



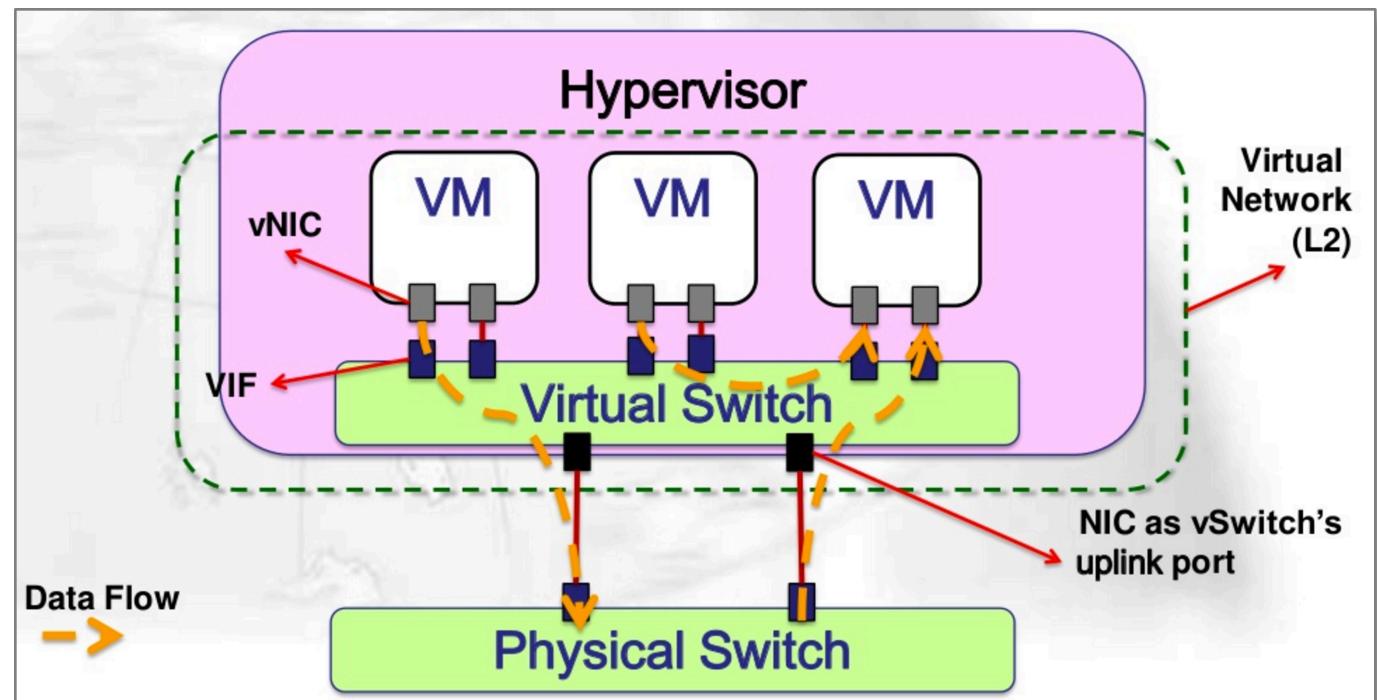
Open vSwitch

2016-study-a-3rd-week 김기덕

Open vSwitch

- 개요

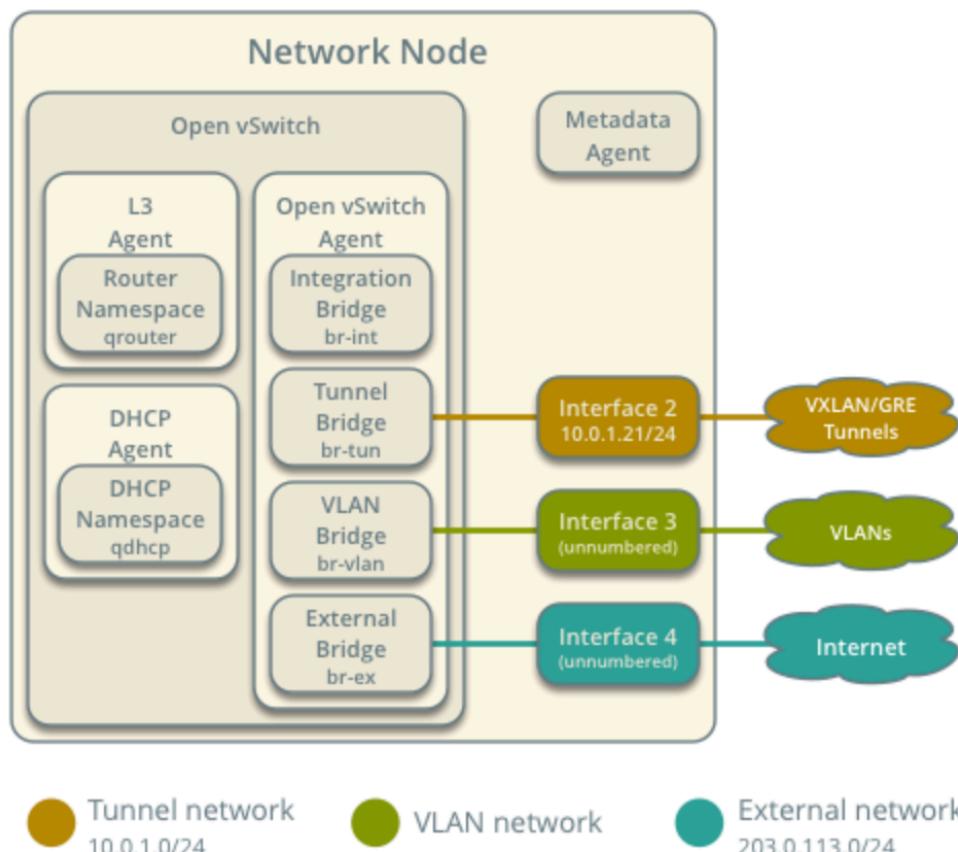
- Virtual Network 환경에서 VM을 위한 advanced switch 기능을 수행(L3+a)
- 리눅스의 브릿지 기능만으로는 네트워크 구성의 한계(network separation) 존재
- User-space(slow-path), Data-path(fast-path)
- Open-Flow 지원



