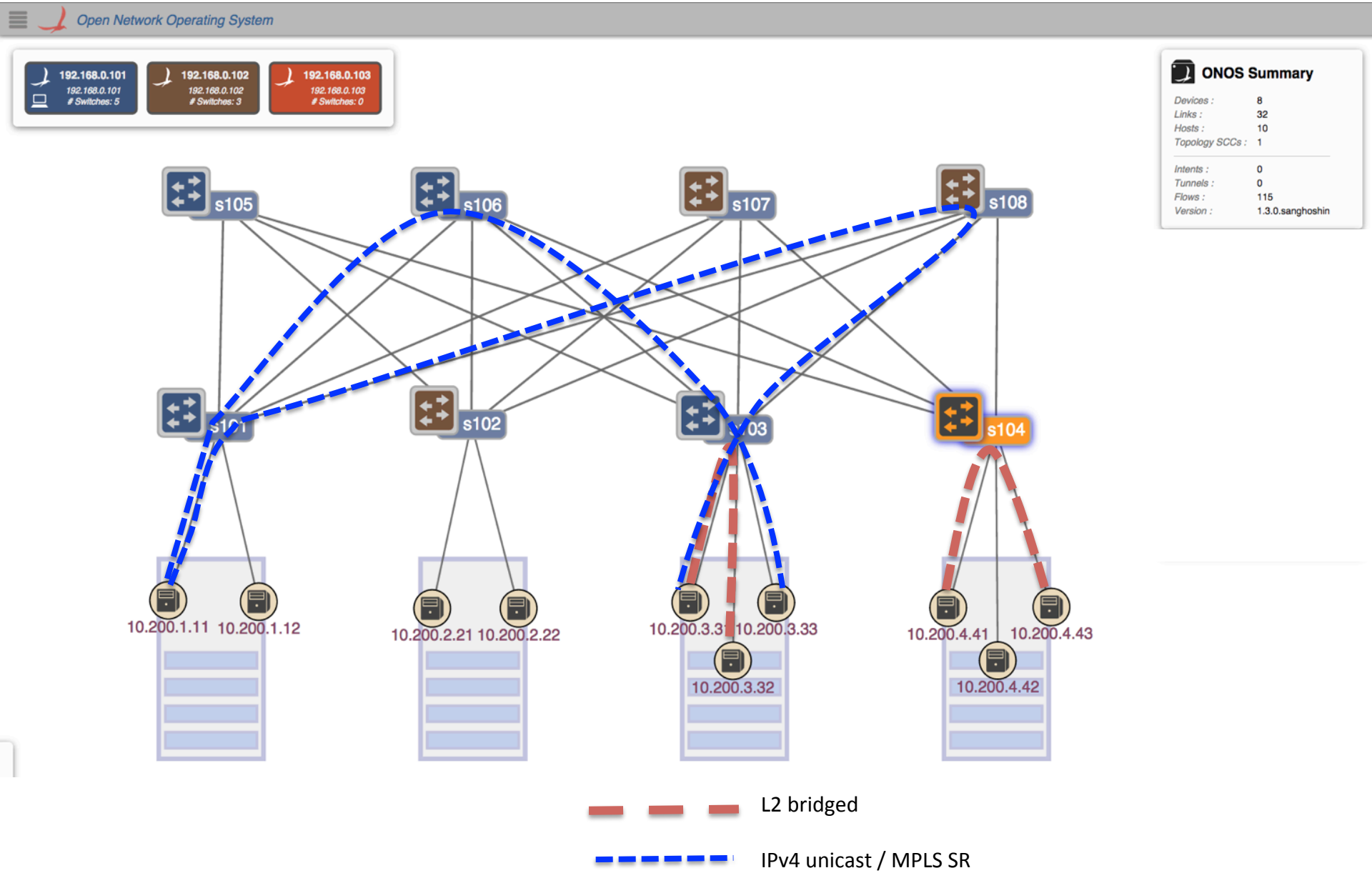
24 x 40G ports downlink to servers
and OLTs

Leaf/Spine Switch Software Stack



OCP: Open Compute Project
ONL: Open Network Linux
ONIE: Open Network Install Environment
BRCM: Broadcom Merchant Silicon ASICs
OF-DPA: OpenFlow Datapath Abstraction

Underlay Fabric Operation

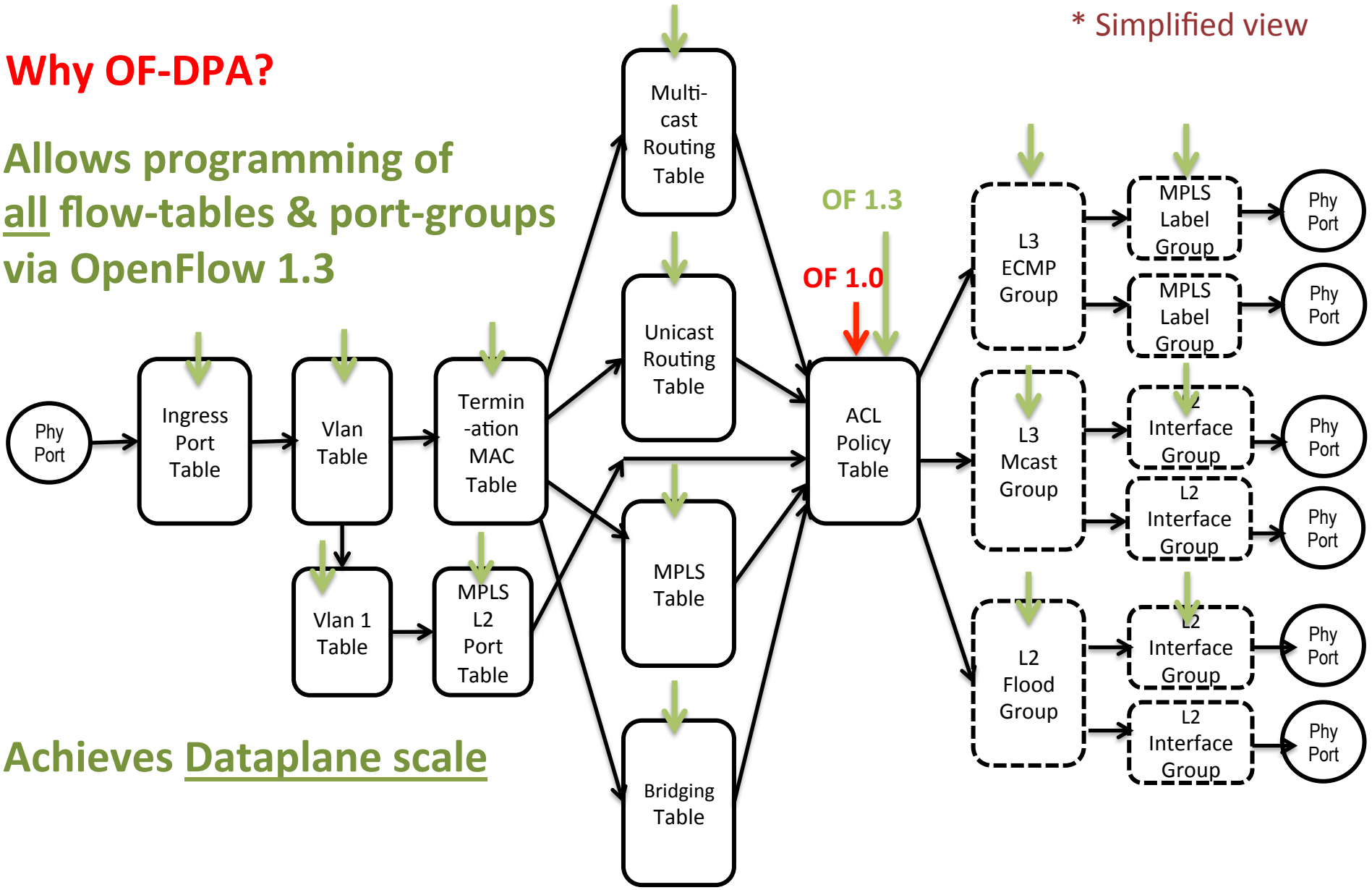


Underlay Fabric ASIC Pipeline* (BRCM's OF-DPA)

* Simplified view

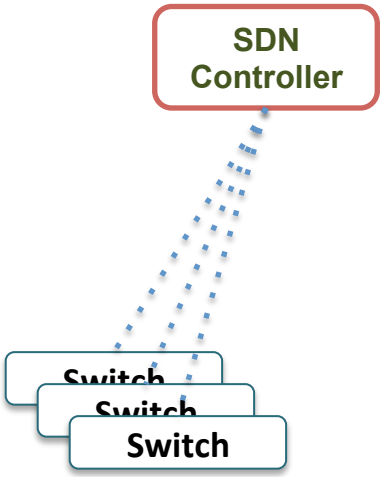
Why OF-DPA?

Allows programming of all flow-tables & port-groups via OpenFlow 1.3



Achieves Dataplane scale

Control Plane HA, Scale & Performance



Classic-SDN Myths:

1. Dataplane packets need to go to controller

Reality: Application designs mode of operation!

