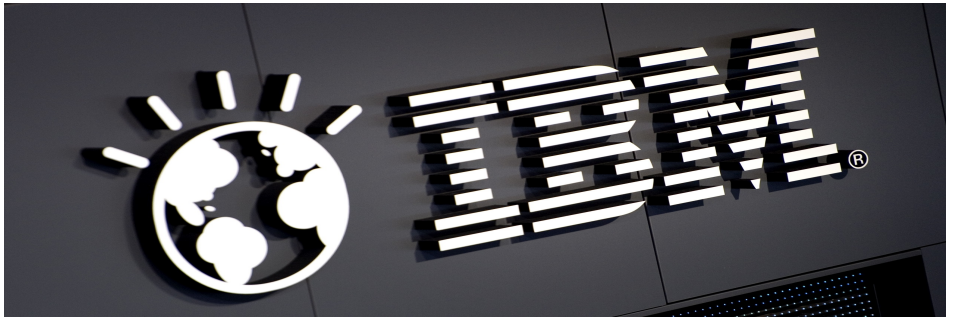
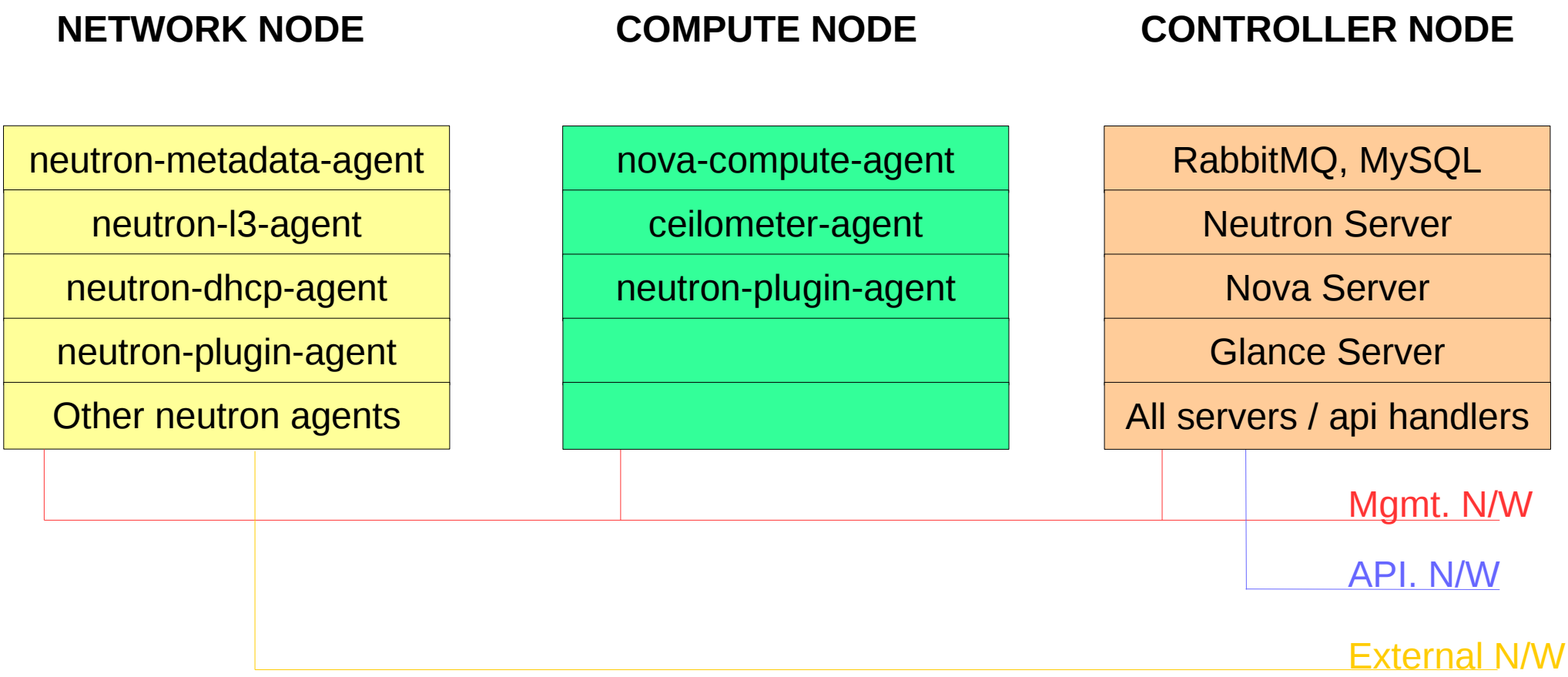
Proof-of-Concept for OpenStack Neutron Agent

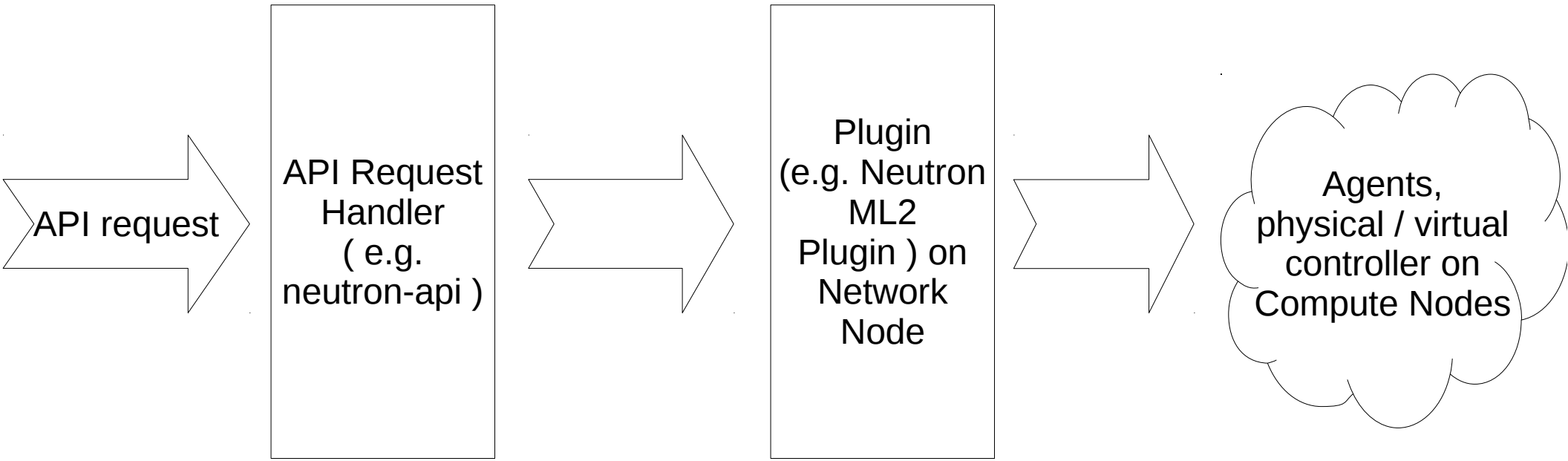
-Piyush Raman Srivastava



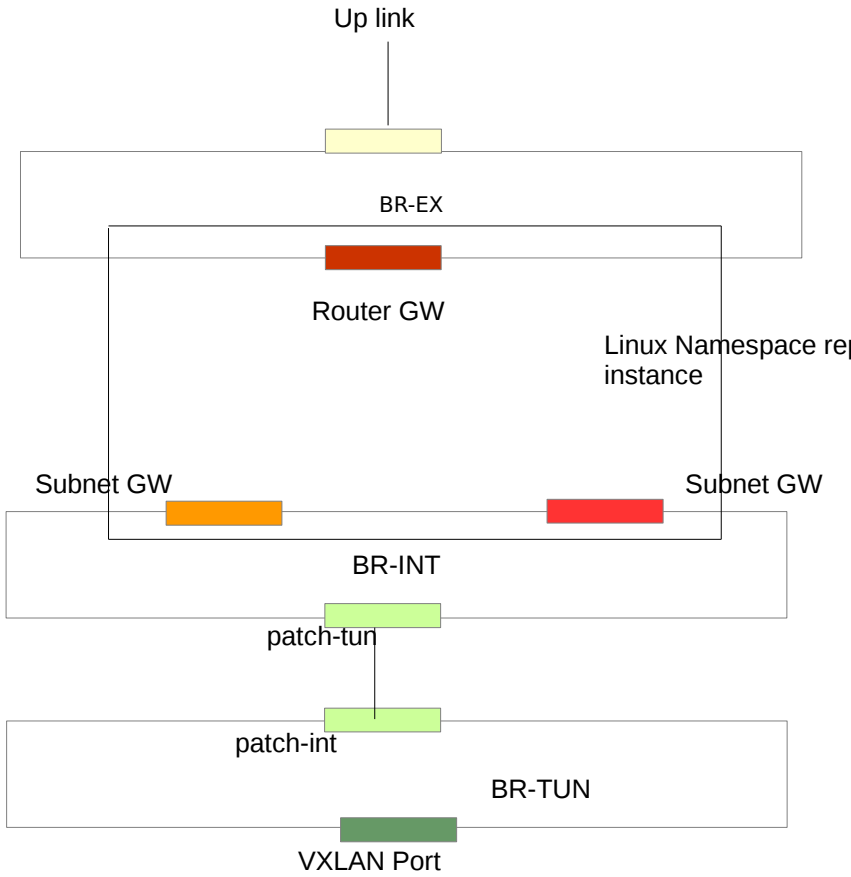
OpenStack Nodes and Data Center Networks



