ONF | ON.LAB

High Availability
In Disaggregated Networks

Saurav Das

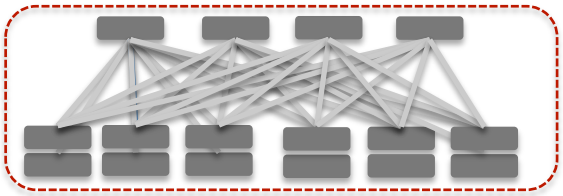
Principal Architect, ONF

With contributions from many others …

Outline

Example Disaggregated Network

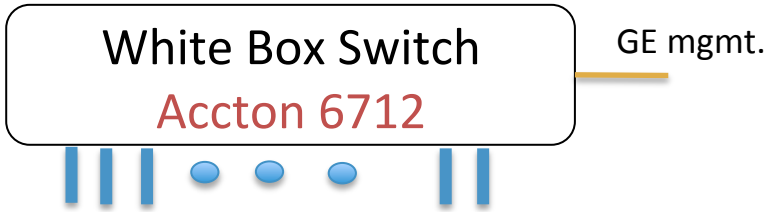
Open-source, SDN-based
Datacenter Leaf-Spine Fabric



High Availability in Disaggregated Networks

Disaggregation 1/2 – Bare-metal + Open-Source

Spine Switch

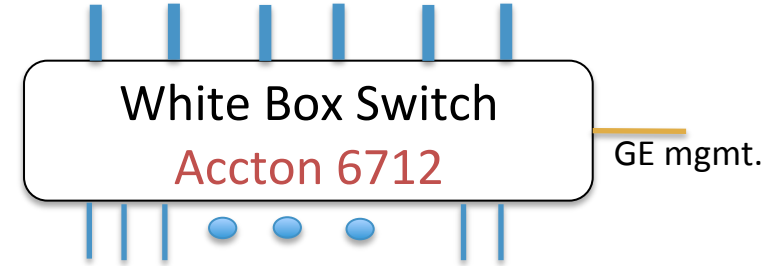


32 x 40G ports downlink to leaf switches

Leaf Switch

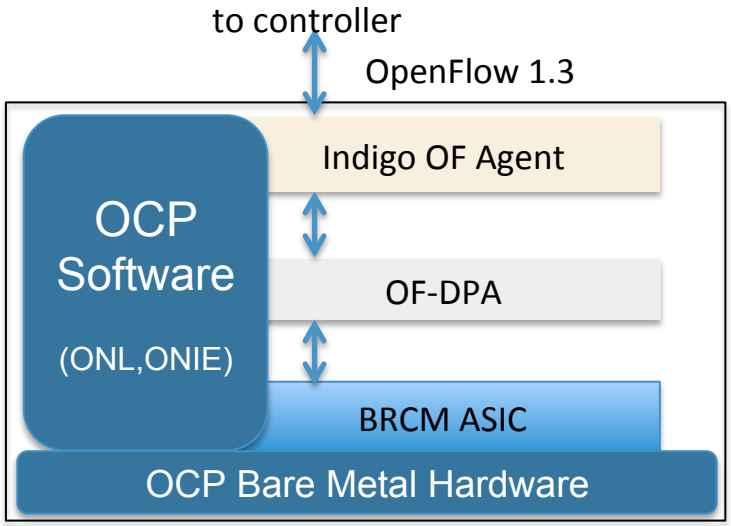
8 x 40G ports uplink to different spine switches

ECMP across all uplink ports



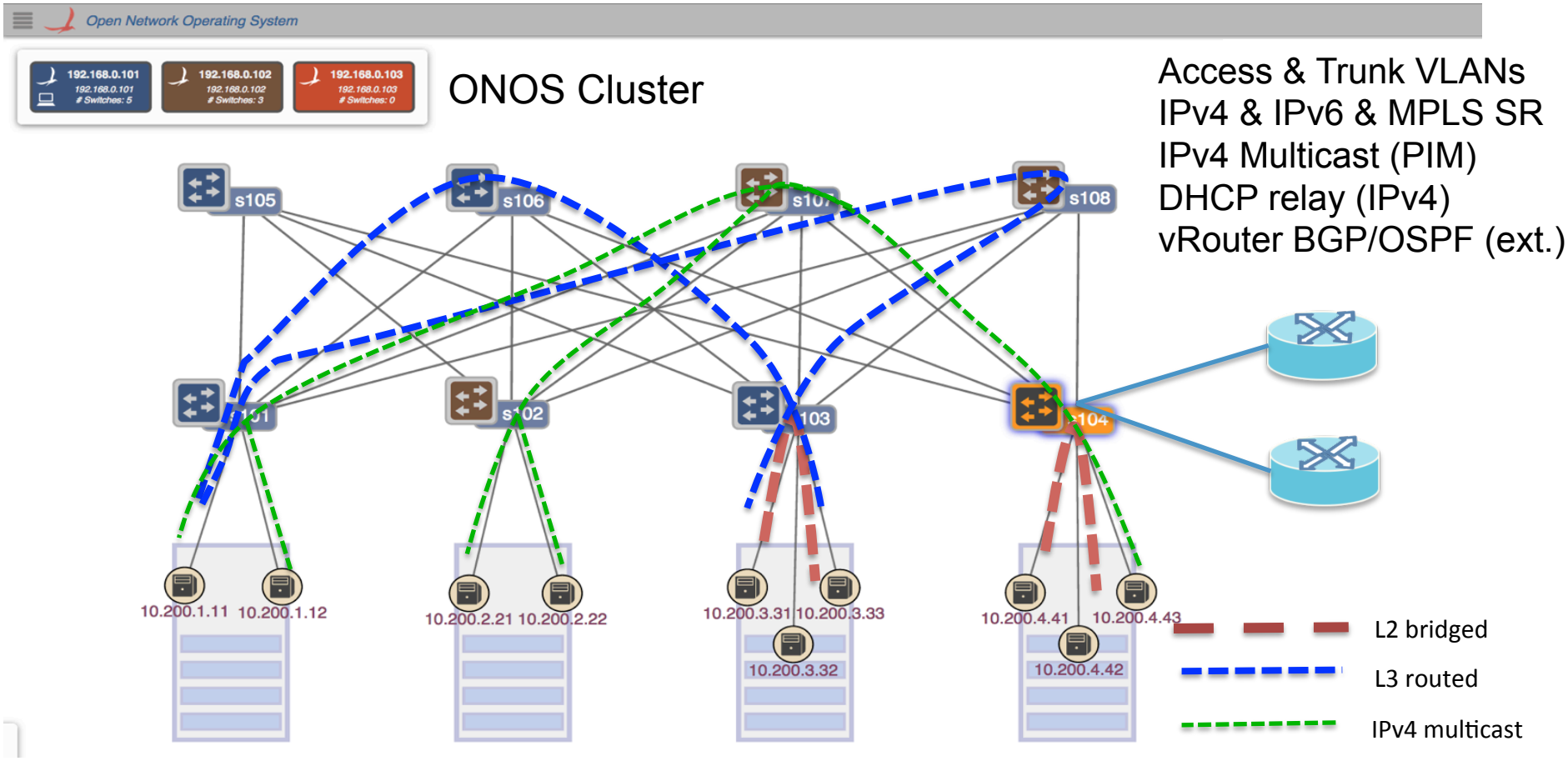
24 x 40G ports downlink to servers

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

Disaggregation 2/2– Bare-metal + Open-Source + SDN

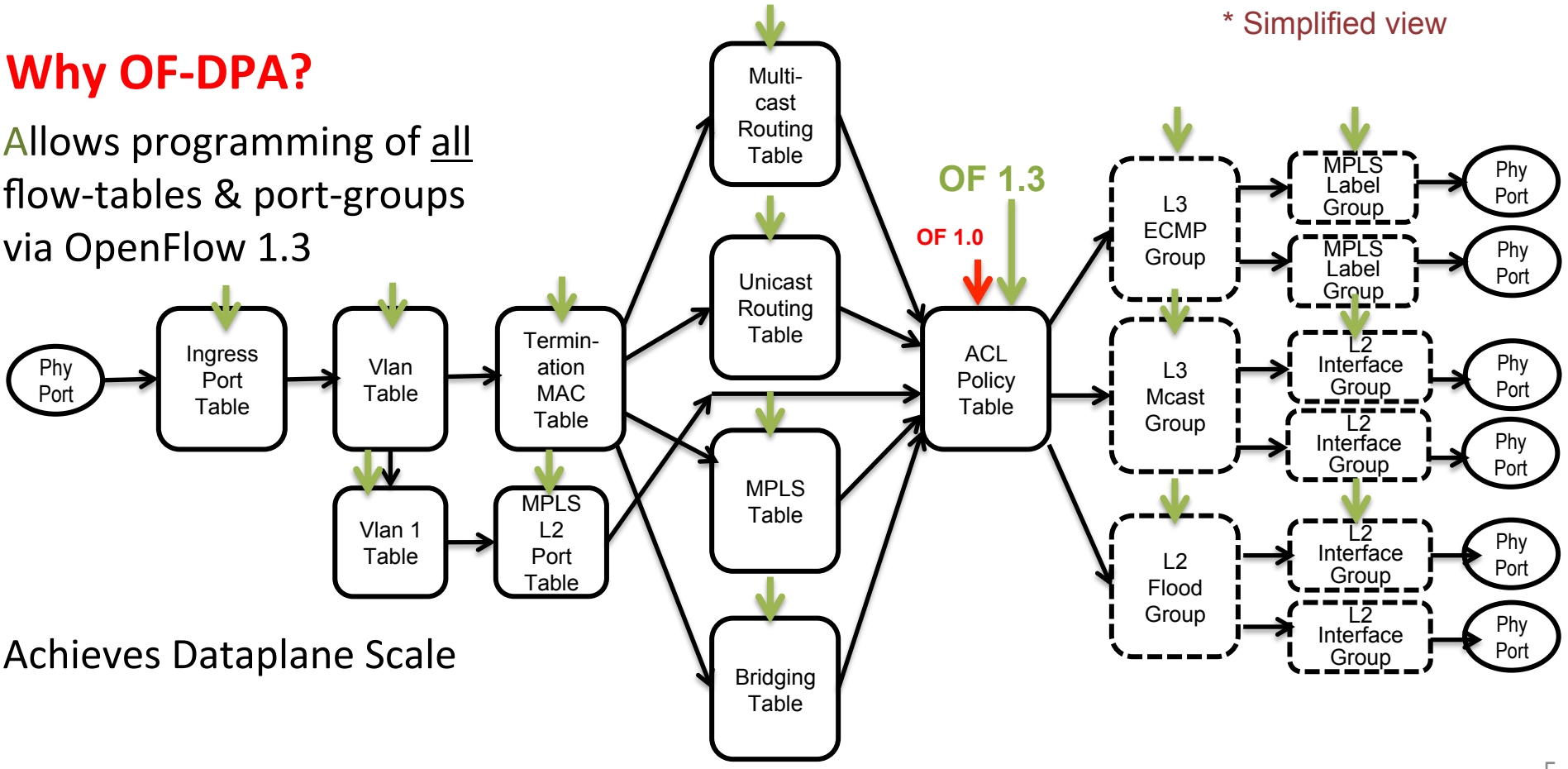


Fabric ASIC Pipeline* (BRCM's OF-DPA)

* Simplified view

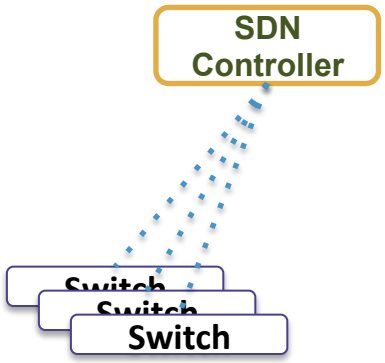
Why OF-DPA?

