



@rony358 

OpenStack Scale-out Networking Architecture

Scaling to 1,000+ servers without Neutron

Abhishek Chanda, Software Engineer
OpenStack Juno Design Summit
May 13th, 2014

Abhishek Chanda






- Engineer at Cloudscaling
- Started with a focus on software defined networking and distributed systems
 - *Ended up as jack of all trades!*
 - *Still bad at configuring network devices!*

Today's Goals

- Our Layer-3 (L3) network architecture
 - Motivation
 - Design goals
 - Network topology
 - Software components layout
 - Under the hood
 - Challenges Faced
 - Future Directions (SDN, Neutron etc.)

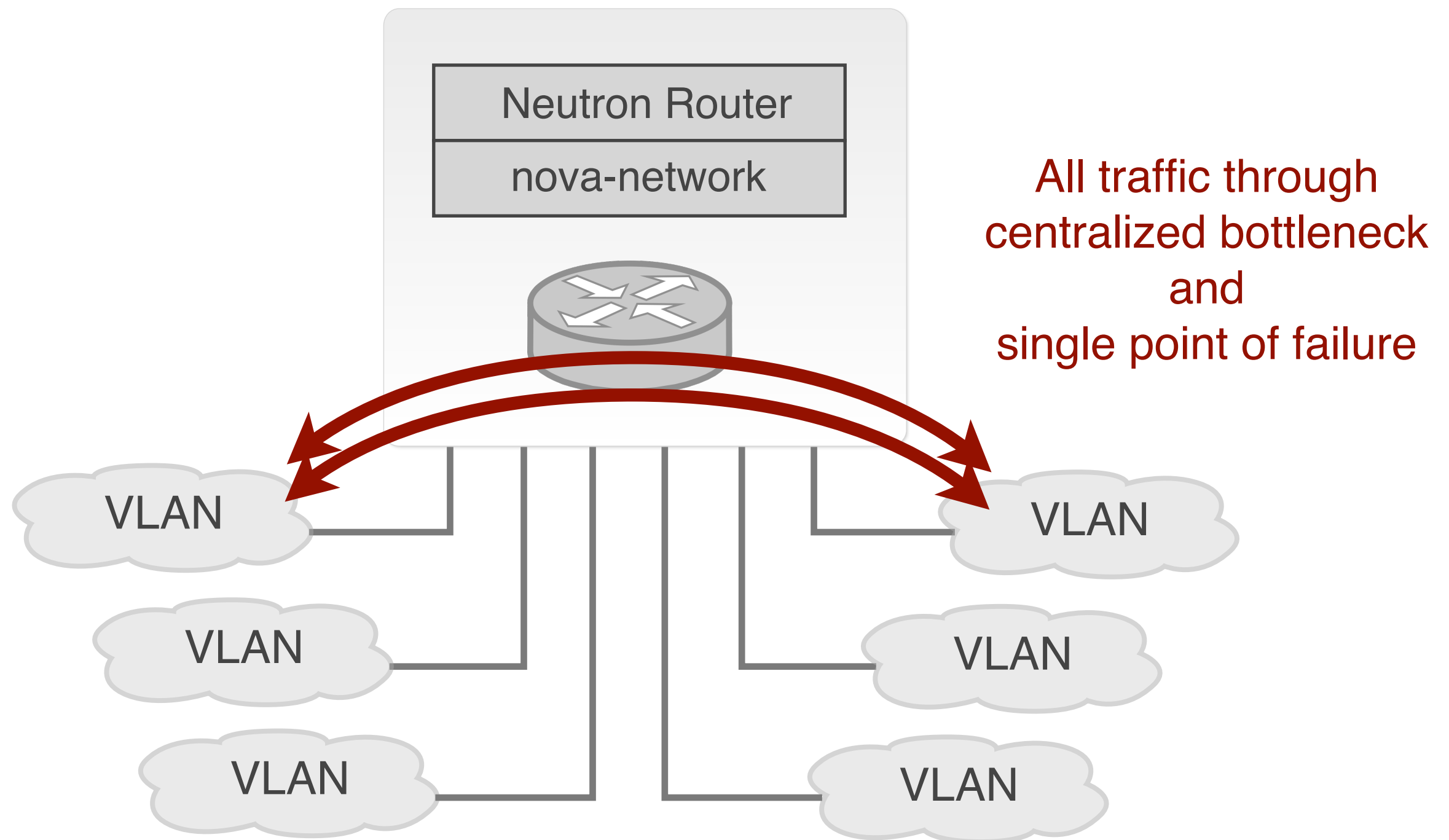
Networking Modes in Vanilla OpenStack using nova-network

OpenStack Networking Options

	Single-Host OpenStack Networking	Multi-Host OpenStack Networking	Flat OpenStack Networking	OCS Classic Networking	OCS VPC Networking
Scalability & Control					
Reference Network Architecture	Layer-2 VLANs w/ STP	Layer-2 VLANs w/ STP	Layer-2 VLANs w/ STP	L3 Spine & Leaf + scale-out NAT	L3 Spine & Leaf Underlay + network virtualization
Design Pattern	Centralized network stack	Decentralized network stack	Centralized network stack	Fully Distributed	Fully Distributed
Core Design Flaw	All traffic through a single x86 server	NAT at Hypervisor; 802.1q tagging	All traffic through a single x86 server	Requires deeper networking savvy	More bleeding edge (scale not proven)
Issues	SPOF; Performance bottleneck; No SDN	Security & QoS issues; No SDN	No control plane scalability; No SDN	No virtual networks	No VLAN support

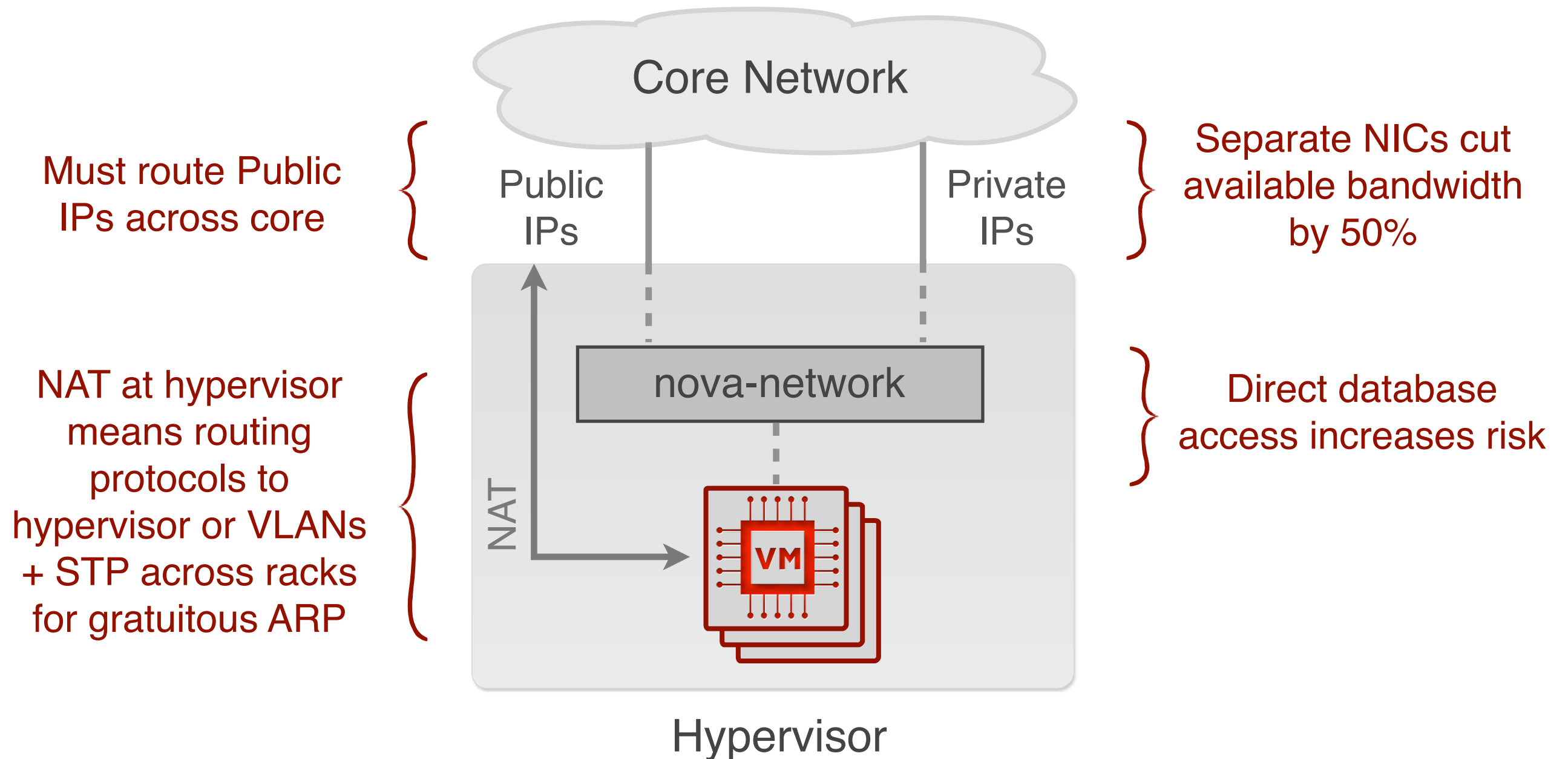
Option 1 - OpenStack Single-Host

Performance bottleneck; non-standard networking arch.