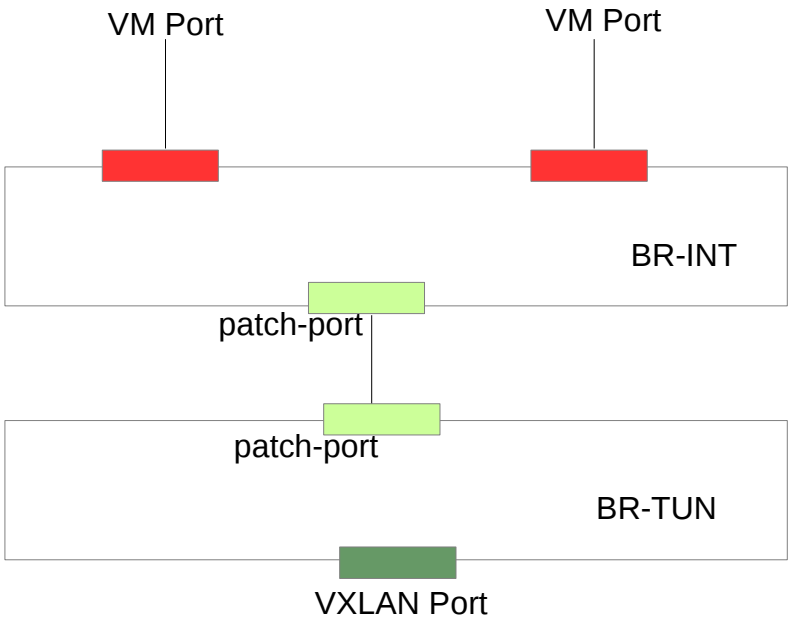
Plugin Architecture- OpenStack Neutron



OpenStack Bridge Setup

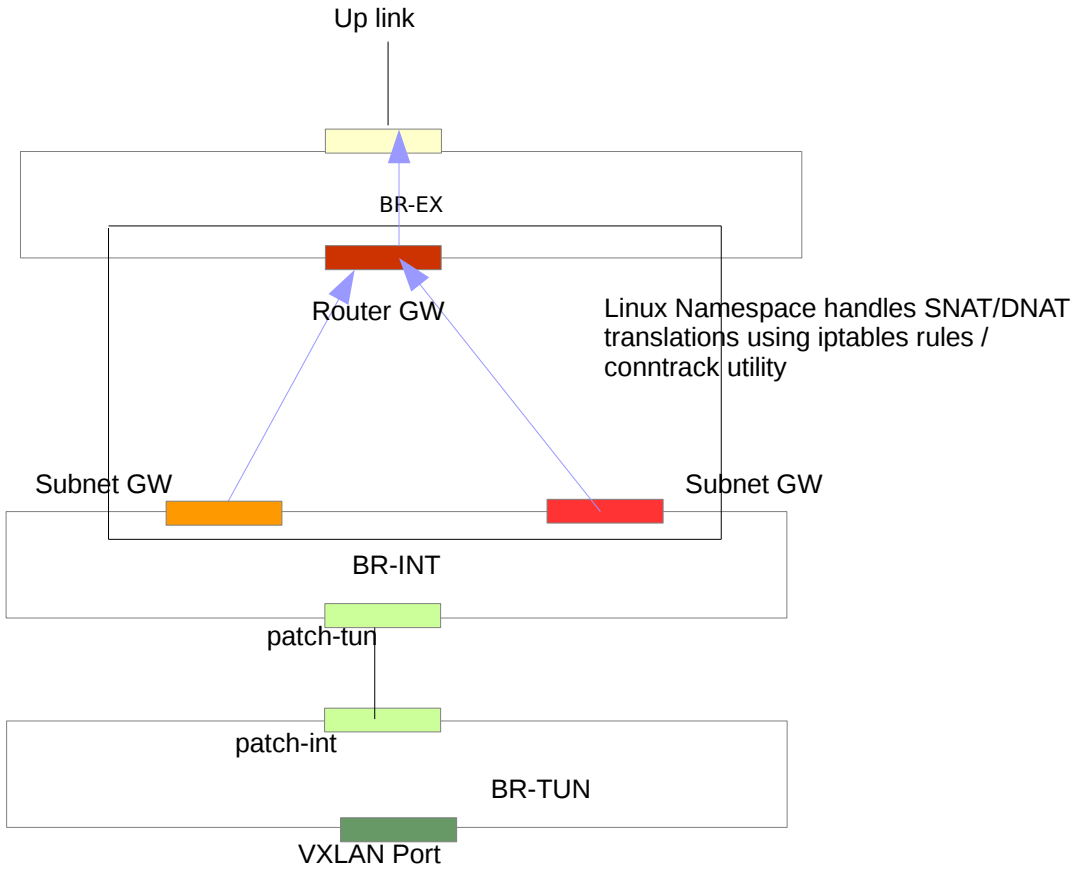


NETWORK NODE

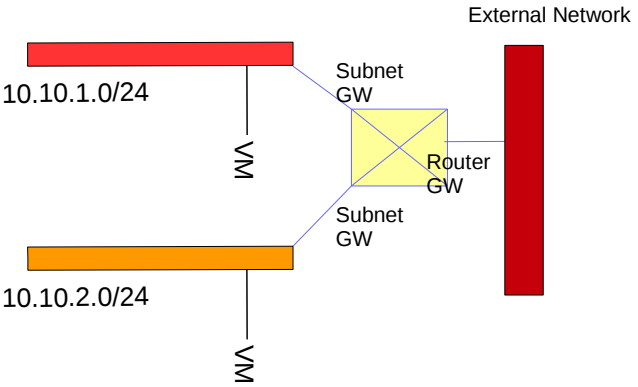


COMPUTE NODE

OpenStack Neutron Router



NETWORK NODE



OpenStack Neutron Virtualization

Proof-Of-Concept Overview

The POC implements a novel OpenStack Neutron Agent which supports L3 routing for overlay network to external (underlay / virtual) network having following functionalities-

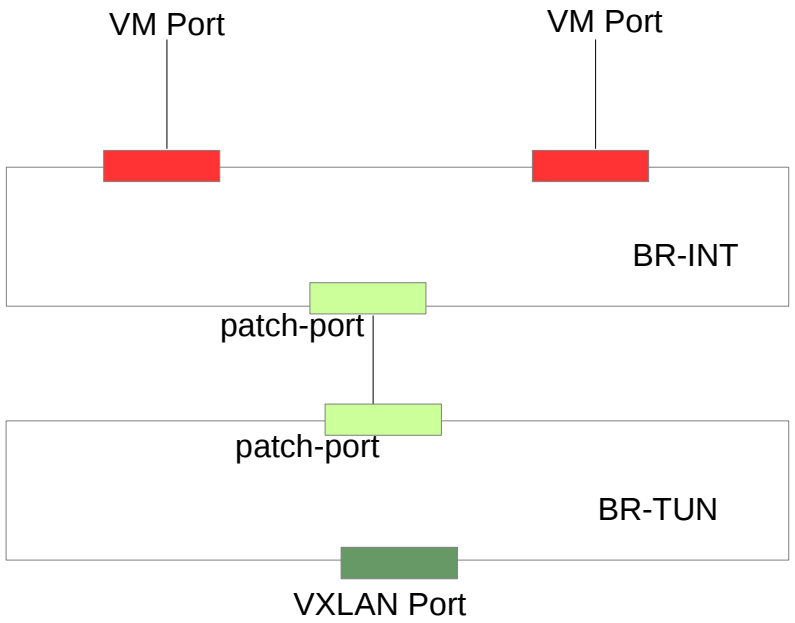
- **Overlay within / across subnet** routing and **NATing** (SNAT/DNAT) via flows
- **Virtualize Neutron Router** using **flows** defined by OpenFlow Protocol
- **Multiple external networks** across / within tenants using **single instance** of the agent
- **De-centralize the overlay across subnet** decision on each compute node
- **ARP Responder / L2 Population** Mechanism on each node to minimize ARP request broadcasts

Openstack Setup Description For POC

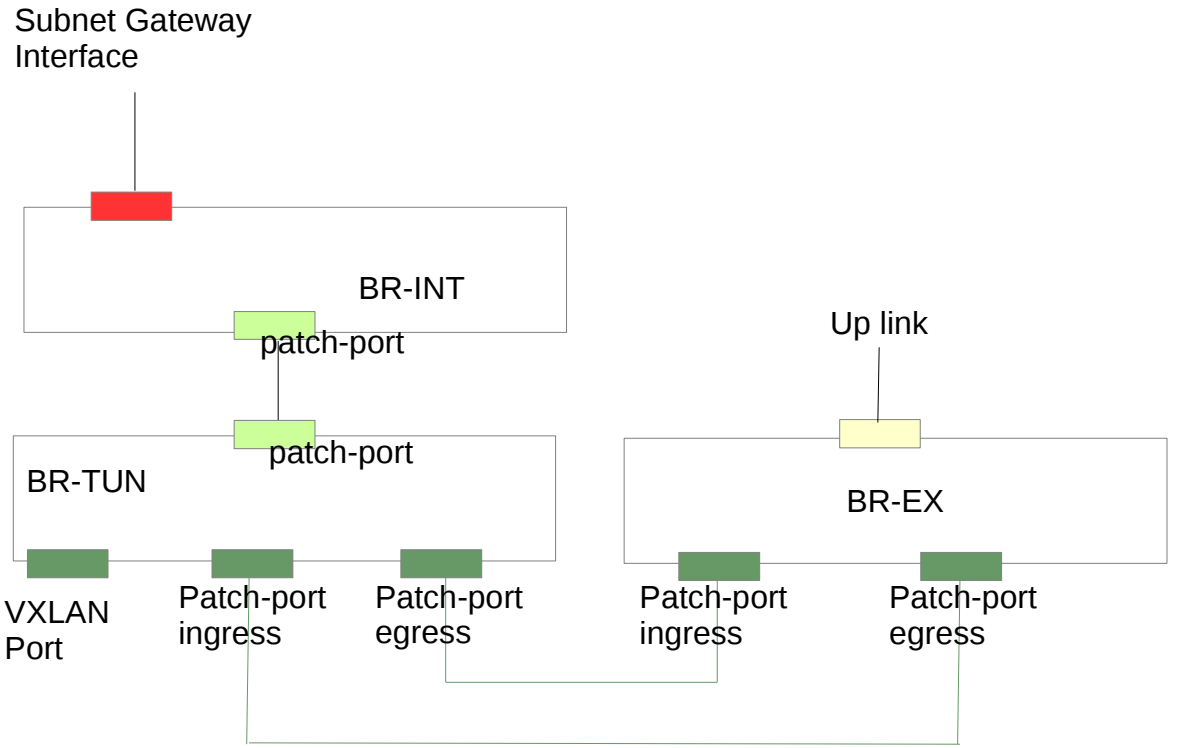
Following is the setup description for POC-

- OpenStack Havana
- ML2 Plugin for OpenStack Neutron
- OVS 2.1
- L2 Population / ARP Responder enabled (introduced in OpenStack Icehouse)
- VXLAN Tunneling (For ‘N’ nodes setup, N-1 VXLAN Tunnel Ports on each node)

OpenStack Neutron Bridge Setup For POC



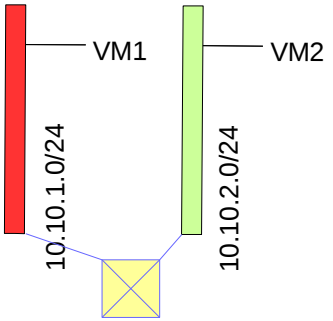
COMPUTE NODE

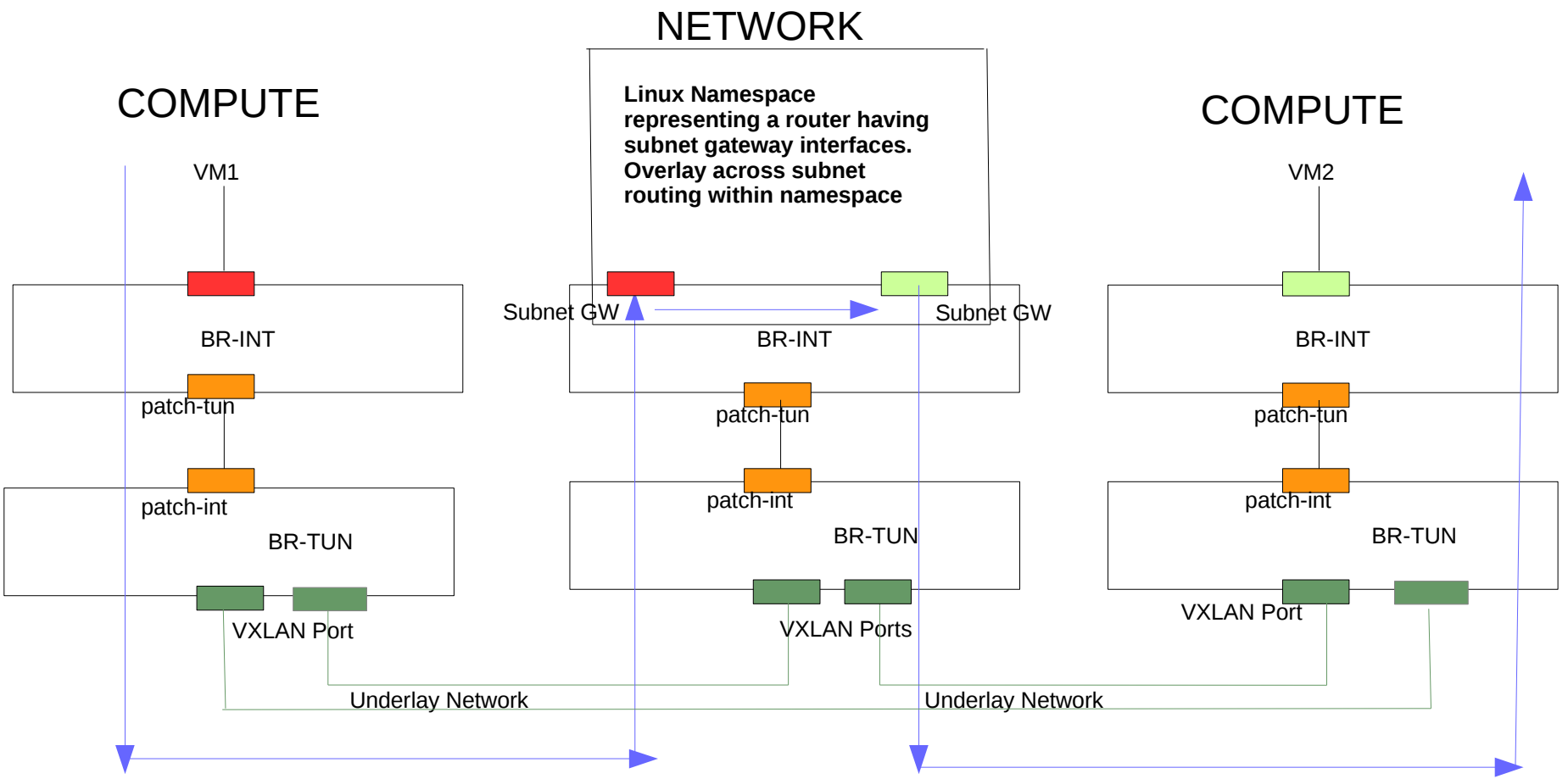


NETWORK NODE

Advantages

- Decentralize overlay across subnet routing decision on each compute node

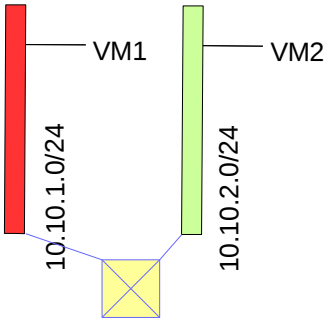




ORIGINAL MECHANISM

Advantages

- Decentralize overlay across subnet routing decision on each compute node (functionality introduced in Openstack Juno)