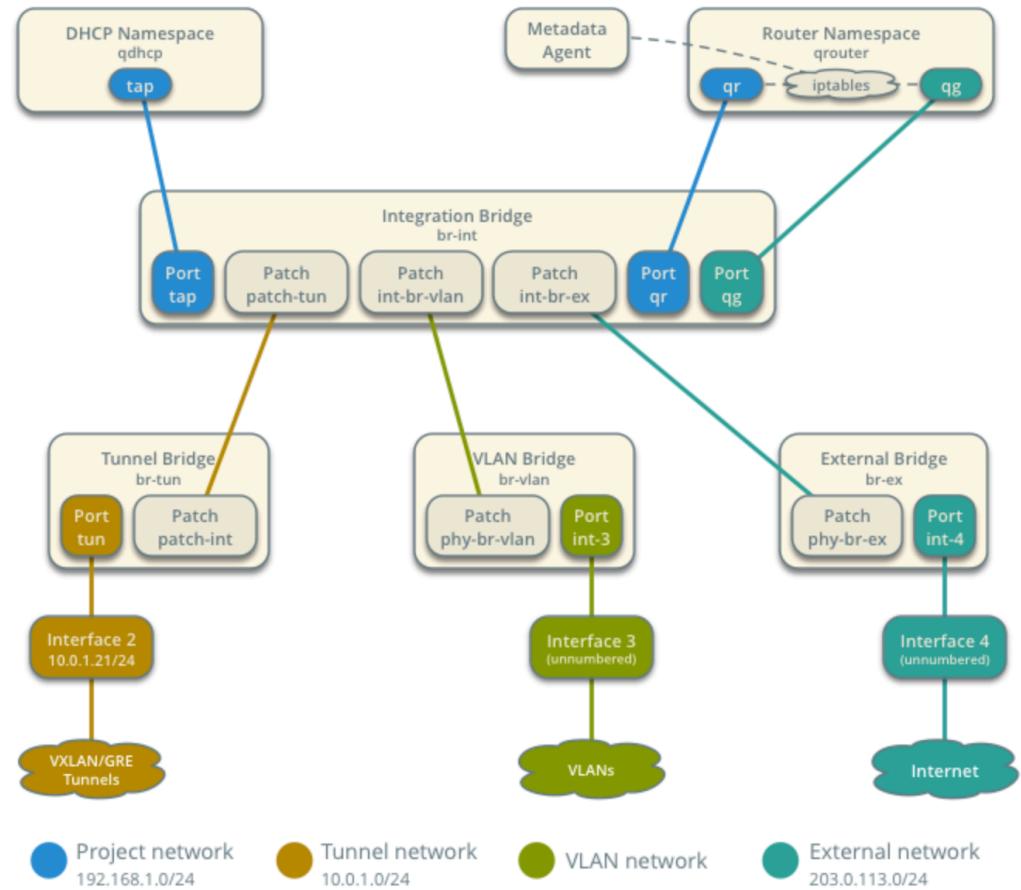
Open vSwitch

- 구조

Network Node Overview

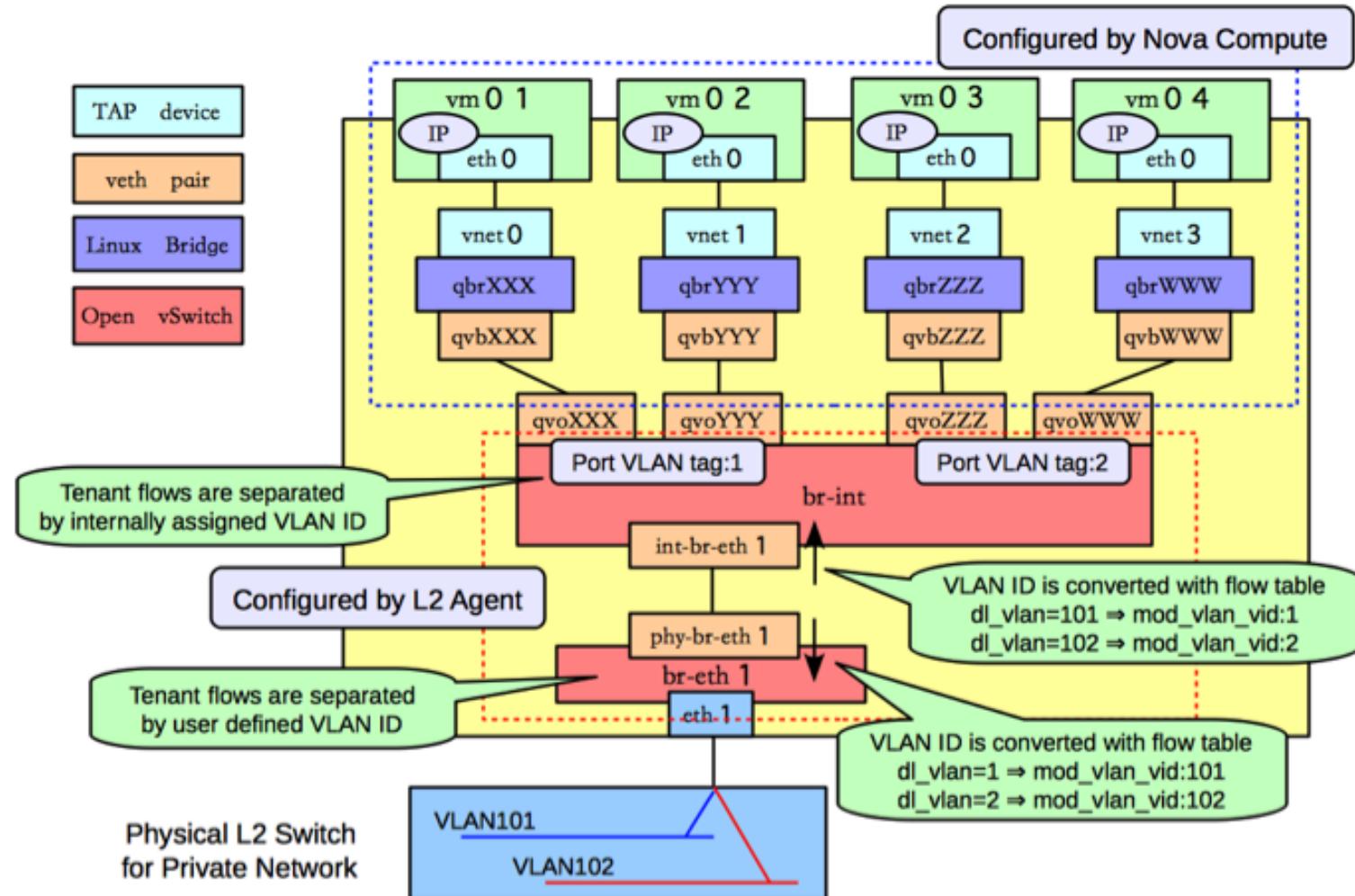


Network Node Components



Open vSwitch

- 구조(계속)



1. Tap device(Test Access Point Device)

- Guest OS와 Hypervisor 사이의 인터페이스 역할
- vnet0 은 KVM/Xen 내부의 VM에 NIC 와(VIF, vNIC) 연결됨

2. veth pair(Virtual ethernet pair)

- 2개 veth의 pair로 구성됨
- veth pair (qvbNNN, qvoNNN) 은 가상 네트워크와 연결되며, virtual bridge(Linux Bridge) 와 연결됨