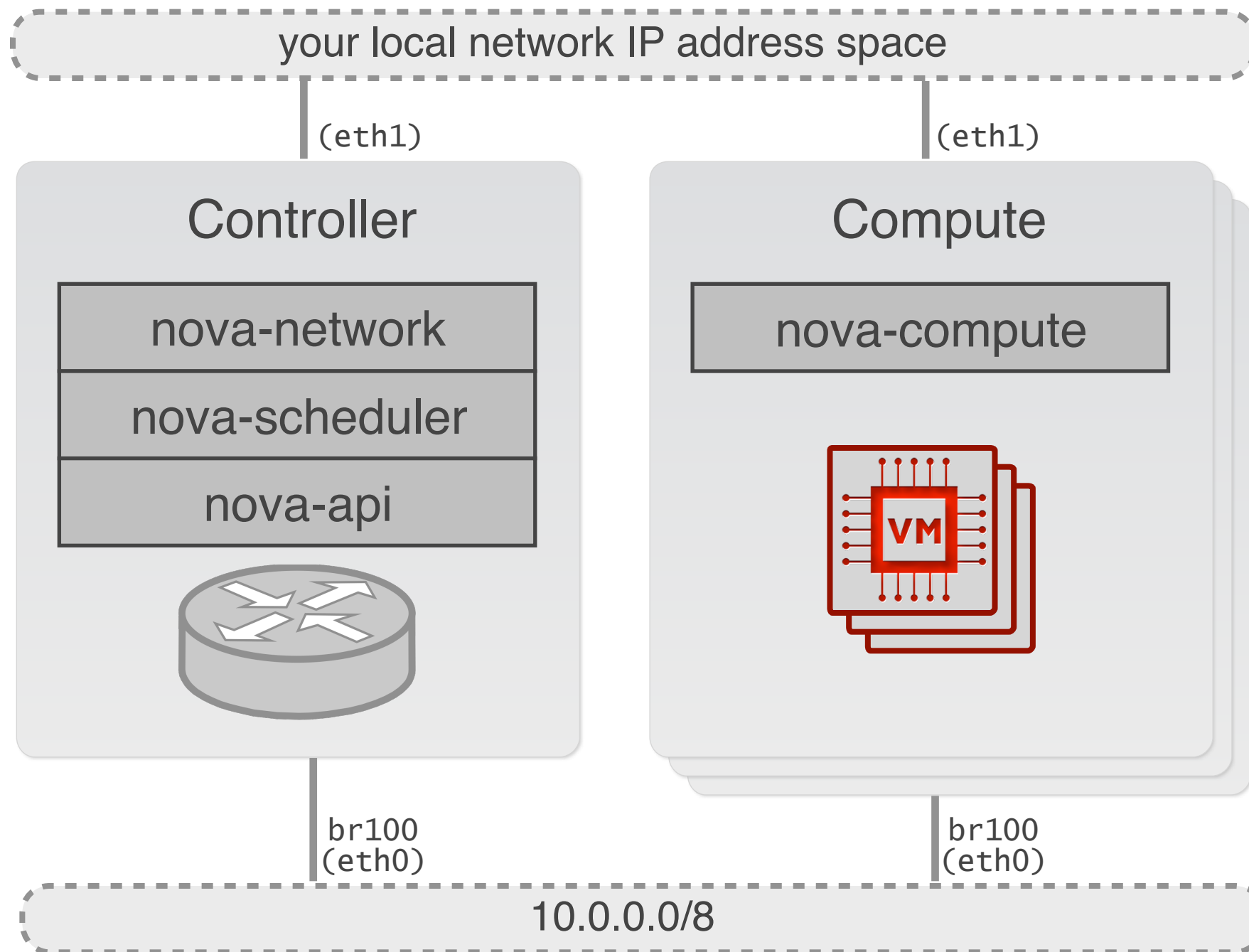
Option 2 - OpenStack Multi-Host

Security problem; non-standard networking arch.



Option 3 - OpenStack Flat

Networking scalability, but no control plane scalability



Uses ethernet adapters configured as bridges to allow network traffic between nodes

Commonly used in POC and development environments

What Path Have Others Chosen?

- HP, CERN, ANL and others
- CERN uses a custom driver which talks to a DB that maps IP to MAC addresses (amongst other attributes)
 - Essentially flat with manually created VLANs assigned to specific compute nodes
- ANL added InfiniBand and VxLAN support to nova-network

The L3 Network Architecture in OCS

Design Goals

- Mimic Amazon EC2 “classic” networking
- Pure L3 is blazing fast and well understood
 - Network/systems folks can troubleshoot easily
- No VLANs, no Spanning Tree Protocol (STP), no L2 mess
- No SPOFs, smaller failure domains
 - E.g. single_host & flat mode
- Distributed “scale-out” DHCP
- No virtual routers!
 - Path is vm->host->ToR->core

This enables a horizontally scalable stateless NAT layer that provides floating IPs

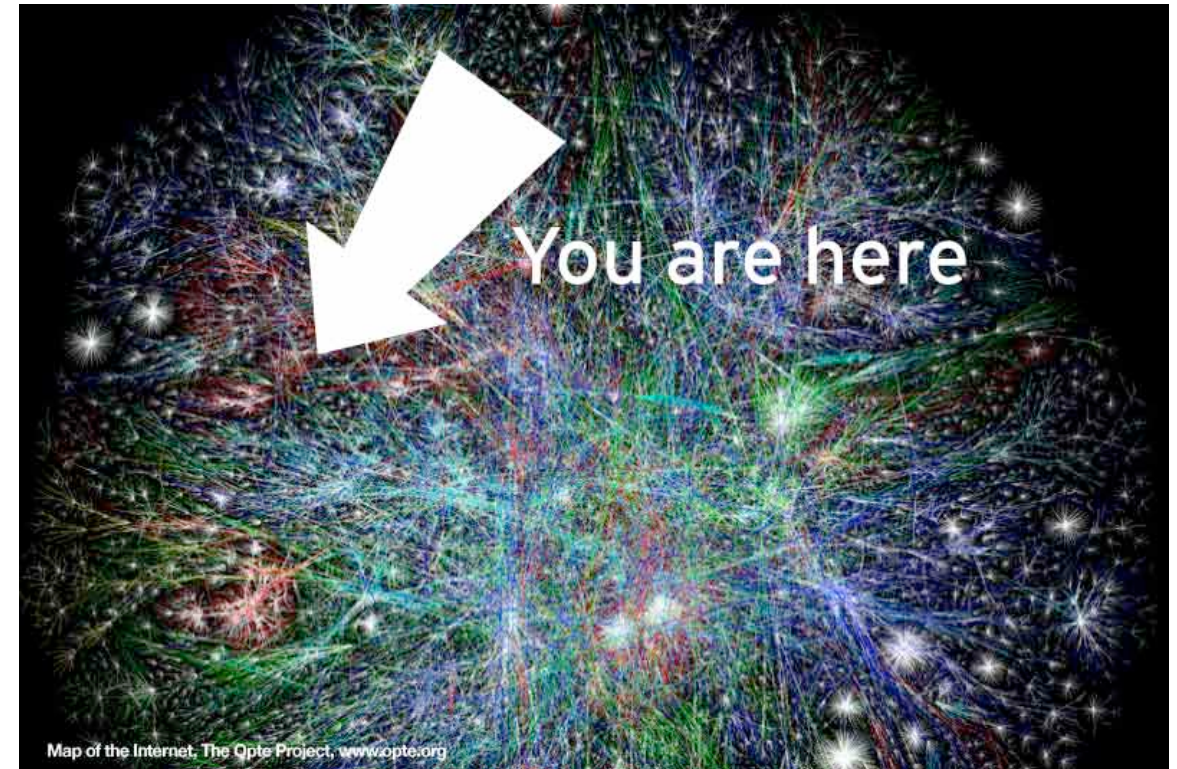
Layer 3 (L3) Networks Scale

The Internet Scales!

