Realize SFC Using ONOS SDN Controller

Mohan Kumar, Senior Software Engineer, Huawei Indian

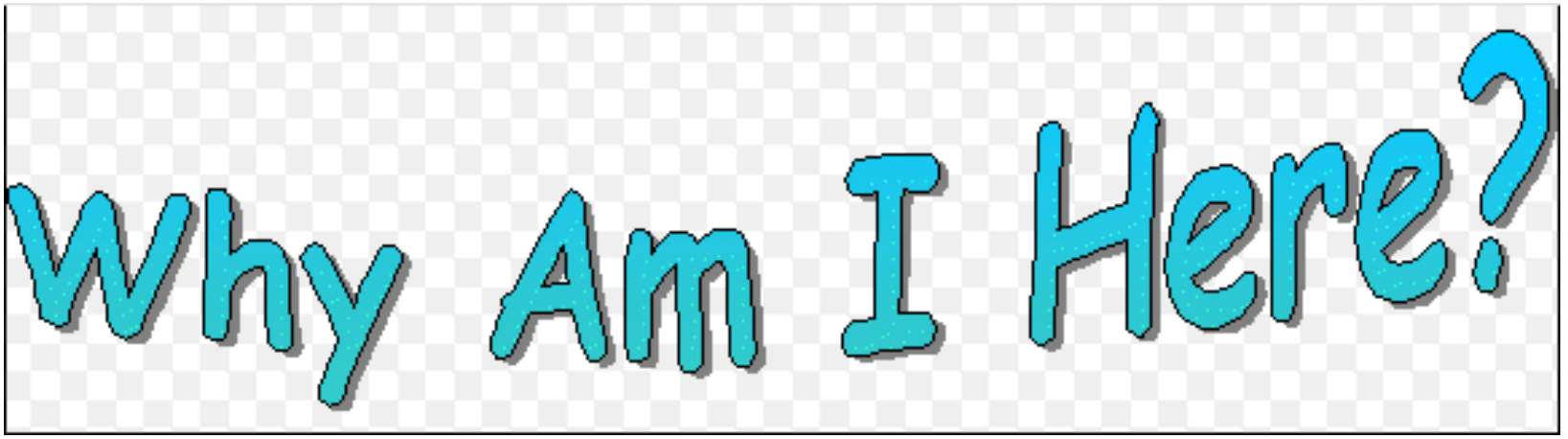
Cathy Zhang, Principal Architect, Huawei USA

Neutron

openstack™

onos

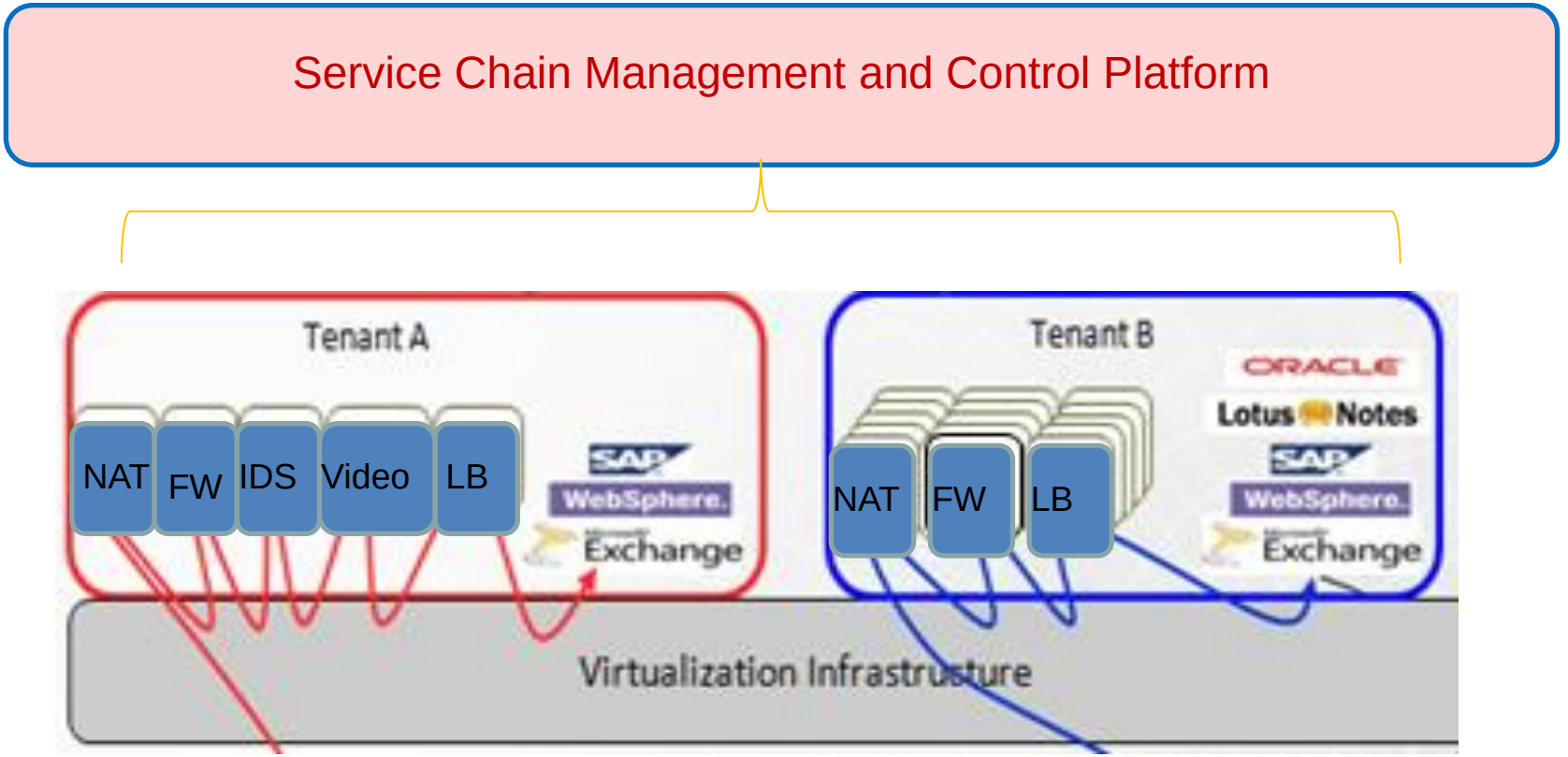
Open Network Operating System

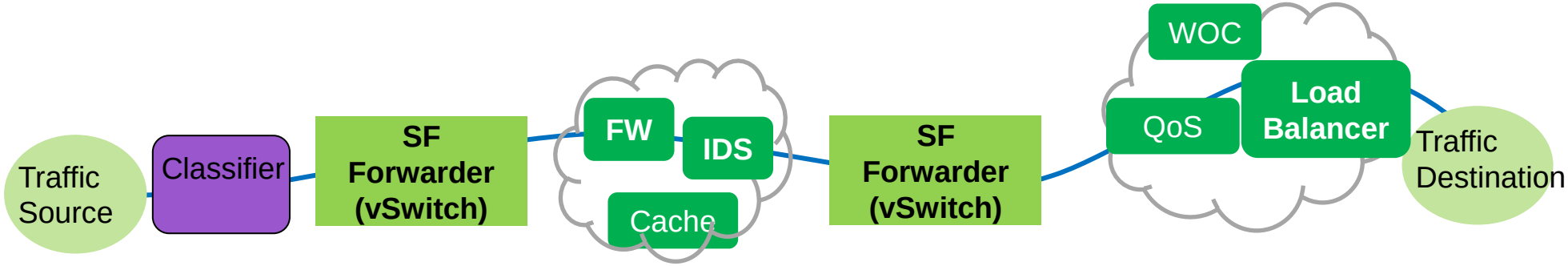


1. Understand OpenStack Neutron SFC Feature: Its Flexible Architecture to Integrate with Multiple SDN Controllers, Its APIs, Its Code Status, Second Phase Roadmap.
2. Understand ONOS Controller (Open Network Operating System) Distributed Architecture For Scalability Support
3. Undertand How Openstack Integrates with ONOS SDN Controller to Realize the SFC Functionality.