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



Achieves Dataplane Scale

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-net (management stations)

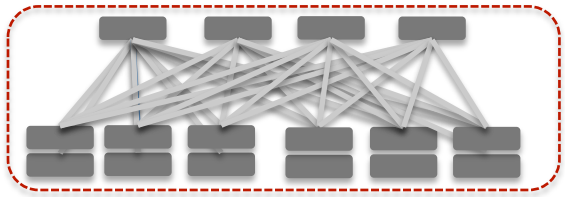
   Reality: Controllers are Network Elements (NEs)!

   - Need to design for redundancy and scale to achieve production readiness

Outline

Example Disaggregated Network

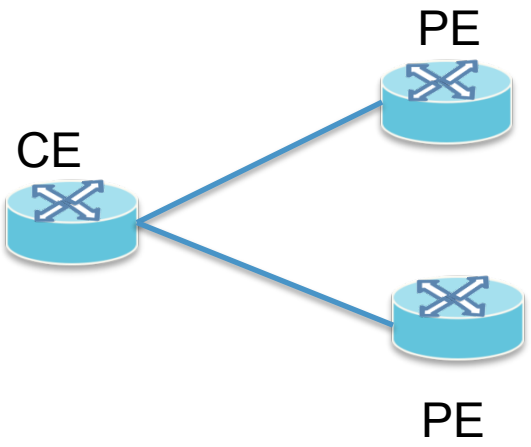
Open-source, SDN-based
Datacenter Leaf-Spine Fabric

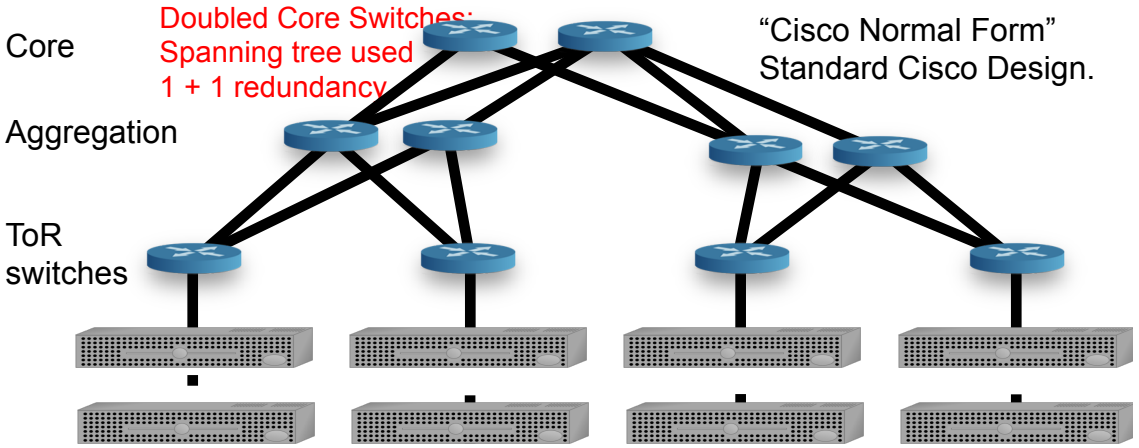


High Availability in Disaggregated Networks

- **Control plane redundancy**
- Data plane redundancy
- Combined data-plane & control-plane failure recovery
- Dual-homing servers
- vRouter HA
- Headless fabric

Redundancy in Networking

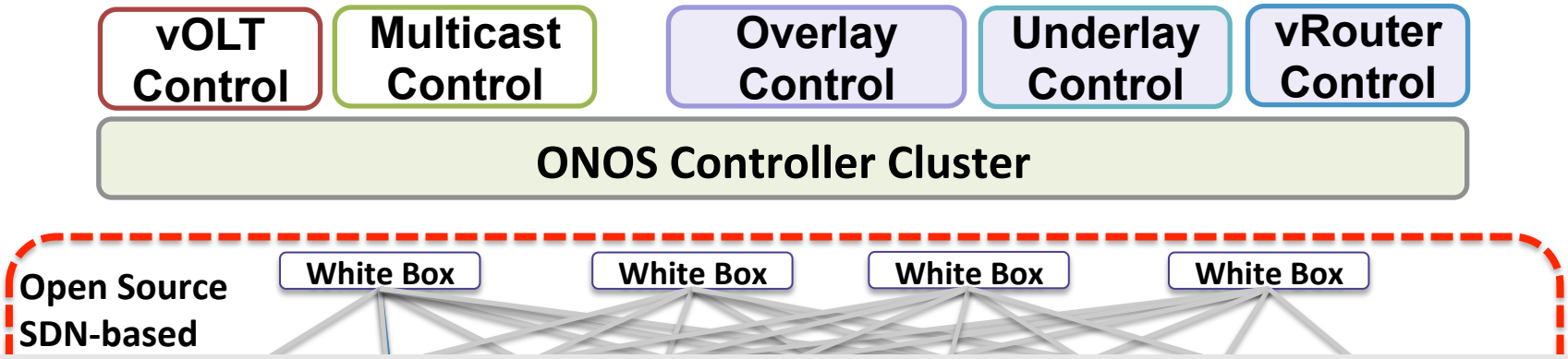




In most of networking

- 2 is a golden number (1+1, 1:1, N:1)
- Acceptible risk/reward – what happens when both routers/switches die – customer network down – low probability, unlikely event

SDN Controller

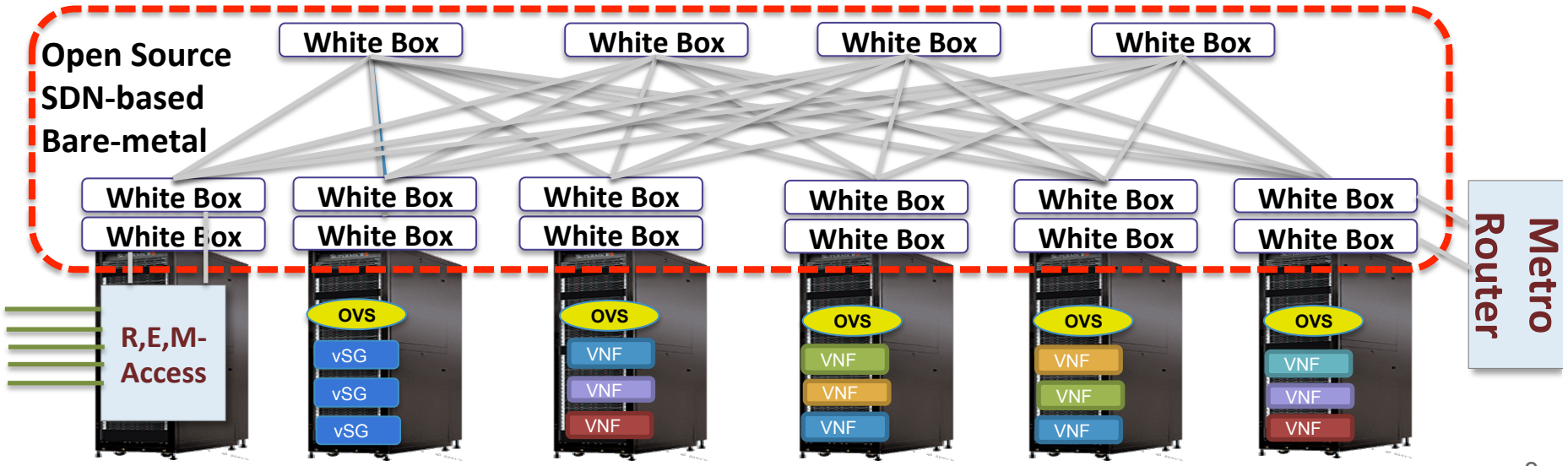


- Tend to show SDN controller as out-of-the-box
- But then it's treated like a management system
  - As a workstation outside of the network.
  - No management system is HA – if it dies, reboot it
  - In the meantime network should still work!

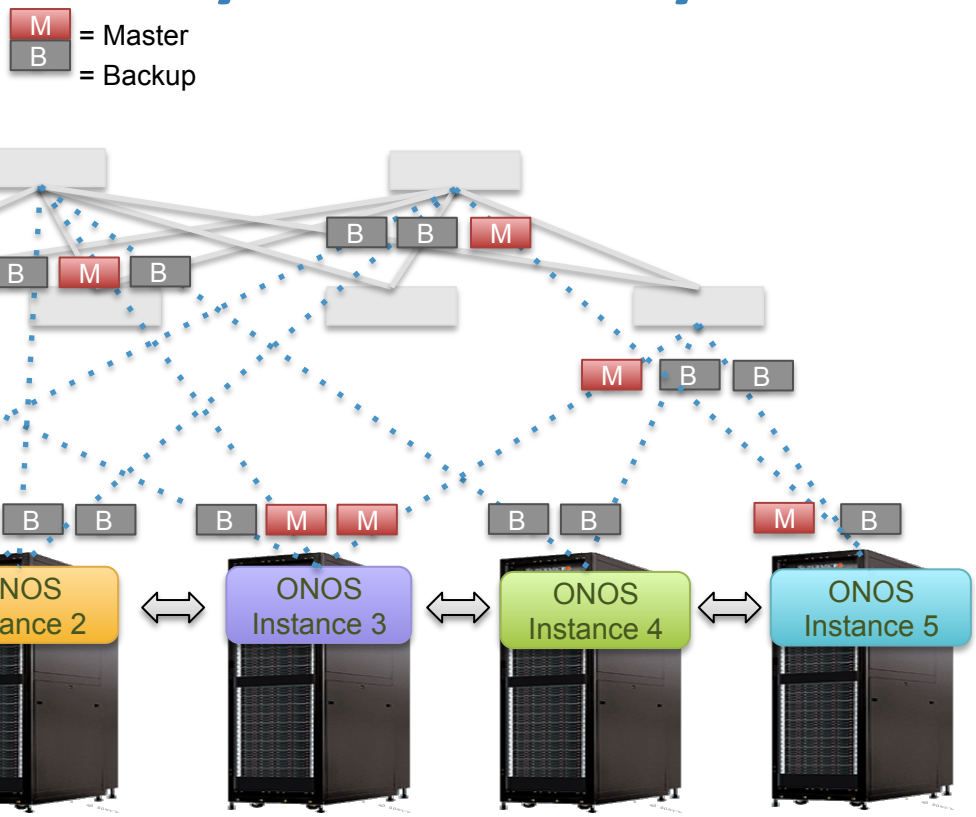
Reality: Controllers are NEs

Need to think of SDN Controller as Network Element (NE)