**Largest cloud operators are
L3 with NO VLANs**

**Cloud-ready apps don't
need or want VLANs (L2)**

**An L3 networking model is
ideal underlay for SDN
overlay**



Why NOT L2?

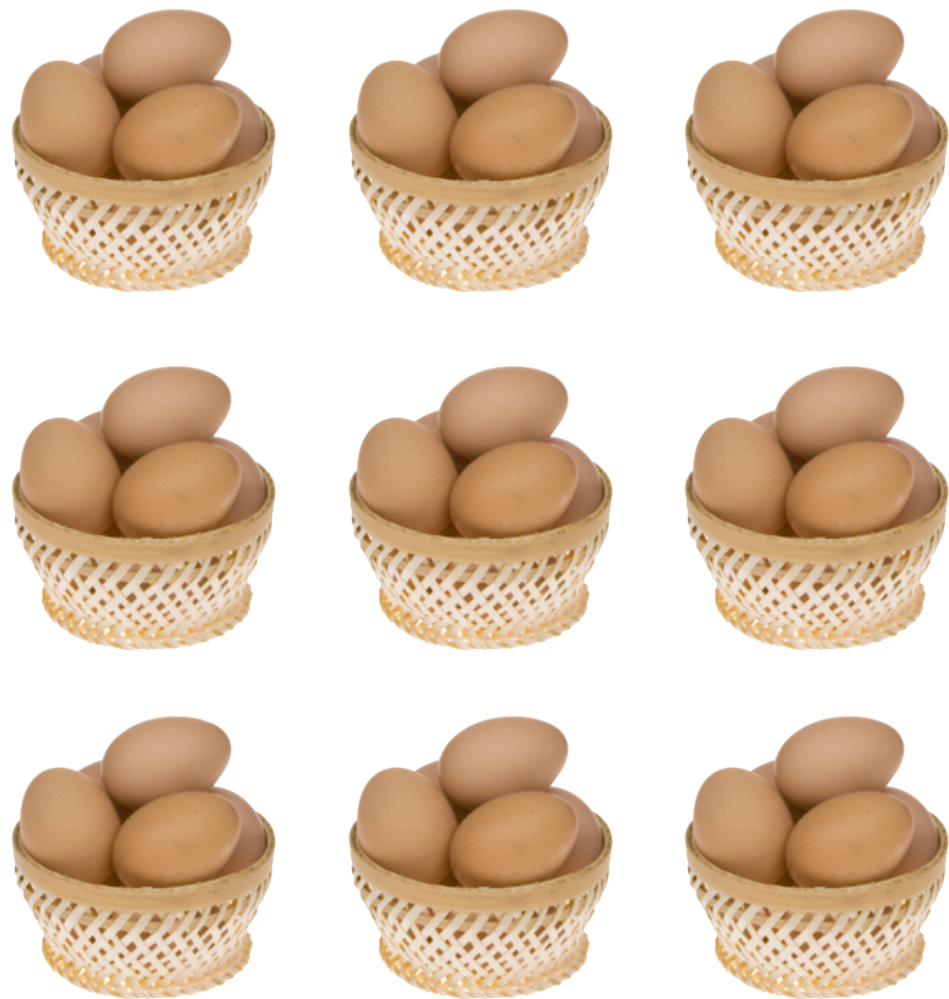
L3

L2

Hierarchical topology	Flat topology
Route aggregation	No route aggregation / everything everywhere
Fast convergence times	Fast convergence times only when tuned properly
Locality matters	Locality disappears
Use all available bandwidth (via ECMP) using multiple paths	Uses half of available bandwidth and most traffic takes a long route
Proven scale	STP/VLANs work at small to medium scale only
The Internet (& ISPs) are L3 oriented	Typical enterprise datacenter is L2 oriented
Best practice for SDN “underlay”	SDN “overlay” designed to provide L2 virt. nets

Smaller Failure Domains

Would you rather have the whole cloud down or just a small bit of it for a short time?



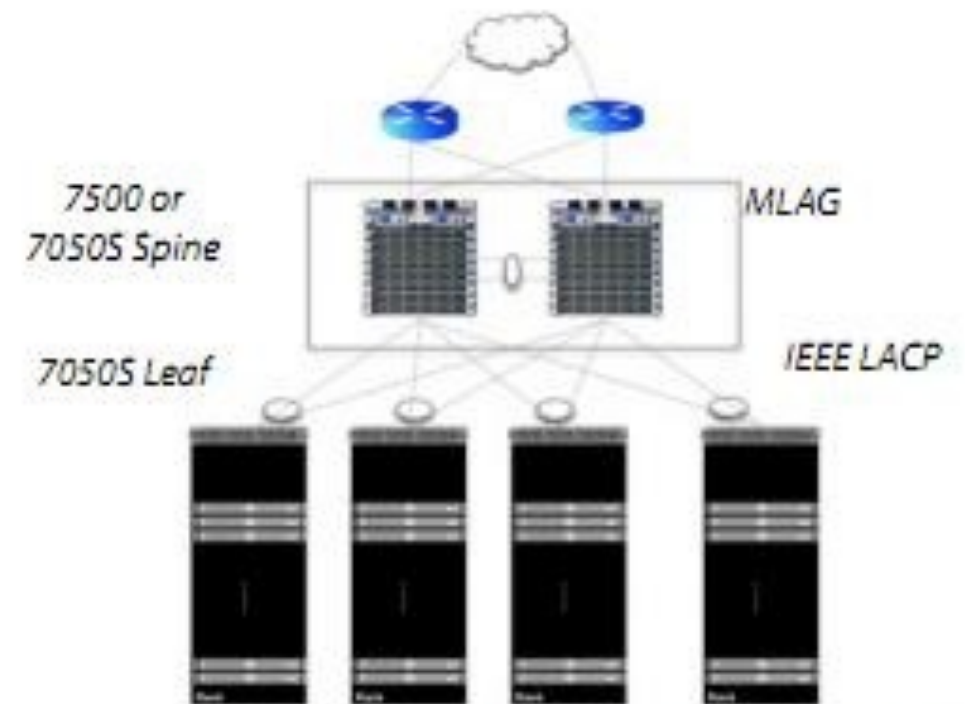
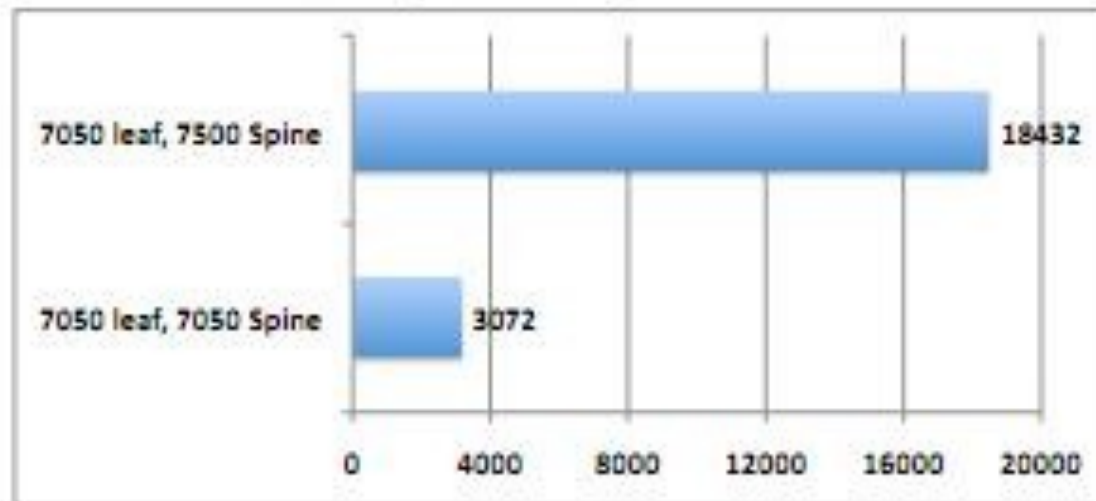
vs