3. Linux Bridge

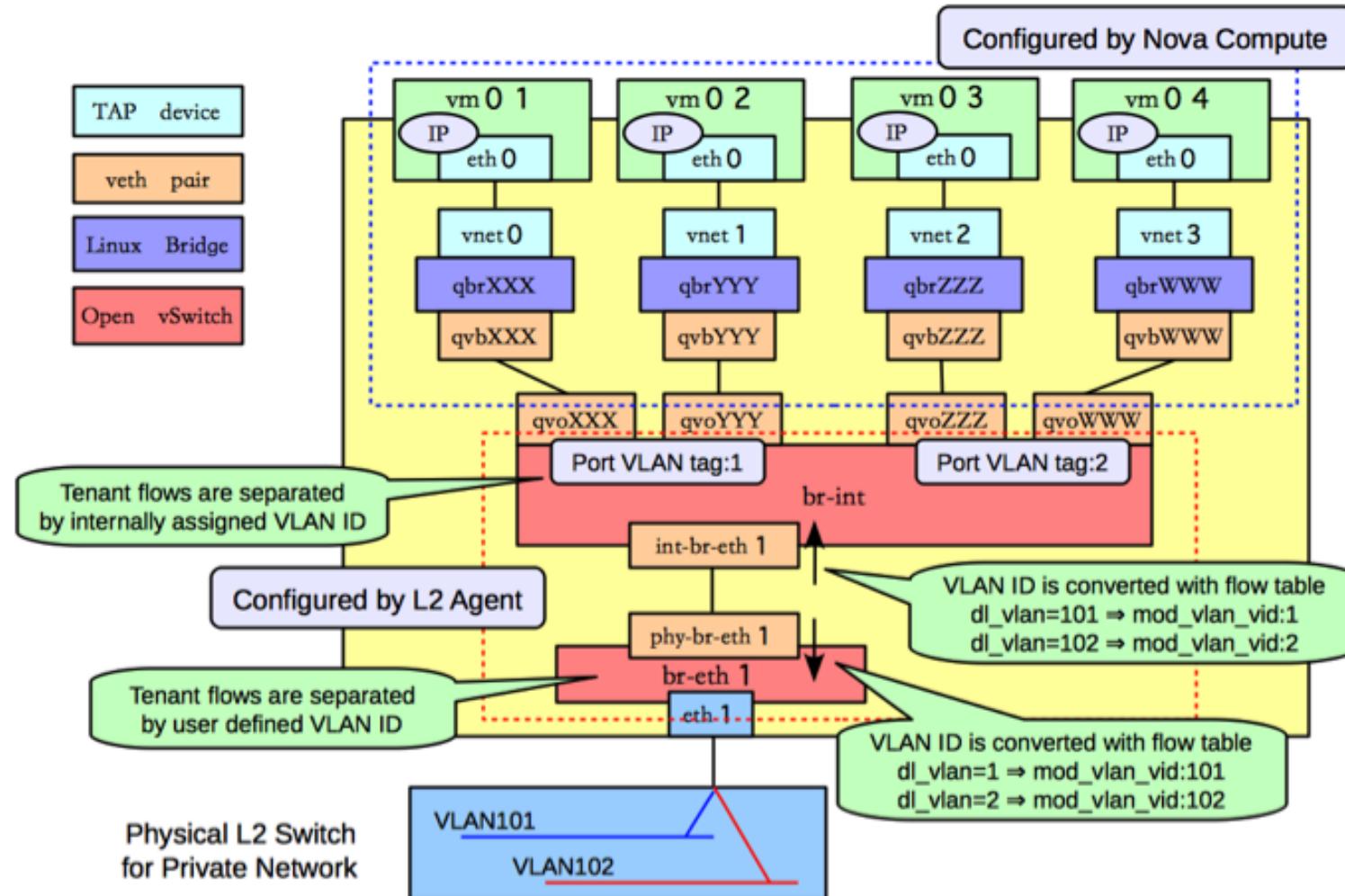
- 간단한 L2 스위칭 기능(MAC learning switch), 여러개 물리/가상 NIC가 연결됨
- 인터페이스는 2개 뿐인가? 나도 잘 모르겠음

4. Open vSwitch

- 가상 스위칭 기능 동작, 물리 포트 등 포트로 연결. GRE, VLAN 등 L2-L4의 일부 영역까지 제어

Open vSwitch

- 구조(계속)