- And like most networking solutions to redundancy, some SDN solutions do 1:1
- ONOS does much much more
  - 3-way, 5-way, 7-way redundancy
  - Bonus: scales the same way
- Spread instances around in DC racks – N-Way redundancy
  - Unlikely event that they all die simultaneously – there are bigger issues if that happens
  - Can design for headless mode

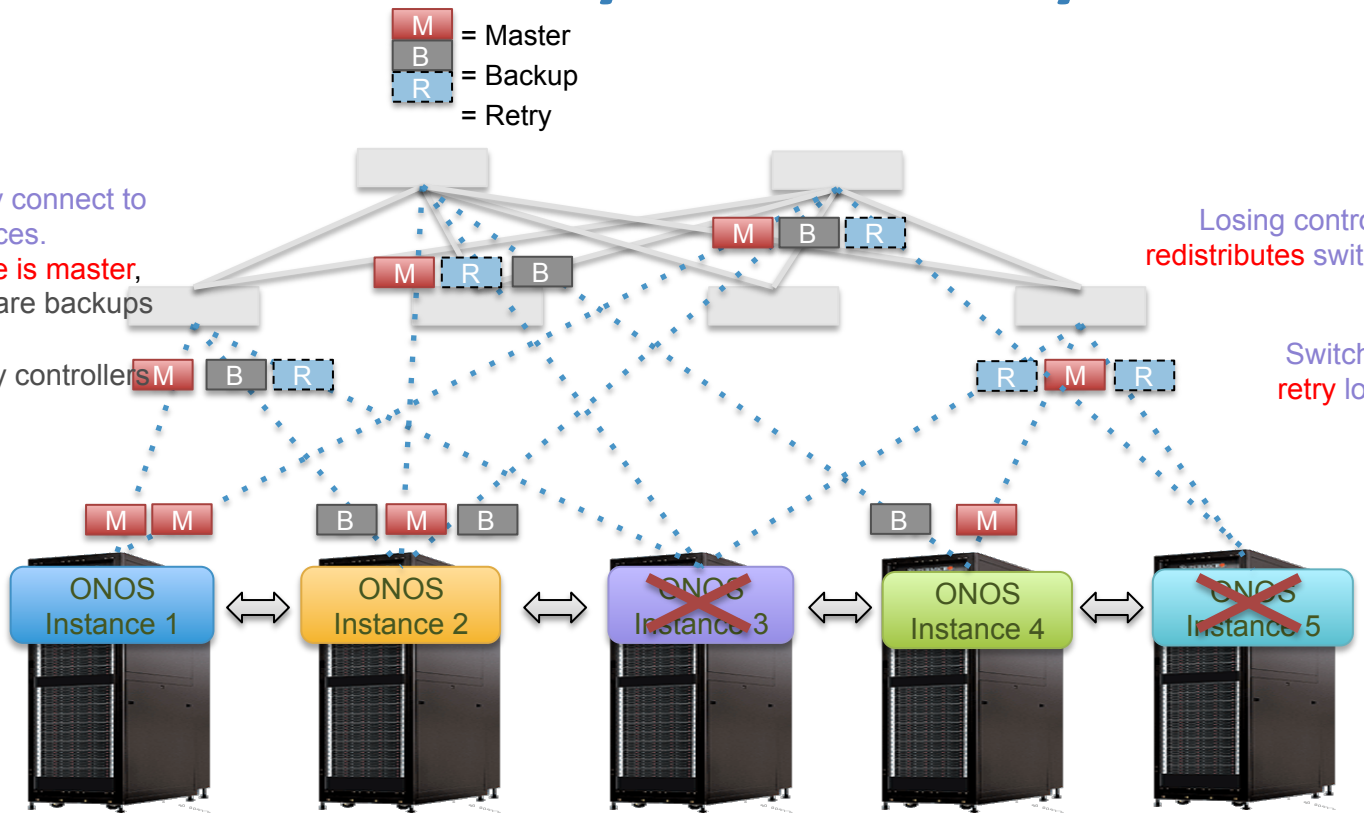


ONOS N-Way Redundancy



Spreading mastership over controller instances contributes to scale

ONOS N-Way Redundancy



Switches simultaneously connect to several controller instances.
only 1 controller instance is master, several other instances are backups

Mastership is decided by controllers
Switches have no say

Controller instances simultaneously connect to several switches.
Any controller instance can be master or backup for any switch

Losing controller instances redistributes switch mastership

Switches continue to retry lost connections

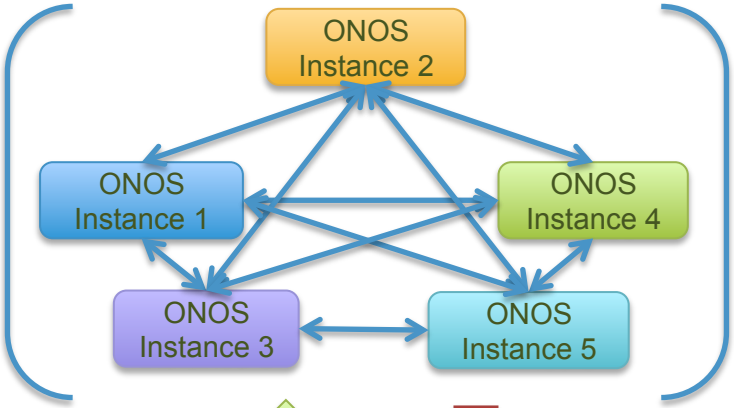
Management watchdog can reboot lost controller instances

Spreading mastership over controller instances contributes to scale

State Synch: Authoritative State

ONOS instances & apps **actively synchronize** with each other using state-of-the-art, fault-tolerant distributed systems algorithms

To the external world the cluster behaves like a **single logical entity**



State:

- Network Topology
- Network Configuration
- Mastership Assignment
- FlowRules / Groups
- Resource Allocations
- Intents
- And many more

Observe

- **Liveness information (up/down)**
- **Statistics**

Program/Enforce

- **FlowRules**
- **Groups**
- **Virtual Ports**
- **Mastership**

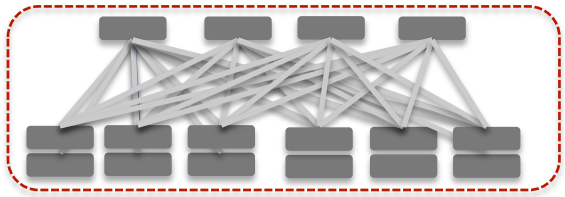
ONOS Cluster Features

- **Failures are the rule not exception**.
- All critical information is **3-way replicated and persisted**. Simple configuration change to enable even higher degrees of replication (if needed).
- **Logically consistent** view of replicated state via state of the art **distributed consensus** and synchronization protocols.
  - **Raft Consensus** for Resources, Mastership, Network Config, …
  - **Primary/Backup** for Flow Rules
  - **Optimistic Replication** for Topology, Data plane stats, …
- Failure handling is **fully automated**.
- Workload is **evenly distributed**. When one node fails, others take over its responsibilities

Outline

Example Disaggregated Network

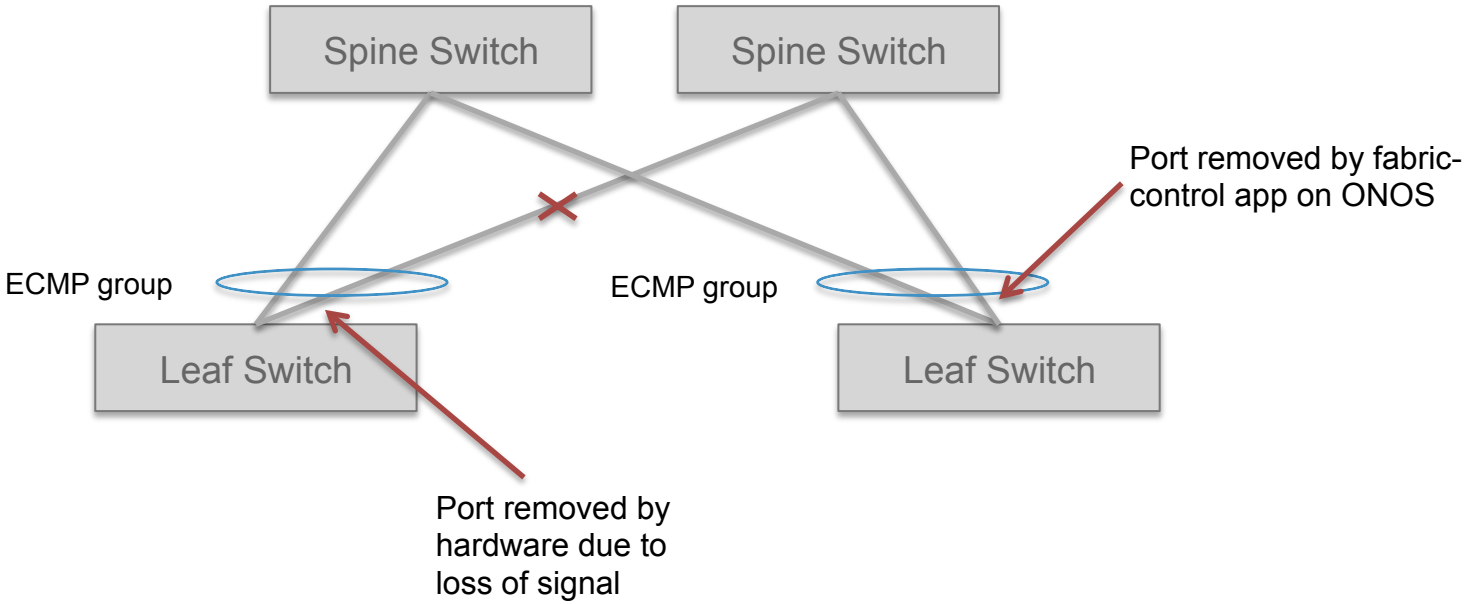
Open-source, SDN-based
Datacenter Leaf-Spine Fabric