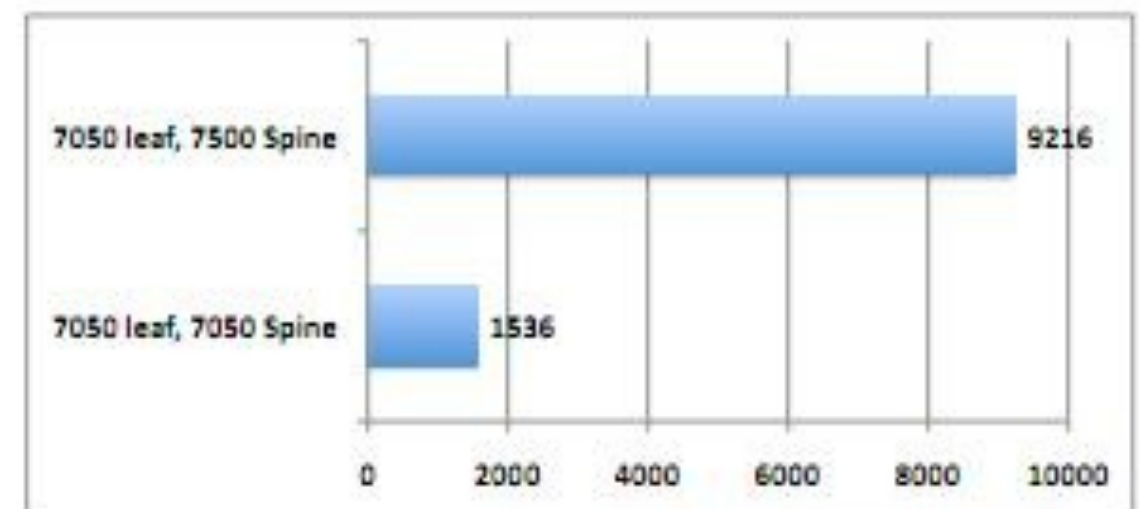
Spine & Leaf



Arista Leaf-Spine Design with L3 ECMP

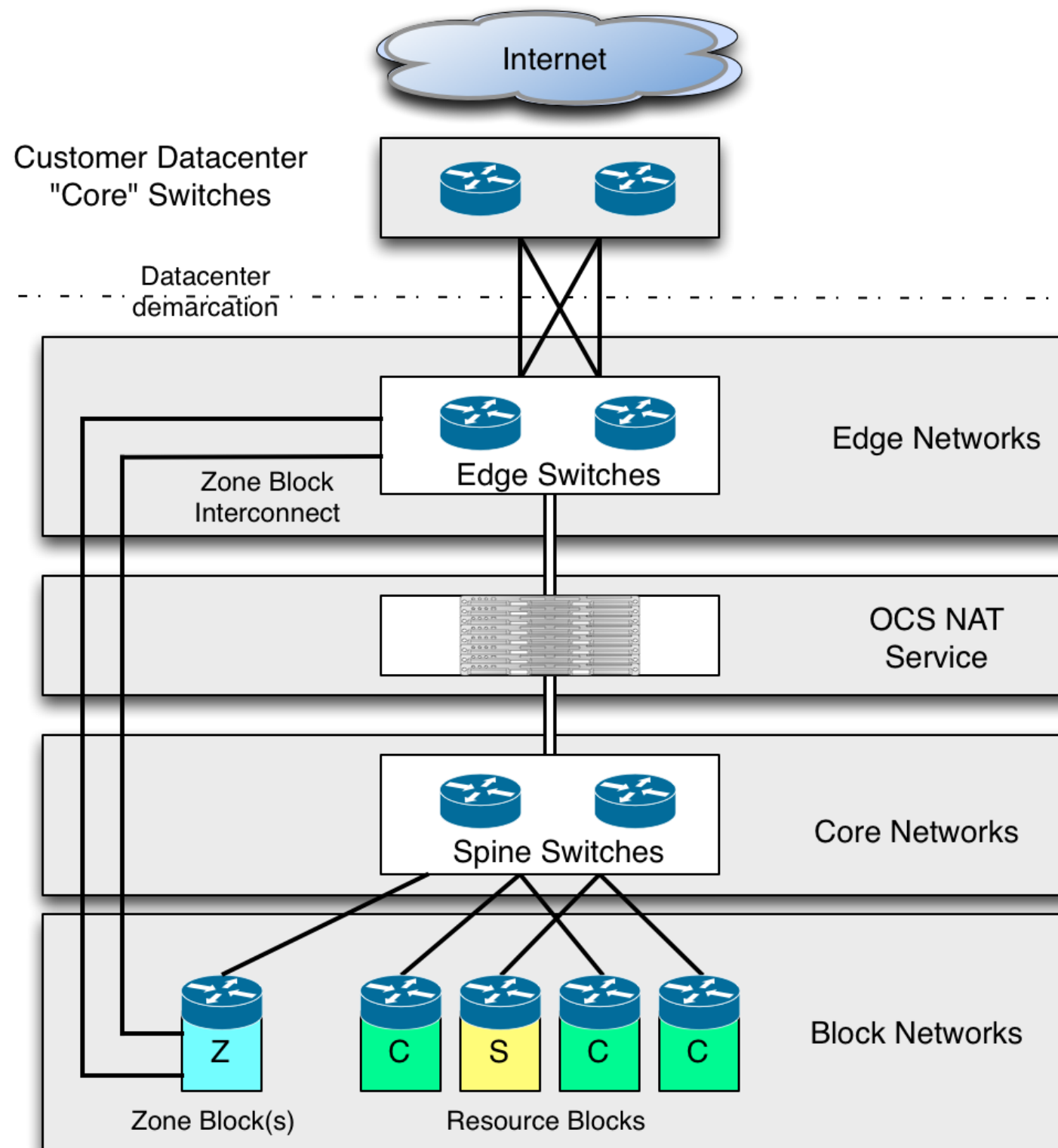


Arista Leaf-Spine Design with L2 MLAG



Number of 10GbE Nodes Interconnected Using Arista Leaf-Spine Designs

Simple Network View



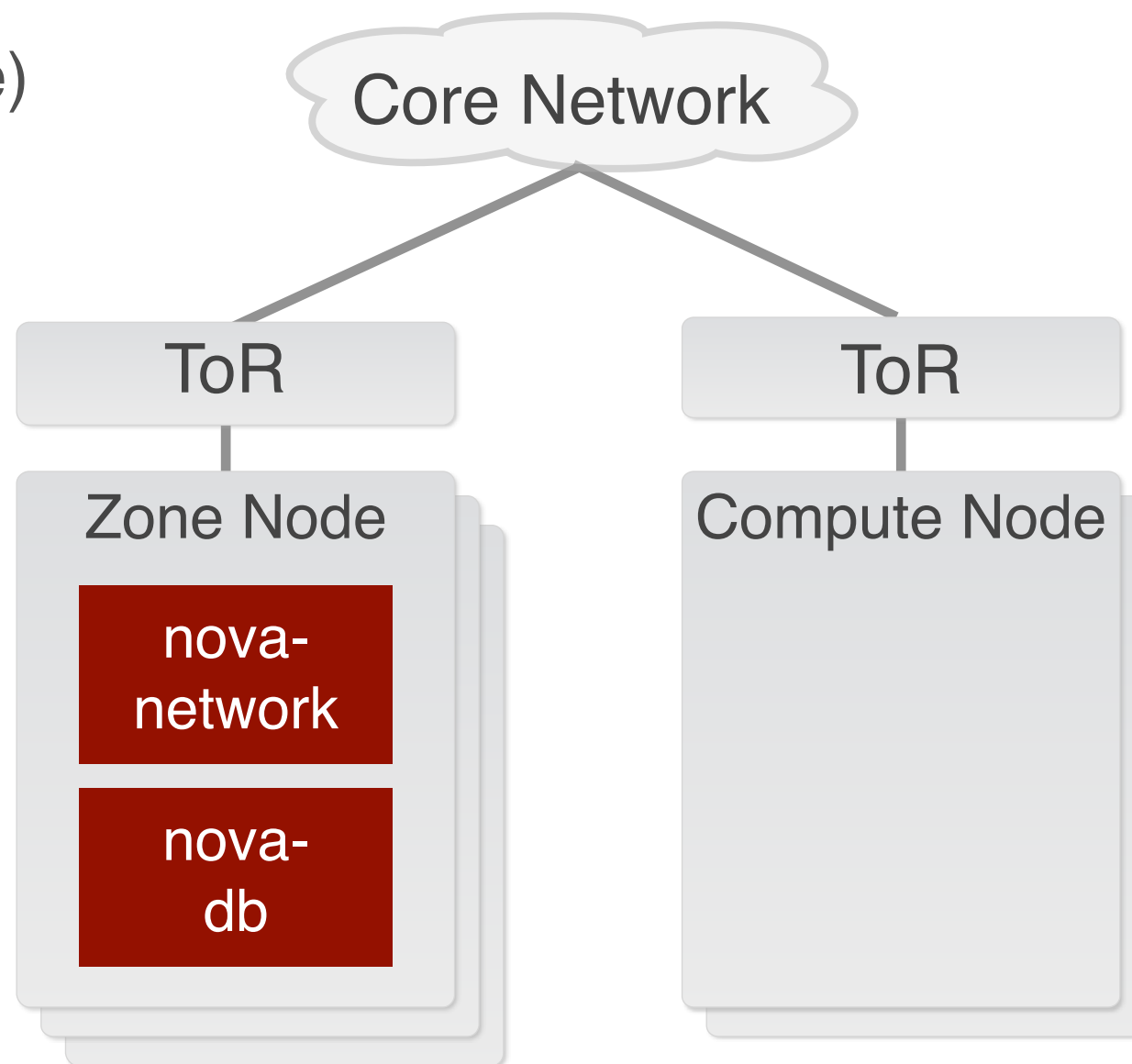
Software Components Schematic Layout

- Nova Network is distributed & synchronized