5. 내부 bridge : br-int

- Bridge 종류 중 하나
- compute node에 모든 VM은 bridge 포트에 연결, 내부 vlan tag을 변환

6. 물리NIC 연결 bridge : br-eth1

- 물리 NIC와 연결, vlan ID

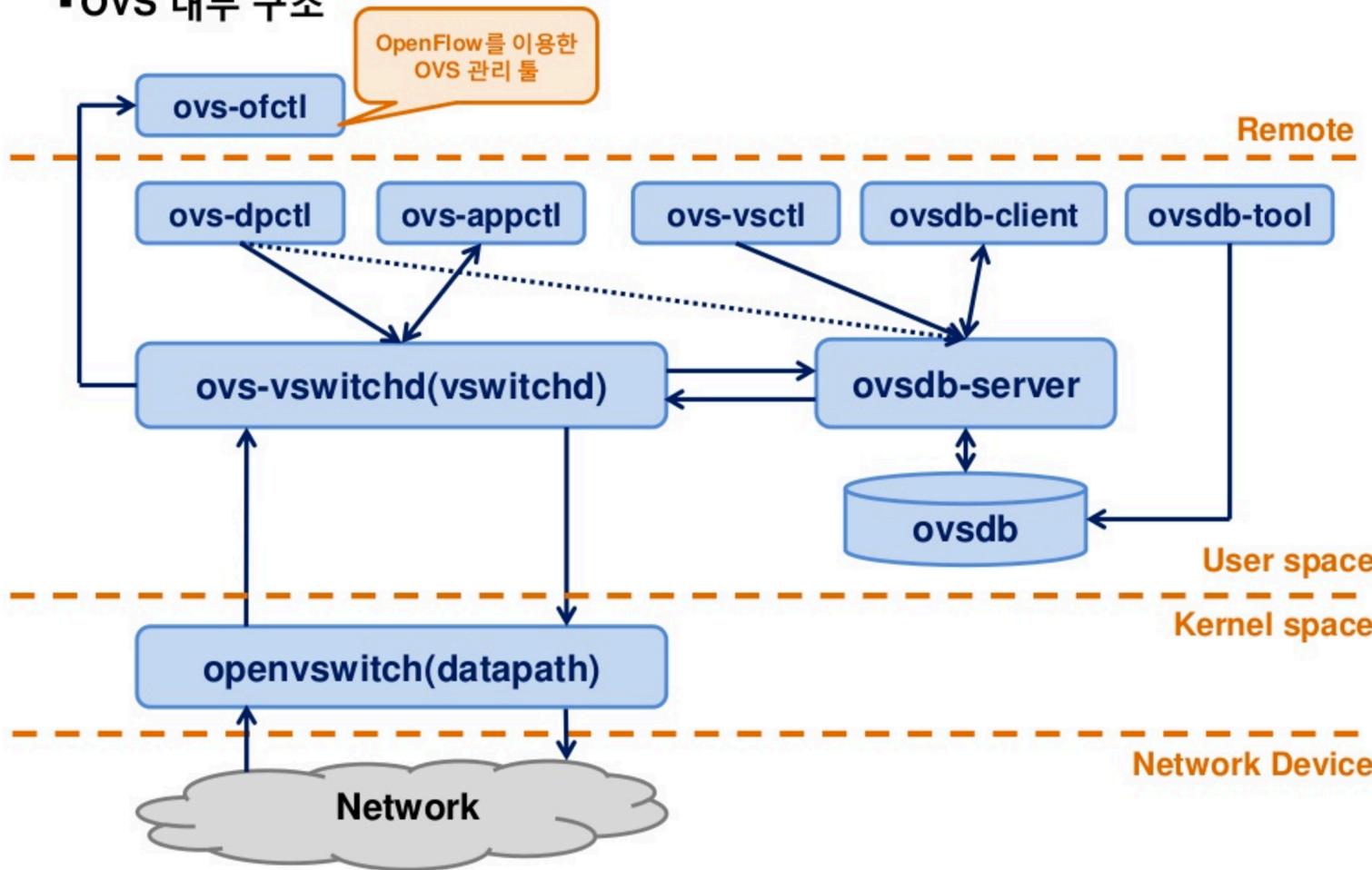
7. VLAN 변환

- br-int : 내부 => 외부 시 vlan ID 변경
- br-eth1 : 외부 => 내부 시 vlan ID 변경

Open vSwitch

- 역할별 구조

▪ OVS 내부 구조



ovsdb-server : ovs의 tunnel, bridge interface 설정을 저장하는 데이터 베이스

ovs-vswitchd : core component

- 패킷 처리에 대한 역할 수행
- ovsdb-server로 부터 packet path에 대한 정보 확보 및 table 생성
- bonding/mirroring 등등 이외의 스위치 기능 수행

ovs kernel module :

- 스위칭, 터널링 제어
- table이 있는 flow에 대한 offloading 수행(Cache)
- 패킷이 들어오면, 할당된 액션을 수행하고 카운팅.
- table이 없는 경우 User space로 전달

Open vSwitch

- 역할별 구조

- ovsdb Table

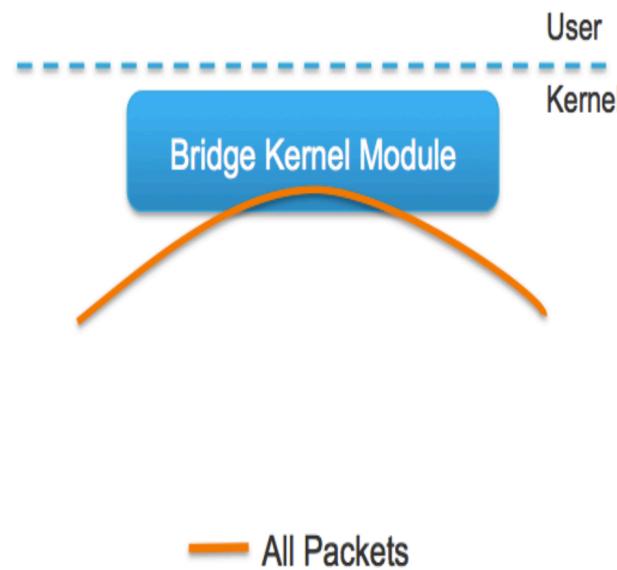
The screenshot shows the ovsdb-table interface for Open vSwitch. It displays three tables:

- Port** table (highlighted by a red circle): Shows interfaces (eth2, br0, eth1, eth3) and their corresponding Open vSwitch port configurations.
- Queue** table (highlighted by a red circle): Shows queue configurations for ports c4 and :6d. The configuration for port c4 is [[{"u'max-rate': u'50000000', "u'min-rate': u'50000000"}]] and for port :6d is [[{"u'max-rate': u'100000000', "u'min-rate': u'100000000"}]].
- QoS** table (highlighted by a red circle): Shows Quality of Service configurations. One entry shows an **other_config** field with [[{"u'max-rate': u'2000000000"}]] and a **queues** field with [[0, [{"u'uuid': u'855b5559-fb96-4f43-a21b-405b8e01134b'}], [1, {"u'uuid': u'69b4e125-9614-4ad6-8b87-40f227db5d42"}]]].

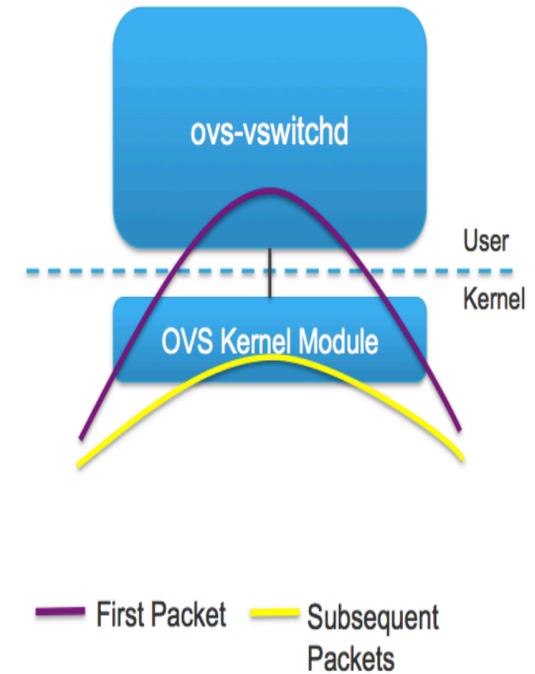
Open vSwitch

- 역할별 구조

- Simple forwarding
- Matches destination MAC address and forwards
- Packet never leaves kernel



- Decision about how to process packet made in userspace
- First packet of new flow goes to ovs-vswitchd, following packets hit cached entry in kernel



Open vSwitch

- 기능

Features

Open vSwitch supports the following features:

- Visibility into inter-VM communication via NetFlow, sFlow(R), IPFIX, SPAN, RSPAN, and GRE-tunneled mirrors
- LACP (IEEE 802.1AX-2008)
- Standard 802.1Q VLAN model with trunking
- Multicast snooping
- IETF Auto-Attach SPBM and rudimentary required LLDP support
- BFD and 802.1ag link monitoring
- STP (IEEE 802.1D-1998) and RSTP (IEEE 802.1D-2004)
- Fine-grained QoS control
- Support for HFSC qdisc
- Per VM interface traffic policing
- NIC bonding with source-MAC load balancing, active backup, and L4 hashing
- OpenFlow protocol support (including many extensions for virtualization)
- IPv6 support
- Multiple tunneling protocols (GRE, VXLAN, STT, and Geneve, with IPsec support)
- Remote configuration protocol with C and Python bindings
- Kernel and user-space forwarding engine options
- Multi-table forwarding pipeline with flow-caching engine
- Forwarding layer abstraction to ease porting to new software and hardware platforms

bridge

By Linux Foundatio... - November 19, 2009 - 6:23pm

[NETWORKING](#)

Translations: [russian](#) , [Turkish](#)