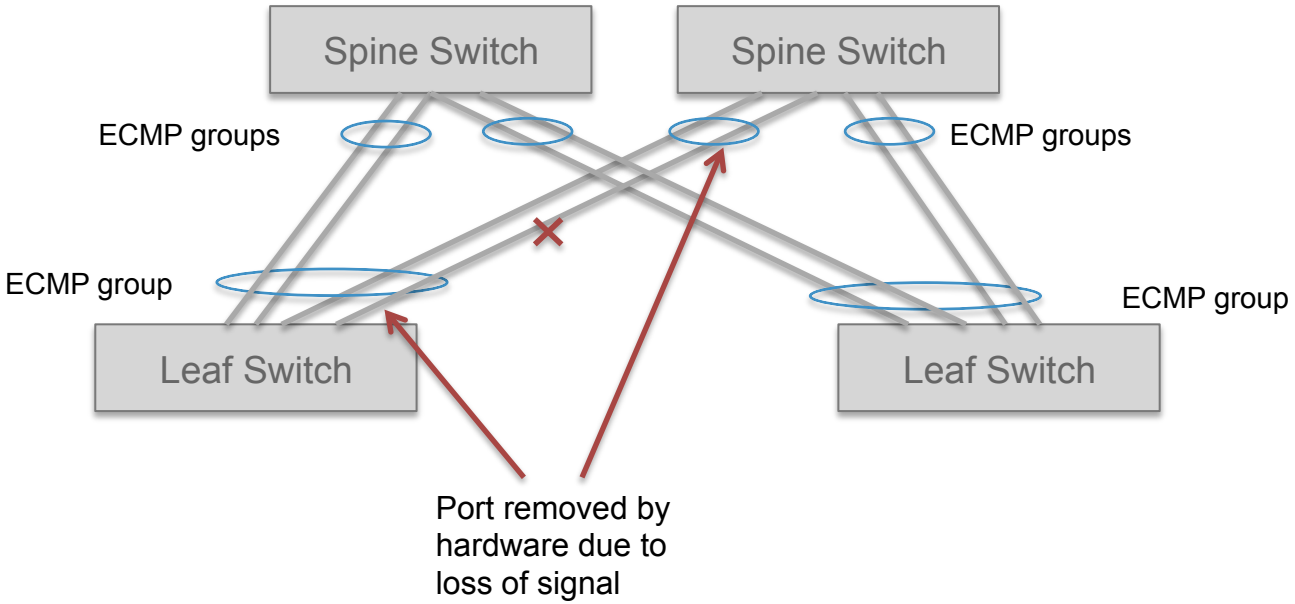
High Availability in Disaggregated Networks

- **Control plane redundancy**
- Data plane redundancy
- Combined data-plane & control-plane failure recovery
- Dual-homing servers
- vRouter HA
- Headless fabric

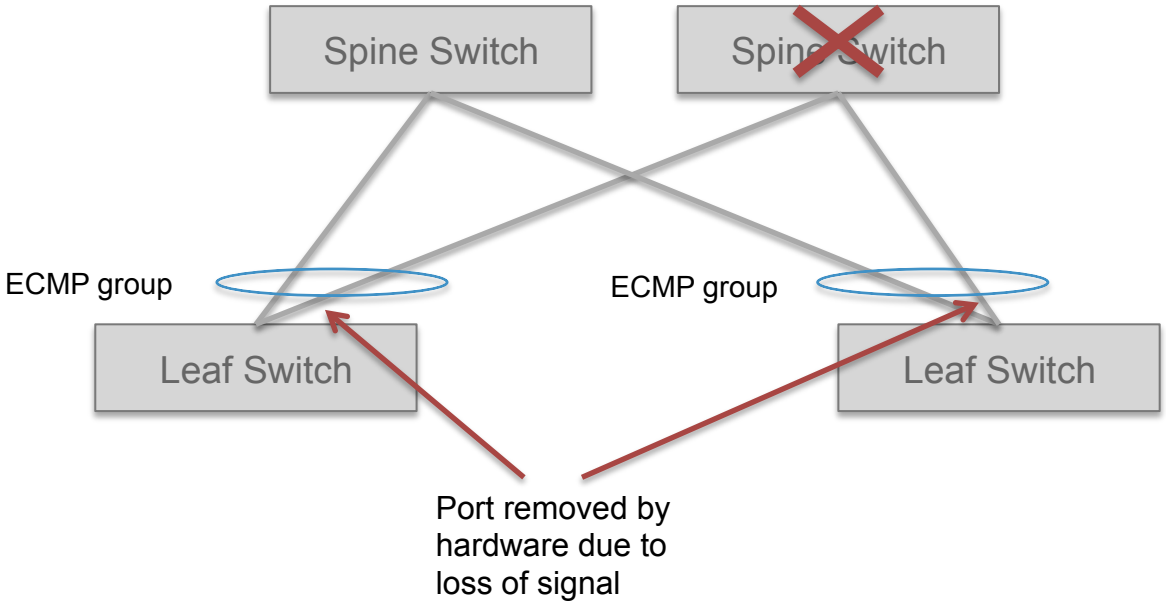
Data Plane Failures – Losing a Link



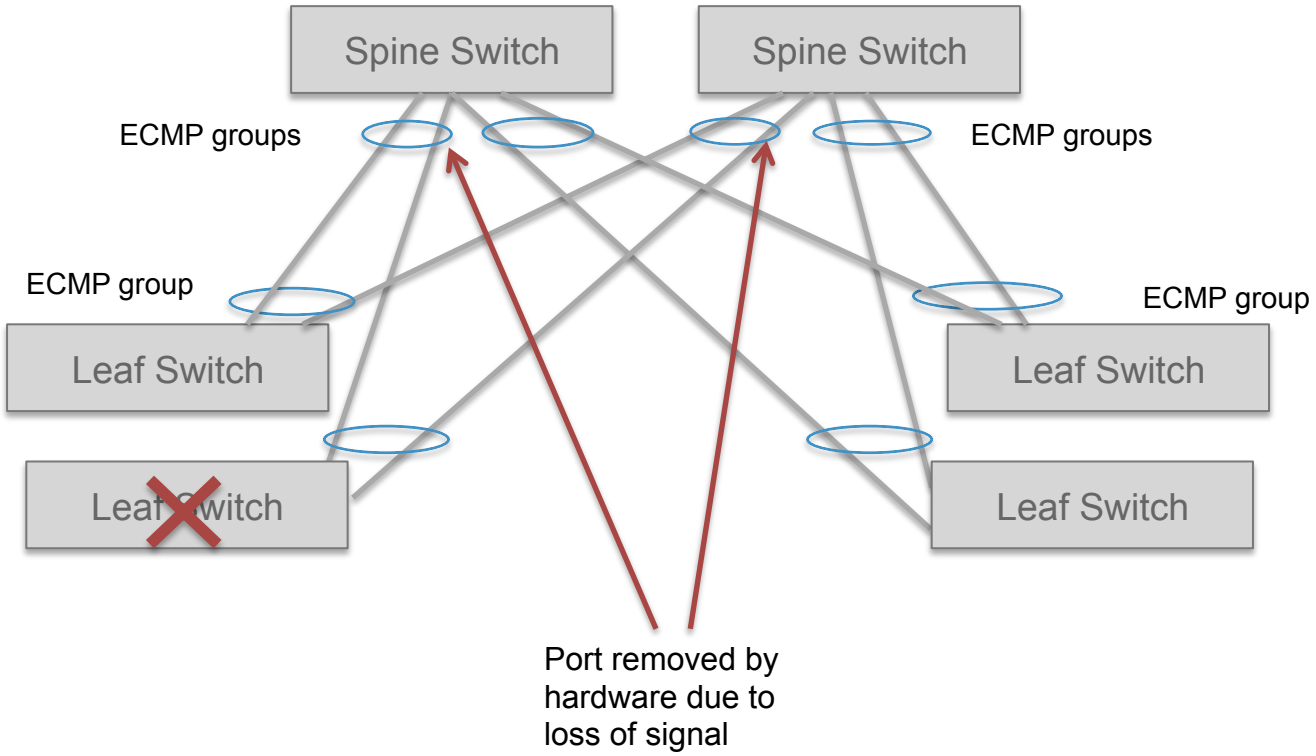
Data Plane Failures – Losing a Link



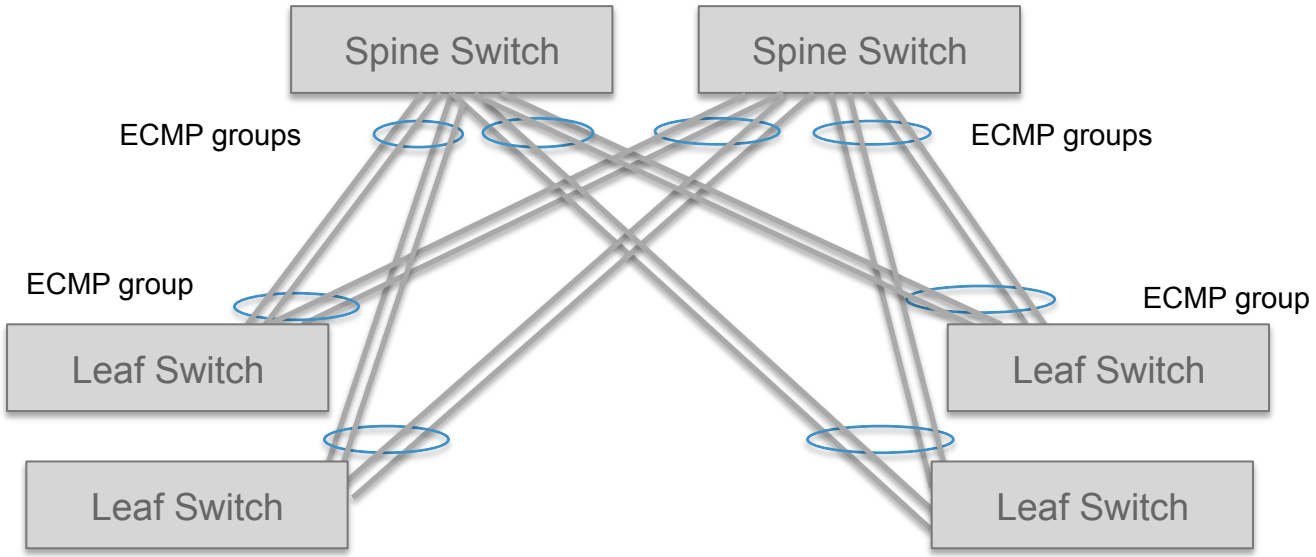
Data Plane Failures – Losing a Spine



Data Plane Failures – Losing a Leaf



Data Plane Failures

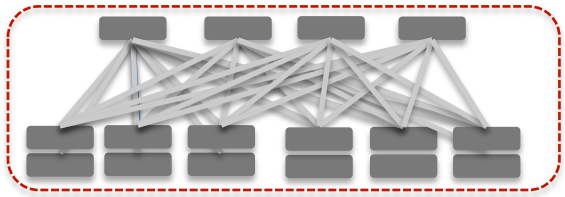


With dual-ToRs, dual-uplinks to same spine, and the use of hash-groups, all single data-plane failures can be handled entirely in hardware without the involvement of the controller