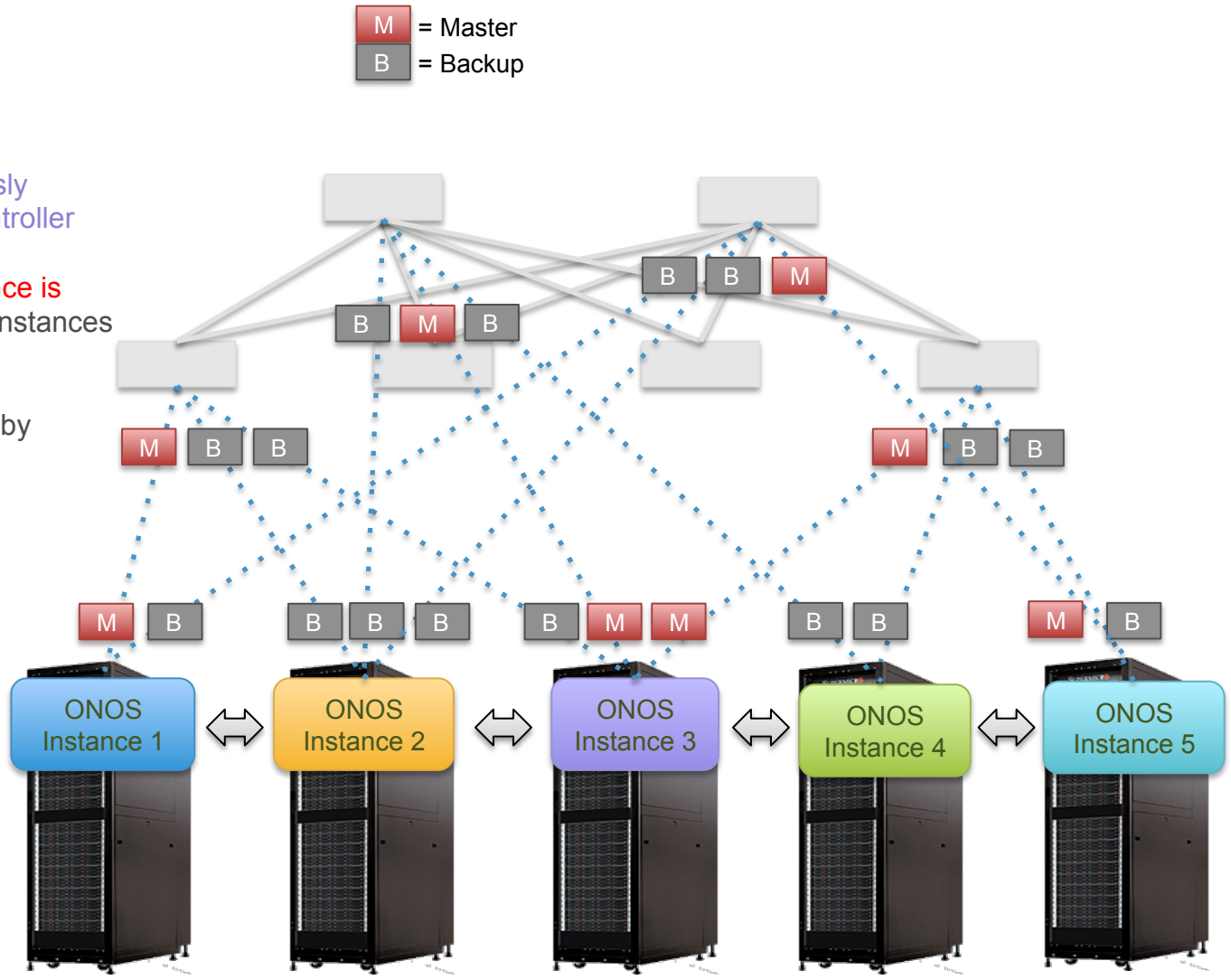
- **Fabric control application designed such that dataplane packets never have to go to the controller.**

2. Controllers are out-of-the-network, like management stations

Reality: Controllers are Network Elements (NEs)!

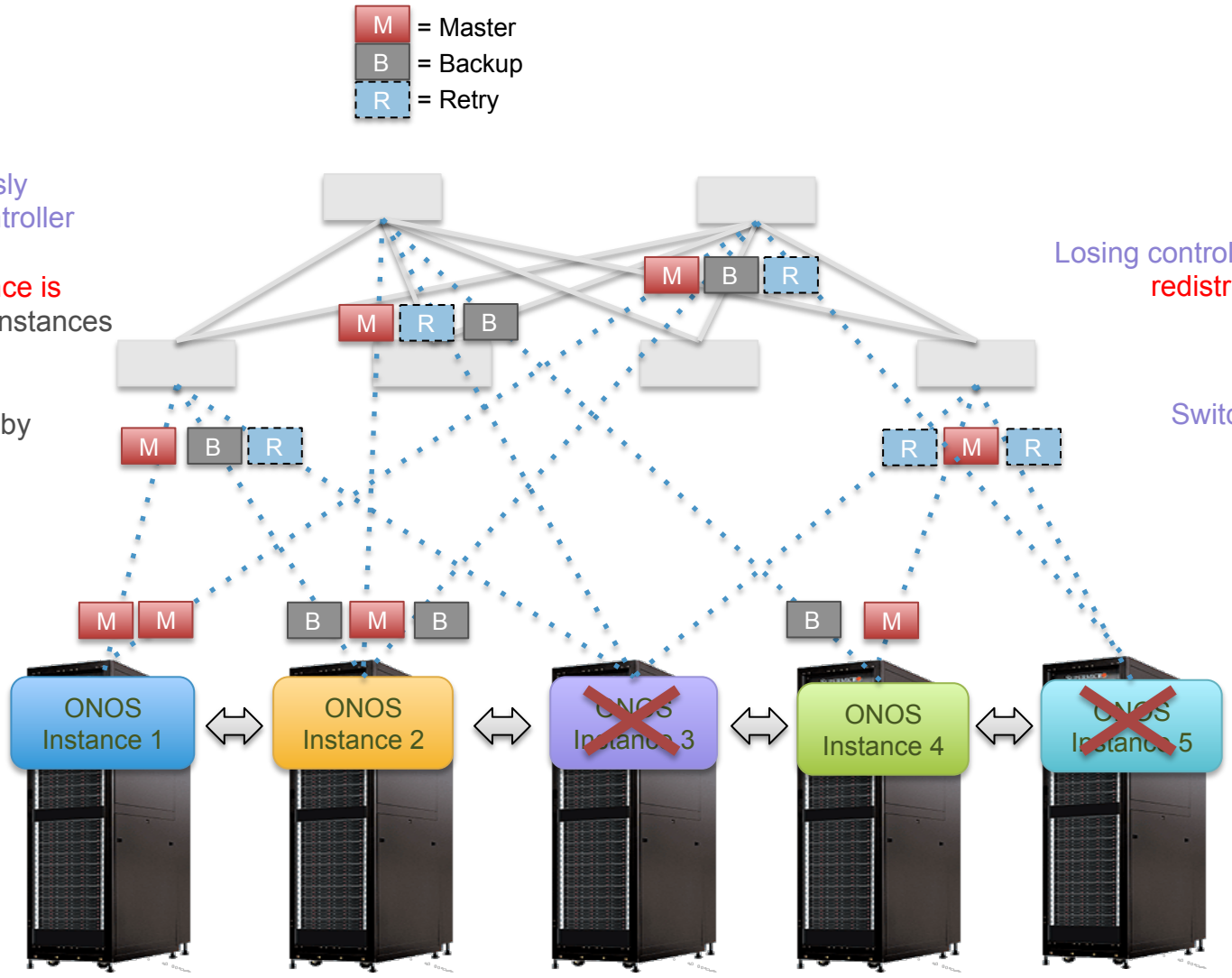
- **Almost all NE redundany is 1:1 or 1+1 (2-way redundancy)**
- **ONOS does much, much more**
  - **3-way, 5-way, 7-way redundancy**
  - **Bonus: scales the same way**