COMPUTE

NETWORK

COMPUTE

VM1

Localized overlay across subnet routing decision on compute node using flows

BR-INT

patch-tun

patch-int

BR-TUN

VXLAN Port

Subnet GW

Subnet GW

BR-INT

patch-tun

patch-int

BR-TUN

VXLAN Ports

VM2

BR-INT

patch-tun

patch-int

BR-TUN

VXLAN Port

Underlay Network

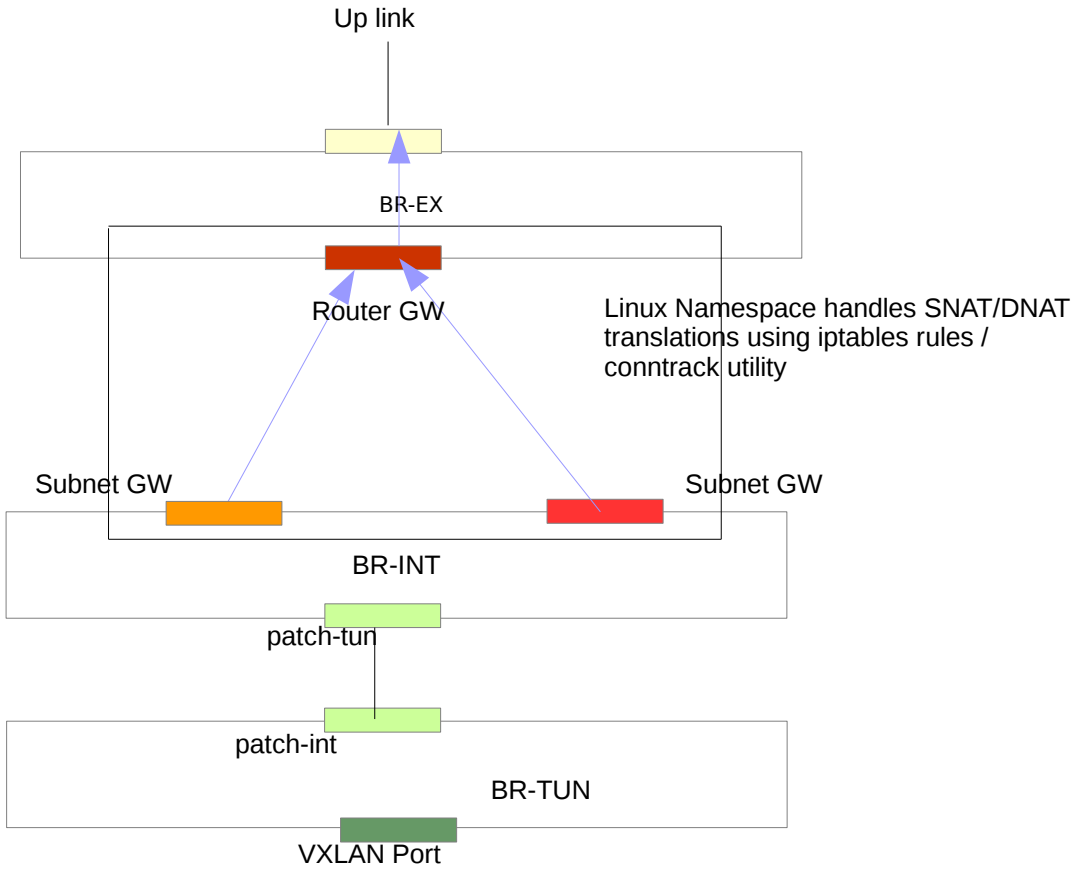
Underlay Network

NEW MECHANISM

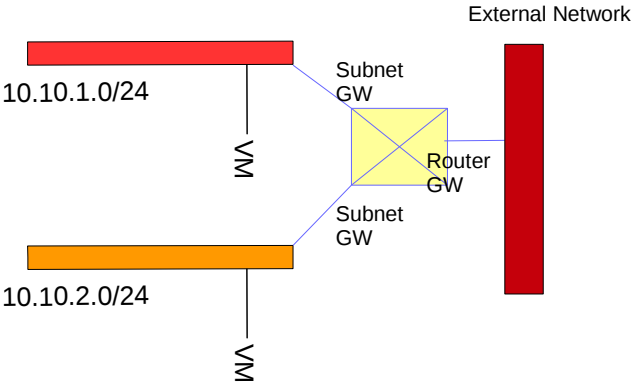
Advantages

- Virtualize Neutron router using flows instead of Linux Namespace, iptables, Host Stack

- SNAT / DNATing using flows



NETWORK NODE

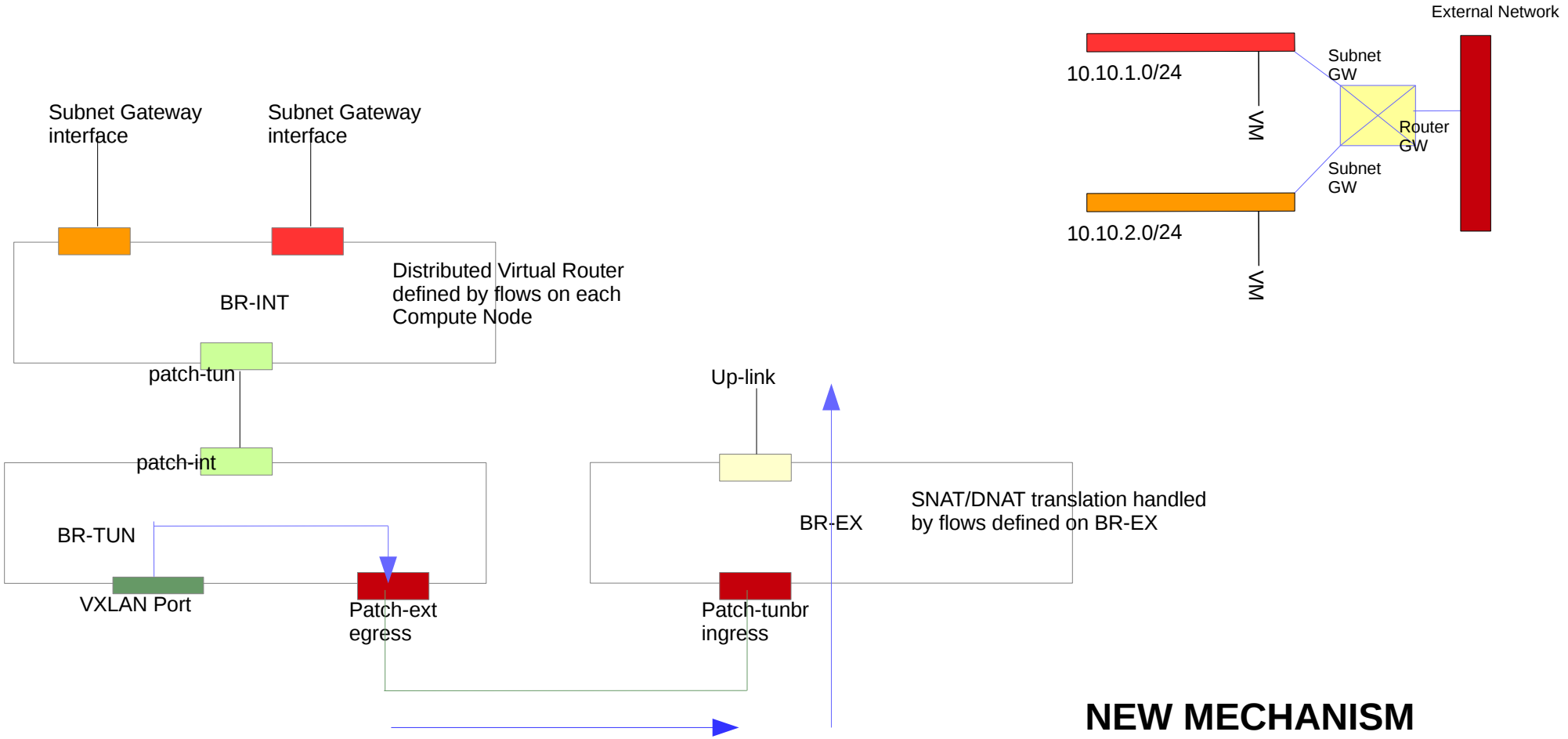


ORIGINAL MECHANISM

Advantages

- Virtualize Neutron router using flows instead of Linux Namespace, iptables, Host Stack

- SNAT / DNATing using flows

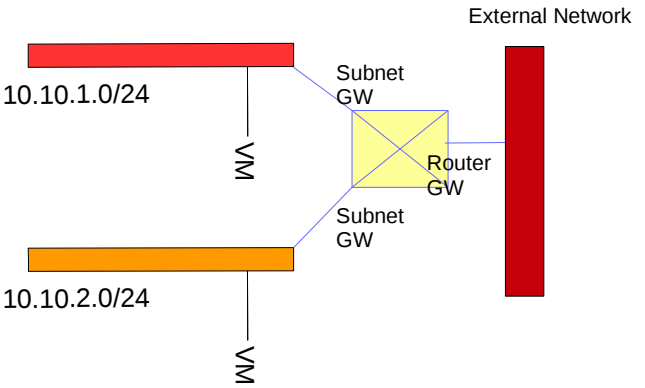


NEW MECHANISM

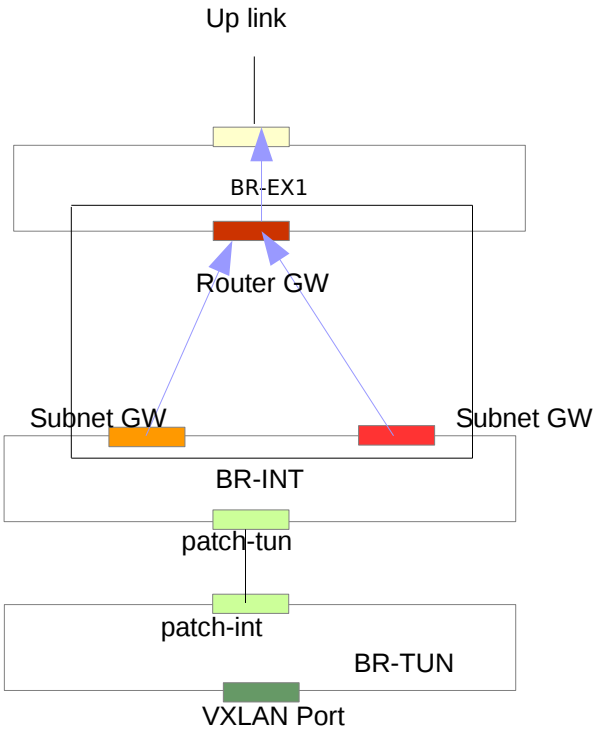
Advantages

- Manage all external networks using single instance of agent

- Multiple external networks can use same bridge

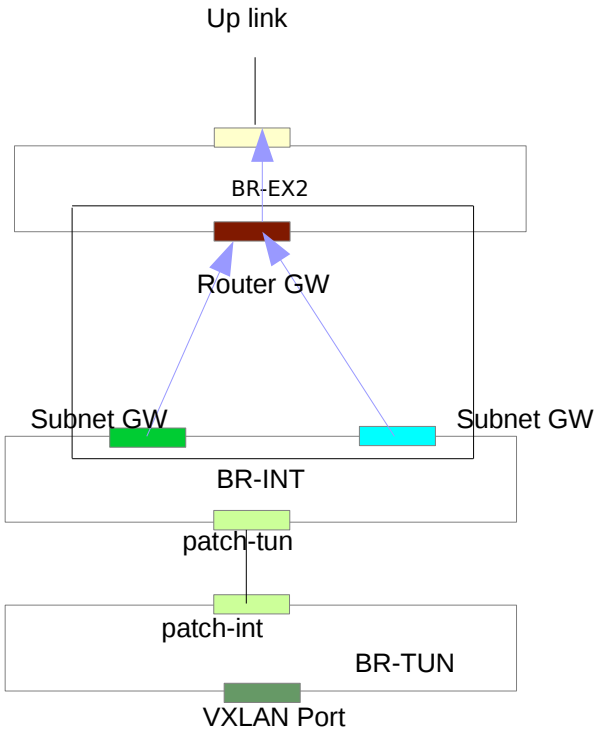


neutron-l3-agent (1st instance)

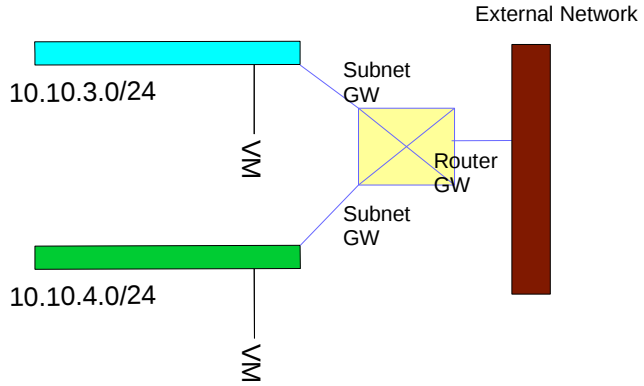


NETWORK NODE

neutron-l3-agent (2nd instance)