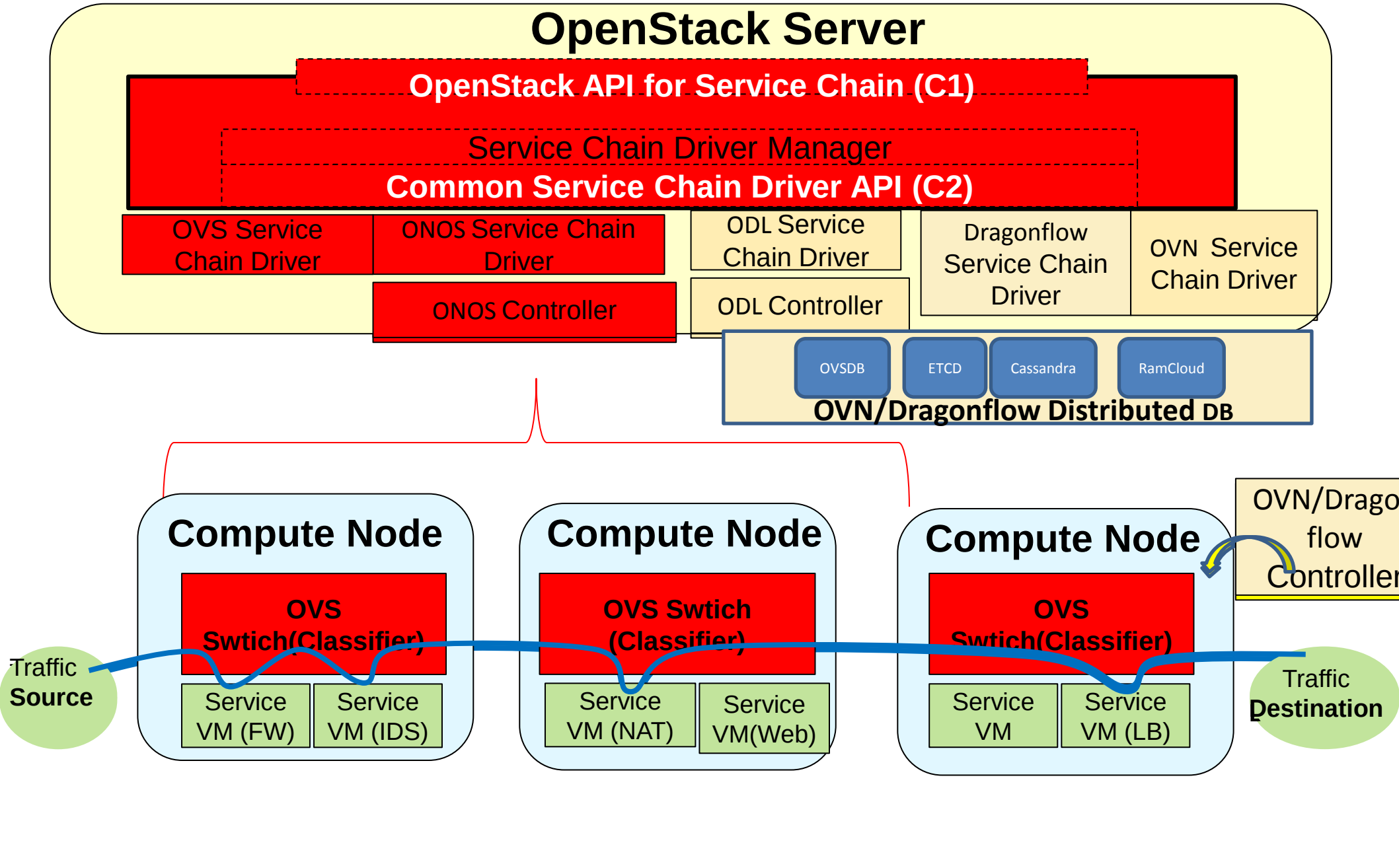
OpenStack Service Chain Overview

What is Service Function Chain ?