- Means we can have many running at once)
- This drives horizontal network scalability by adding more network managers

Hardware Components

Software Components

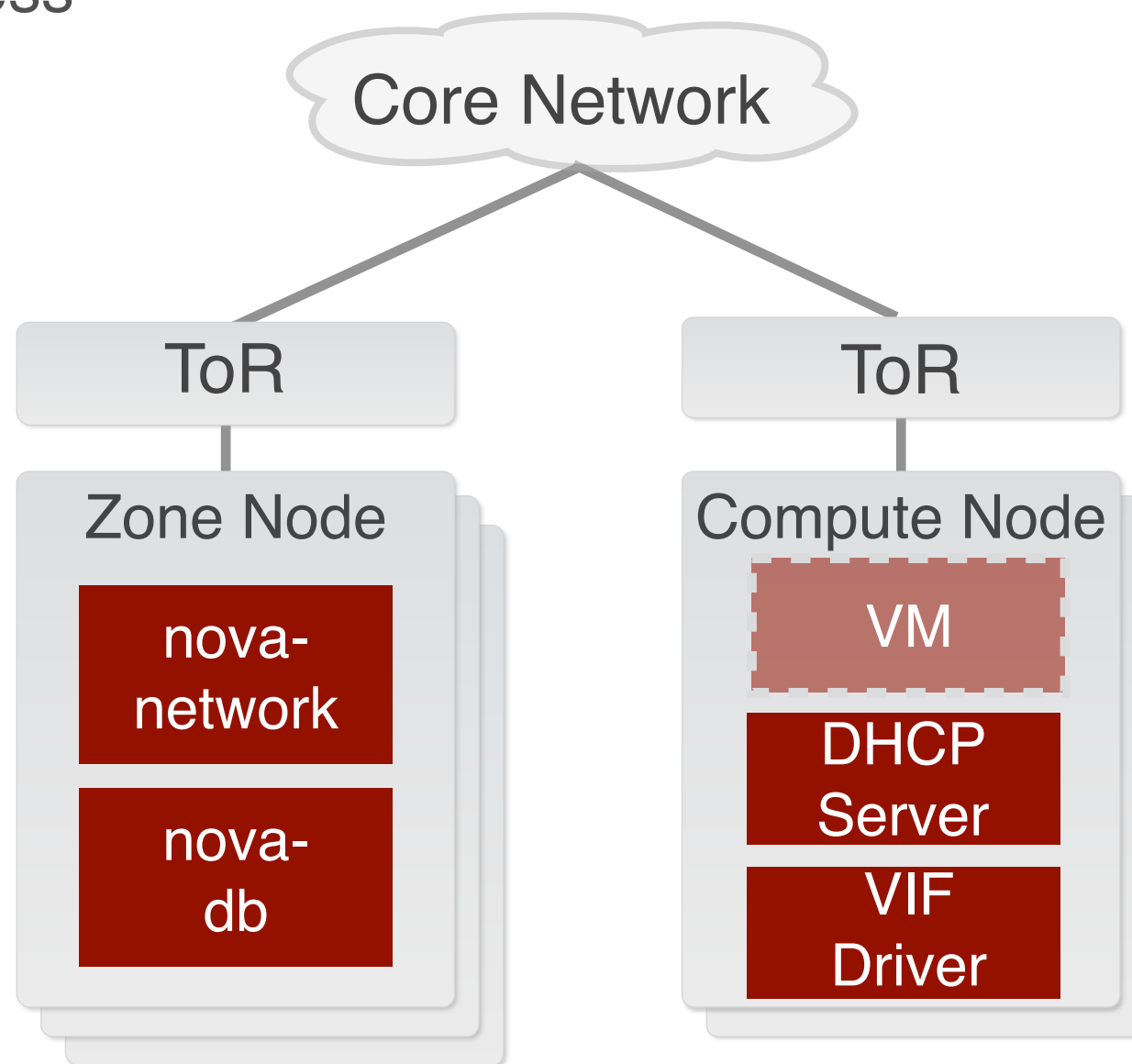


Software Components Schematic Layout

- VIF driver on each compute node
 - Bridge creation on each vm (/30)
 - Enhanced iptables rules
 - Per vm udhcpd process
 - Configures routing