ONOS N-Way Redundancy



Spreading mastership over controller instances contributes to scale

ONOS N-Way Redundancy



Spreading mastership over controller instances contributes to scale

Outline

Underlay Fabric

- Bare-metal + Open-source = White-Box
- L2/L3 Leaf-Spine with SDN Control

Virtual Network Overlay

- OVS and VXLAN with SDN Control
- Service chaining

vRouter

- Distributed Virtual Routing
- Multicast handling

Virtual Network Overlay

Tenant Green Virtual Network

Service Y Virtual Network

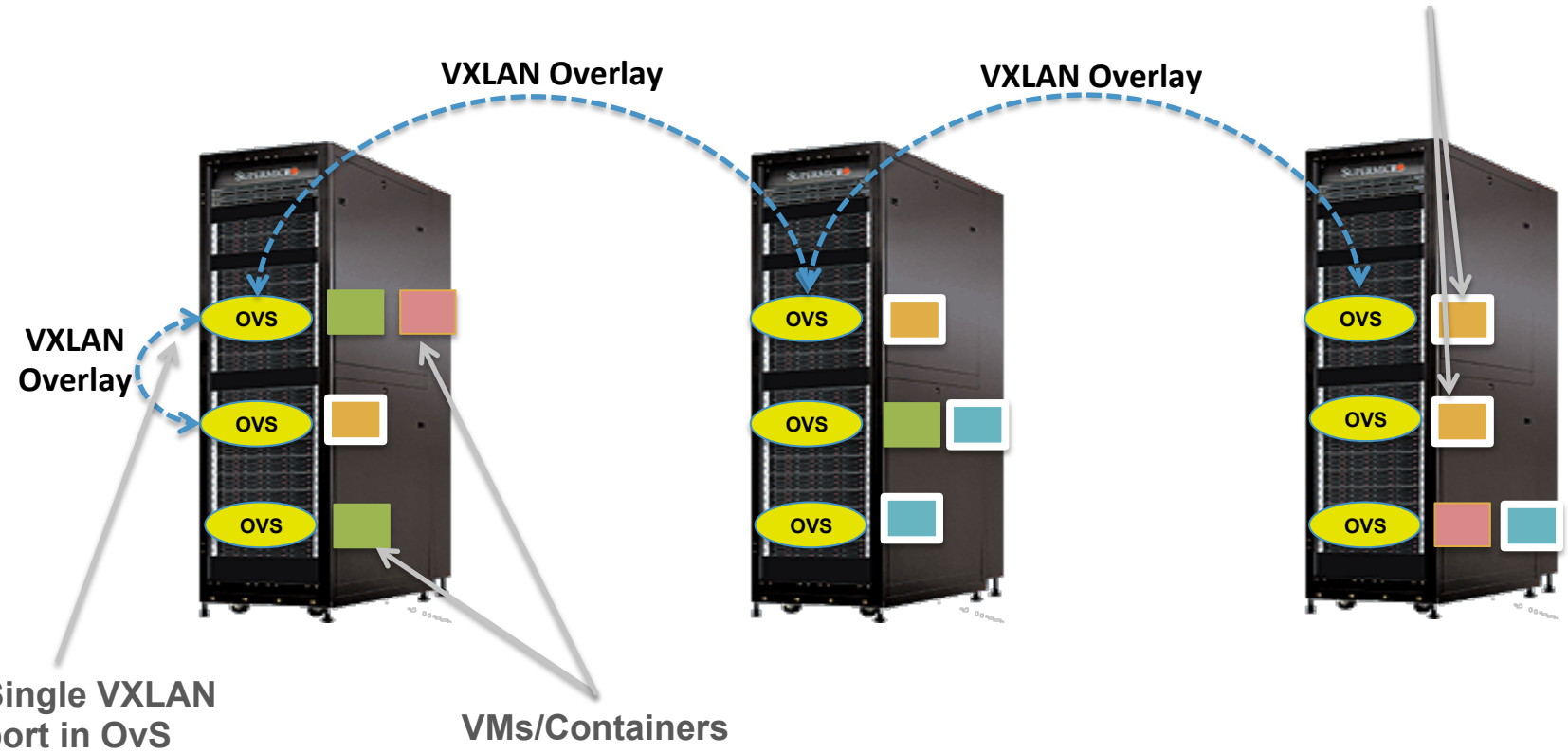
Service B Virtual Network

Tenant Red Virtual Network

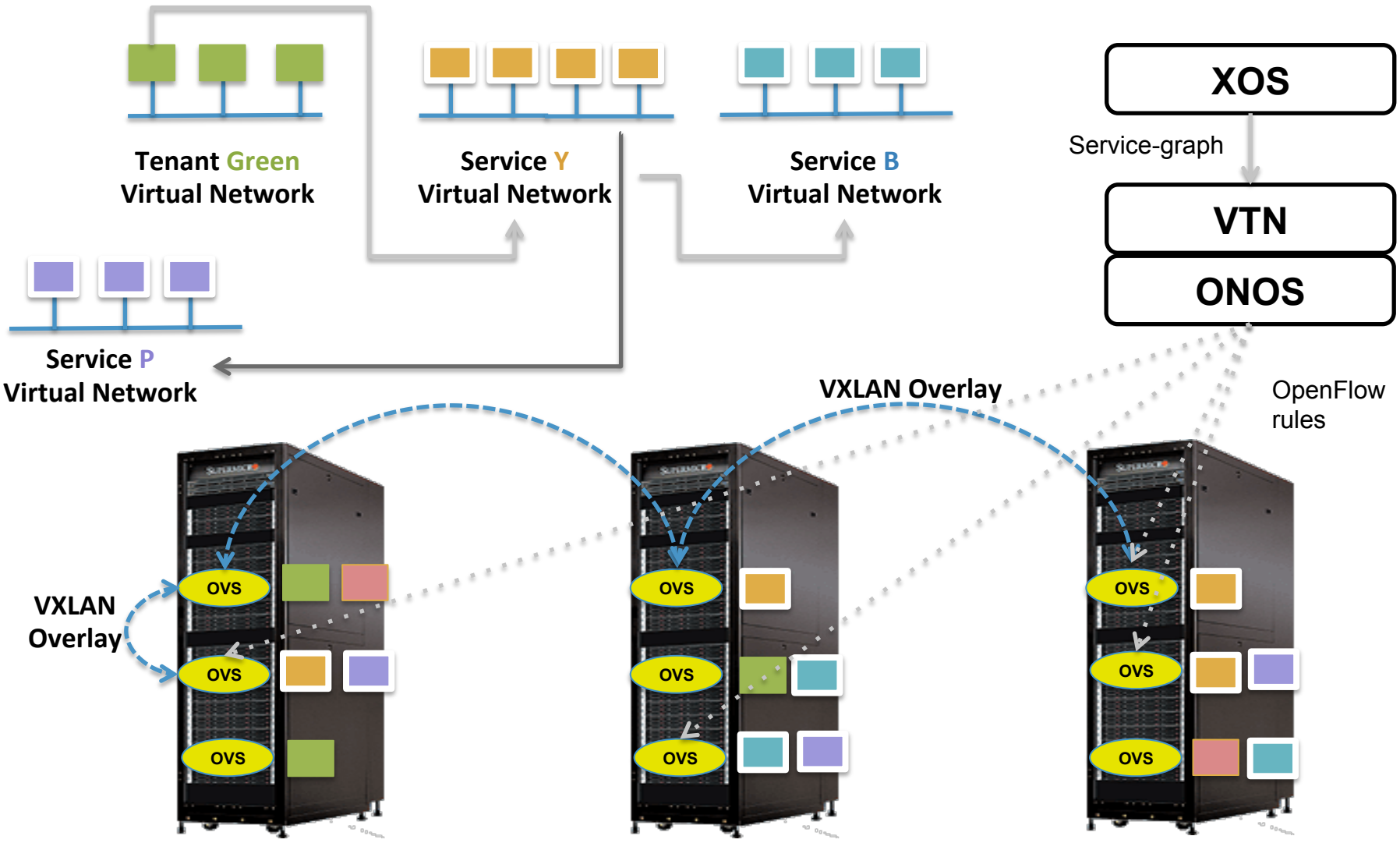
Overlapping address space
Connectivity isolation

Service VNFs & vNets
Non-overlapping addresses

Services can dynamically grow or shrink



Service Composition Impl

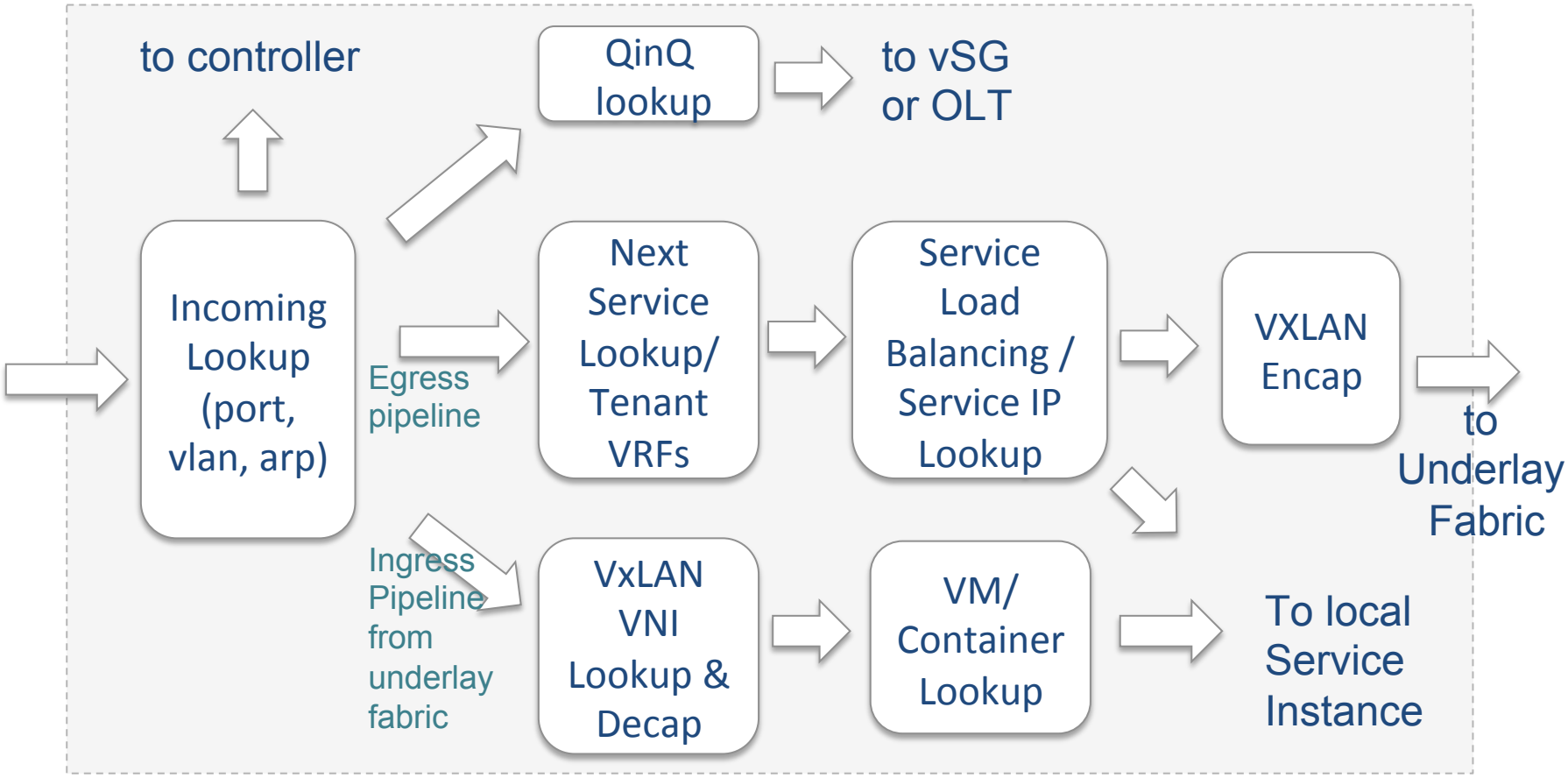


Service graph is created by XOS and programmed into OvS by overlay-control app (VTN)

Distributed load-balancers exist for each service in each OvS

OVS Pipeline* (Vnets & Service-Chaining)