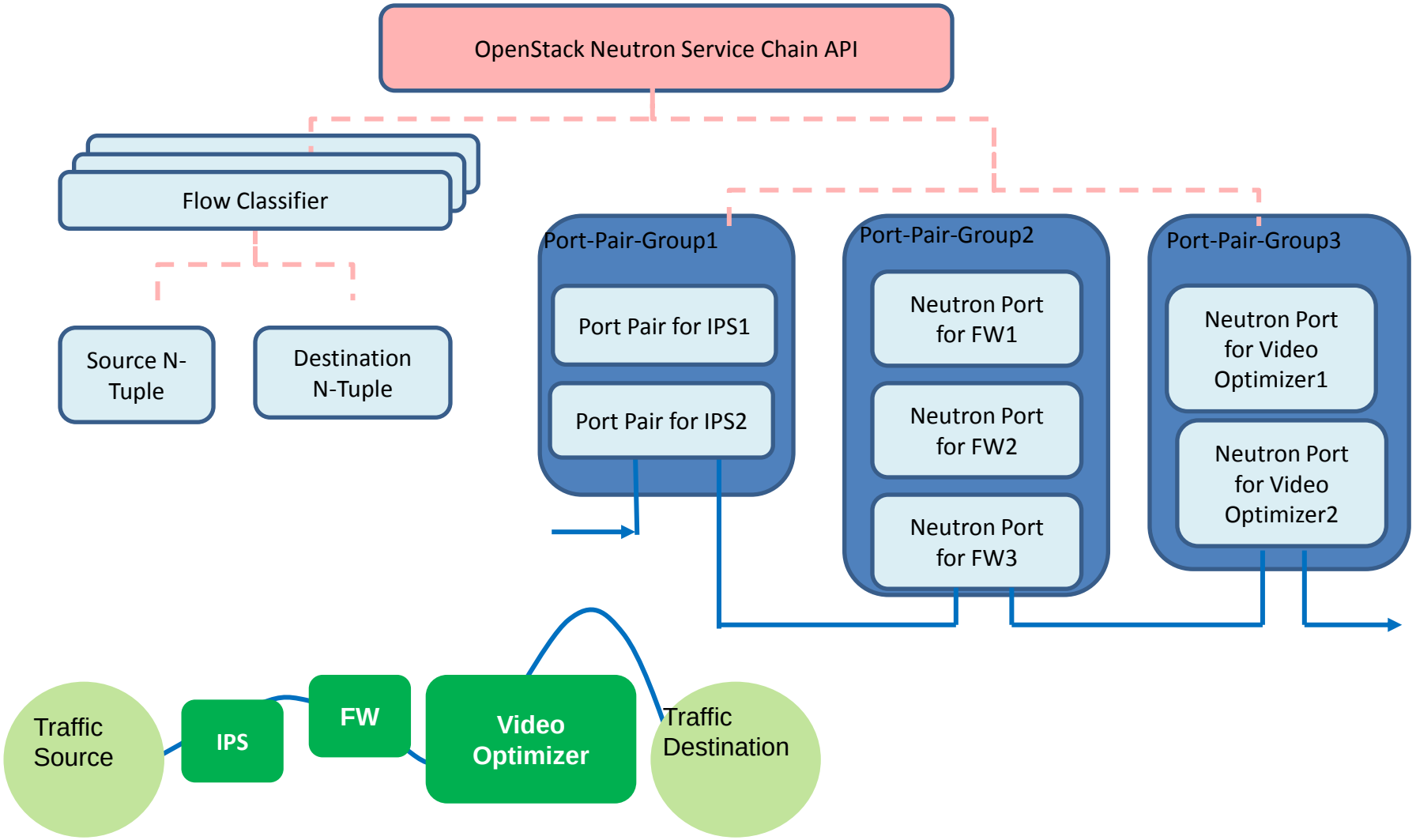




OpenStack Neutron Service Chain Architecture



OpenStack Service Chain API Overview



Networking-sfc Project Information

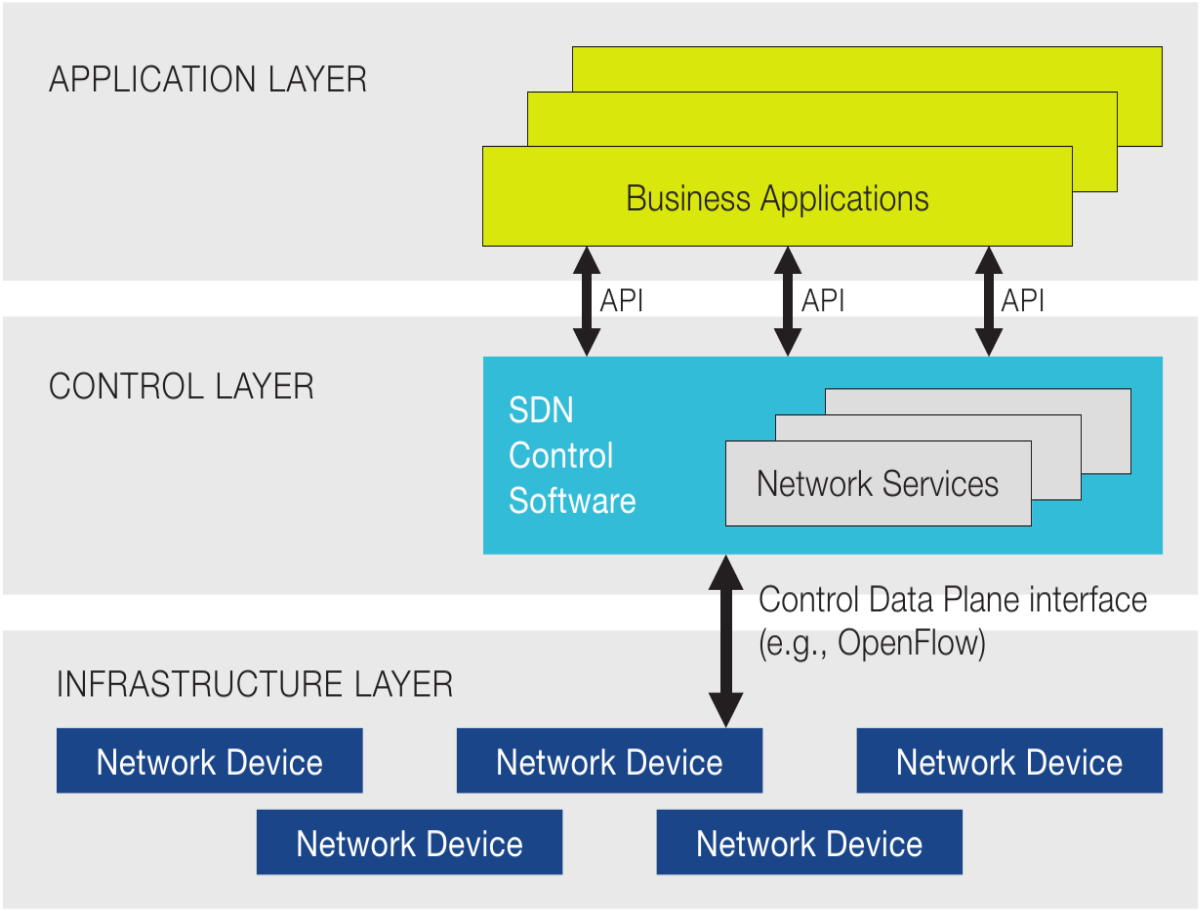
- First Release in Feb 2016
- Architecture and API Specification Link:
  - http://docs.openstack.org/developer/networking-sfc/
- Project Wiki Page:
  - https://wiki.openstack.org/wiki/Neutron/APIForServiceChaining
- Weekly IRC Meeting:
  - Thursday 1700 UTC on #openstack-meeting-4
  - https://wiki.openstack.org/wiki/Meetings/ServiceFunctionChainingMeeting

Second Phase Road Map of Networking-SFC

- **Add Support for a Chain of SFs Hosted on Container**
- **Add Support for a Chain of SFs Hosted on Physical Device**
- **Integrate with VNFM Tacker**
- **Add ODL SFC Driver, OVN SFC Driver, Dragonflow SFC Driver to Support the Implementation path on these Open Source SDN Controllers.**
- **Support for IETF NSH Encapsulation**
- **Support for Symmetric SFC Path**