Outline

Example Disaggregated Network

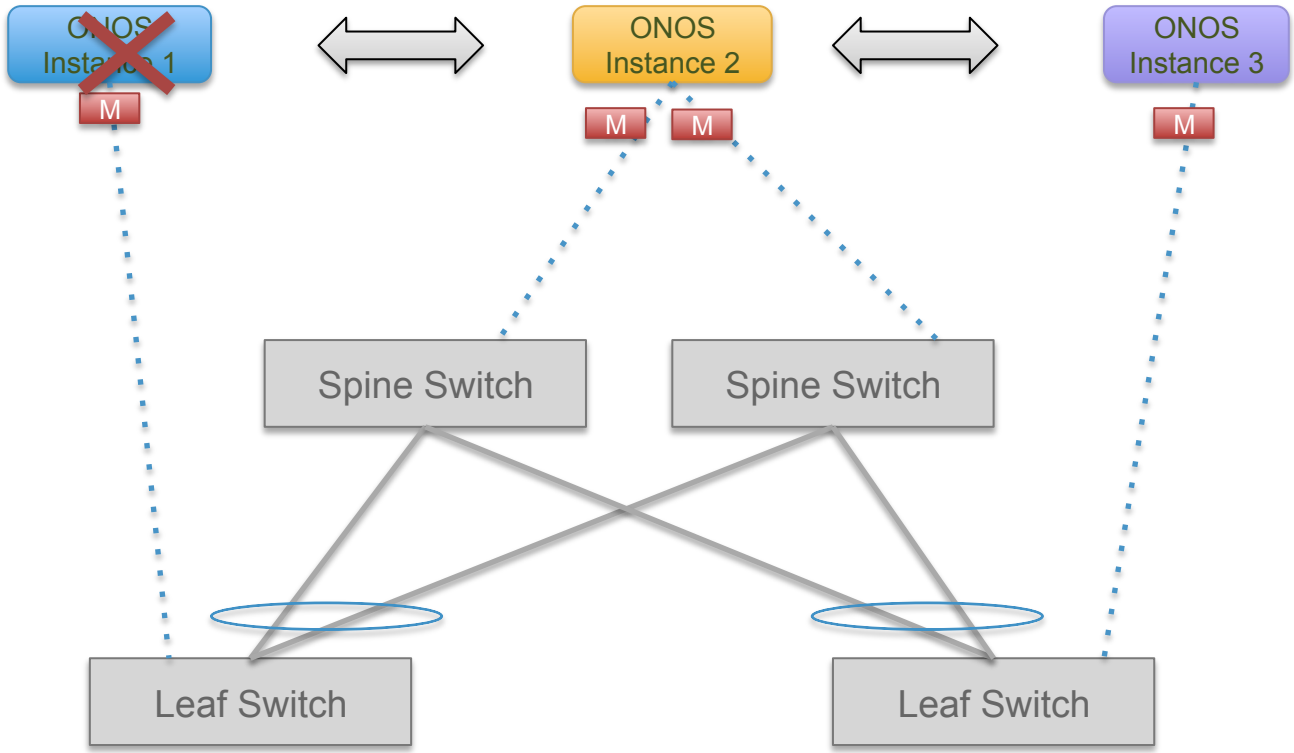
Open-source, SDN-based
Datacenter Leaf-Spine Fabric



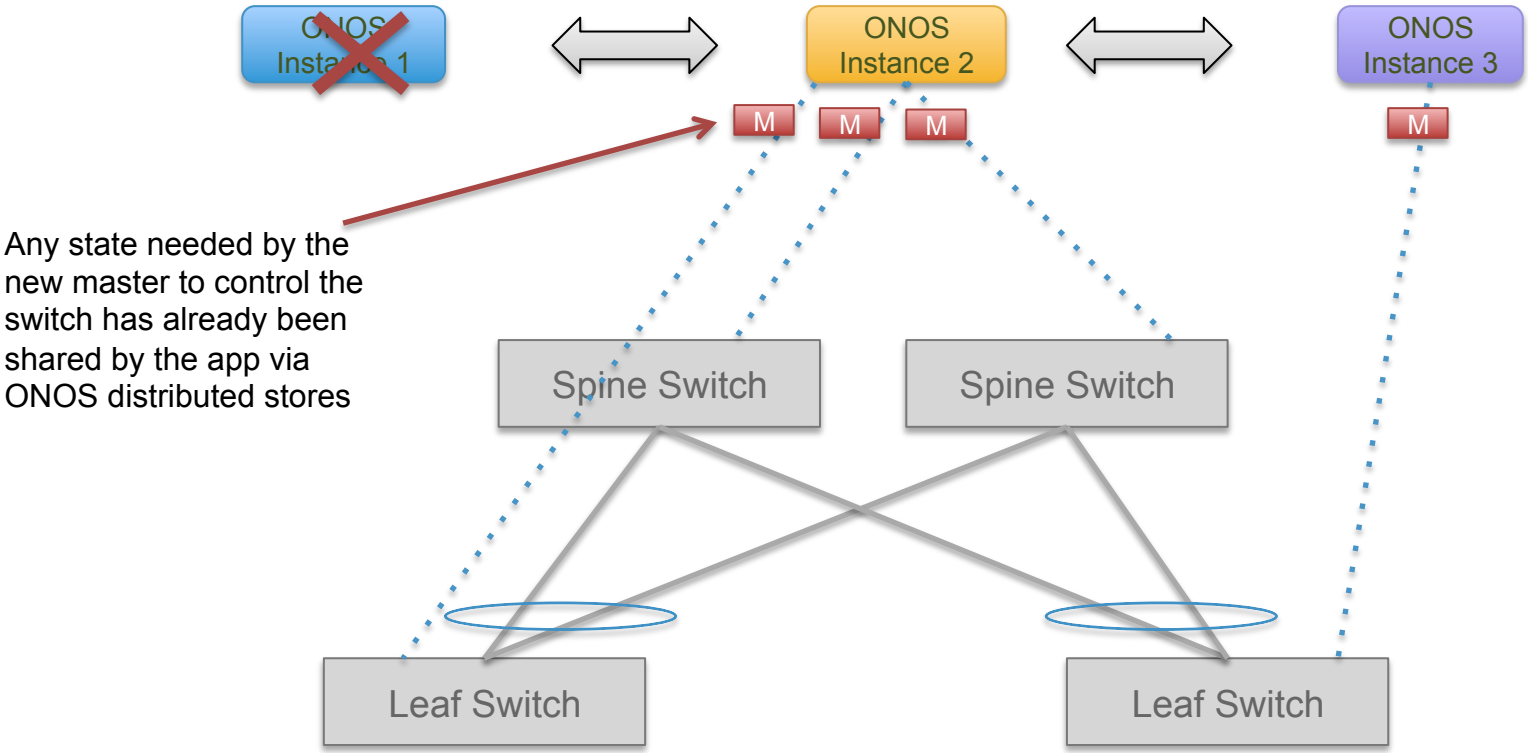
High Availability in Disaggregated Networks

- Control plane redundancy
- **Data plane redundancy**
- Combined data-plane & control-plane failure recovery
- Dual-homing servers
- vRouter HA
- Headless fabric