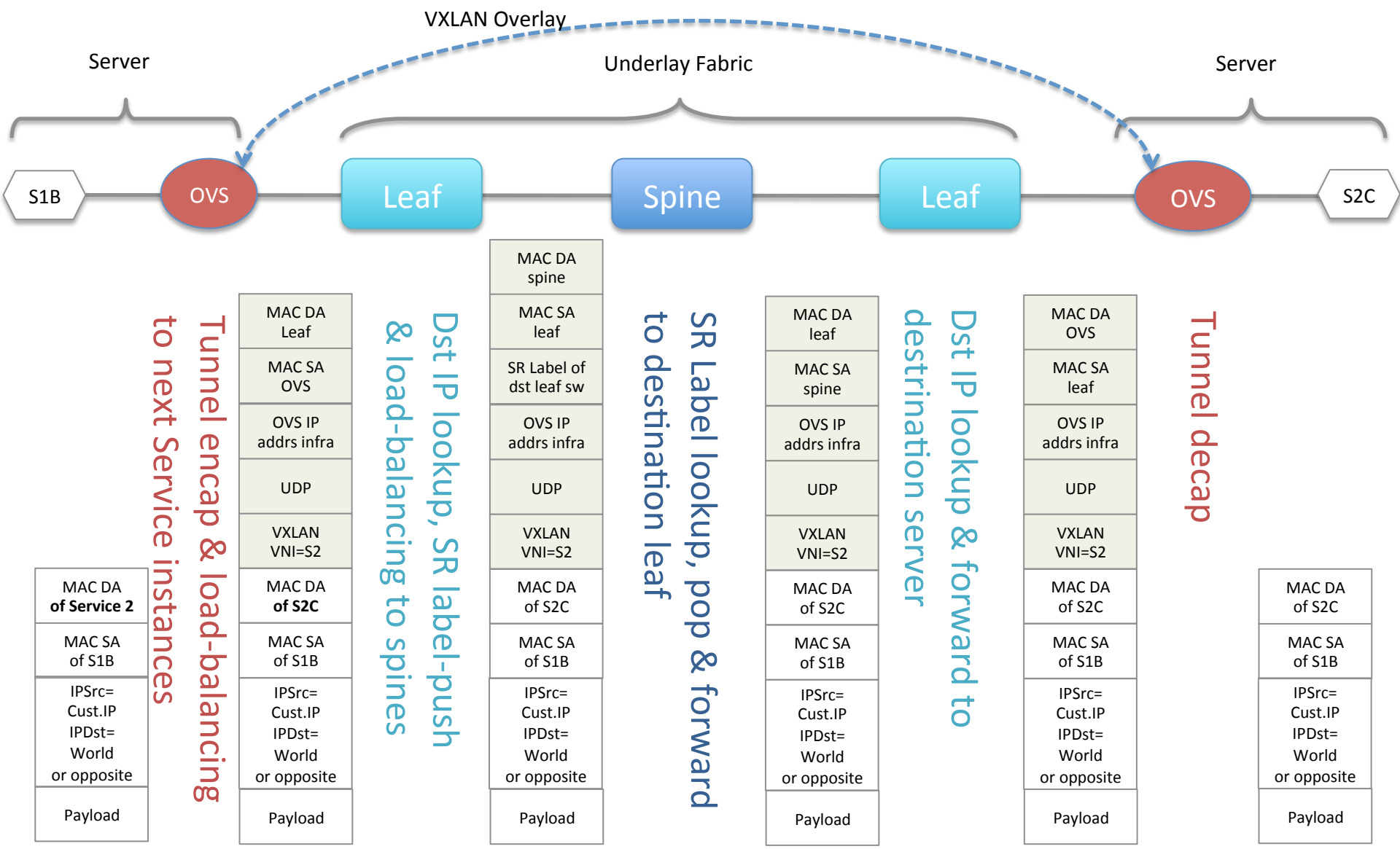
* Simplified view



All flow-tables & port-groups are programmed using OpenFlow 1.3

VxLAN tunnel ports are created using OVSDB protocol

Trellis Overlay & Underlay



Outline

Underlay Fabric

- Bare-metal + Open-source = White-Box
- L2/L3 Leaf-Spine with SDN Control

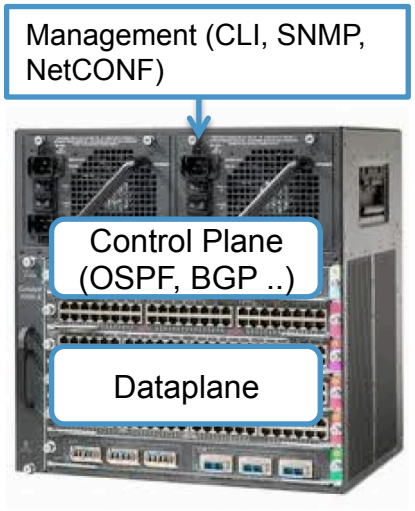
Virtual Network Overlay

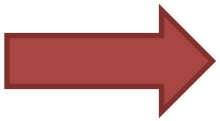
- OVS and VXLAN with SDN Control
- Service chaining

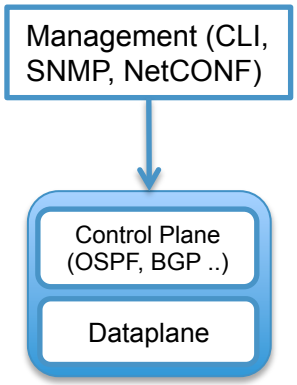
vRouter

- Distributed Virtual Routing
- Multicast handling

vRouter as a VNF?



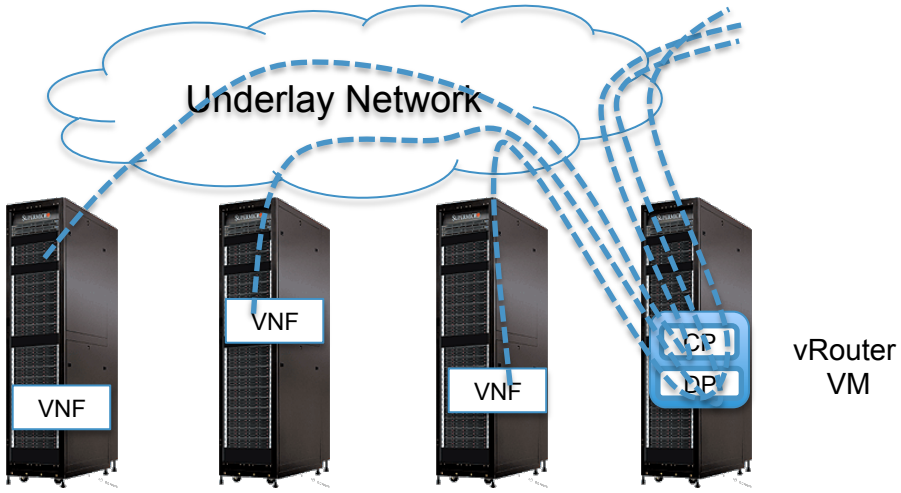


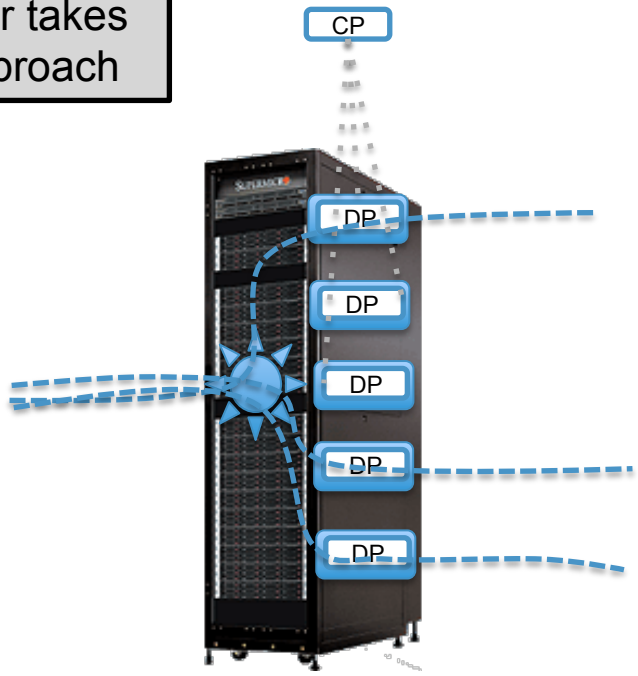


VNFM (VNF Manager)

VNF = vRouter VM (vCPE, vBNG, vPGW, vBRAS)