*Hardware
Components*

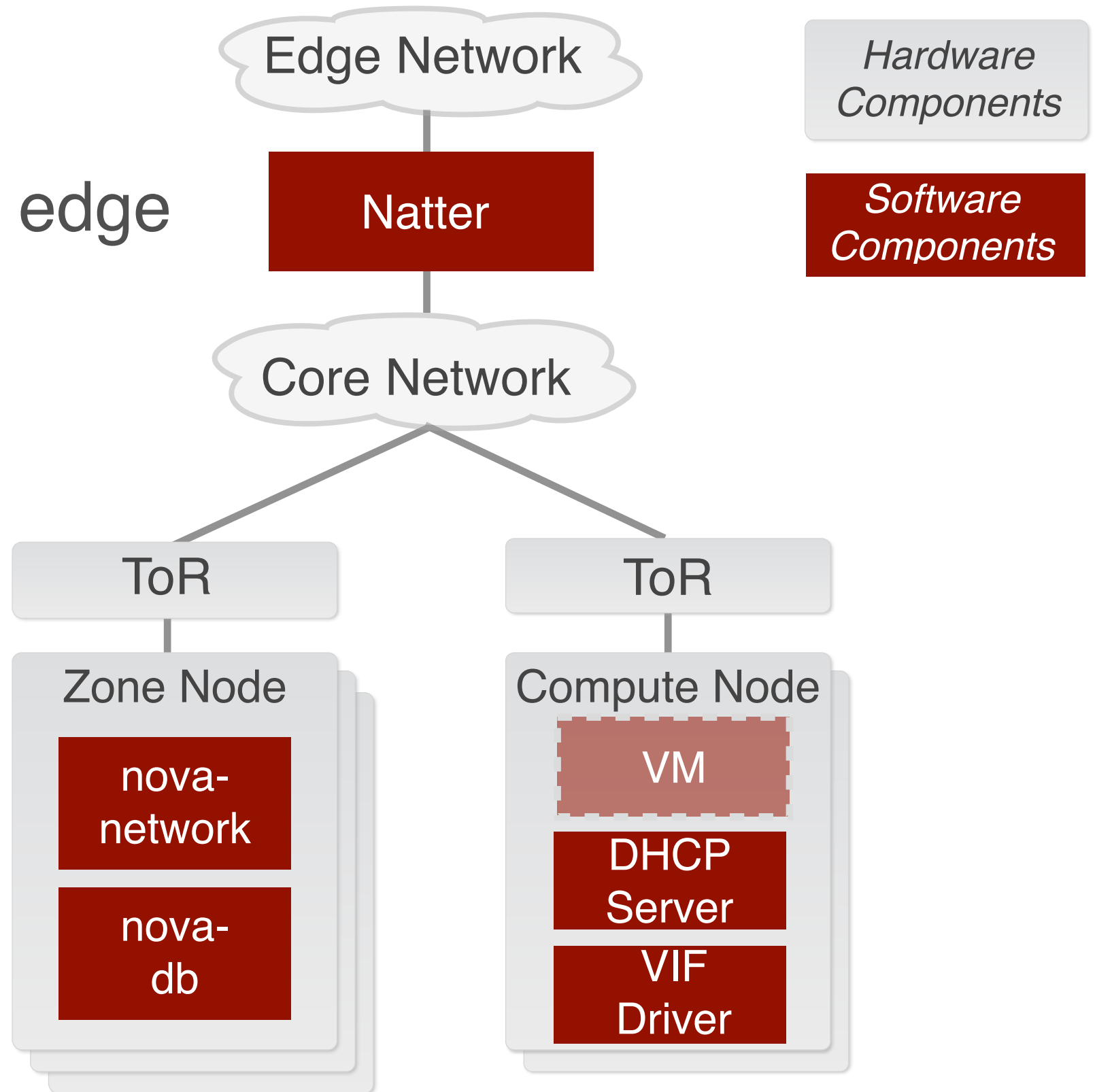
*Software
Components*



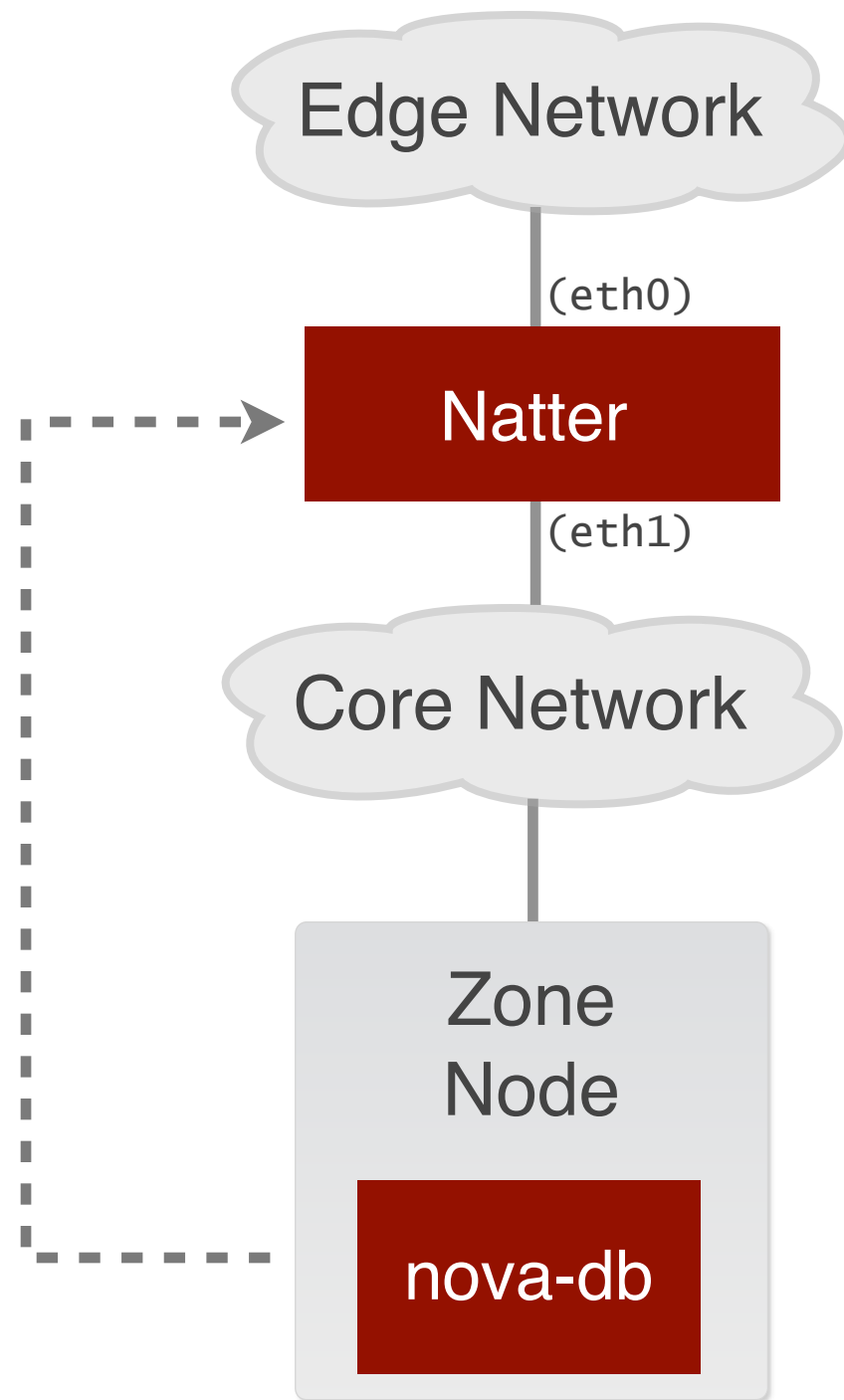
Software Components Schematic Layout

- NAT service on the edge
 - Provides on demand elastic IP service

Provides utilities to create a number of /30 networks per host and pin them to host (l3addnet)



Under the Hood: Natter



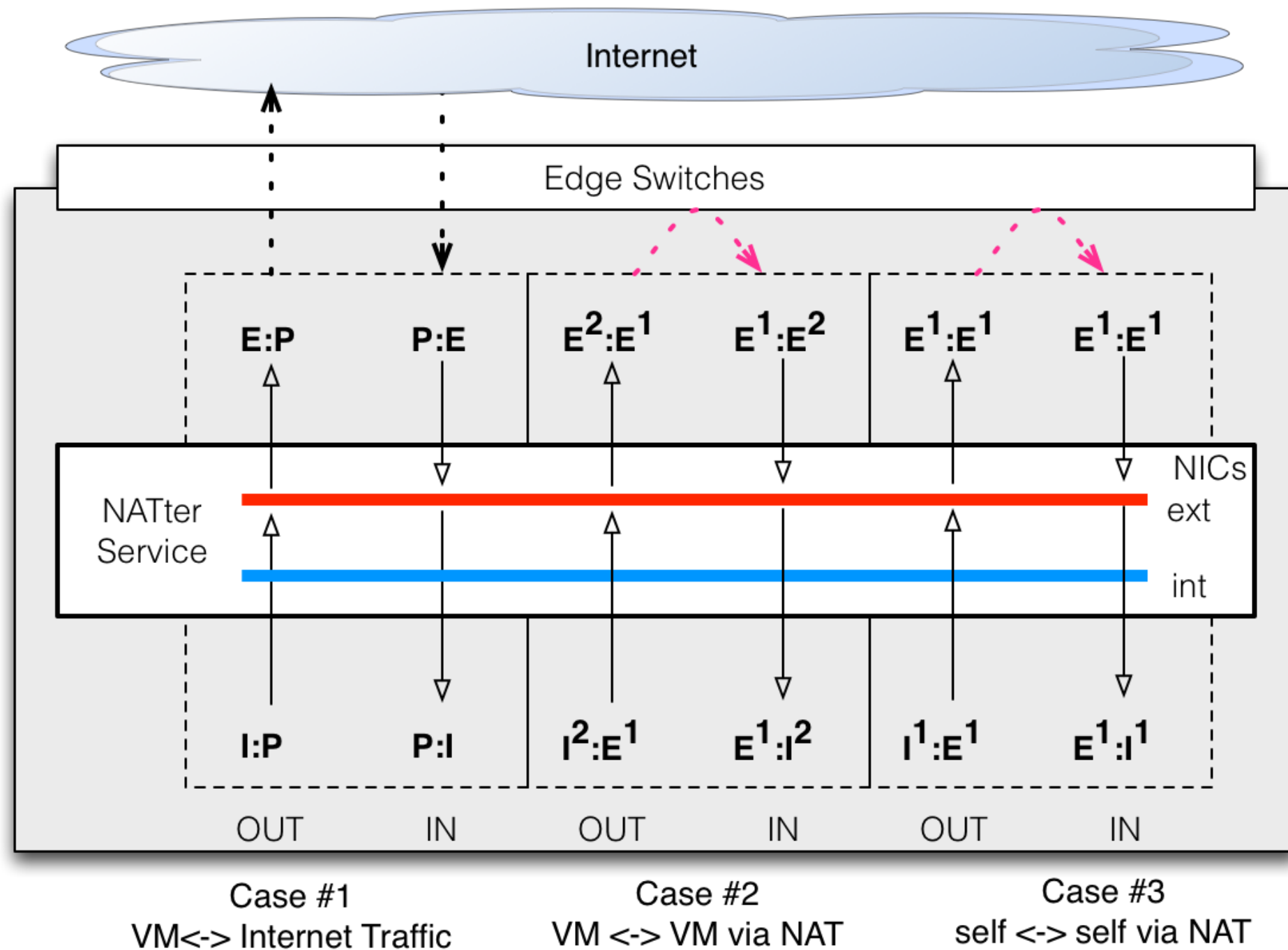
1 Polls nova-db for new `<floating_ip, private_ip>` tuples