NETWORK NODE

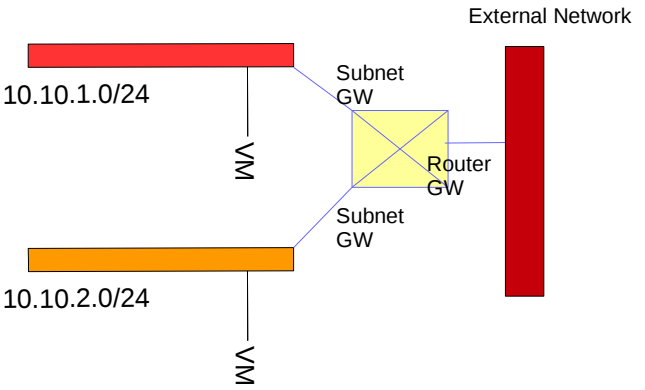


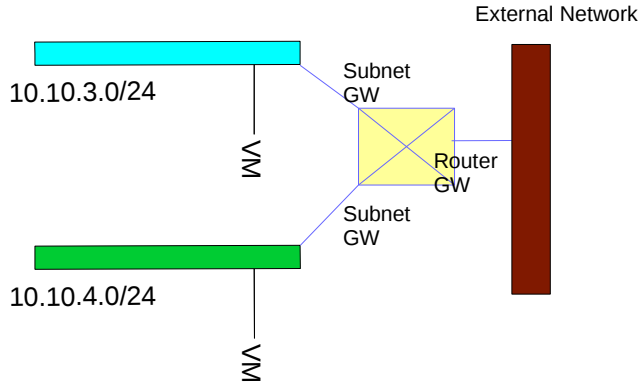
ORIGINAL MECHANISM

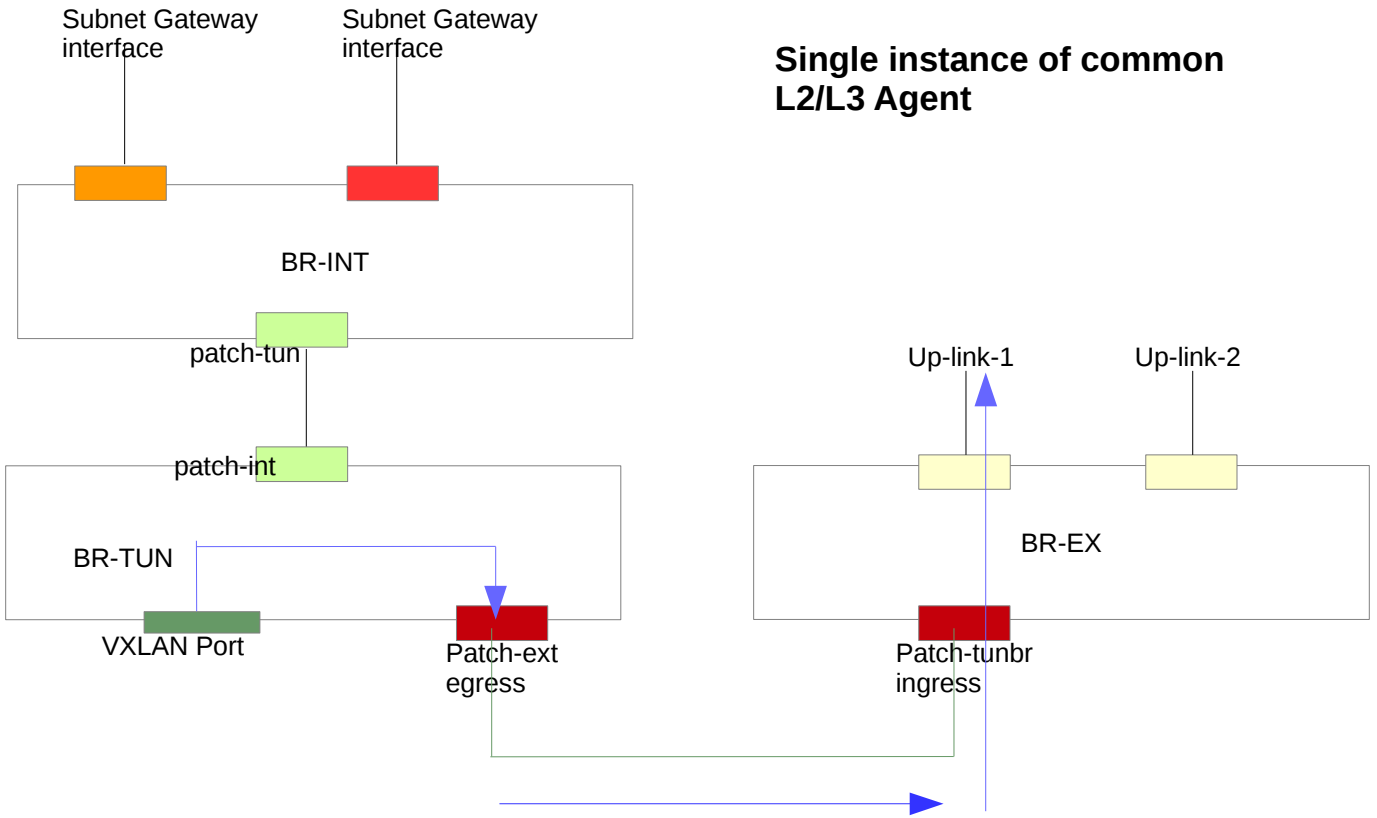
Advantages

- Manage all external networks using single instance of agent (functionality introduced in Openstack Icehouse)

- Multiple external networks can use same bridge







Single instance of common L2/L3 Agent

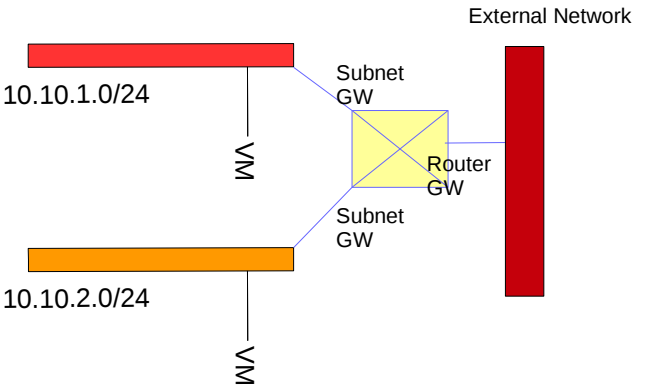
NEW MECHANISM

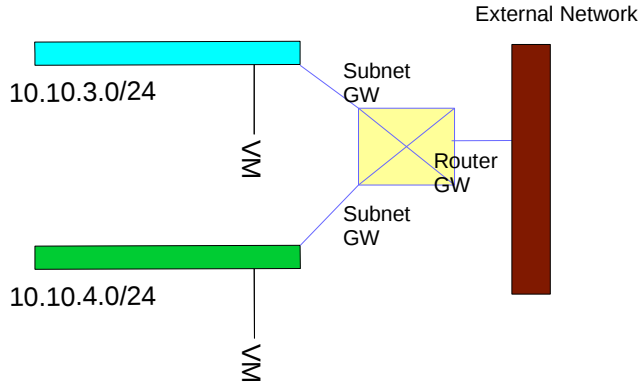
NETWORK NODE

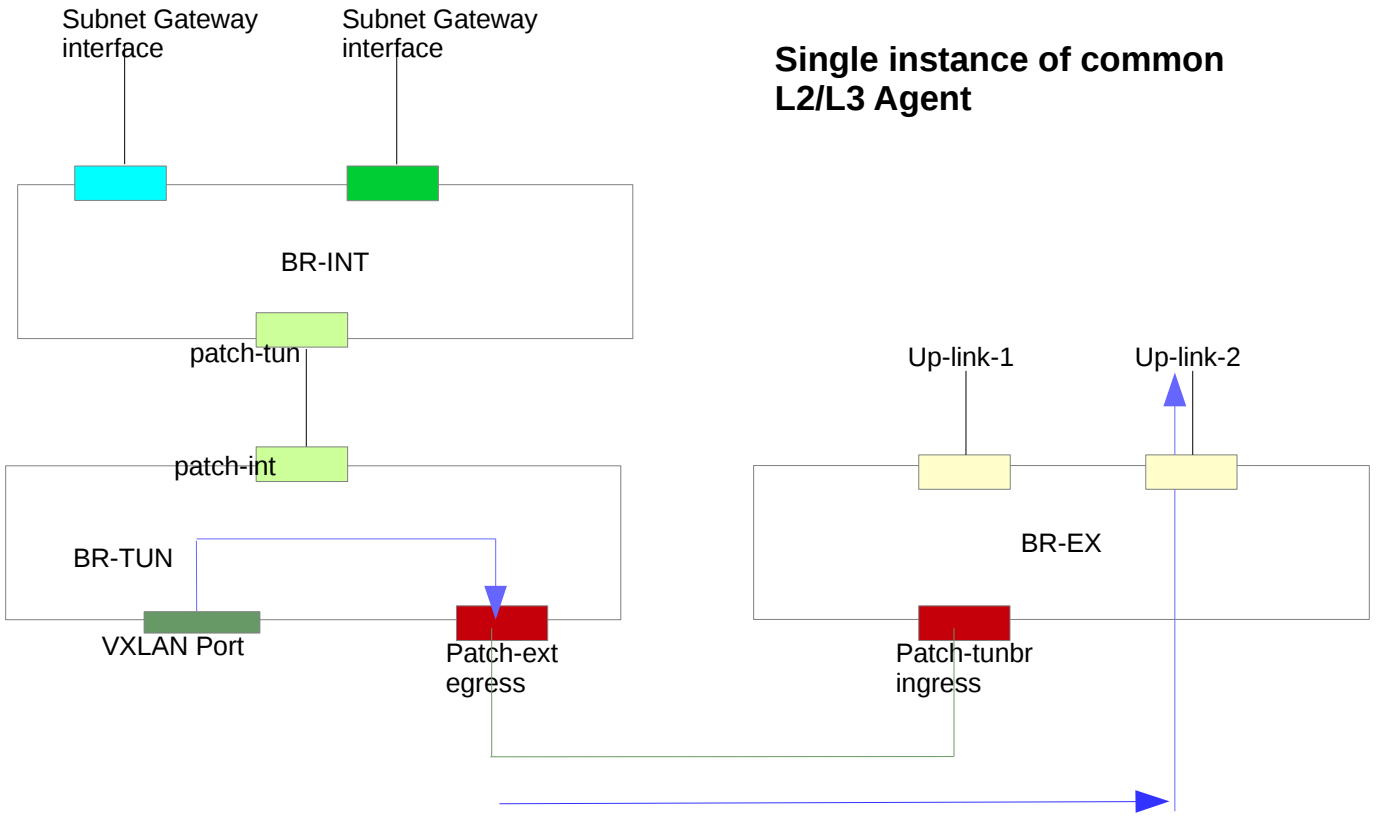
Advantages

- Manage all external networks using single instance of agent (functionality introduced in Openstack Icehouse)

- Multiple external networks can use same bridge







Single instance of common L2/L3 Agent

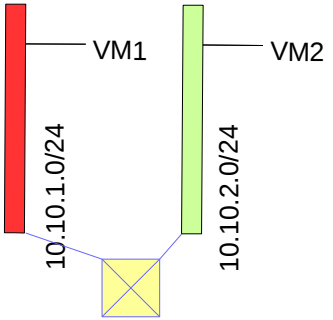
NETWORK NODE

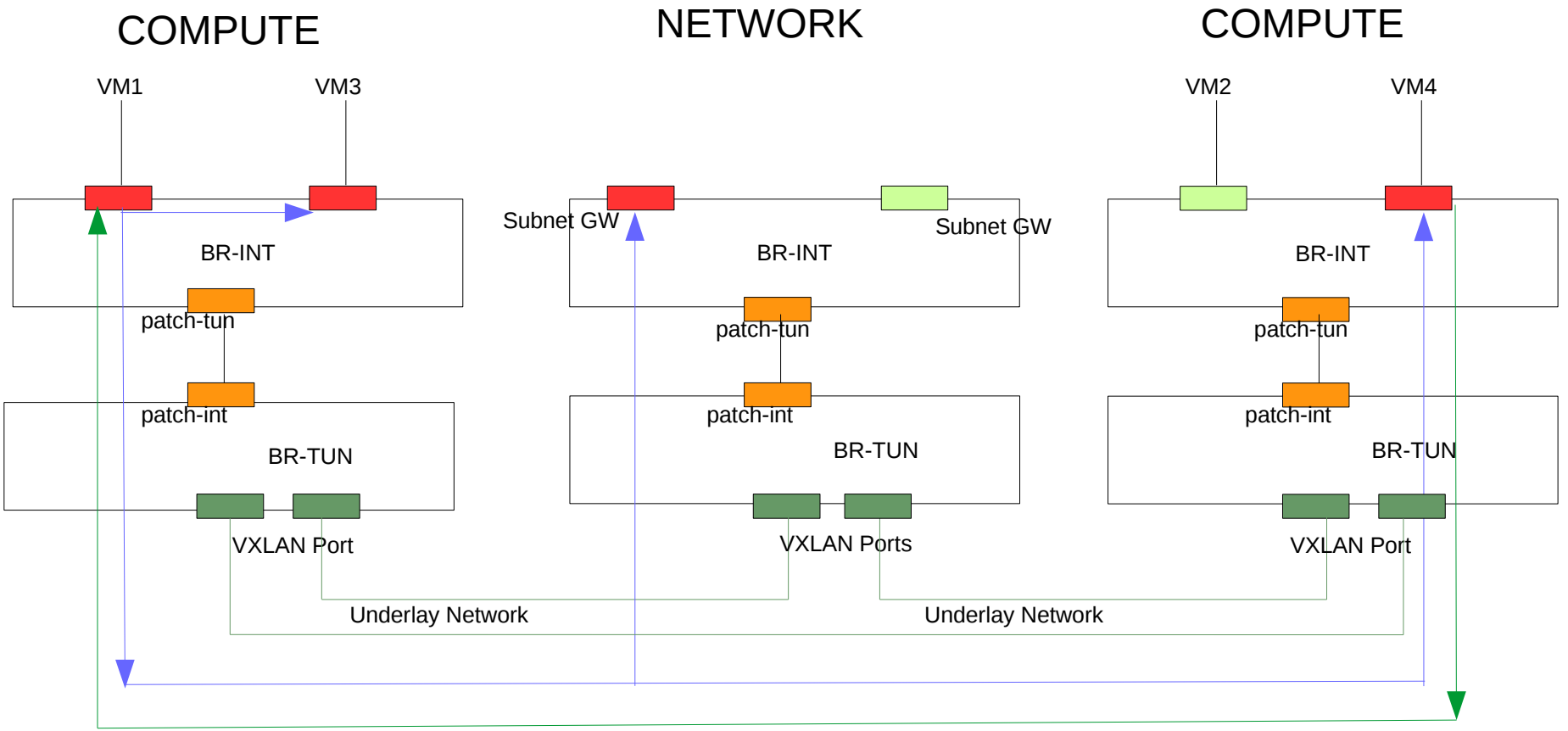
NEW MECHANISM

Advantages

- Minimize ARP Request flooding using ARP Responder / L2 Population

(functionality derived from Openstack Havana)