Trellis vRouter takes a different approach

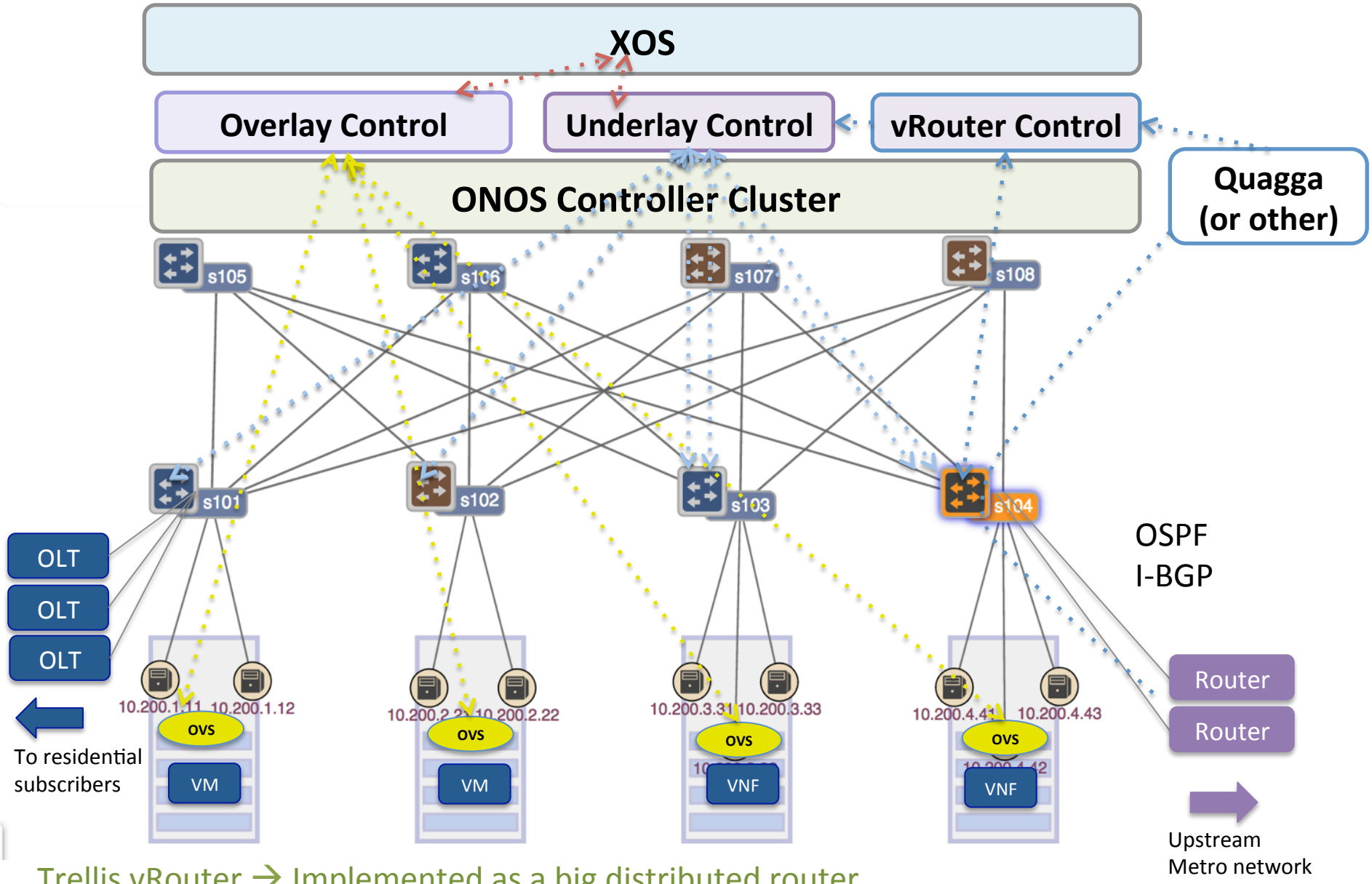




Issues: Hairpinning, Embedded control plane complexity for scale-out

Issues: Still hairpinning through a load-balancer appliance

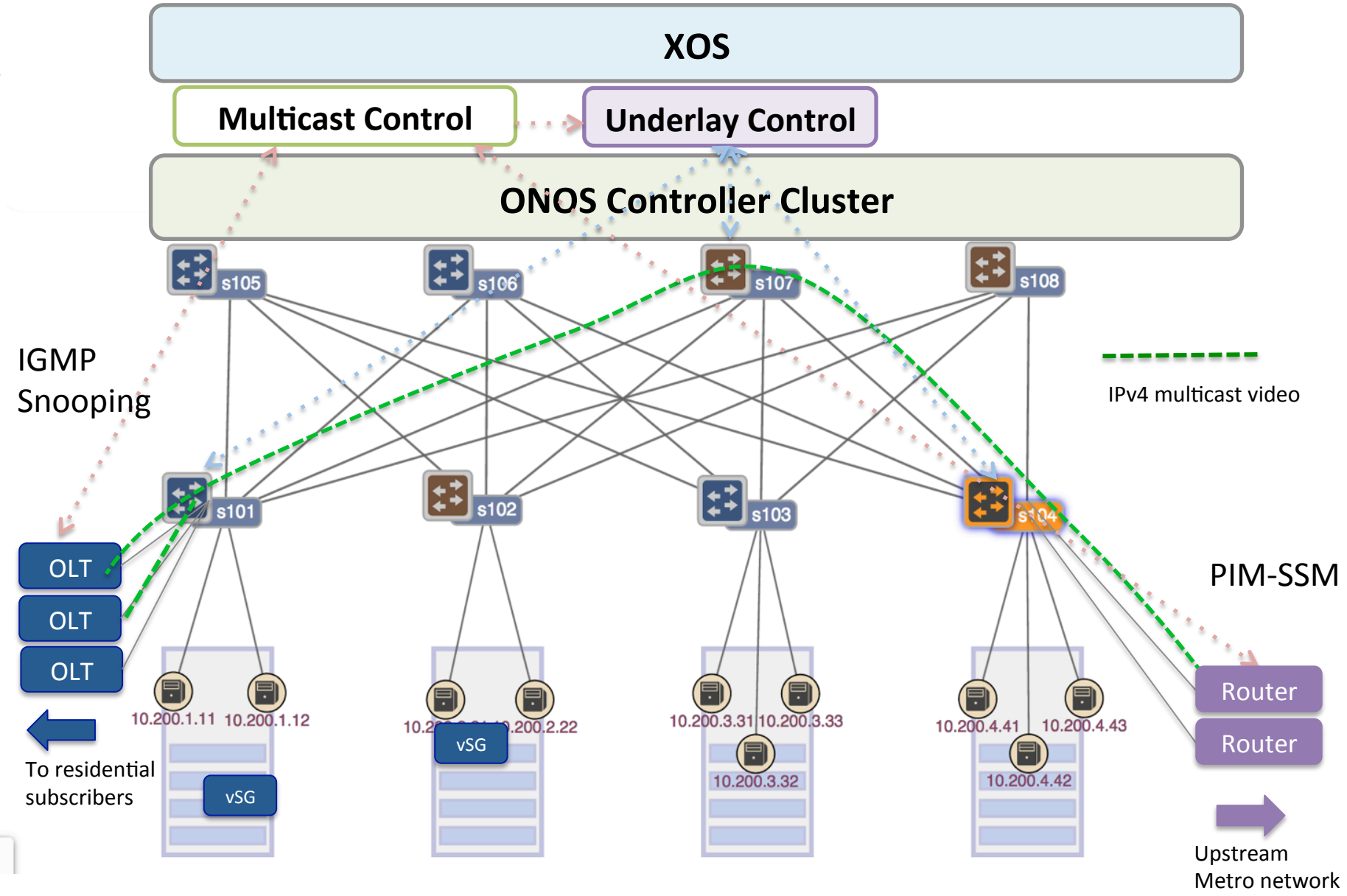
Trellis vRouter



Trellis vRouter → Implemented as a big distributed router

→ Presents entire n/w infrastructure as a single router to outside world

Optimized Multicast in R-CORD



R-CORD Multicast video streams never need to go through any software switch or VNF

Trellis Summary

TRELLIS: DC Fabric Underlay + Virtual Network Overlay + Unified SDN Control

Underlay Fabric

- **L2/L3 spine-leaf** fabric built on bare-metal hw and open-source software
  - Preferred architecture in modern DCs: horizontal scaling & high bisection bandwidth
  - Designed to scale up to 16 racks (~ 40 hardware switches)
  - Low cost ~ $5k per switch
- **SDN control plane** - no distributed protocols
  - Proactive mode of operation
  - Highly available & scalable – ONOS N-way control plane redundancy & scale
- **Modern ASIC data plane** - 1.28 Tbps switching bandwitdth for each switch
  - Broadcom's OF-DPA + OpenFlow 1.3 use of multi-tables & port-groups => dataplane scale

Virtual Network Overlay

- **Designed for NFV** (service-chained VNFs) with best principles of cloud
  - Orchestration, agility, elasticity, micro-services
- **Overlay control application** (VTN) implements/maintains service graph presented by XOS
  - SDN control plane – mostly proactive, HA, no distributed protocols
  - Designed to scale up to 400 virtual switches
- **OvS + VXLAN data plane**
  - Standard VXLAN dataplane encap/decap with entropy-hashing
  - Custom OvS pipeline for tenant virtual networks, service-chaining & per-service load-balancing

Unified SDN Control

- **Common control** provides opportunites for optimized service delivery
  - vRouter – Distributed routing using SDN control and hardware fabric
  - Multicast IPTV service delivery in R-CORD
  - More to follow

BACKUP

CORD Infrastructure Choices: #Racks

Source: Ciena

CORD comes in many sizes → Suitable for many different deployment scenarios
→ Underlay leaf-spine fabric is designed to scale horizontally

CORD Infrastructure Choices: Underlay Fabric

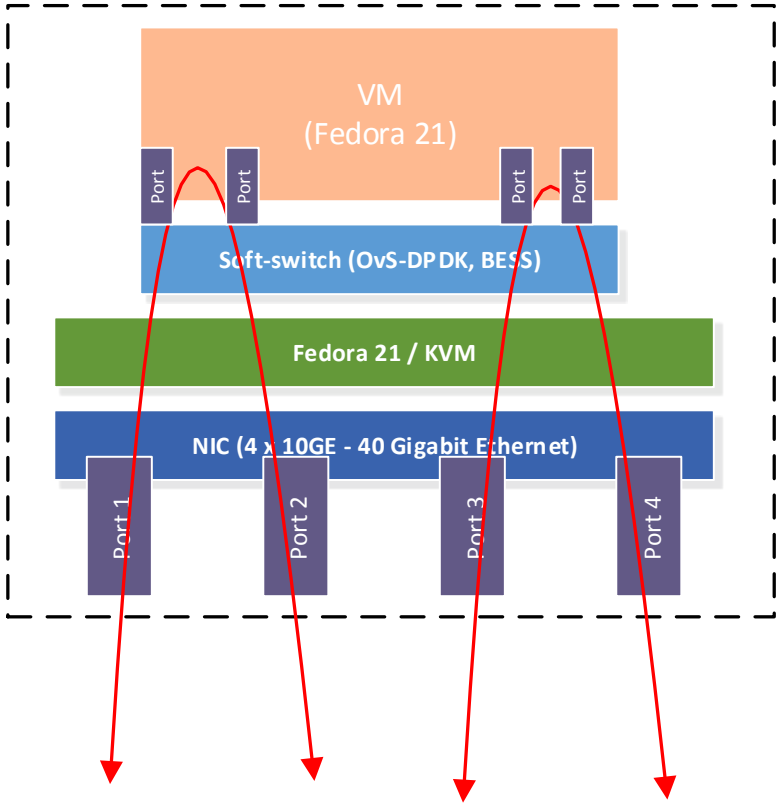
White Box SDN	Grey Box SDN	White Box Td	Grey Box Td	Black Box Td
Open Source SDN Software * OF-DPA API * ONL * Indigo Bare Metal Hardware Choices * Accton * Delta * Dell * Wedge * Quanta * Interface Masters * Celestica	Dell * Dell/DNI hardware * OF-DPA * FTOS / OS10 BigSwitch * Switch Light OS * Most bare-metal hw Pica8 * PicOS * Most bare-metal hw Google * Jupiter fabric	OpenSwitch * HP initiative * Most bare-metal hw SONIC * Microsoft initiative * Most bare-metal hw FBOSS * Facebook initiative * Wedge hw	Cumulus * Cumulus Linux * Most bare-metal hw Pluribus * Netvisor software * Freedom bare-metal hw LinkedIn * Pigeon switch * Bare metal hw Broadcom * FastPath software * Bare metal hw	Cisco & others
Open Source SDN Software on Bare-Metal Hardware	Proprietary SDN Software on Bare-Metal Hardware	Open Source Traditional n/w Software on Bare-Metal Hardware	ProprietaryTraditional n/w Software on Bare-Metal Hardware	ProprietaryTraditional n/w Software on Proprietary Hardware