ONOS for Openstack

SDN Architecture



OpenStack
Networking-SFC

ONOS Controller

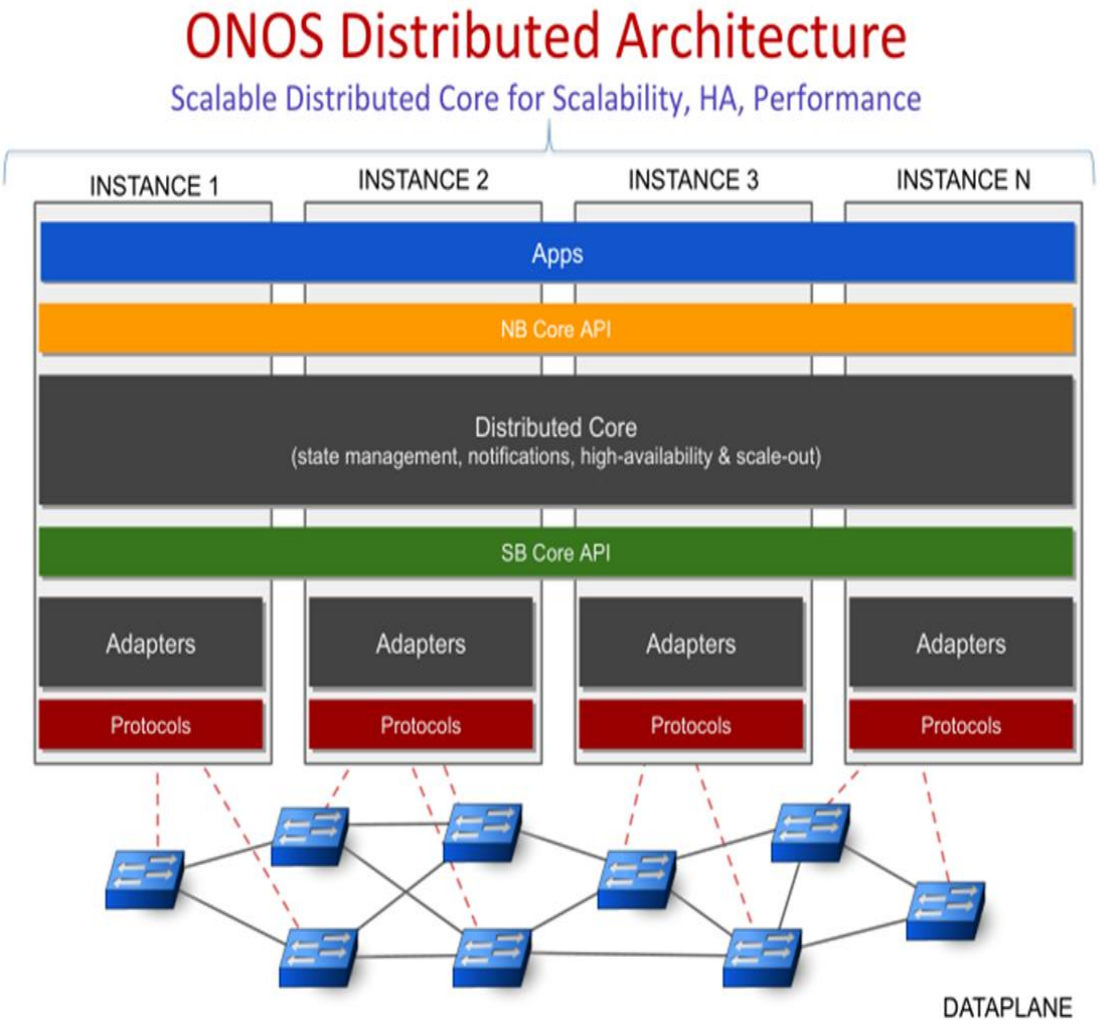
vSwitch, Service
Function
VM/Container/
Physical Device

What is Modular ONOS?

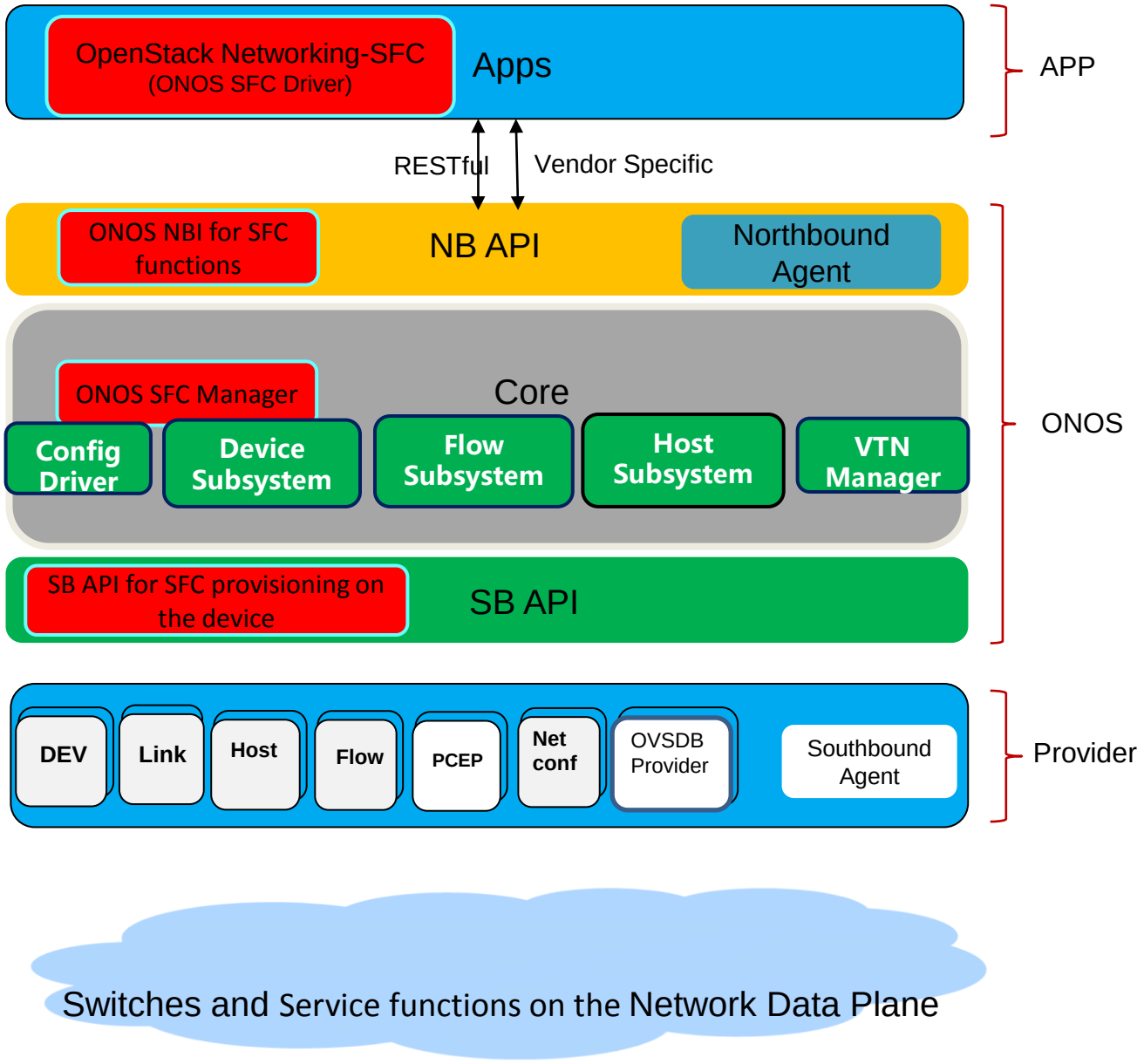
A new carrier-grade SDN network operating system designed for

- high availability
- performance
- scale-out.

Mission: “to produce the Open Source Network Operating System that will enable service providers to build real Software Defined Network”