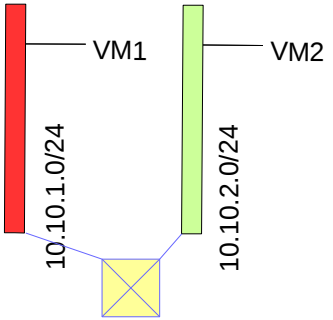


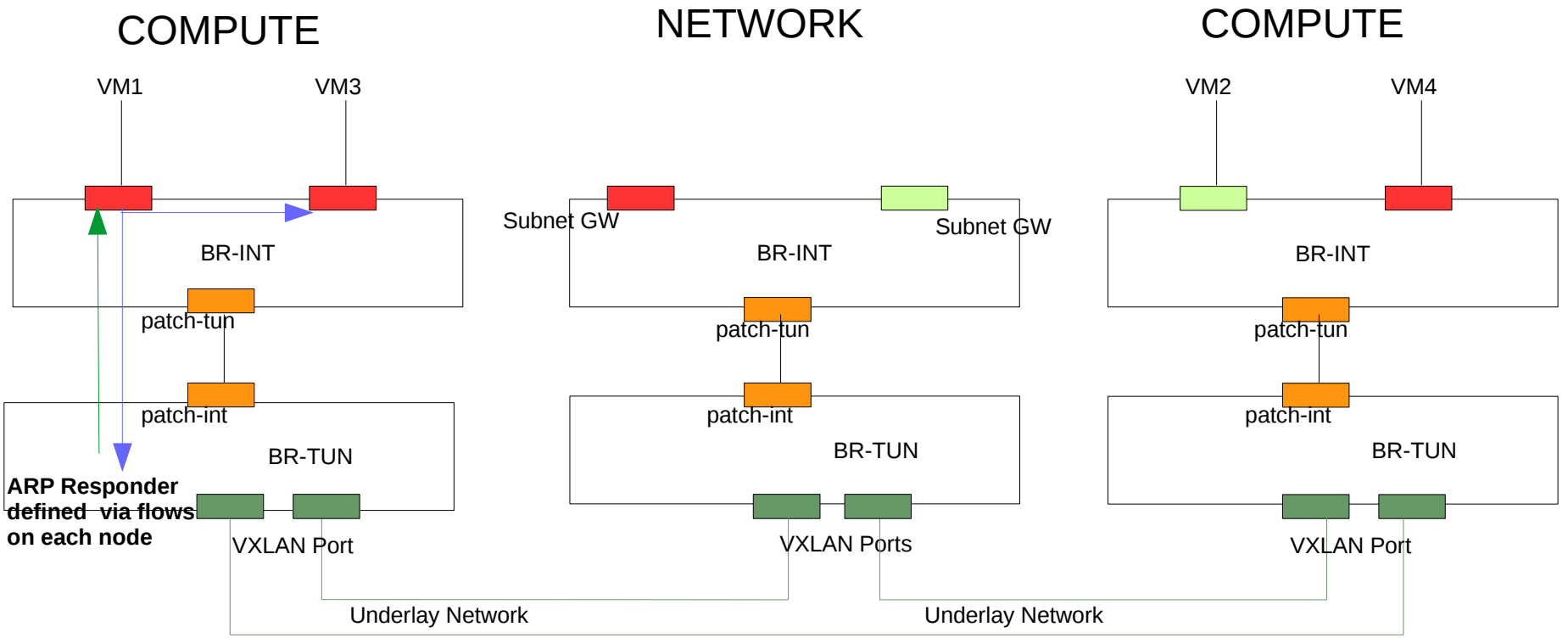
ORIGINAL MECHANISM (FLOODING OF ARP PACKET)

Advantages

- Minimize ARP Request flooding using ARP Responder / L2 Population

(functionality derived from Openstack Havana)



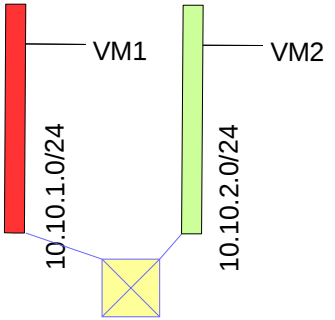


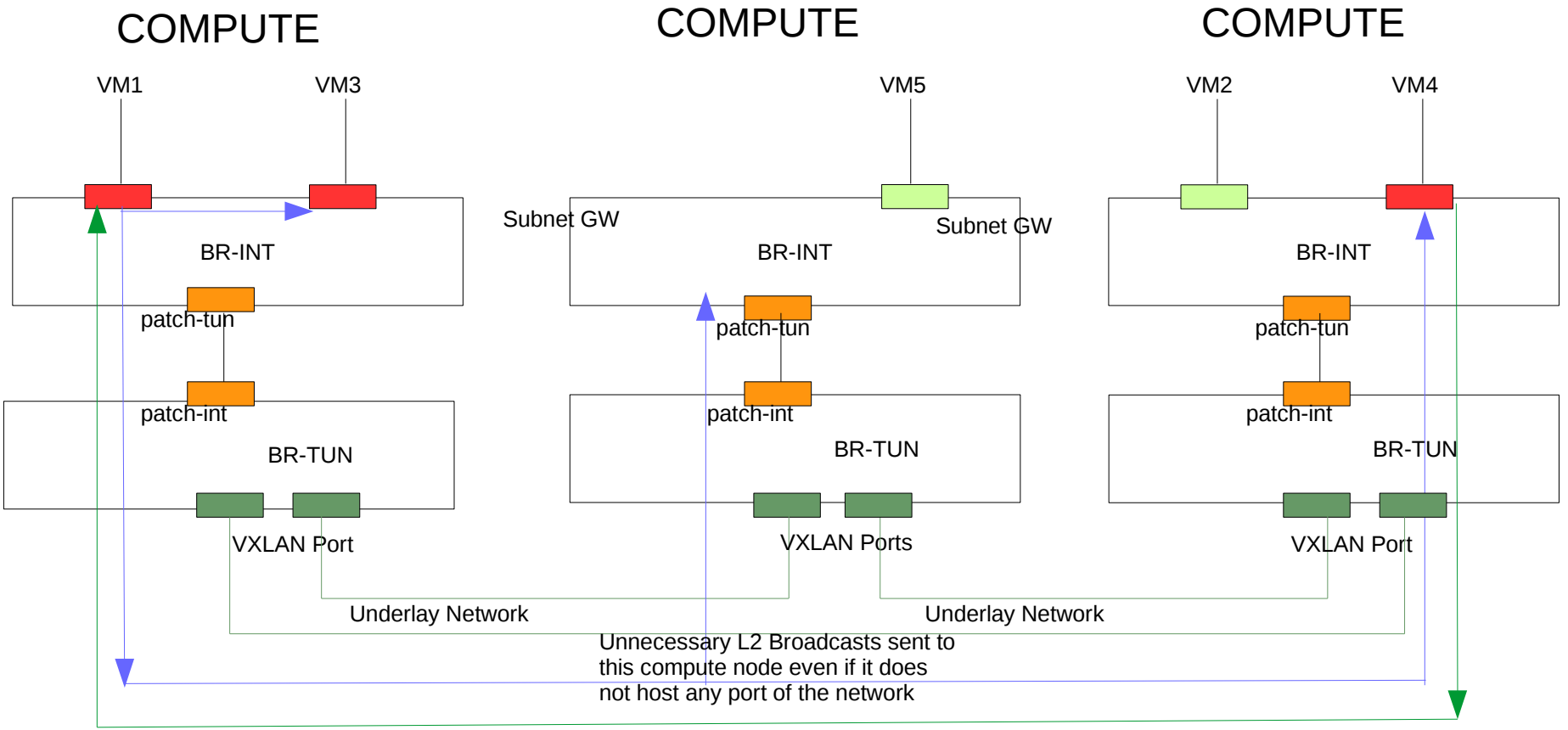
NEW MECHANISM (ARP RESPONDER)

Advantages

- Minimize ARP Request flooding using ARP Responder / L2 Population

(functionality derived from Openstack Havana)



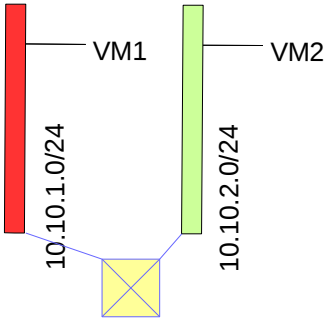


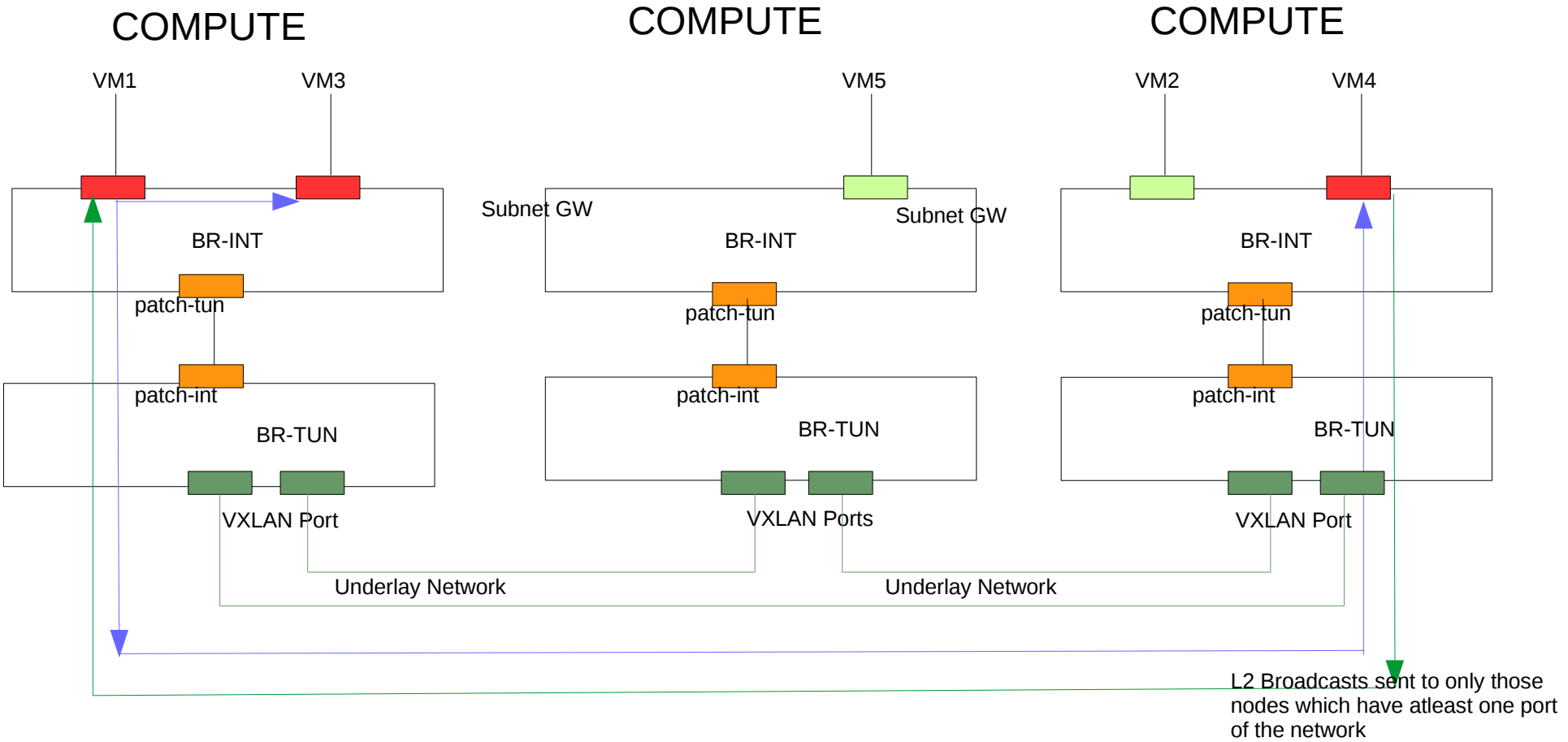
ORIGINAL MECHANISM

Advantages

- Minimize ARP Request flooding using ARP Responder / L2 Population

(functionality derived from Openstack Havana)





NEW MECHANISM (With L2 Population)

Performance Analysis

Hardware Specification-

- 4 nodes setup (1 controller / network and 3 compute nodes)