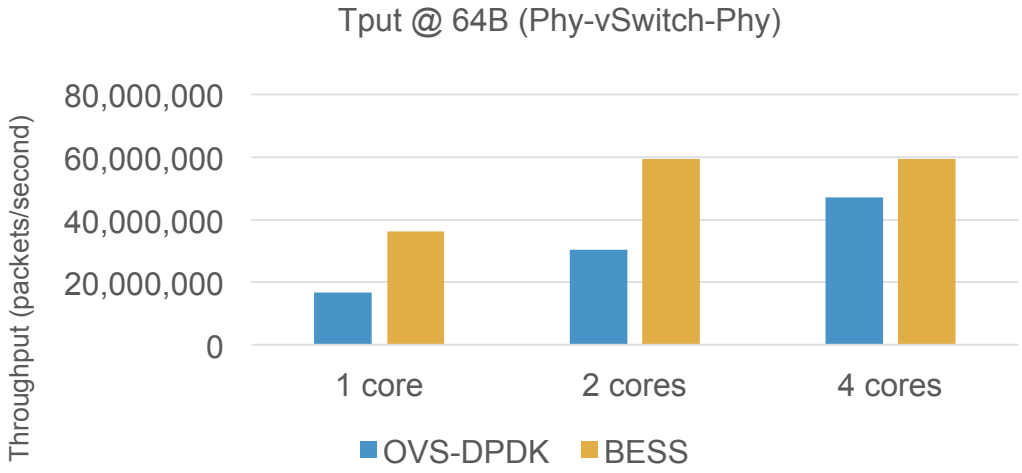
CORD uses White Box SDN → 1) Simpler 2) Easier to introduce new features 3) Lower TCO
+ Significant advantages of common SDN control over underlay and overlay.

Progressively less agile, slower innovation, more complex as we move from Left → Right

CORD Infrastructure Choices: Software Switches

Two choices: OVS + DPDK
BESS + DPDK



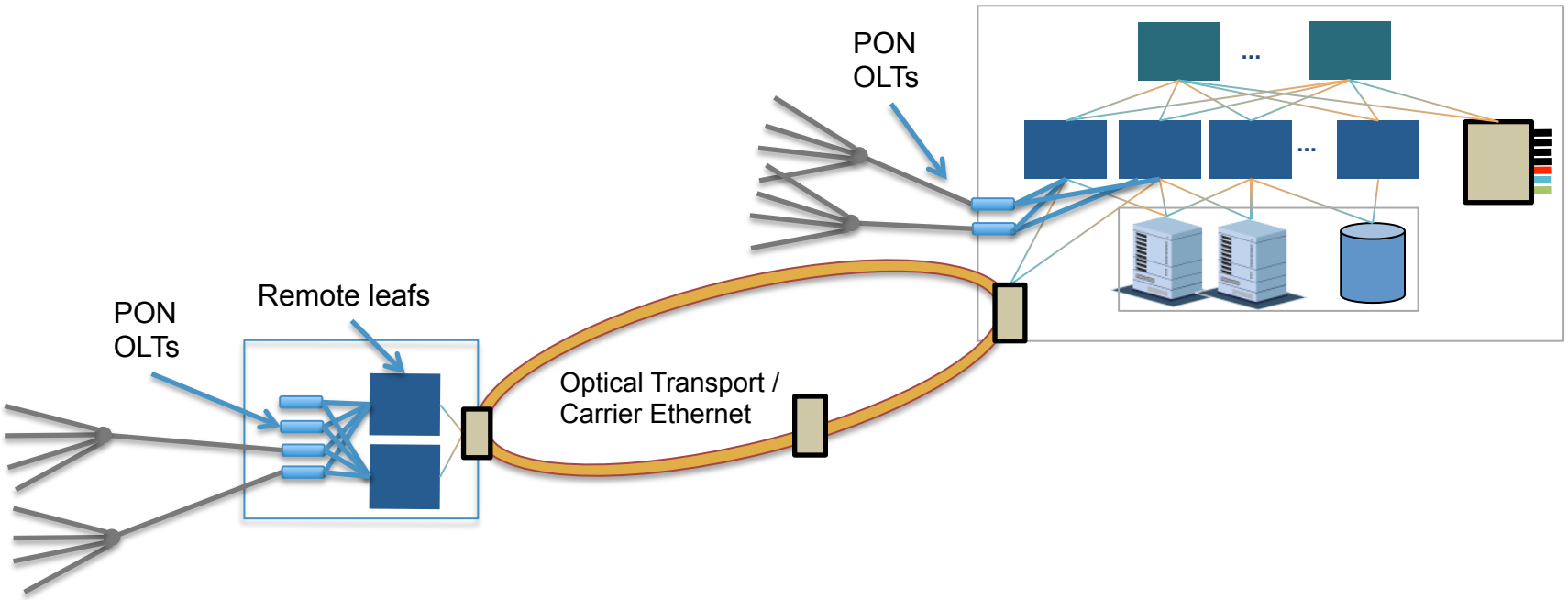


Source: Joshua Reich, AT&T

Open Questions:

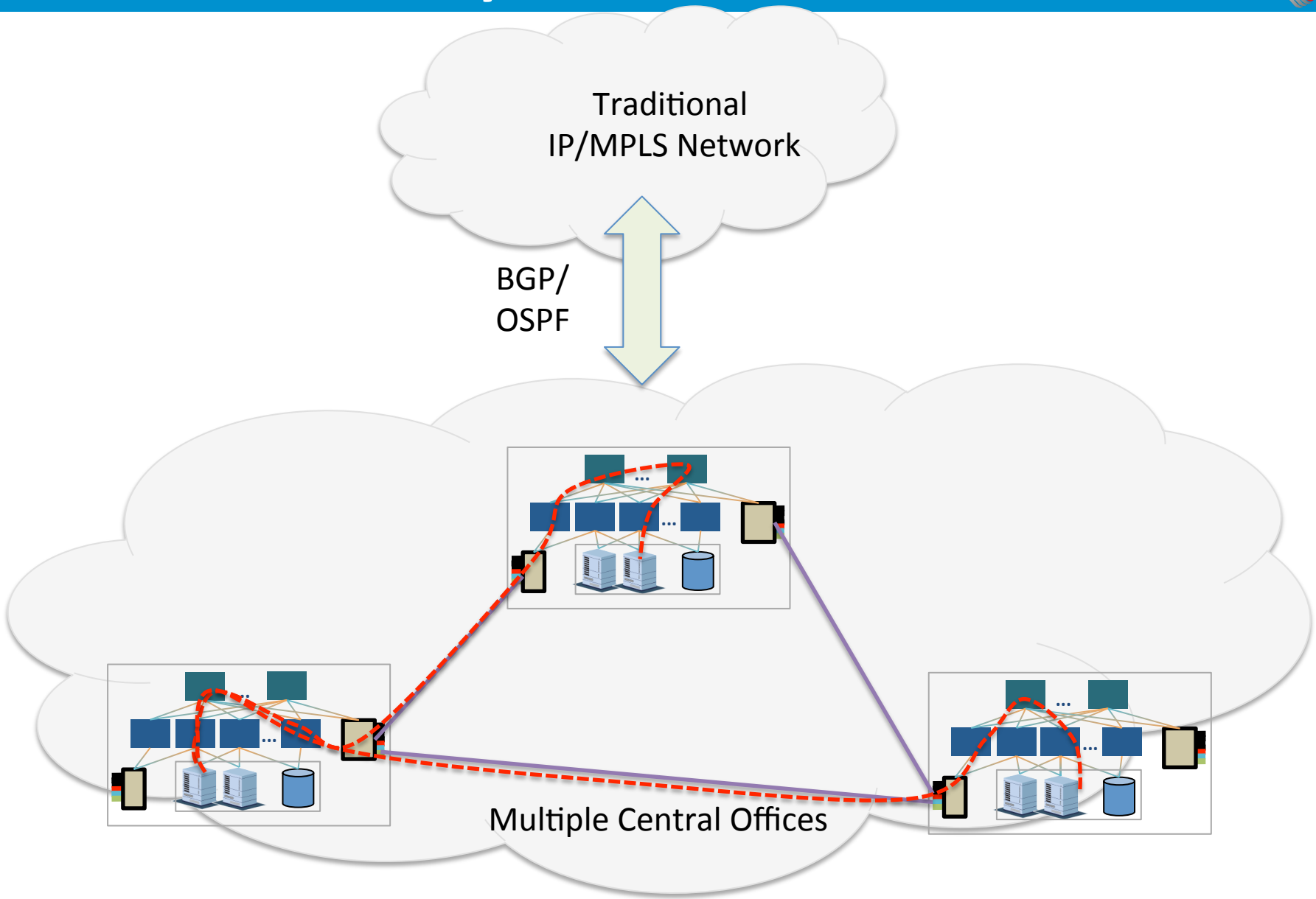
- How do both perform with VxLAN encap?
- And do more than simple bridging?