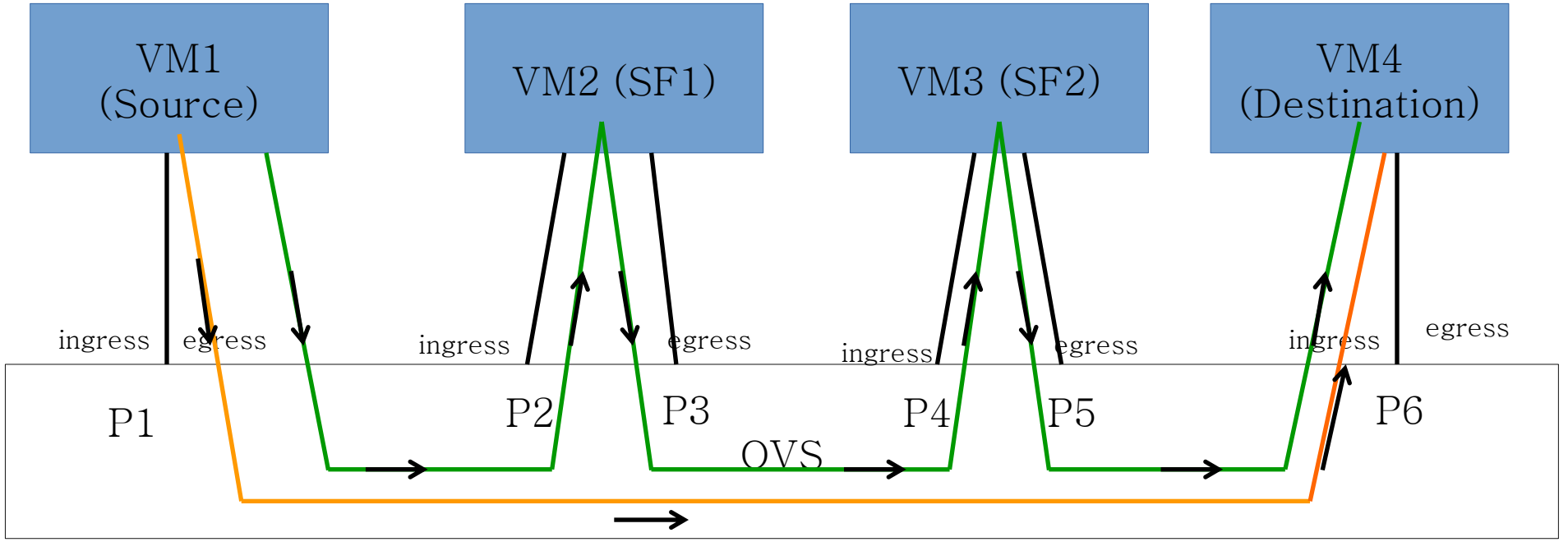


SFC in ONOS Architecture




Demo

Demo Topology

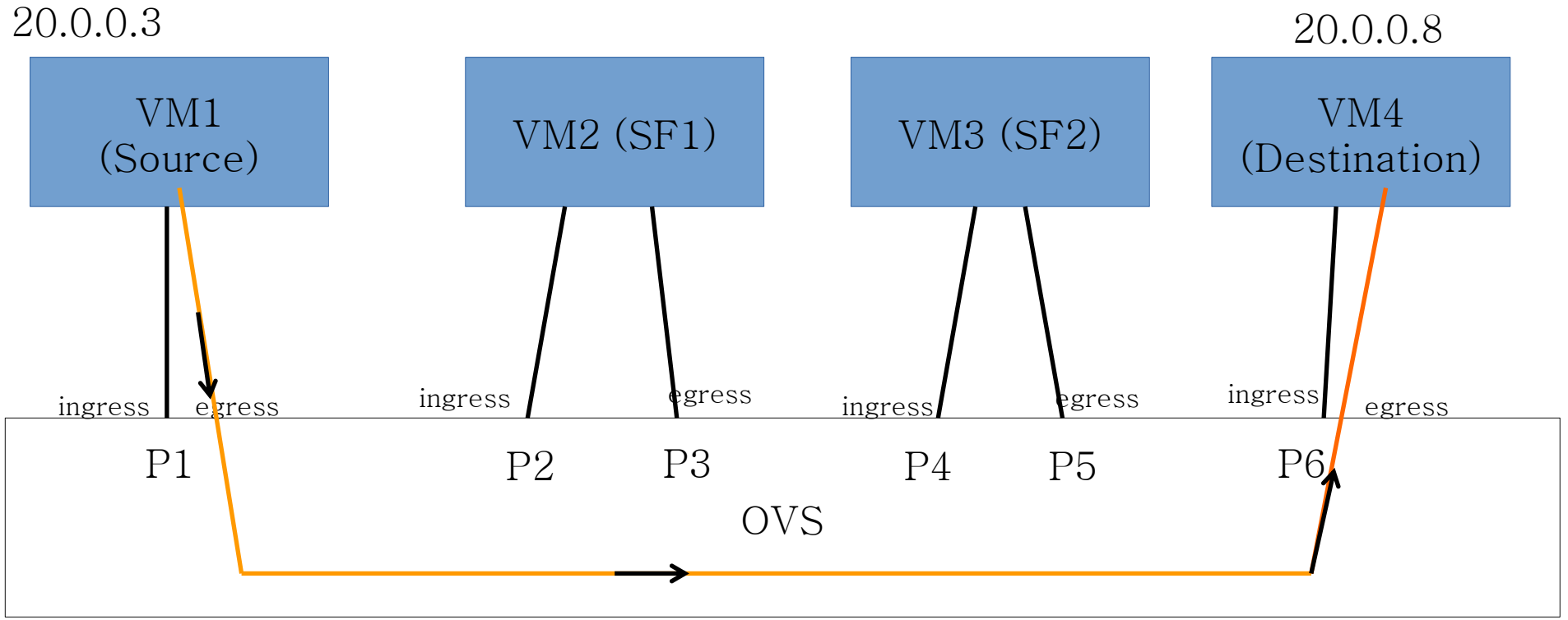


Packet path before installing SFC
VM1 -> VM4

Packet path after installing SFC
VM1 -> VM2 -> VM3 -> VM4

- In our demonstration we have source VM, destination VM and a set of service functions VM's spawned using openstack network API
- We use ping packet as data transfer between source and destination.
- Before installing SFC, the packet will directly go to the destination
- After installing SFC the packet will take the defined SFC path and is processed at each service function before reaching destination.