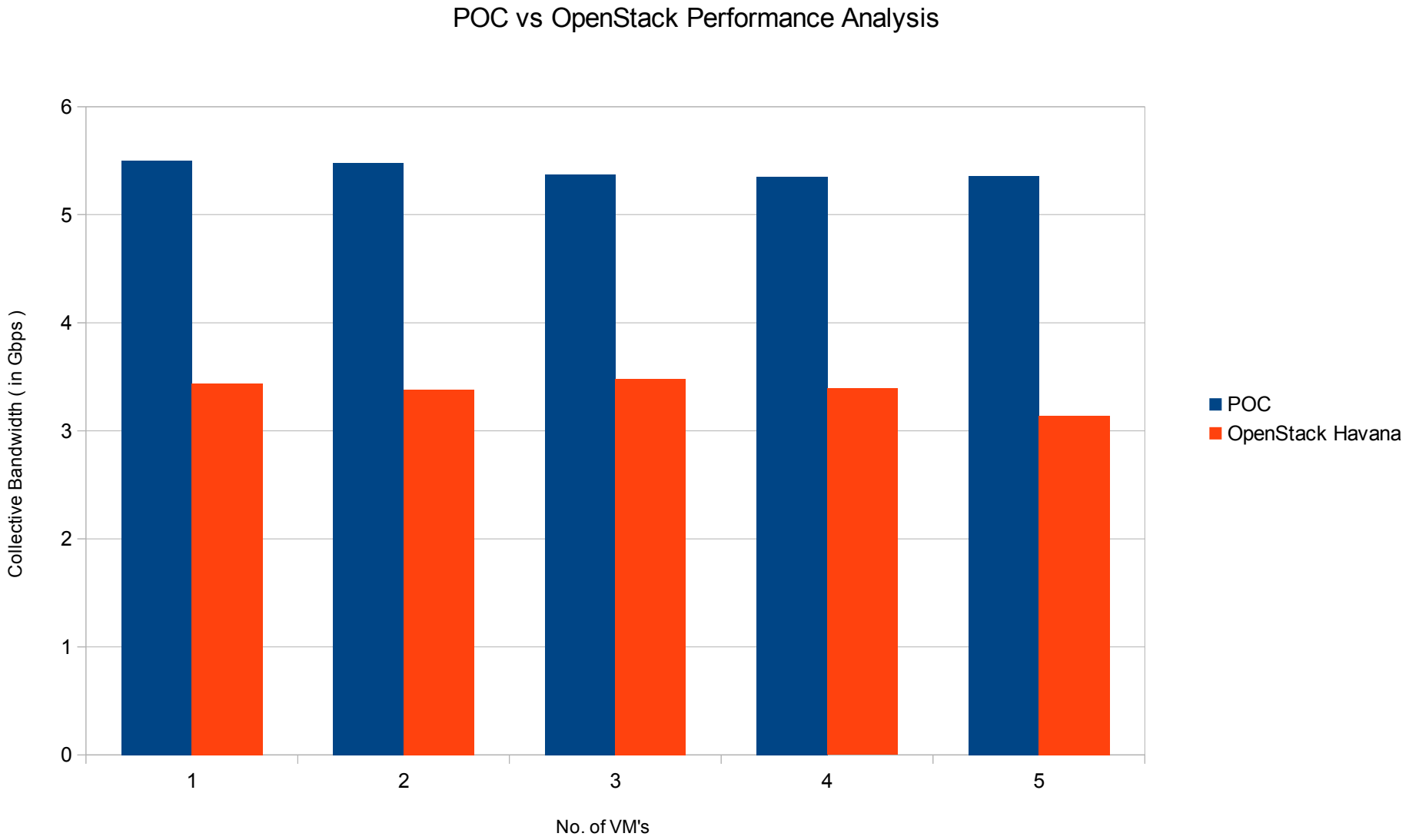
- Each node connected to Tunnel Network via 10 Gbps link