2 Use `tc` to install 1:1 NAT rules in `eth0`

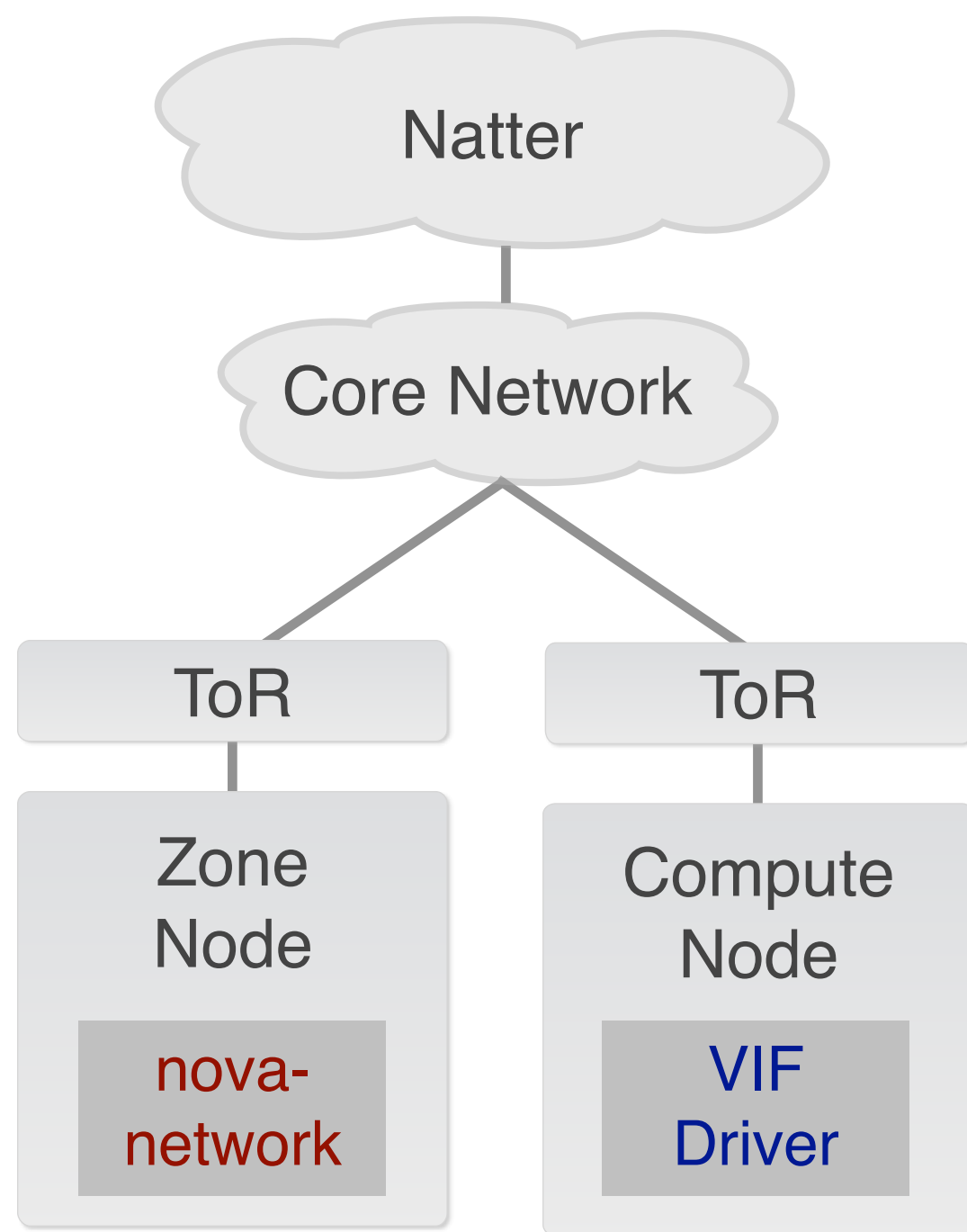
— Connection

- - - -> Control Plane Data

Under the Hood: Natter



Under the Hood: VIF Driver and Nova-Network



VM Provisioning

- 1) VIF: Build linux bridge
- 2) NM: Get host
- 3) NM: Get all available networks for host
- 4) NM: Find first unused network for the host
- 5) VIF: Create a VIF
- 6) VIF: Sets up and starts udhcpd on host per VM
MAC is calculated based on the IP
- 7) VIF: Creates a /30 network for the VM, assigns one address to the bridge, one to the VM
- 8) VIF: Adds routes to enable VM to gateway traffic
- 9) VIF: Adds iptables rules to enable blocked networks and whitelisted hosts

VM Decommissioning

- 1) VIF: Stop udhcpd for the bridge the VM is attached to and remove config
- 2) NM: Delete all IPs in all VIFs
- 3) NM: Cleanup linux bridge
- 4) NM: Cleanup all /30 networks

Under the Hood: l3addnet

Used by cloud admins to pin networks to hosts

Wrapper around nova-manage network create

```
root@z2.b0.z1:~# l3addnet
Usage: l3addnet cidr host01 dns1 dns2
root@z2.b0.z1:~# l3addnet 10.50.0.0/24 10.18.1.12 8.8.8.8 8.8.4.4
```

- Breaks down the input CIDR into /30 blocks
- Loops through each block and calls the nova-manage API to create a network on that compute host

Network Driver Challenges

- OpenStack releases are moving targets
 - Plugin interfaces change
 - Library dependencies change
- Database API not rich/efficient enough
 - Straight to SQL to get what we needed
- nova-network supposed to be deprecated?
 - First in Folsom or Grizzly? Then Havana??
 - Have to figure out our Neutron strategy

Why Not Neutron Now?