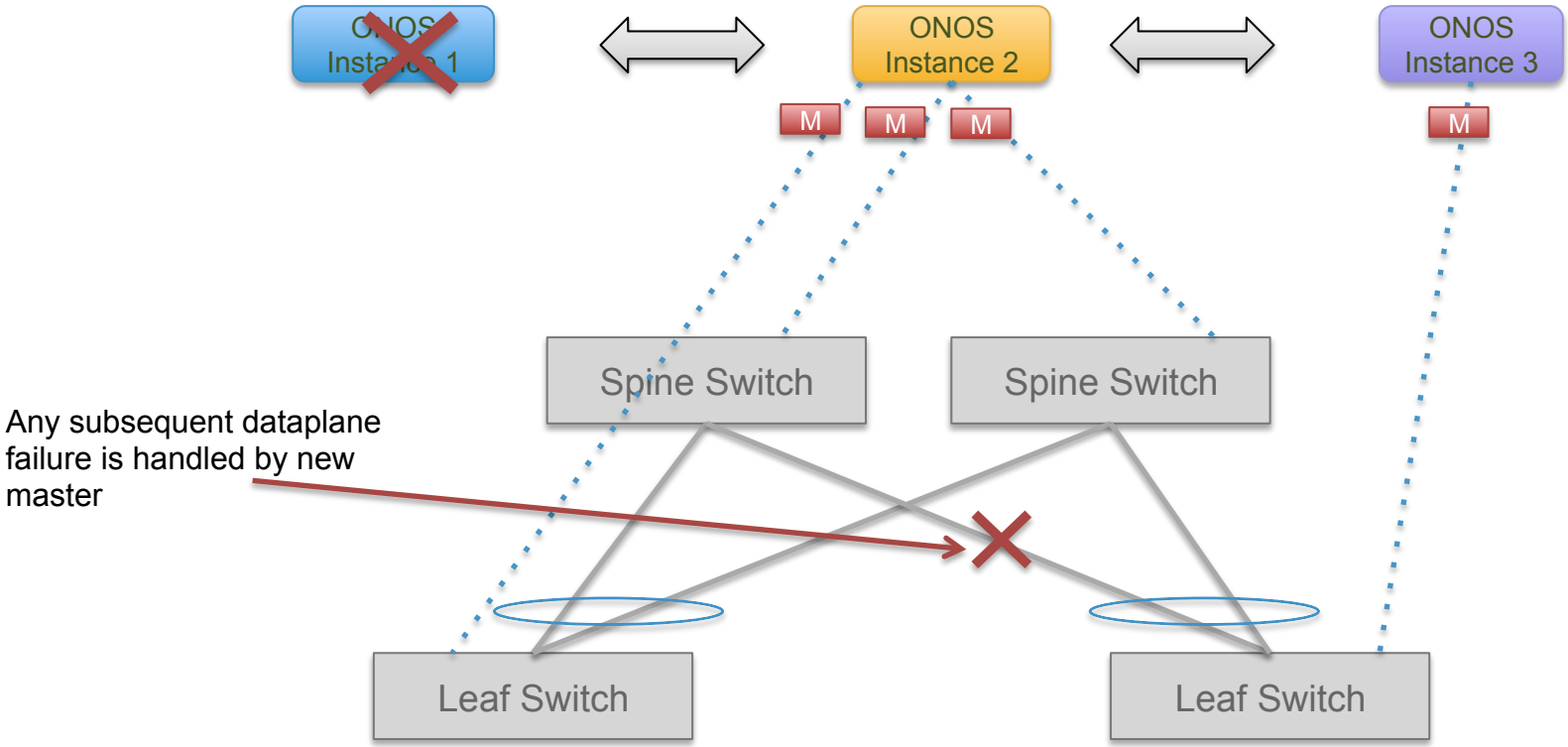
Combined DP & CP Failures



Combined DP & CP Failures



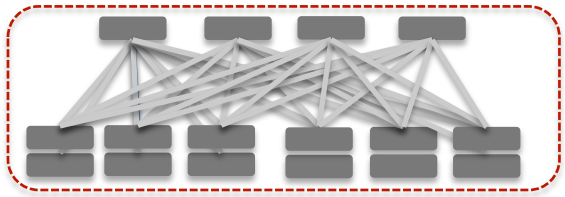
Combined DP & CP Failures



Outline

Example Disaggregated Network

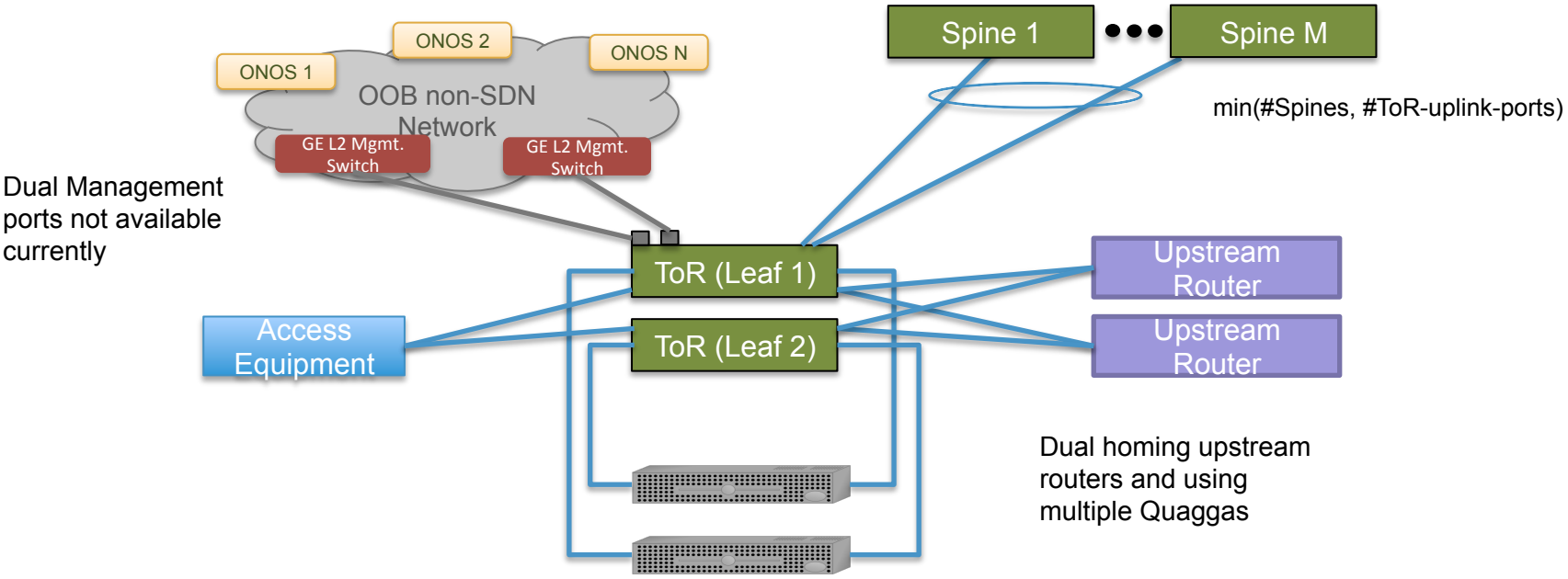
Open-source, SDN-based
Datacenter Leaf-Spine Fabric



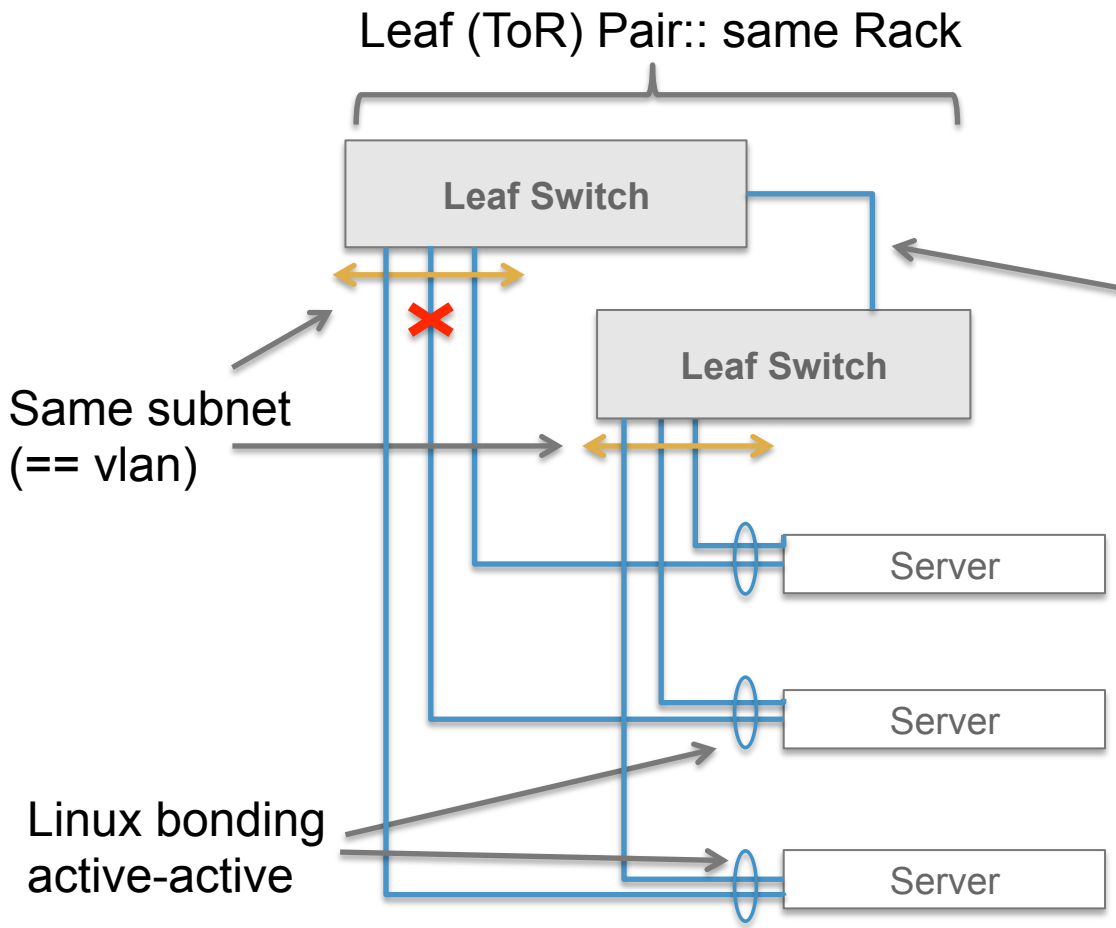
High Availability in Disaggregated Networks

- Control plane redundancy
- Data plane redundancy
- **Combined data-plane & control-plane failure recovery**
- Dual-homing servers
- vRouter HA
- Headless fabric

2-way Redundancy Everywhere Else



Server Dual Homing



Bridging:

- Unicast & Broadcast traffic
- Redirection over **pair-link** in the event of failure

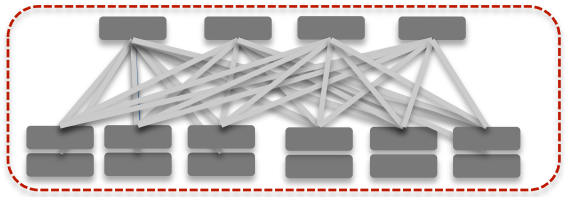
Routing:

- Access port failure handled as a localized change (using pair-link)
- Does not require rerouting elsewhere in the fabric.

Outline

Example Disaggregated Network

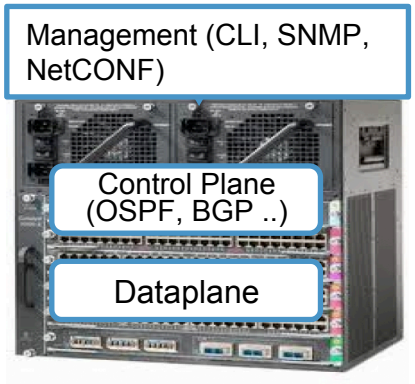
Open-source, SDN-based
Datacenter Leaf-Spine Fabric

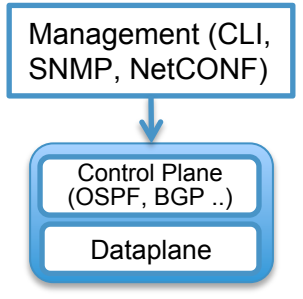


High Availability in Disaggregated Networks

- Control plane redundancy
- Data plane redundancy
- Combined data-plane & control-plane failure recovery
- **Dual-homing servers**
- vRouter HA
- Headless fabric

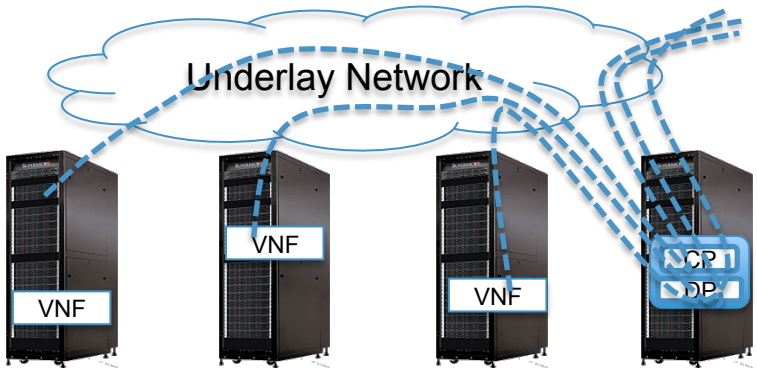
vRouter in Software?