Scenario 1

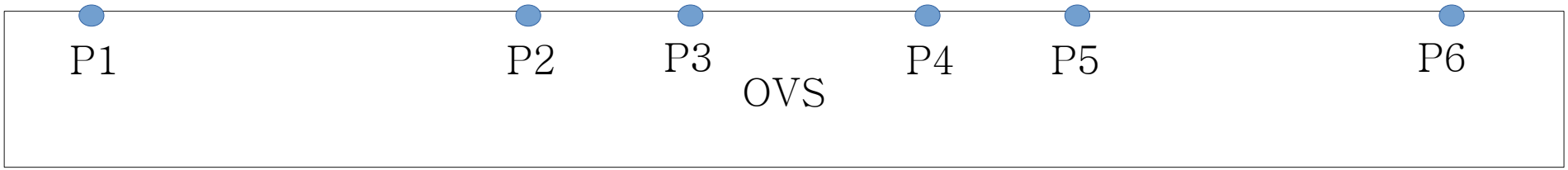


Test : Ping from VM1 to VM4

Packet path before installing SFC, VM1 -> VM4

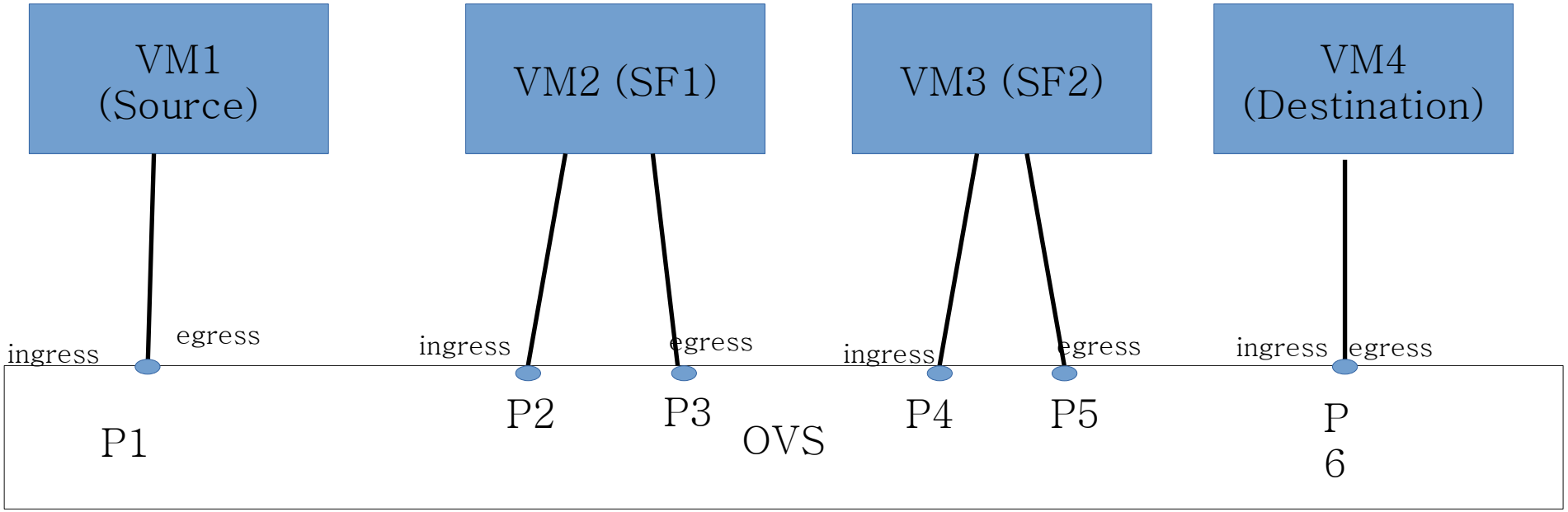
Creating port

- Use neutron networking CLI to create ports on OVS (All ports are created on the same network)
- neutron port-create --name p1 net1
- neutron port-create --name p2 net1
- neutron port-create --name p3 net1
- neutron port-create --name p4 net1
- neutron port-create --name p5 net1
- neutron port-create --name p6 net1
- neutron port-create --name p7 net1
- neutron port-create --name p8 net1

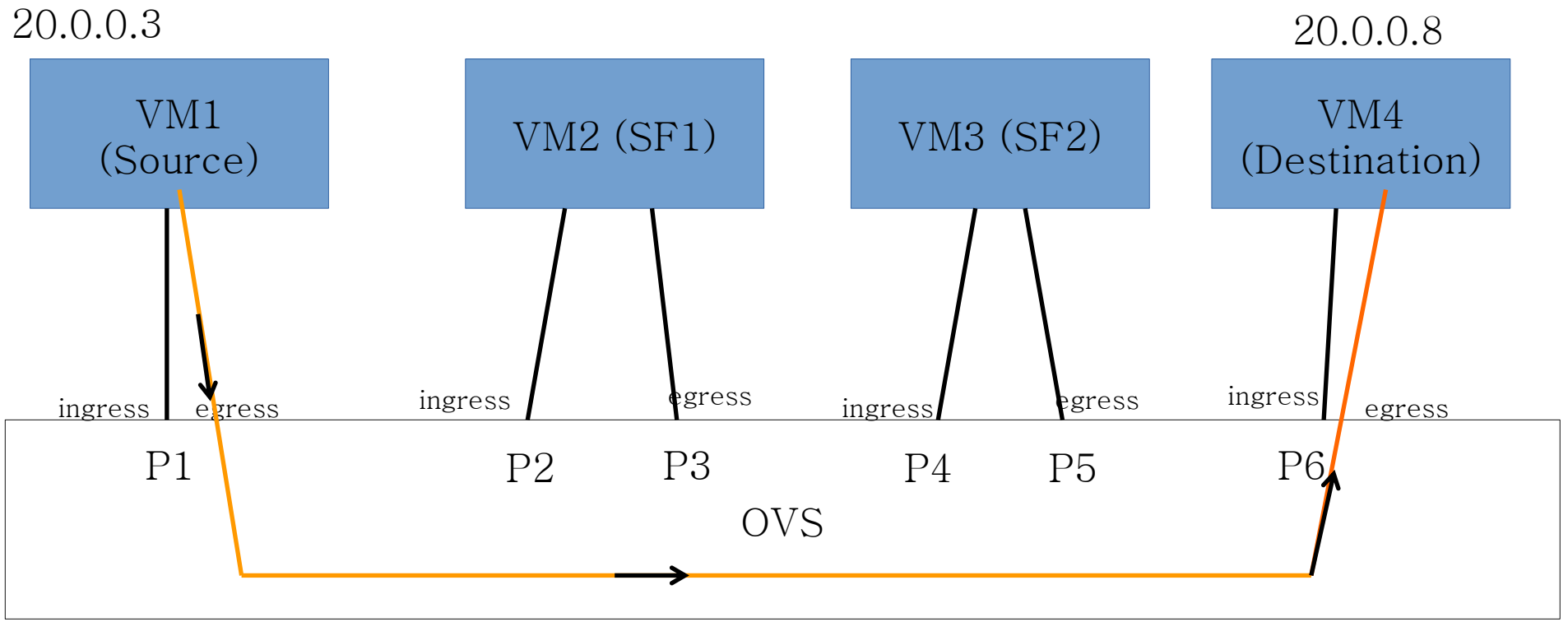


Spawning VM's

- Spawn the VM's with the created ports
- Use nova CLI to spawn the VM's
- nova boot --image cirros-0.3.4-x86_64-uec --flavor m1.small --nic port-id=<ingress port> --nic port-id=<egress port> <vm name>