- Created in Diablo timeframe
- Neutron still not stable
 - API changes and interfaces are actively hostile
 - No multi-host support
 - Complicated, non-intuitive maintenance procedures
 - Not all plugins are equally stable
 - *many are outright buggy*

SDN in OCS

- OpenContrail only one to meet our rqmts
- OCS ref network arch ideal “underlay”
 - SDN underlays usually are spine and leaf
 - L3 routing does not interfere with supporting encapsulation or tunneling protocols
- Customers can choose network model
 - VPC or L3

A large customer who wants to seamlessly support autoscaling for its tenants is a perfect use-case for VPC

Example Packet Path - L3*

* natters not shown for simplicity purposes

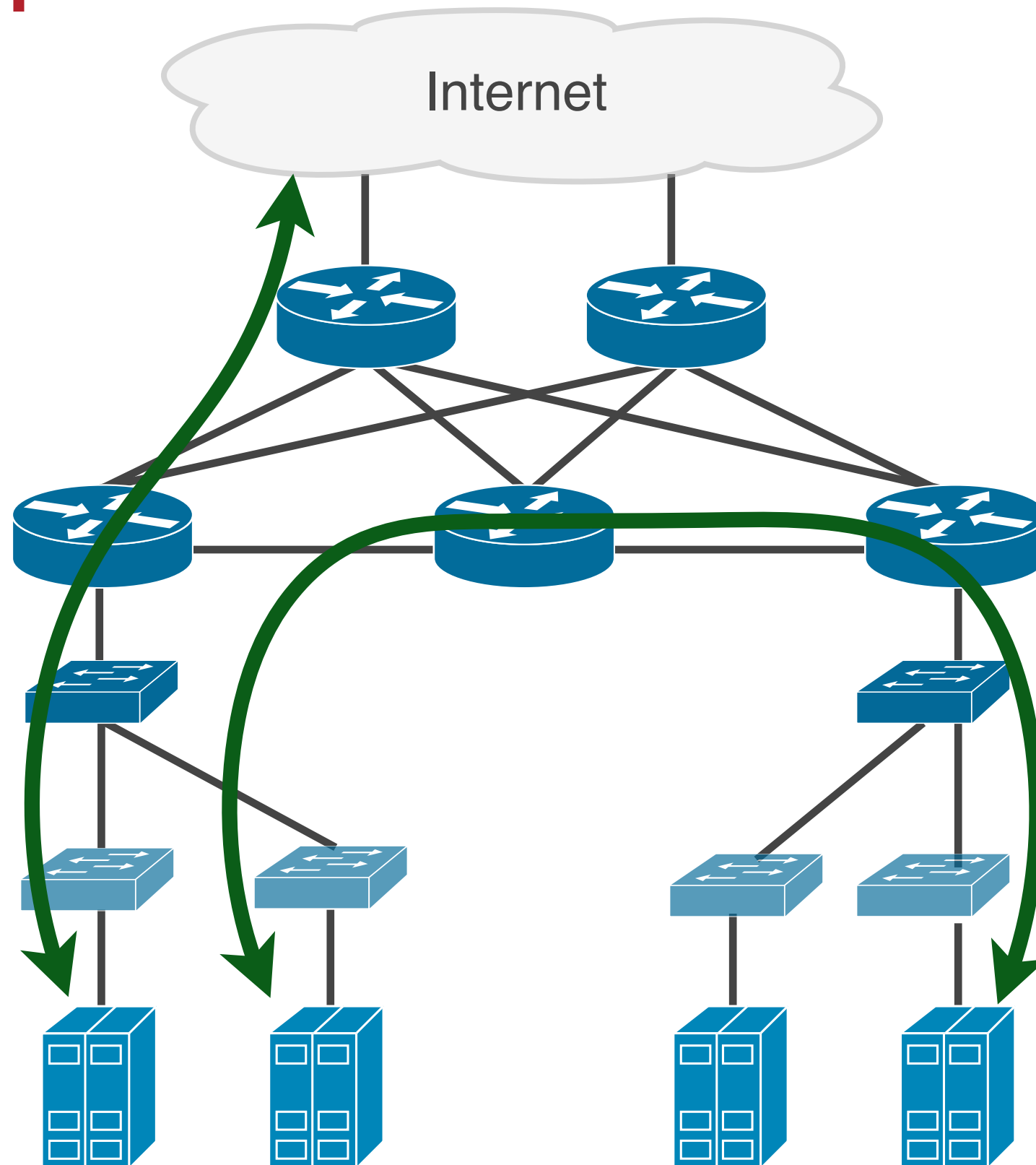
Edge Routers
("egress")

Core Routers
("spine")

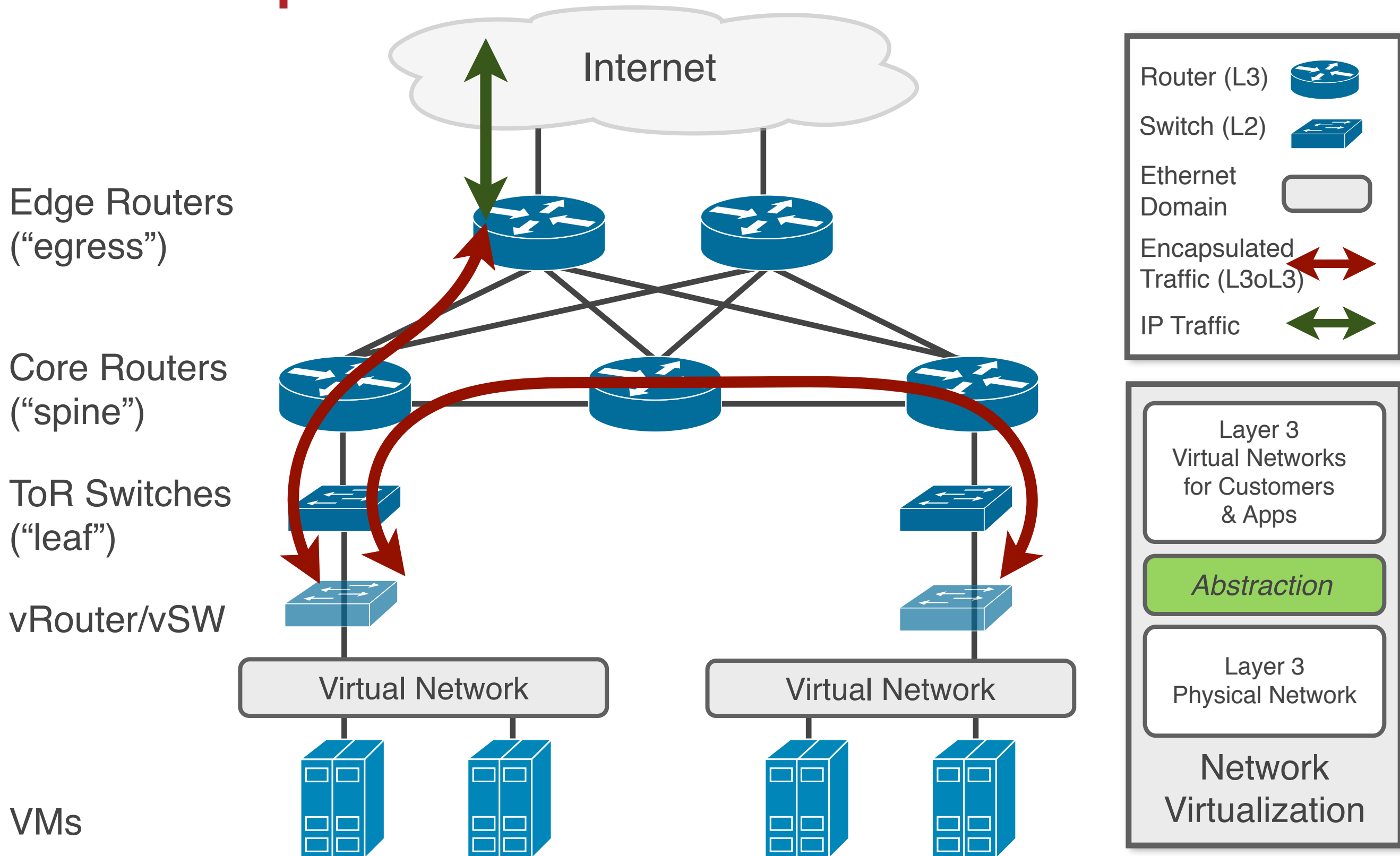
ToR Switches
("leaf")

Linux Bridge
on compute
node

VMs




Example Packet Path (🌥️) OPENCONTRAIL



Future Directions

- OCS L3 networking migrates to Neutron
 - As networking plugin (beyond nova-network replacement)
- OCS VPC w/ more advanced SDN capabilities
 - NFV combined with Service Chaining for Carriers
 - Support existing physical network assets with Service Chaining



@rony358 
Abhishek Chanda

Questions?