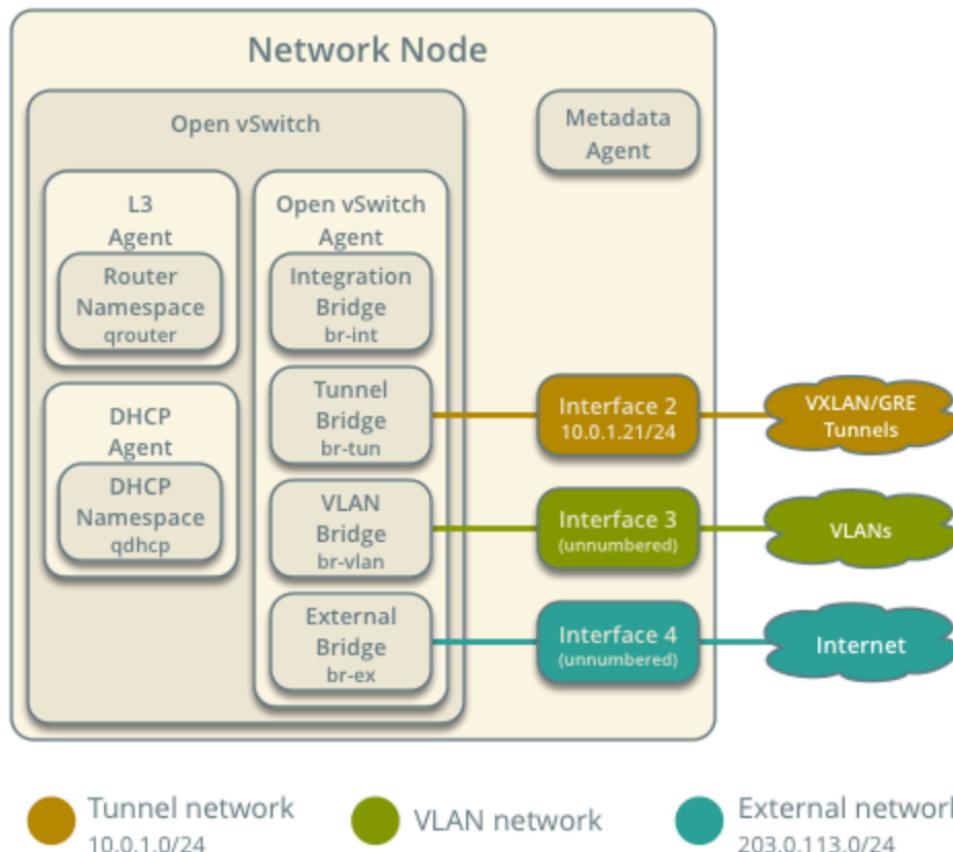
A bridge is a way to connect two [Ethernet](#) segments together in a protocol independent way. Packets are forwarded based on Ethernet address, rather than IP address (like a router). Since forwarding is done at Layer 2, all protocols can go transparently through a bridge.

The Linux bridge code implements a subset of the ANSI/IEEE 802.1d standard. [\[1\]](#). The original Linux bridging was first done in Linux 2.2, then rewritten by Lennert Buytenhek. The code for bridging has been integrated into 2.4 and 2.6 kernel series.