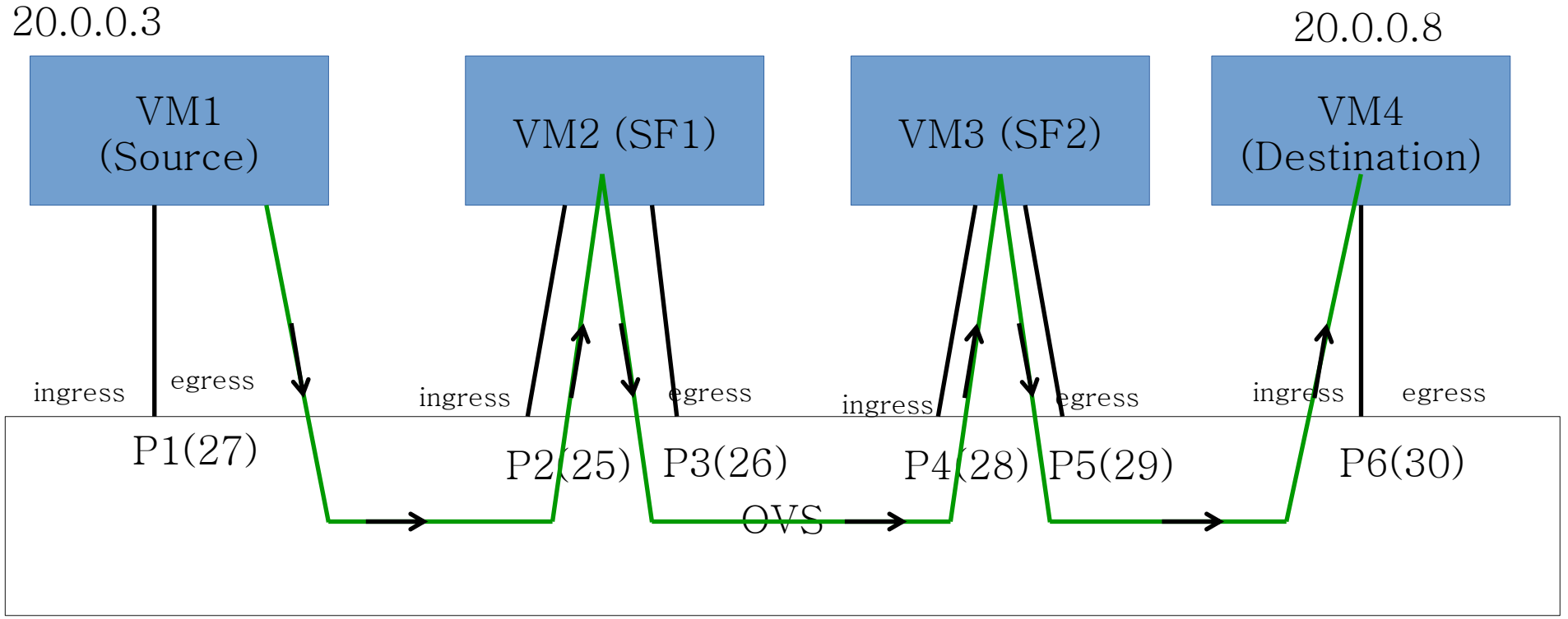
Ping from VM1 to VM4 Without SFC



Packet path before installing SFC
VM1 -> VM4

Scenario 2

Packet path after installing SFC(Create port chain)
VM1 -> VM2 -> VM3 -> VM4

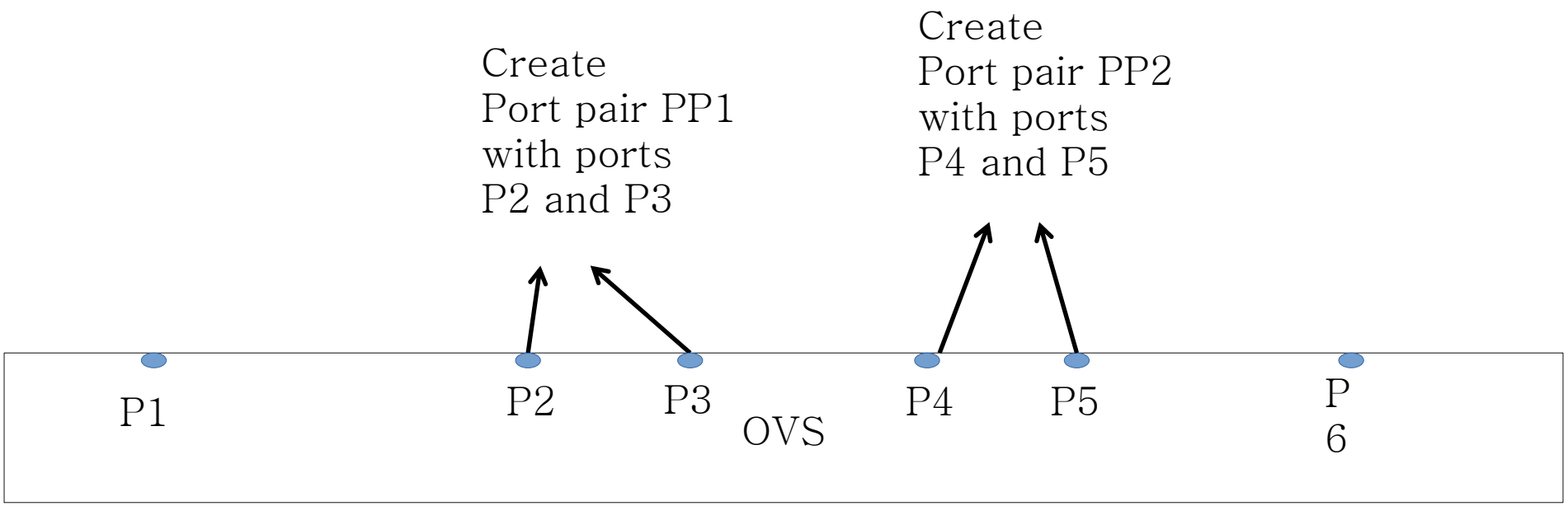


Test : Ping from VM1 to VM4

Note: 5, 6, 7, 8.. are the OVS ports on which the VM's tap interfaces are created

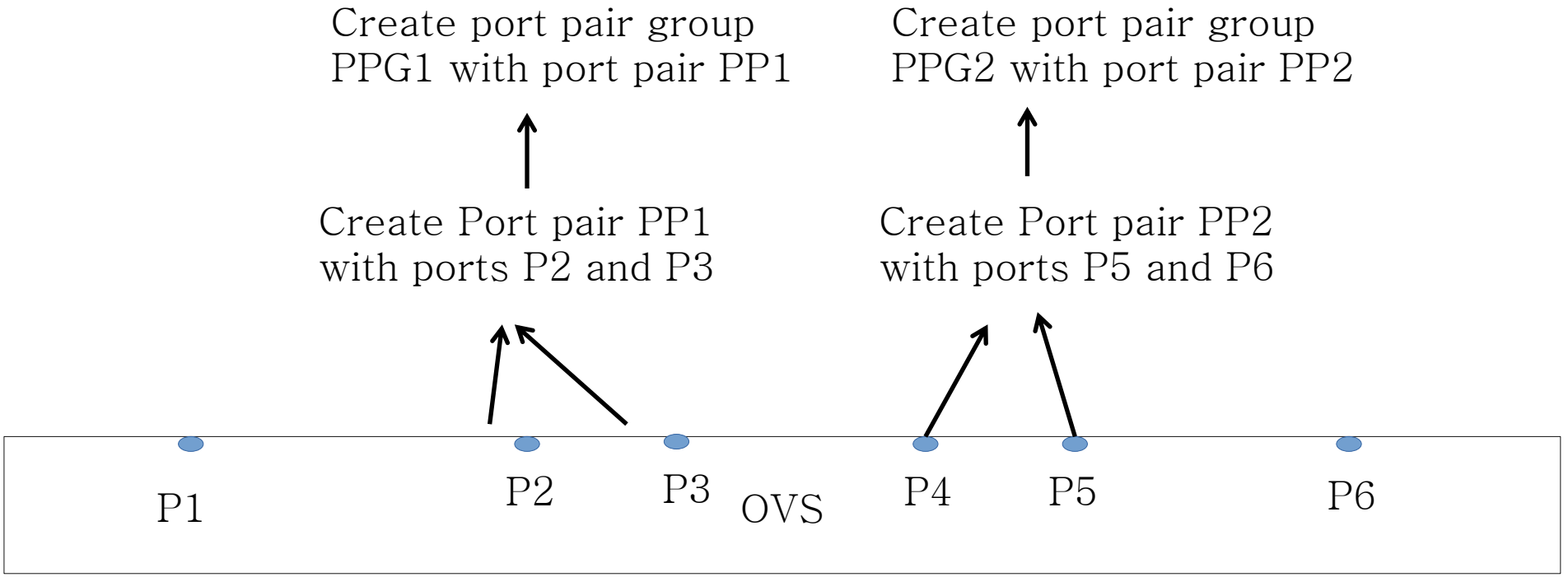
Creating SFC Port pair

- Use Networking-SFC CLI to create port pairs
- neutron port-pair-create <port pair name> --ingress <port id> --egress <port id>
- When a port pair is successfully created, neutron SFC will send a create request to ONOS rest API.
- ONOS will store the respective port pair details in its DB