CORD System: Access Choices



- Remote leafs can be managed in-band by ONOS at CO node
- Remote leafs perform
  - Aggregation of ‘pizza-box’ OLT traffic
    - Assumes transport equipment can deliver double-vlan tagged Ethernet frames from remote-leaf to CO
  - Encapsulation of frames (if necessary)
    - E-Lines implemented using MPLS PW

CORD System: Metro Choices



CORD Infrastructure Roadmap

Plat-forms
XOS
ONOS

Underlay
Overlay
Apps
Multicast
vOLT
vRouter

VNF
vSG

Accton
Hw
Celestica

OF-DPA
Switch software
ONL
Indigo
Quagga
OVS

CORD 1st Release

Dual Homing
- Servers to 2 ToRs
- OLTs to 2 TorRs
- ToRs to 2 Upstream routers

DPDK + OVS/BESS
- with VxLAN
- and VTN pipeline

More VNFs
- Closed & Open source
- Different service chains

More Access Tech
- G.Fast & other vendors

ARP Delegation

Troubleshooting Tools
- Everywhere

sFlow Integration

Scale Testing
- Users, switches, servers

PNF Integration
- Leaf VTEP

ISSU

IPv6

More VNF Integration E-CORD & M-CORD

Inter Central Office

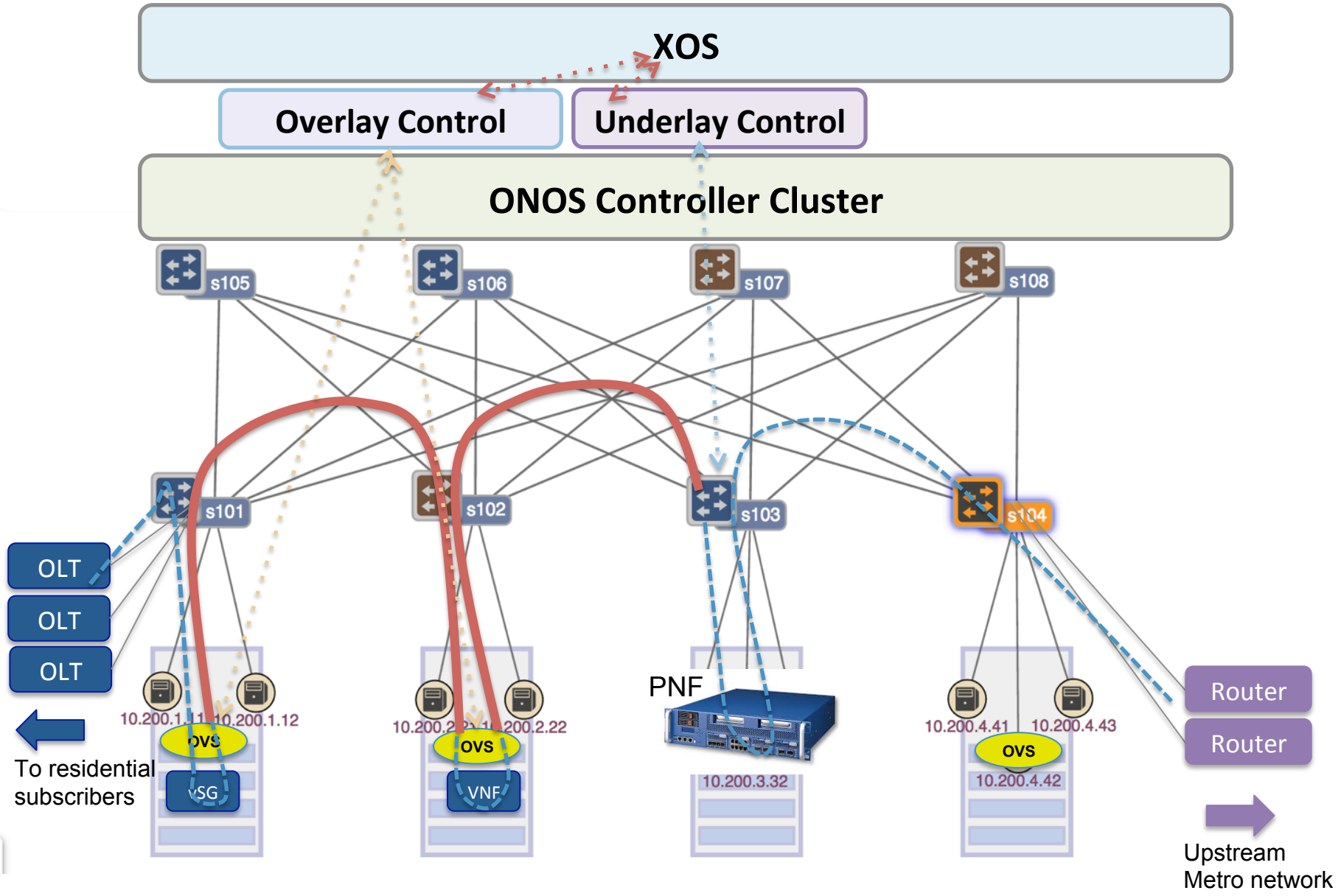
Remote Field Office
- Requires in-band control

Flat network

P4

Q2'16 Q3'16 Q4'16 2017

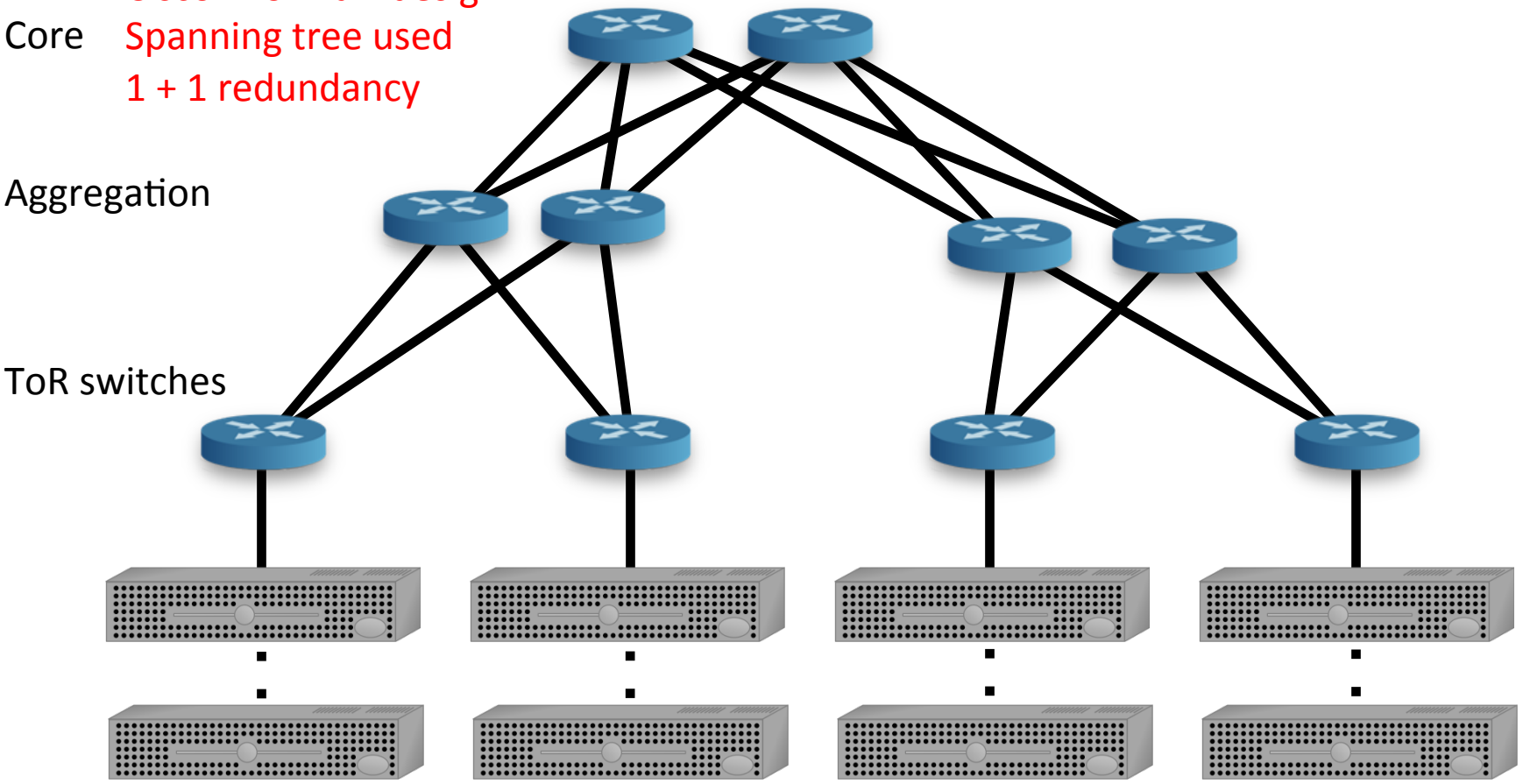
CORD System: PNFs



CORD Service graphs are implemented by both overlay & underlay fabrics

Why DC Leaf-Spine Fabrics?

Cisco “normal” design
Spanning tree used
1 + 1 redundancy



Old design not good for east-west (dominant) traffic in modern data-centers

Why Bare Metal Switches?

Cost

400,000 servers
20,000 switches
$5k vendor switch = $100M
$1k bare-metal switch = $20M
Savings in 10 data centers = $800M

Control

Tailor network to applications
Proprietary behavior
Quickly debug
No vendor lock-in

ONL, SwitchLight, PicOS, FBOSS, SONIC, OpenSwitch, Cumulus, Pigeon, FastPath, SnapRoute

Optional Remote API

Linux

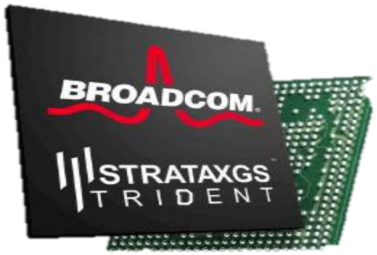
Accton
Delta
Dell
Wedge
Quanta
Interface Maste
Celestica

OF-DPA, OpenNSL, SAI, P4

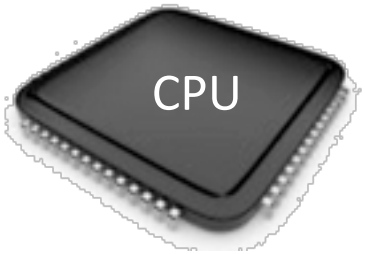
API/Driver

Boot Loader

ONIE



xChip



Power supply + fans

Why SDN?