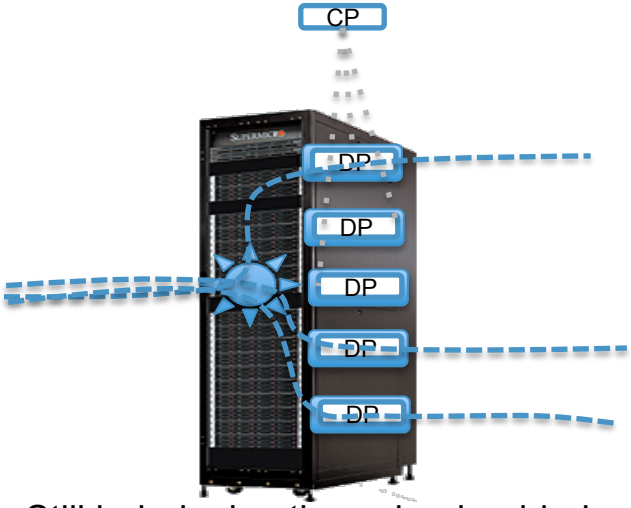


VNFM
(VNF Manager)

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



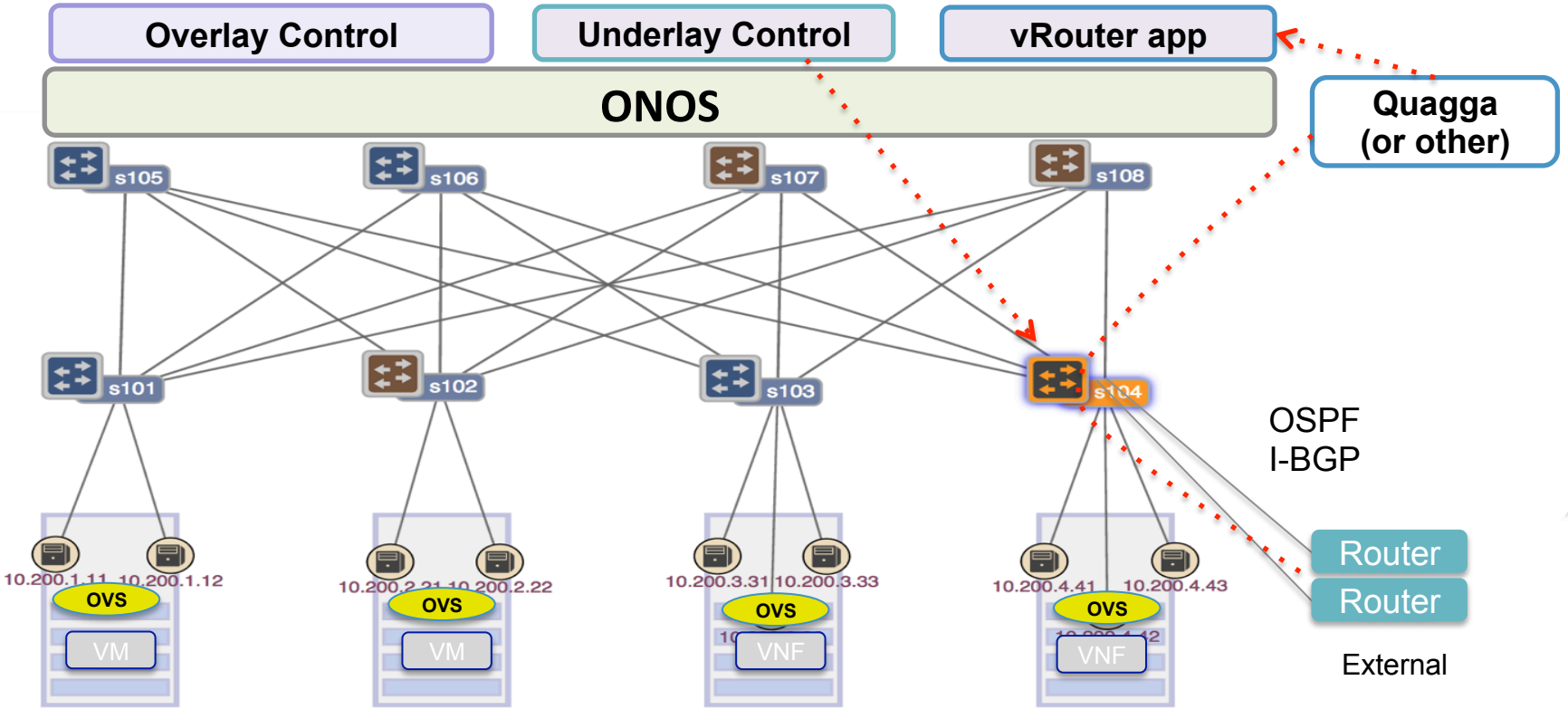
vRouter
VM



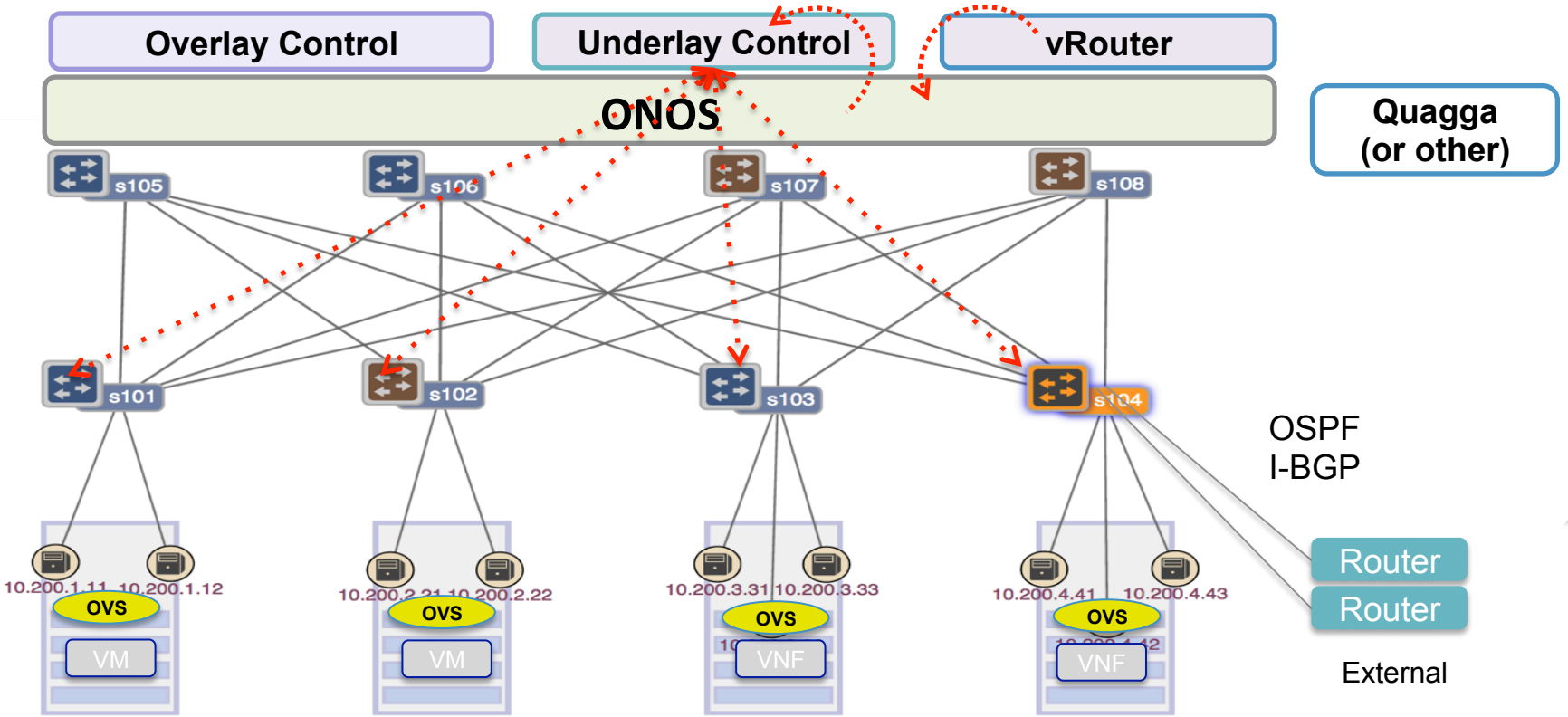
Issues: Hairpinning
Embedded control plane complicates scale-out