Creating Port pair group

- Use Networking-SFC CLI to create port pair group
- neutron port-pair-group-create --port-pairs <port pair name> <port pair group name>
- Once the port pair group is successfully created, neutron will send a create request to ONOS rest API for port pair group.
- ONOS will store the respective port pair details in its DB

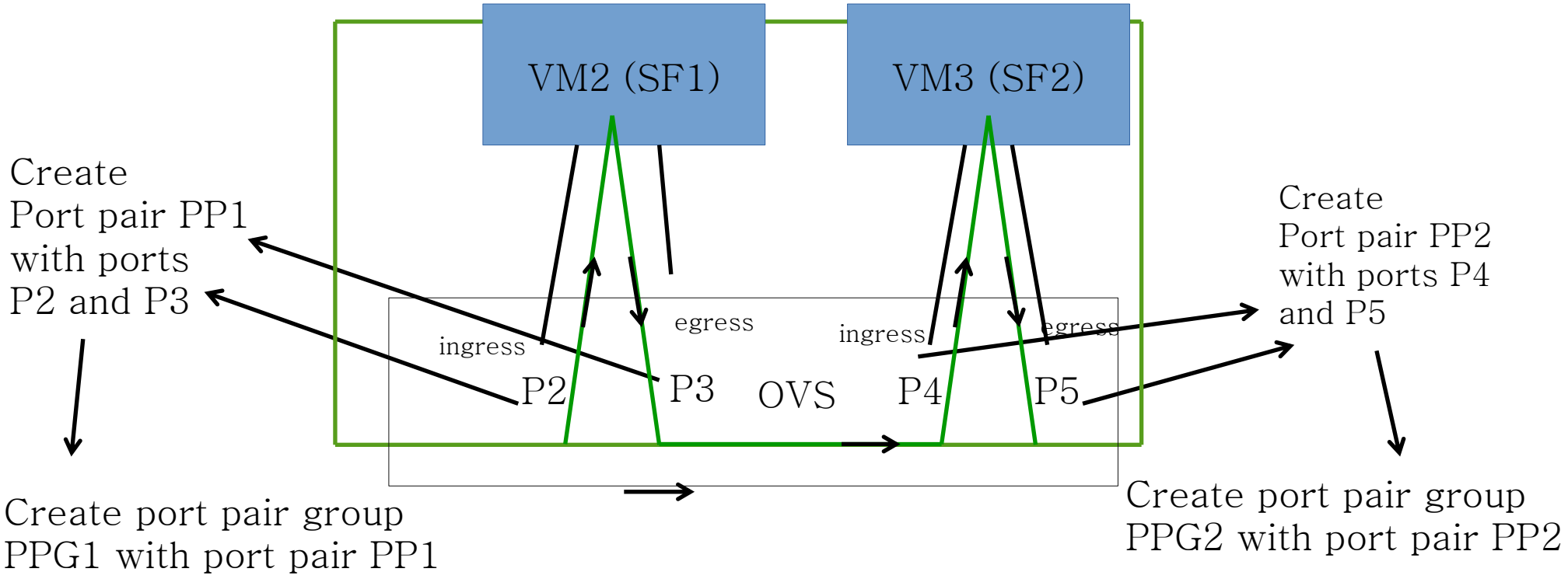


Create flow classifier

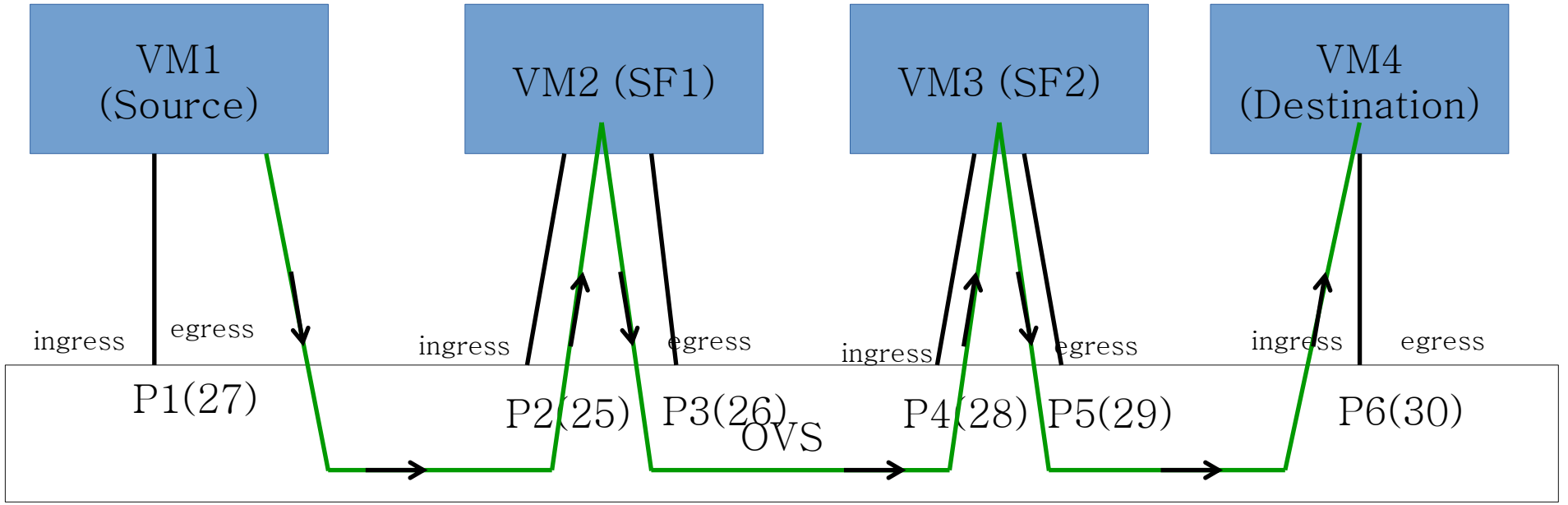
- Use Networking-SFC CLI to create flow classifier.
- Classifier rule is used to select traffic that originates from source with IP prefix 20.0.0.3/32 and goes to destination with IP prefix 20.0.0.8/32 (Source and destination with in the same network), and the chain ingress port is set as the Source VM's neutron port p1
- neutron flow-classifier-create --source-ip-prefix 20.0.0.3/32 --destination-ip-prefix 20.0.0.8/32 -logical-source-port p1 FC1
- Once the flow classifier is successfully created, neutron will send a create request to ONOS rest API
- ONOS will store the respective flow classifier details in its DB

Create Port chain

- Use Networking-SFC CLI to create port chain
- Create port chain with VM2 as SF1 and VM3 as SF2
- neutron port-chain-create --port-pair-group PPG1 --port-pair-group PPG2 --flow-classifier FC1 <port chain name>
- Once the port chain is successfully created, neutron will send a chain creation request to ONOS rest API for port chain.
- ONOS will store the respective port chain details in its DB and initiates event to generate and download required flow rules to the switches for setting up the SFC traffic steering path.



Ping from VM1 to VM4 With SFC



- When the ping packet is coming out from VM1, it will meet the classifier rule and the packet is forwarded to VM2.
- There is a SF1 running in VM2, which will receive the packet, process it and send it out on the VM2's egress port
- The packet coming out from the Egress port of VM2 will qualify the pre-programmed forwarding rule and be forwarded to VM3.
- There is a SF2 running in VM3, which will receive the packet process it, and send it out on the egress port of VM3
- The packet coming out from Egress port of VM3 will satisfy the normal forwarding rules to the destination, and forwarded to VM4.

Thank You