Benefits of Classic SDN

1. Simpler Control with Greater Flexibility

- Networks work because we can master complexity, but what we should be doing is extracting simplicity, with the right abstractions

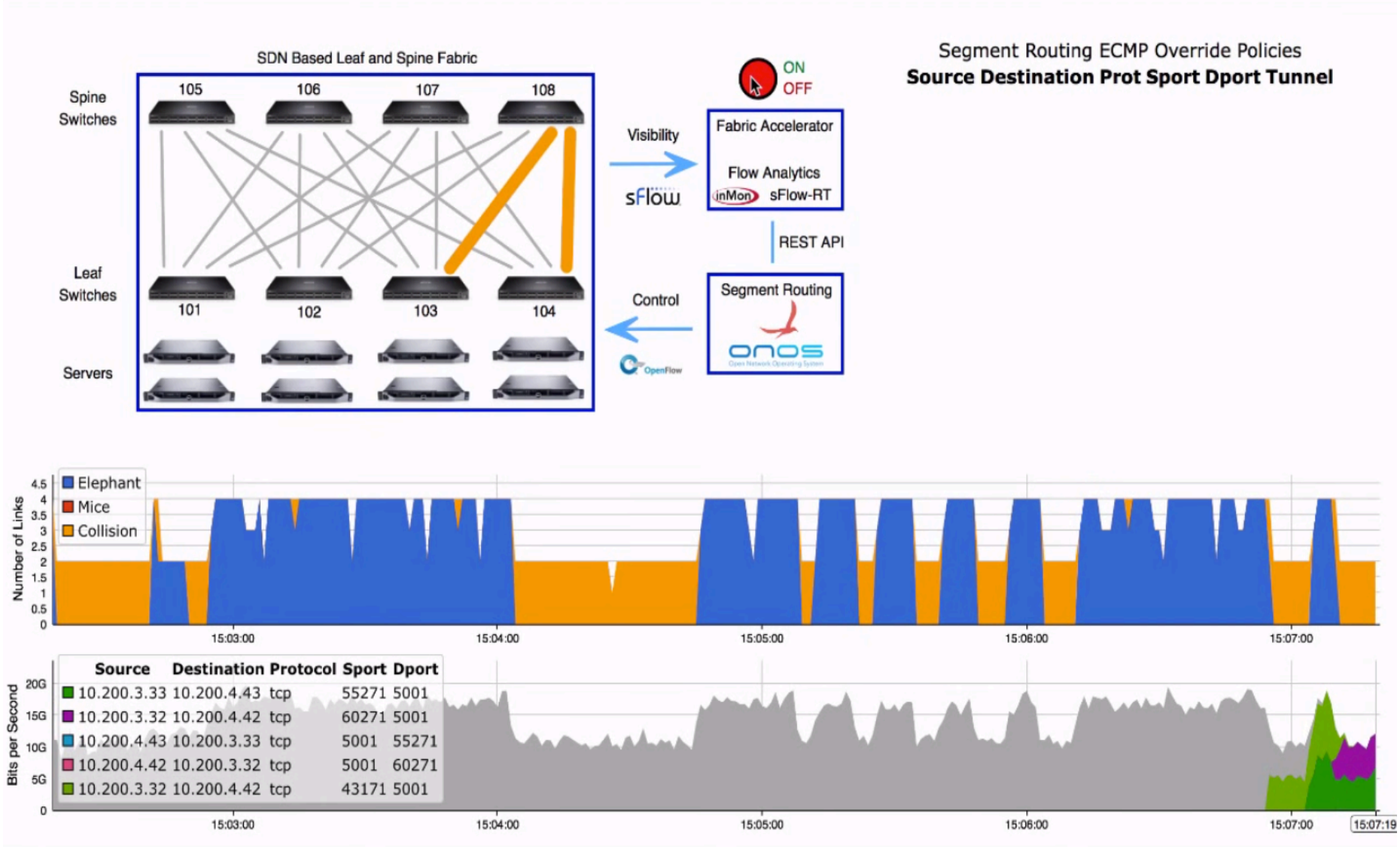
2. Speed of Innovation, Ease of Service Insertion & Faster Time to Market

- Does not involve changing/creating a fully distributed protocol

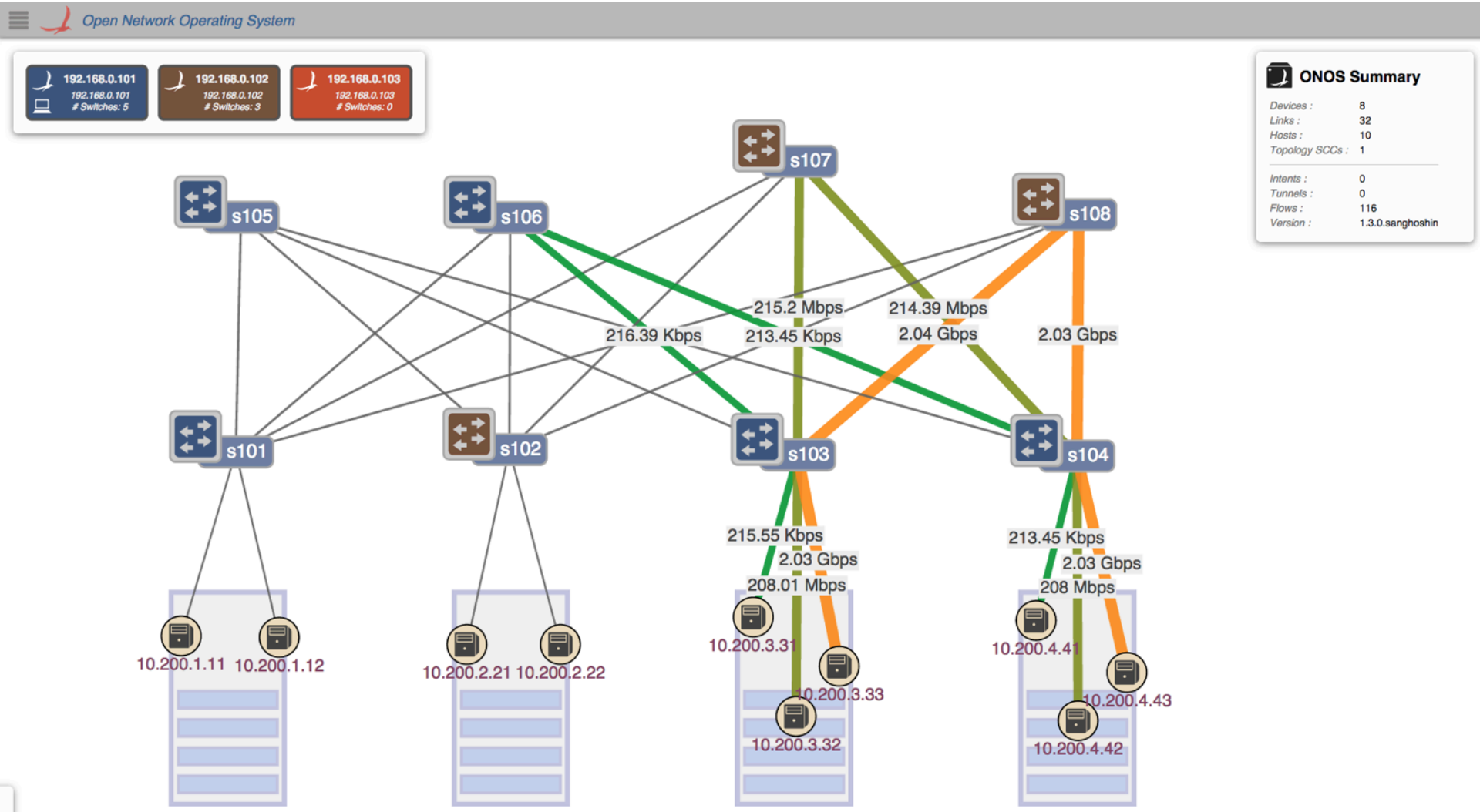
3. Lower Total Cost of Ownership (TCO)

- Lower Opex – easier to manage, troubleshoot, emulate
- Lower Capex – replacing proprietary hardware

Analytics Driven Traffic Engineering



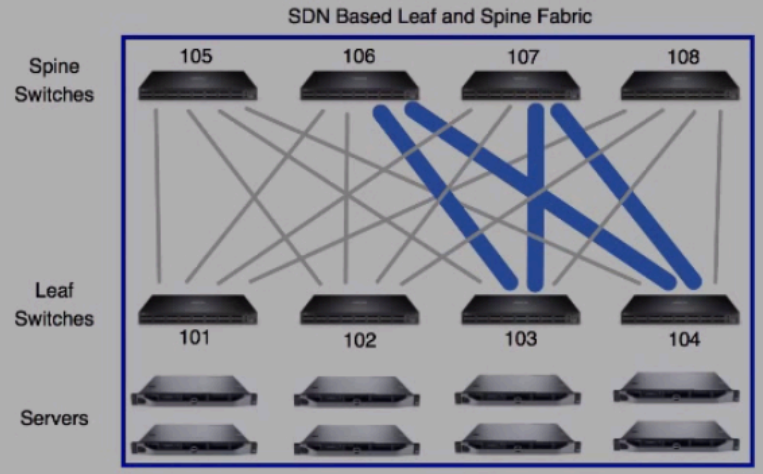
Policy Driven Traffic Engineering

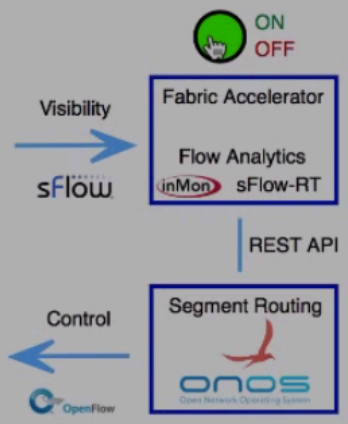



```
onos> srpolicy-add p1 1000 10.200.3.31/32 10.200.4.41/32 TUNNEL_FLOW sr34
onos> srpolicy-list
  ID    TYPE          PRIORITY  SRC_IP           DST_IP           TUNNEL_ID
  p1 _  TUNNEL_FLOW   1000      10.200.3.31/32   10.200.4.41/32   sr34
```

```
onos> srtunnel-list
   ID  GROUP    LABELS
sr34   175      [103, 106, 104]
```

Analytics Driven Traffic Engineering





Segment Routing ECMP Override Policies

Source	Destination	Prot	Sport	Dport	Tunnel
10.200.3.33	10.200.4.43	tcp	36262	5001	103,107,104
10.200.3.32	10.200.4.42	tcp	56522	5001	103,106,104