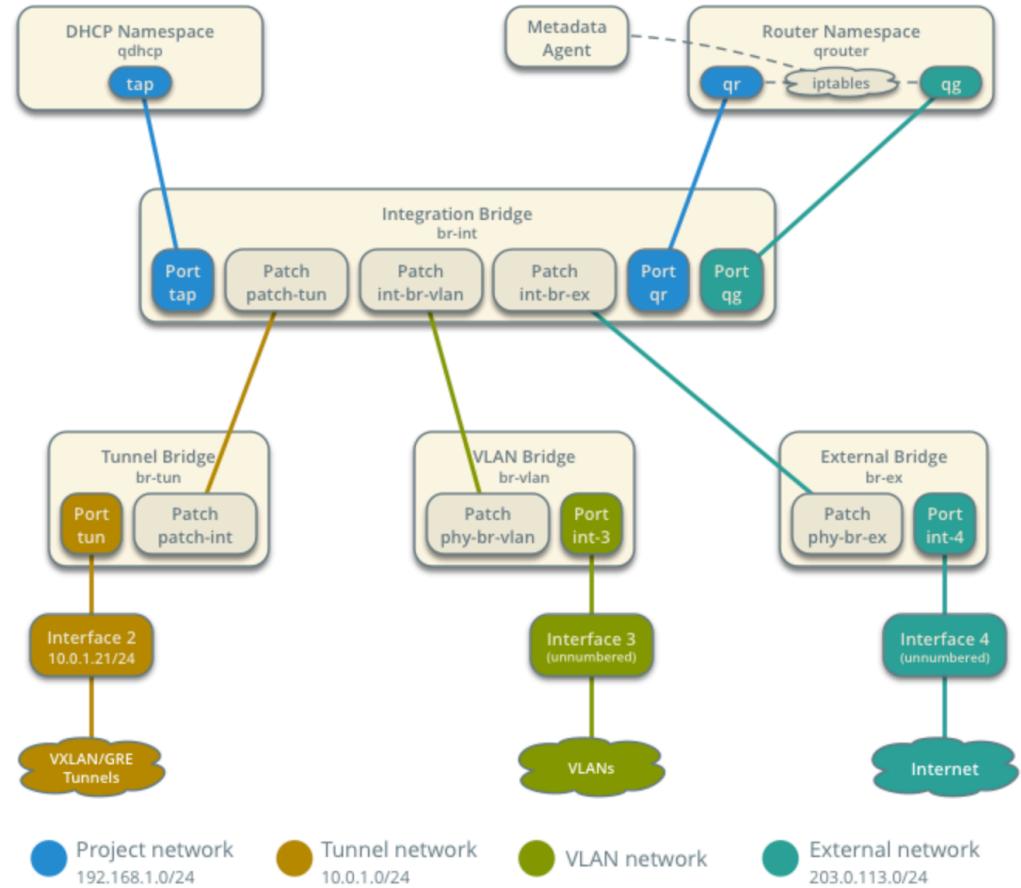
Open vSwitch

- 시나리오 : Scenario: Legacy with Open vSwitch

Network Node Overview



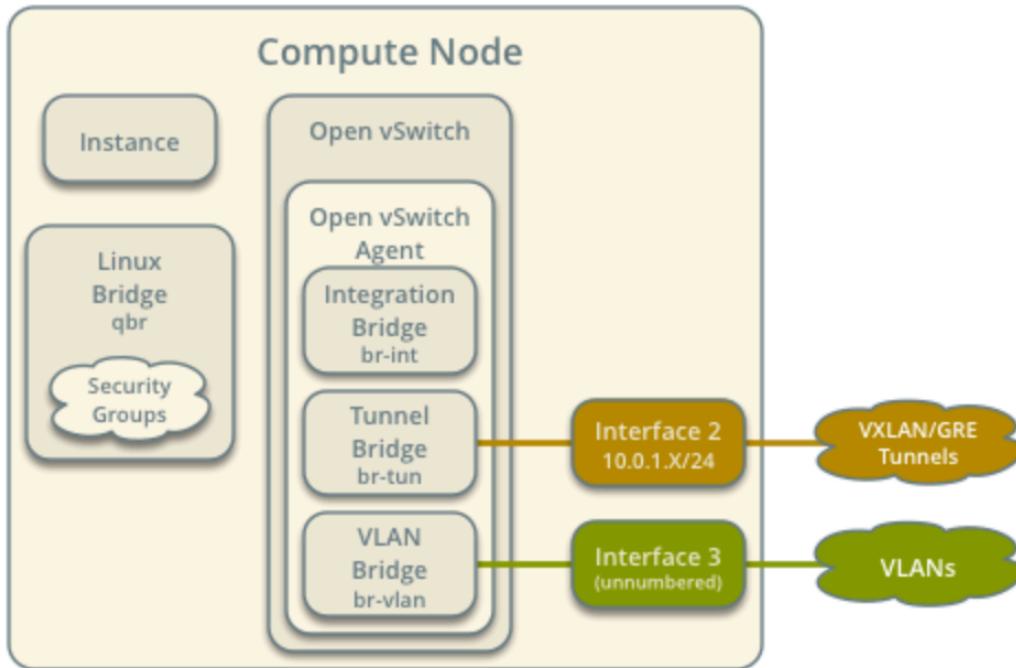
Network Node Components