Issue: Still hairpinning through a load-balancer

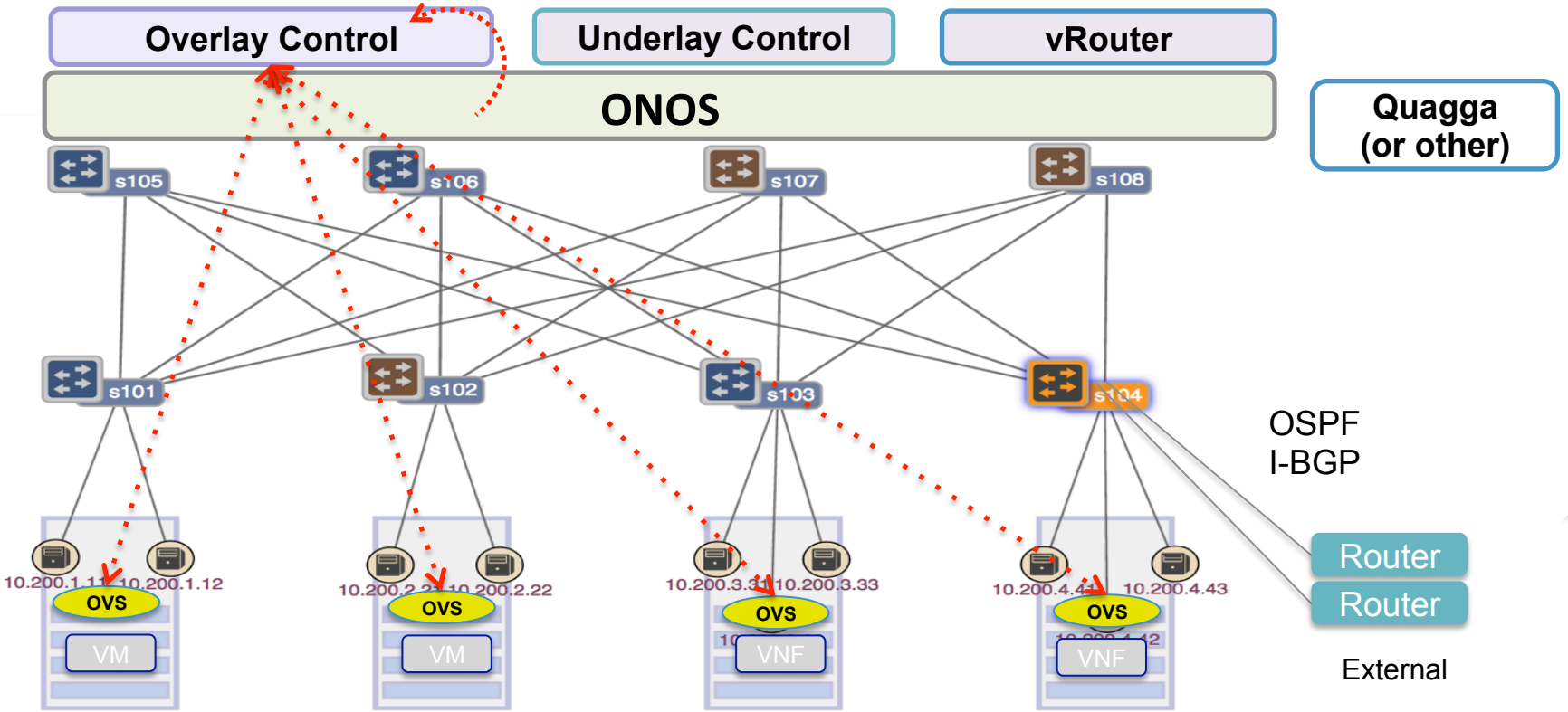
vRouter Operation



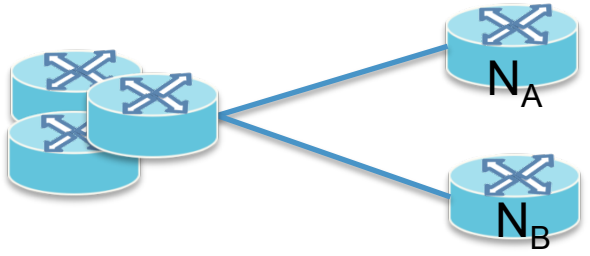
vRouter Operation

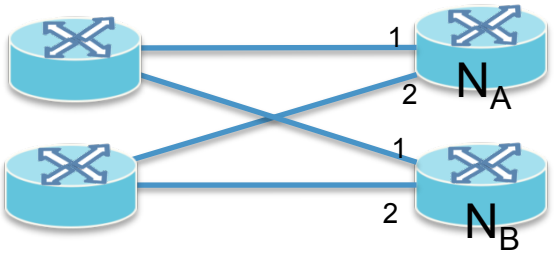


vRouter Operation

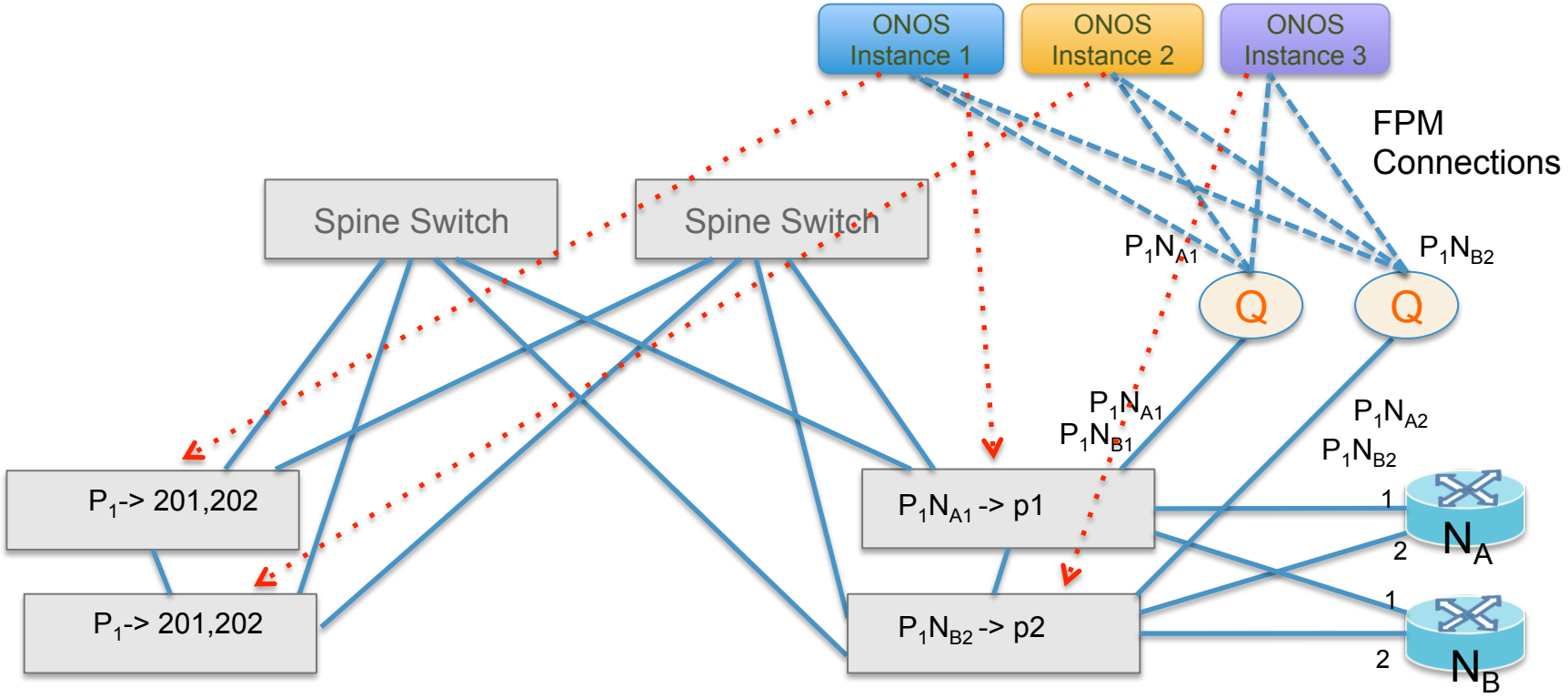


vRouter Dual-Homing





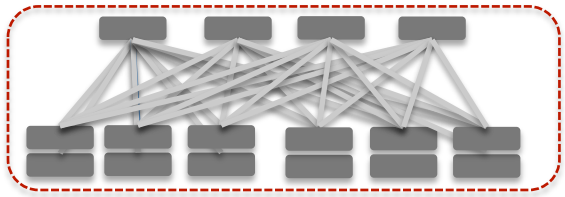
vRouter Dual-Homing



Outline

Example Disaggregated Network

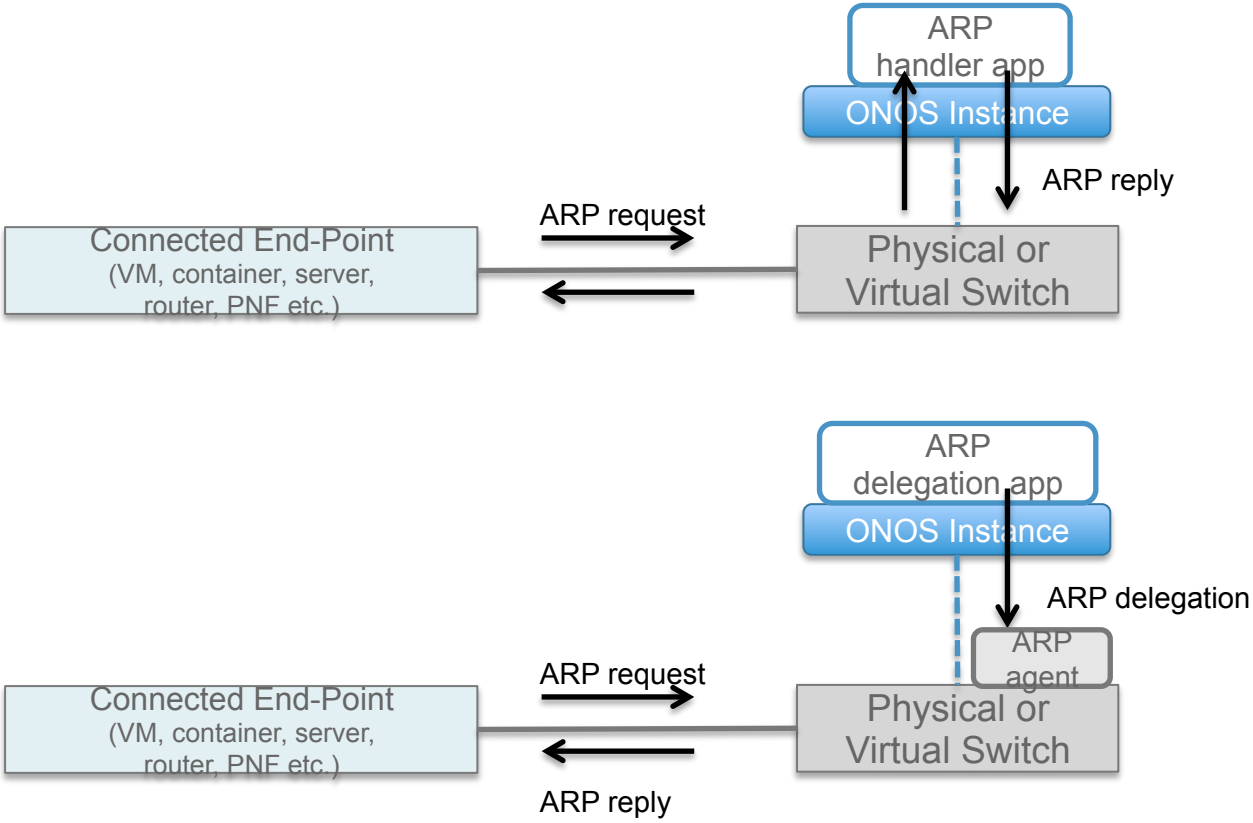
Open-source, SDN-based
Datacenter Leaf-Spine Fabric



High Availability in Disaggregated Networks

- Control plane redundancy
- Data plane redundancy
- Combined data-plane & control-plane failure recovery
- Dual-homing servers
- **vRouter HA**
- Headless fabric

Headless Mode



Summary

- Disaggregated Network – Bare-metal + Open-source + SDN-based Leaf-Spine Fabric
  - Vlan, IPv4, IPv6, MPLS SR, IPv4 Multicast, DHCP Relay, vRouting
- ONOS is a Network Element, not a management station
  - ONOS strength: N-way redundancy & scale
  - 3R recovery: Redistribute mastership, Retry connections, Reboot instances
  - Low probability of all controller instances dying simultaneously
- 2-way redundancy everywhere else
  - Dual-ToRs, Servers-to-ToRs, Routers-to-ToRs, RPYs-to-ToRs, Dual-management ports
  - Acceptable in most/all networking scenarios
- Headless mode: in the unlikely event that all controllers are lost
  - Existing network & services continue to operate as before
    - Even if there are subsequent data-plane failures
    - Requires dual uplinks to same spine, local agents for ARP delegation, and pair-link reroute.
  - New services, subscribers and routes, hosts (dhcp) cannot be provisioned
    - Acceptable/tolerated in interim state