- Network Node connected to External Network via 10 Gbps link

- VM configuration – 2 VCPUs, 2048 MB RAM, CentOS

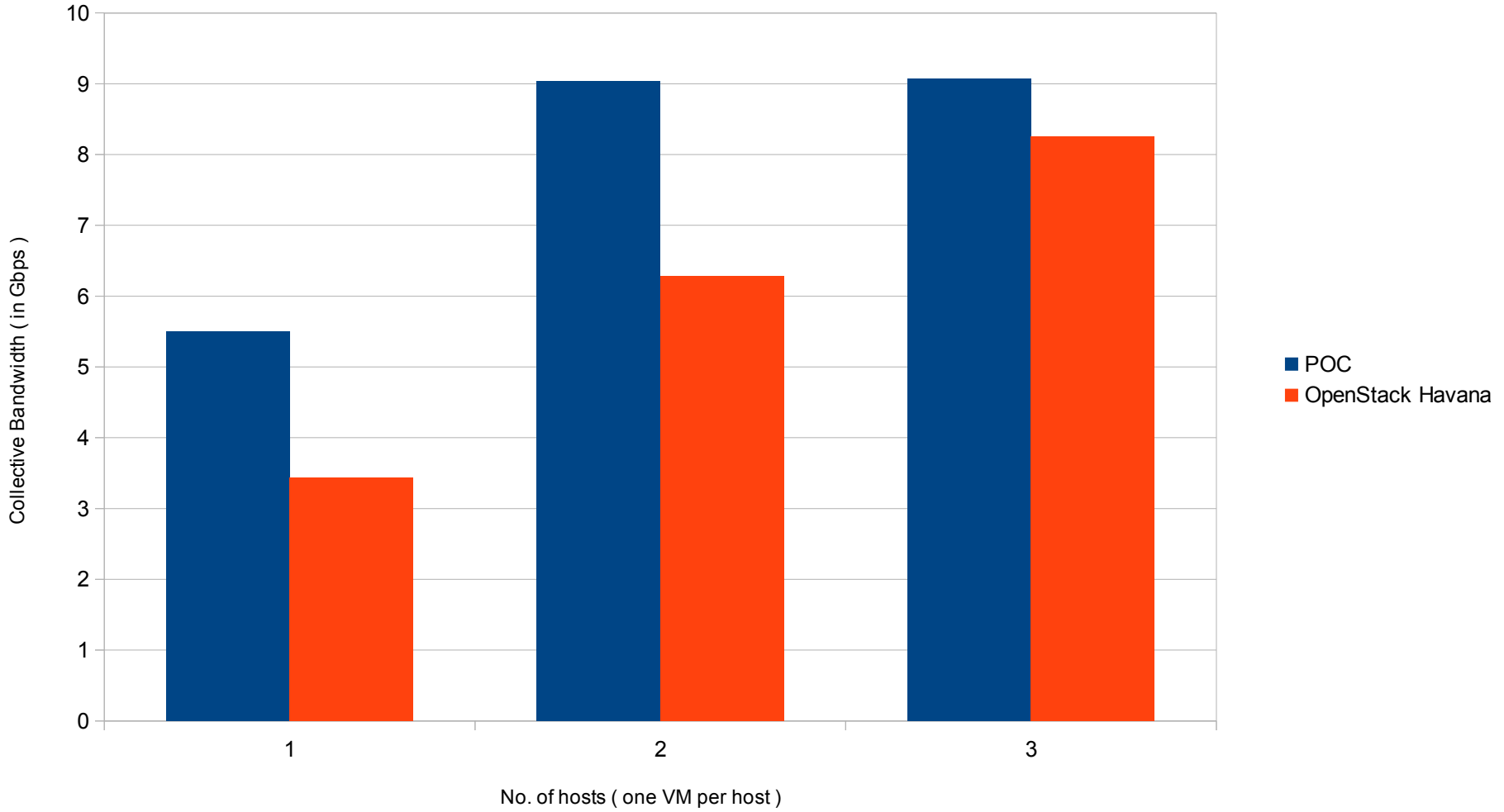
Performance Analysis

Use Case - SNAT for single host



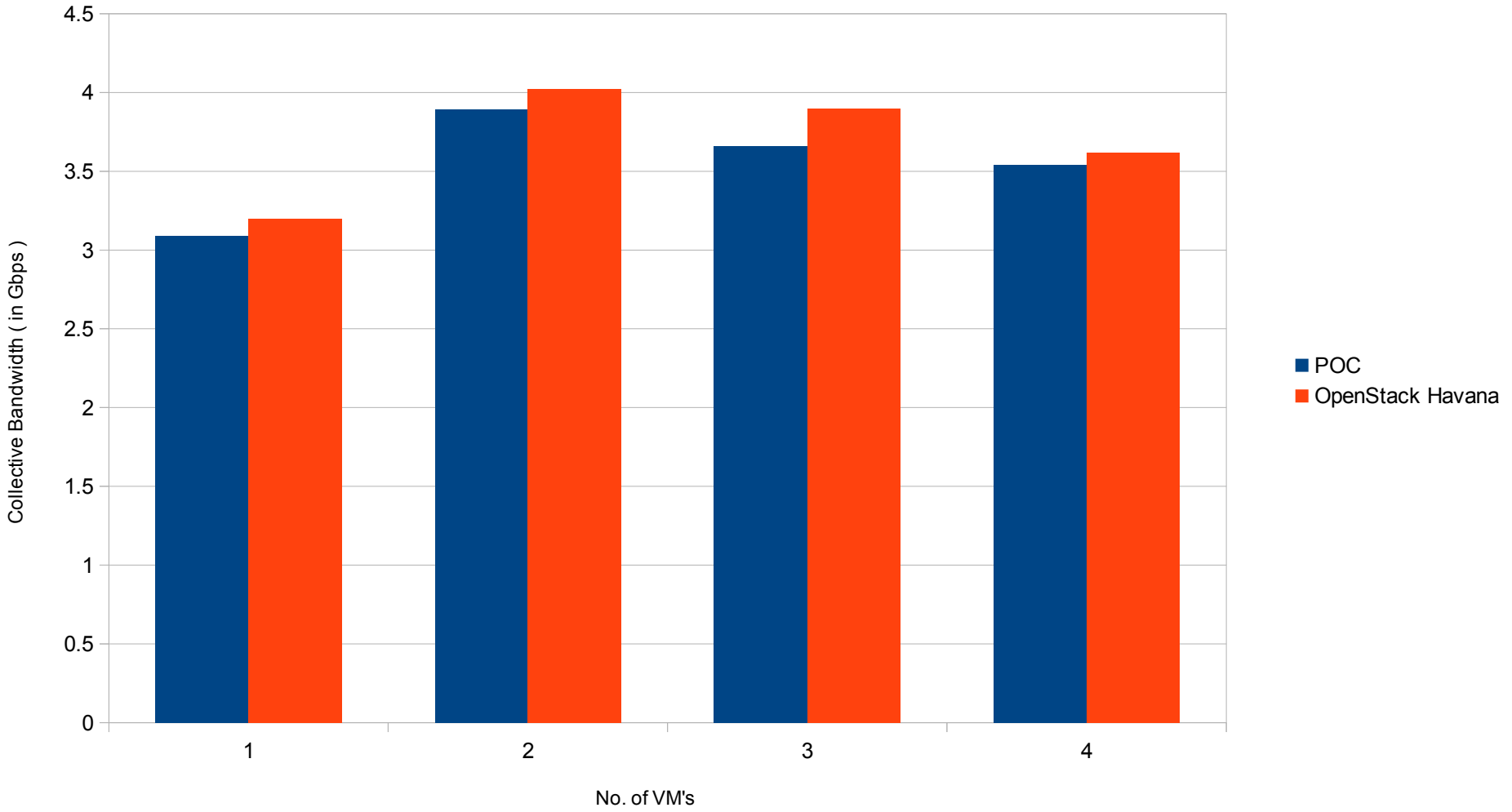
Performance Analysis

Use Case - SNAT for multiple hosts



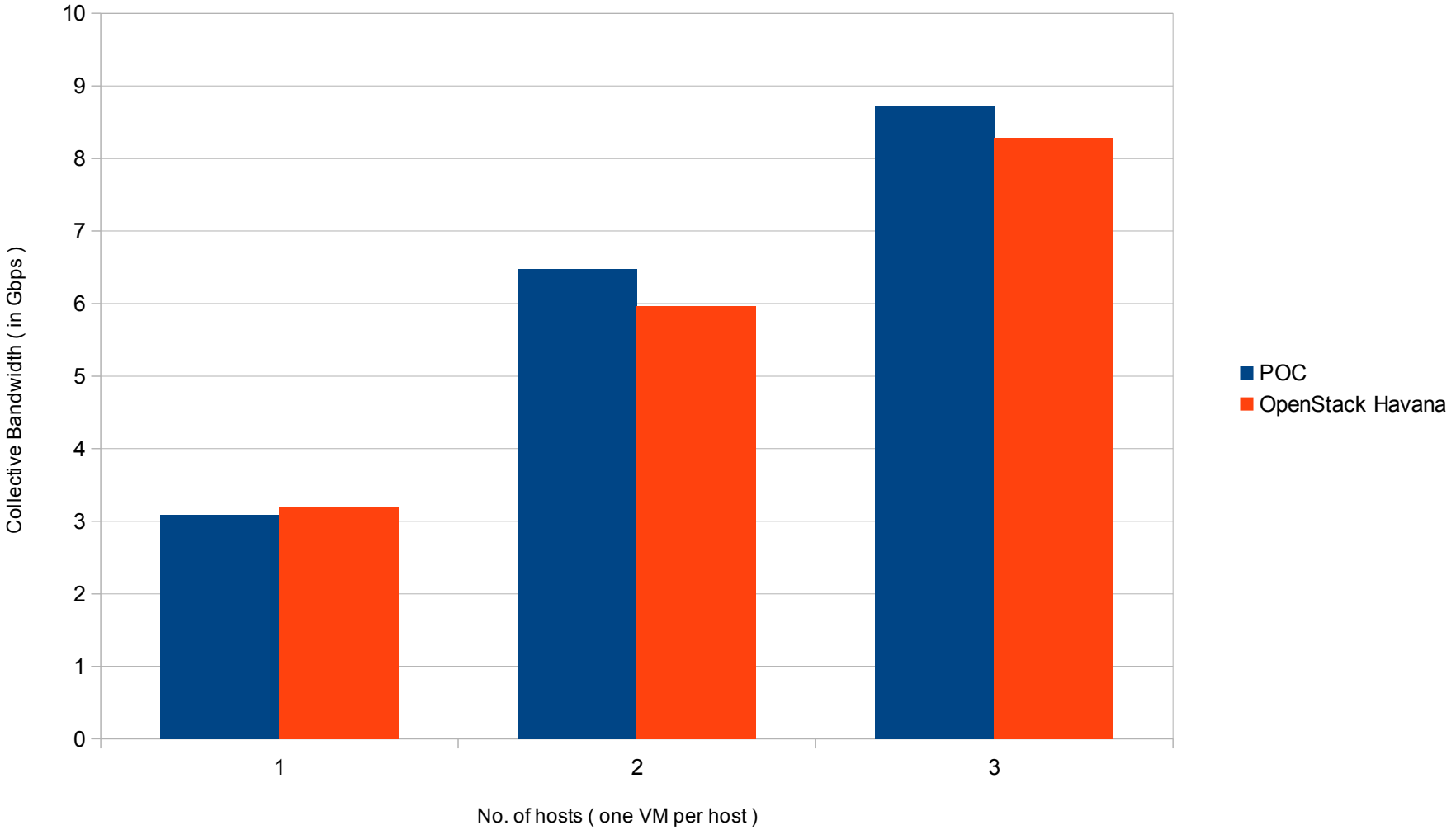
Performance Analysis

Use Case - DNAT for single host



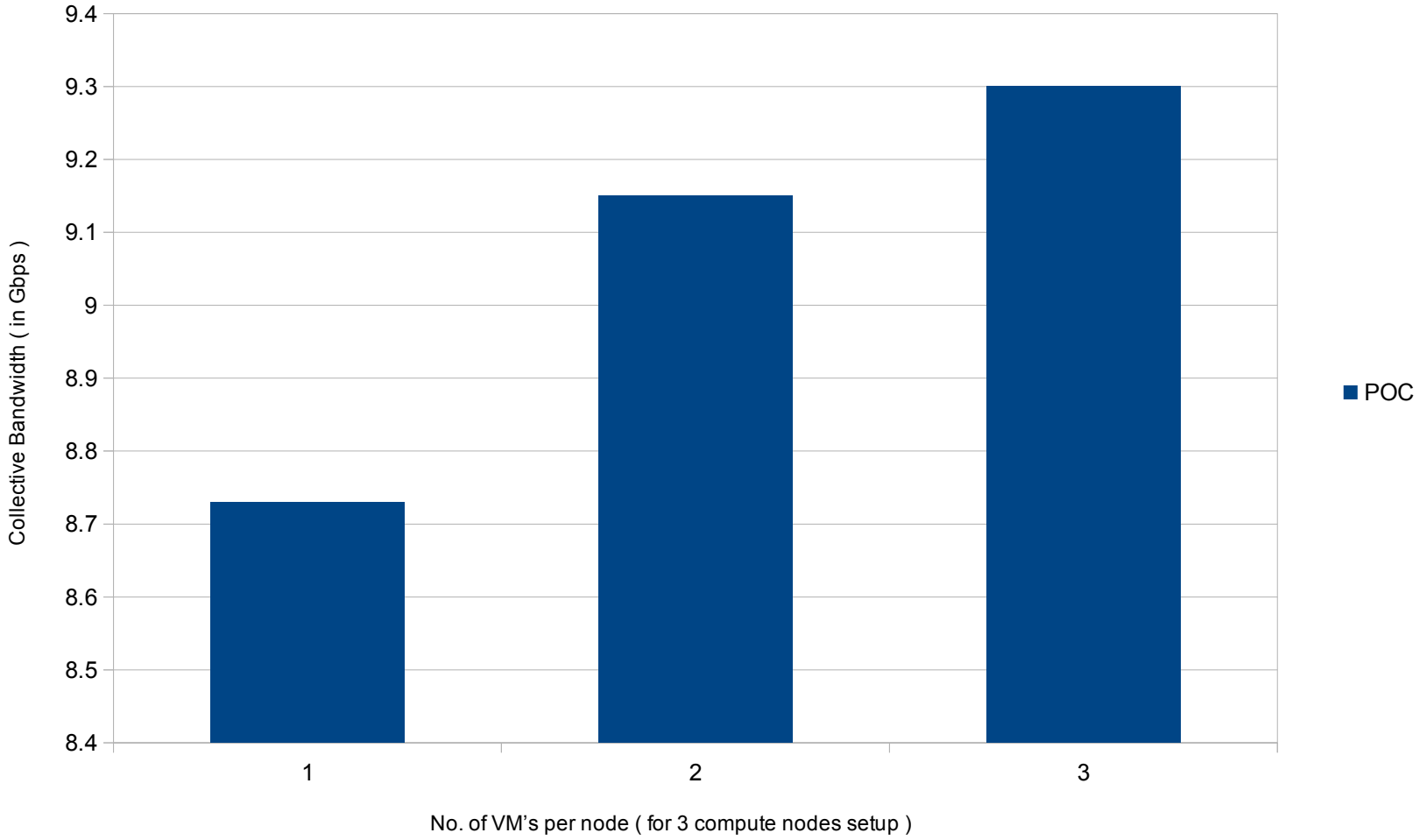
Performance Analysis

Use Case - DNAT for multiple host



Performance Analysis

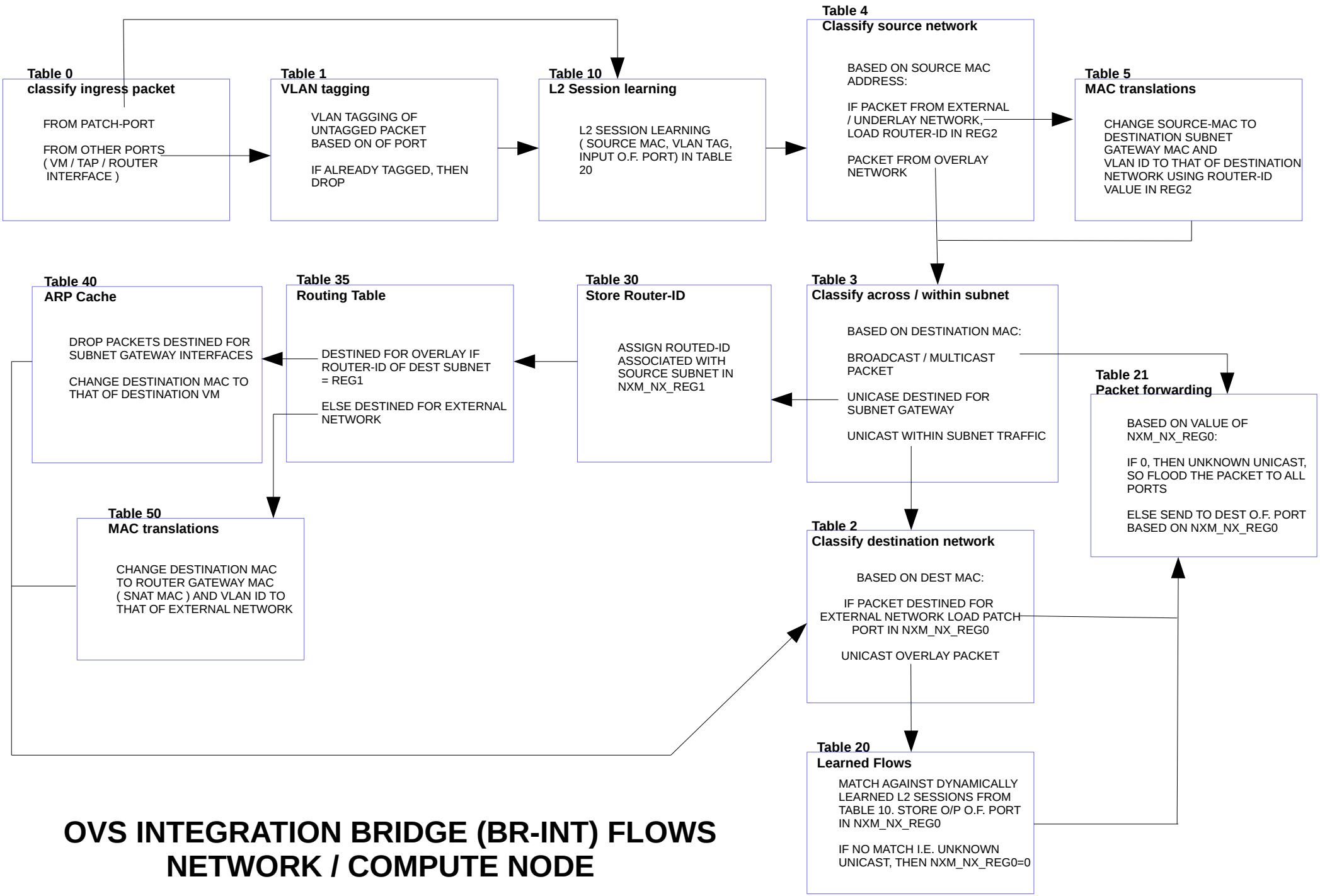
Use Case - DNAT for multiple host (POC only)



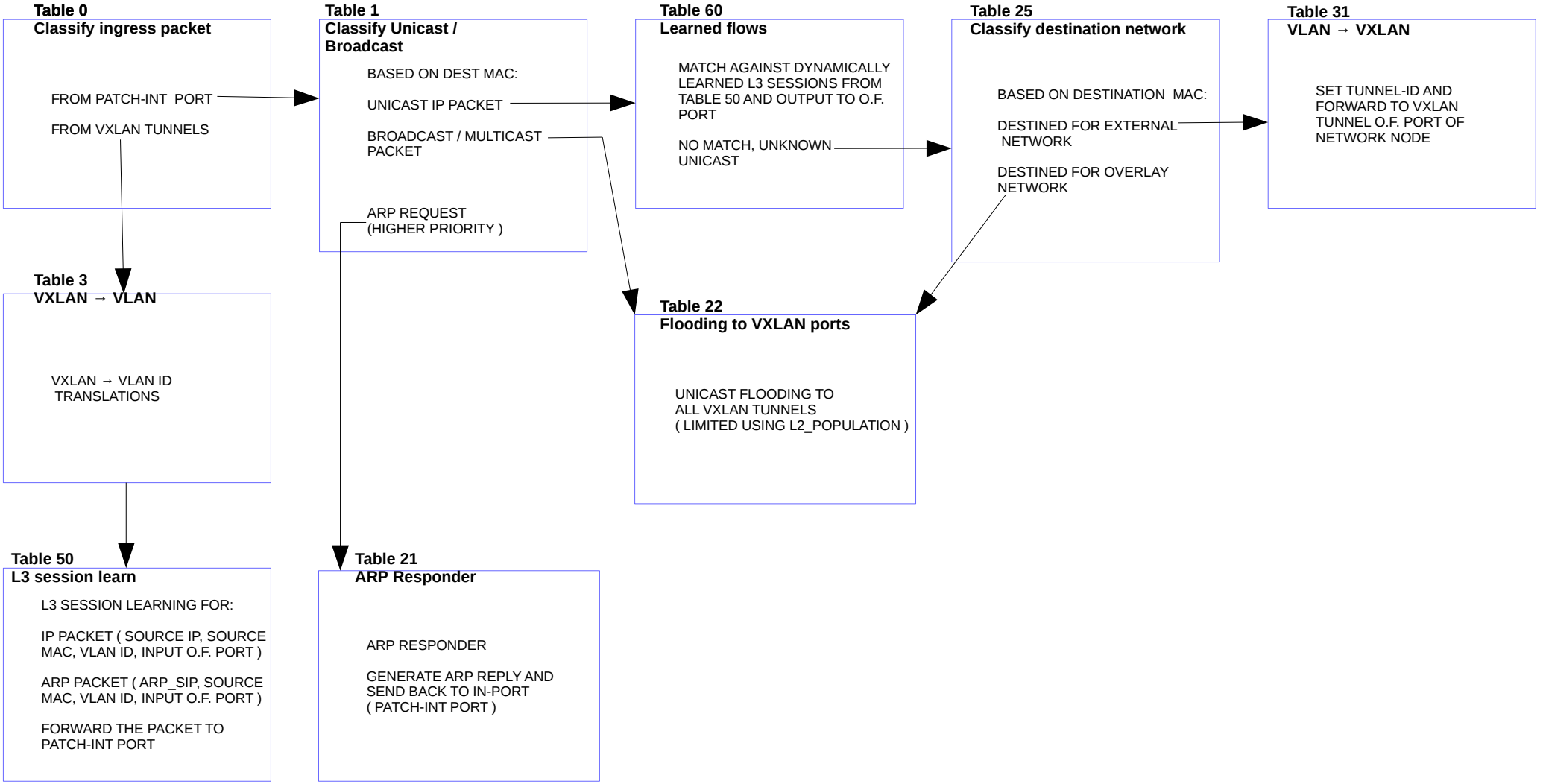
Future Work

- ICMP support for router interfaces by generating ICMP replies using the new Agent
- SNAT support for ICMP packets. Implement OpenFlow protocol to allow access to ICMP identifier field and use port mapping for ICMP and TCP/UDP sessions
- Implement OpenStack Nova Security Groups using flows on OVS integration bridge, thus, mapping VM’s directly on the OVS integration bridge
- Updating ARP cache for external network at OVS external bridge programatically
- Updating ARP Responder / Cache on OVS tunnel / integration bridge programatically
- Deletion of flows in a programmatic manner

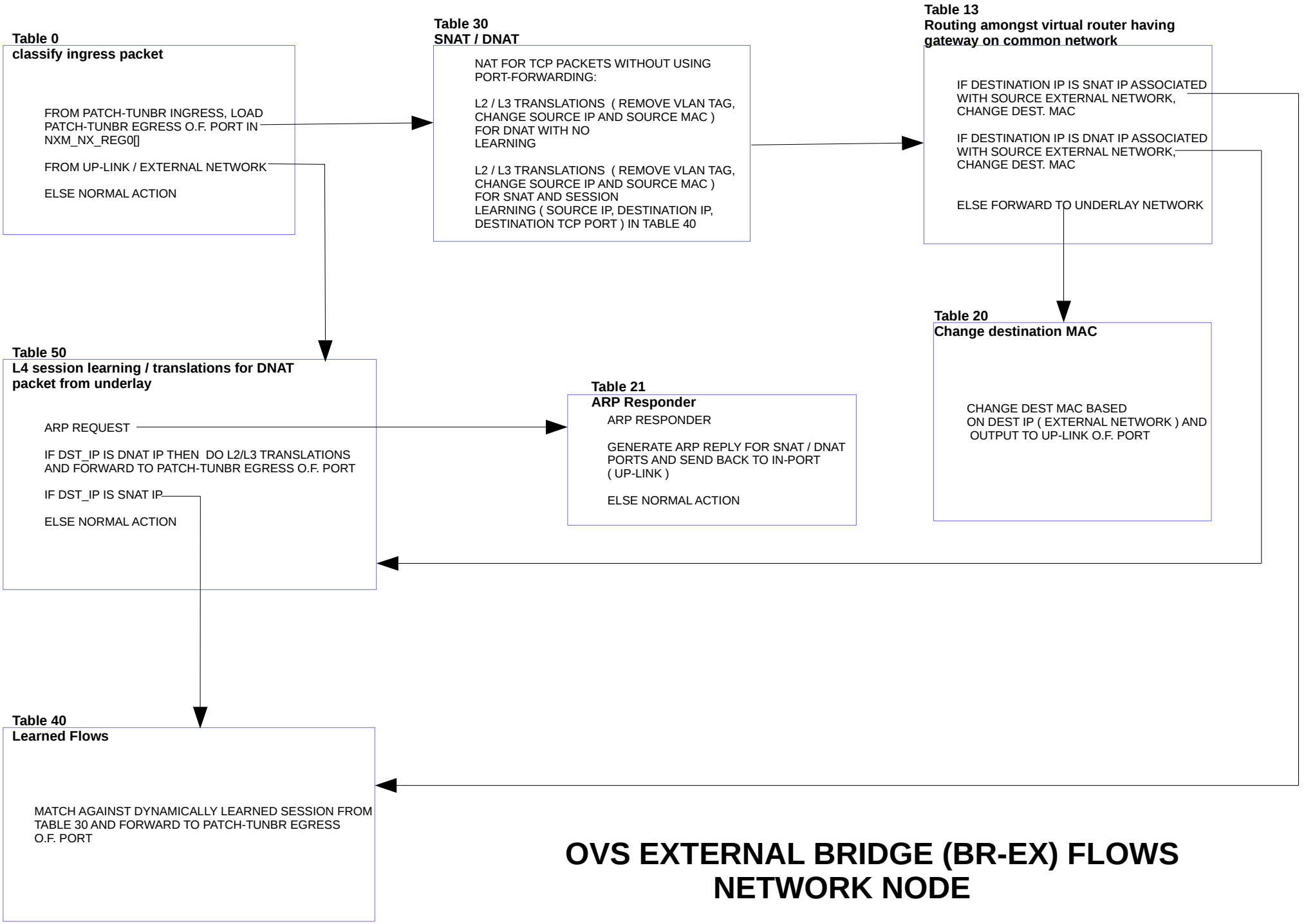
BACKUP SLIDES



OVS INTEGRATION BRIDGE (BR-INT) FLOWS
NETWORK / COMPUTE NODE



OVS TUNNEL (BR-TUN) FLOWS
COMPUTE NODE



OVS EXTERNAL BRIDGE (BR-EX) FLOWS NETWORK NODE

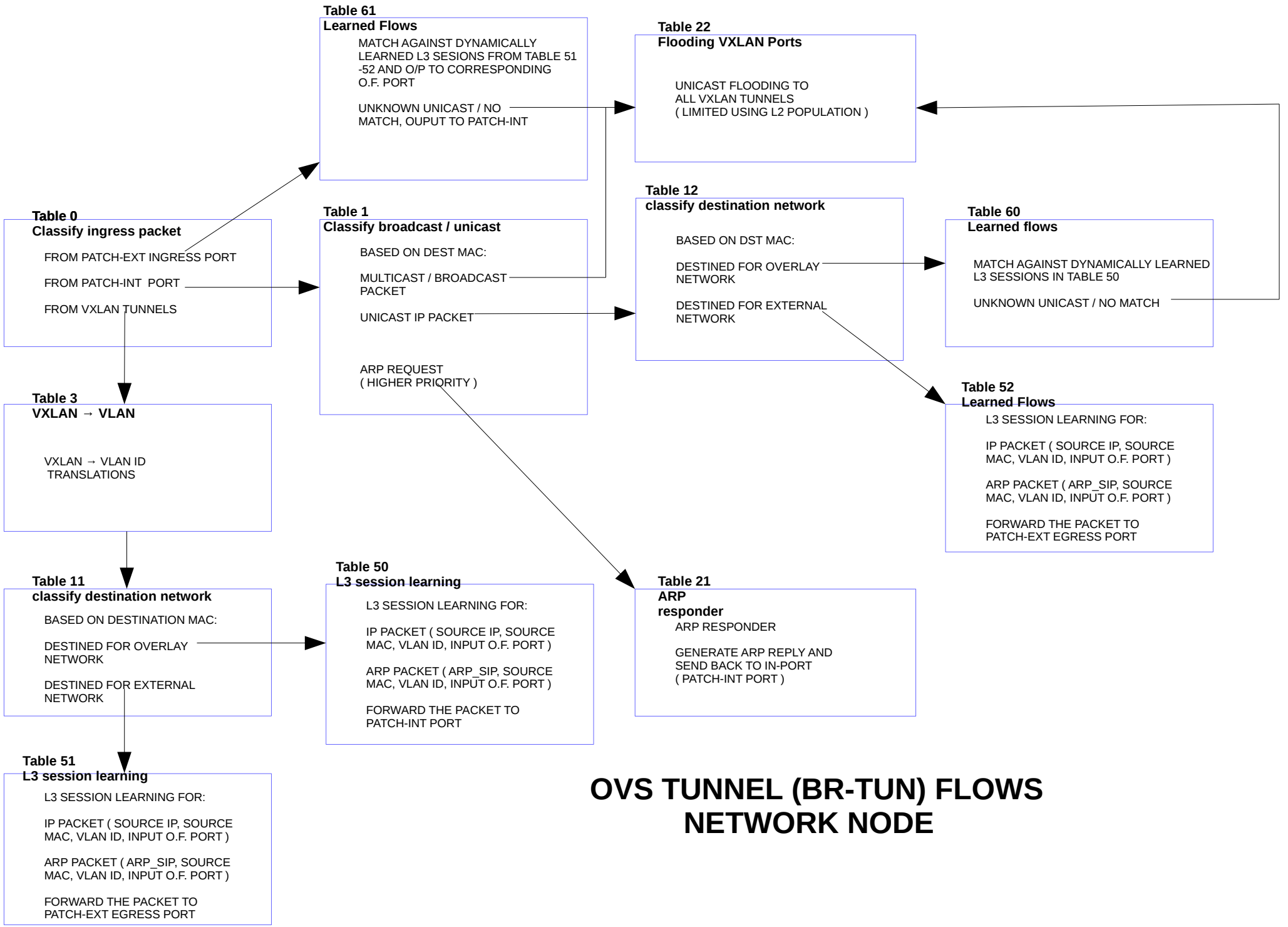


Table 61
Learned Flows
MATCH AGAINST DYNAMICALLY LEARNED L3 SESIONS FROM TABLE 51 -52 AND O/P TO CORRESPONDING O.F. PORT
UNKNOWN UNICAST / NO MATCH, OUPUT TO PATCH-INT
Table 22
Flooding VXLAN Ports
UNICAST FLOODING TO ALL VXLAN TUNNELS (LIMITED USING L2 POPULATION)
Table 0
Classify ingress packet
FROM PATCH-EXT INGRESS PORT
FROM PATCH-INT PORT
FROM VXLAN TUNNELS
Table 1
Classify broadcast / unicast
BASED ON DEST MAC:
MULTICAST / BROADCAST PACKET
UNICAST IP PACKET
ARP REQUEST (HIGHER PRIORITY)
Table 12
classify destination network
BASED ON DST MAC:
DESTINED FOR OVERLAY NETWORK
DESTINED FOR EXTERNAL NETWORK
Table 60
Learned flows
MATCH AGAINST DYNAMICALLY LEARNED L3 SESSIONS IN TABLE 50
UNKNOWN UNICAST / NO MATCH
Table 3
VXLAN → VLAN
VXLAN → VLAN ID TRANSLATIONS
Table 52
Learned Flows
L3 SESSION LEARNING FOR:
IP PACKET (SOURCE IP, SOURCE MAC, VLAN ID, INPUT O.F. PORT)
ARP PACKET (ARP_SIP, SOURCE MAC, VLAN ID, INPUT O.F. PORT)
FORWARD THE PACKET TO PATCH-EXT EGRESS PORT
Table 11
classify destination network
BASED ON DESTINATION MAC:
DESTINED FOR OVERLAY NETWORK
DESTINED FOR EXTERNAL NETWORK
Table 50
L3 session learning
L3 SESSION LEARNING FOR:
IP PACKET (SOURCE IP, SOURCE MAC, VLAN ID, INPUT O.F. PORT)
ARP PACKET (ARP_SIP, SOURCE MAC, VLAN ID, INPUT O.F. PORT)
FORWARD THE PACKET TO PATCH-INT PORT
Table 21
ARP responder
ARP RESPONDER
GENERATE ARP REPLY AND SEND BACK TO IN-PORT (PATCH-INT PORT)
Table 51
L3 session learning
L3 SESSION LEARNING FOR:
IP PACKET (SOURCE IP, SOURCE MAC, VLAN ID, INPUT O.F. PORT)
ARP PACKET (ARP_SIP, SOURCE MAC, VLAN ID, INPUT O.F. PORT)
FORWARD THE PACKET TO PATCH-EXT EGRESS PORT
OVS TUNNEL (BR-TUN) FLOWS
NETWORK NODE

Performance Analysis

Use Case - SNAT for single host (POC)

No. of VM's	Bandwidth (Gbps)
1	5.5
2	5.48
3	5.37
4	5.35
5	5.36

Use Case - SNAT for single host (OpenStack)

No. of VM's	Bandwidth (Gbps)
1	3.44
2	3.38
3	3.48
4	3.39
5	3.14

Performance Analysis

Use Case - SNAT for multiple hosts (POC)

No. of hosts	No. of VMs	Bandwidth (Gbps)
2	2 (one per host)	9.03
3	3 (one per host)	9.07

Use Case - SNAT for multiple hosts (OpenStack)

No. of hosts	No. of VMs	Bandwidth (Gbps)
2	2 (one per host)	6.28
3	3 (one per host)	8.25

Performance Analysis

Use Case - DNAT for single host (POC)

No. of VM's	Bandwidth (Gbps)
1	3.09
2	3.89
3	3.66
4	3.54

Use Case - DNAT for single host (OpenStack)

No. of VM's	Bandwidth (Gbps)
1	3.2
2	4.02
3	3.9
4	3.618

Performance Analysis

Use Case - DNAT for multiple hosts (POC)

No. of hosts	No. of VM's	Bandwidth (Gbps)
2	2 (one per host)	6.48
3	3 (one per host)	8.73
3	6 (two per host)	9.15
3	9 (three per host)	9.3

Use Case 4- DNAT for multiple hosts (OpenStack)

No. of hosts	No. of VM's	Bandwidth (Gbps)
2	2 (one per host)	5.97
3	3 (one per host)	8.29