Open vSwitch

- 시나리오 : Scenario: Legacy with Open vSwitch

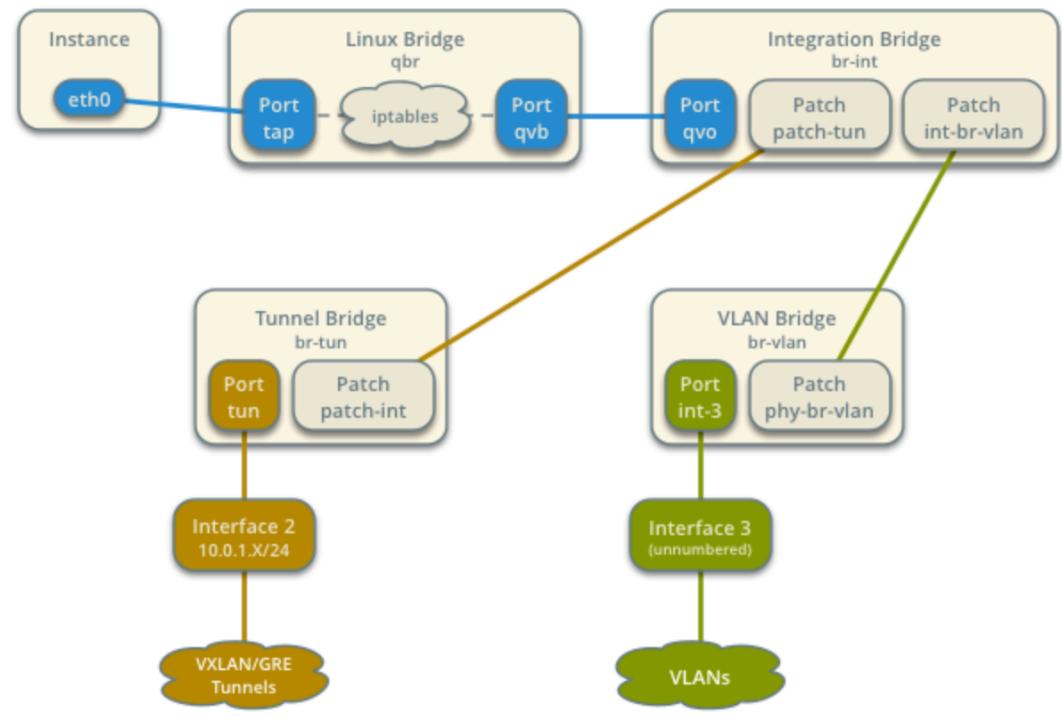
Compute Node Overview



Tunnel network
10.0.1.0/24

VLAN network

Compute Node Components



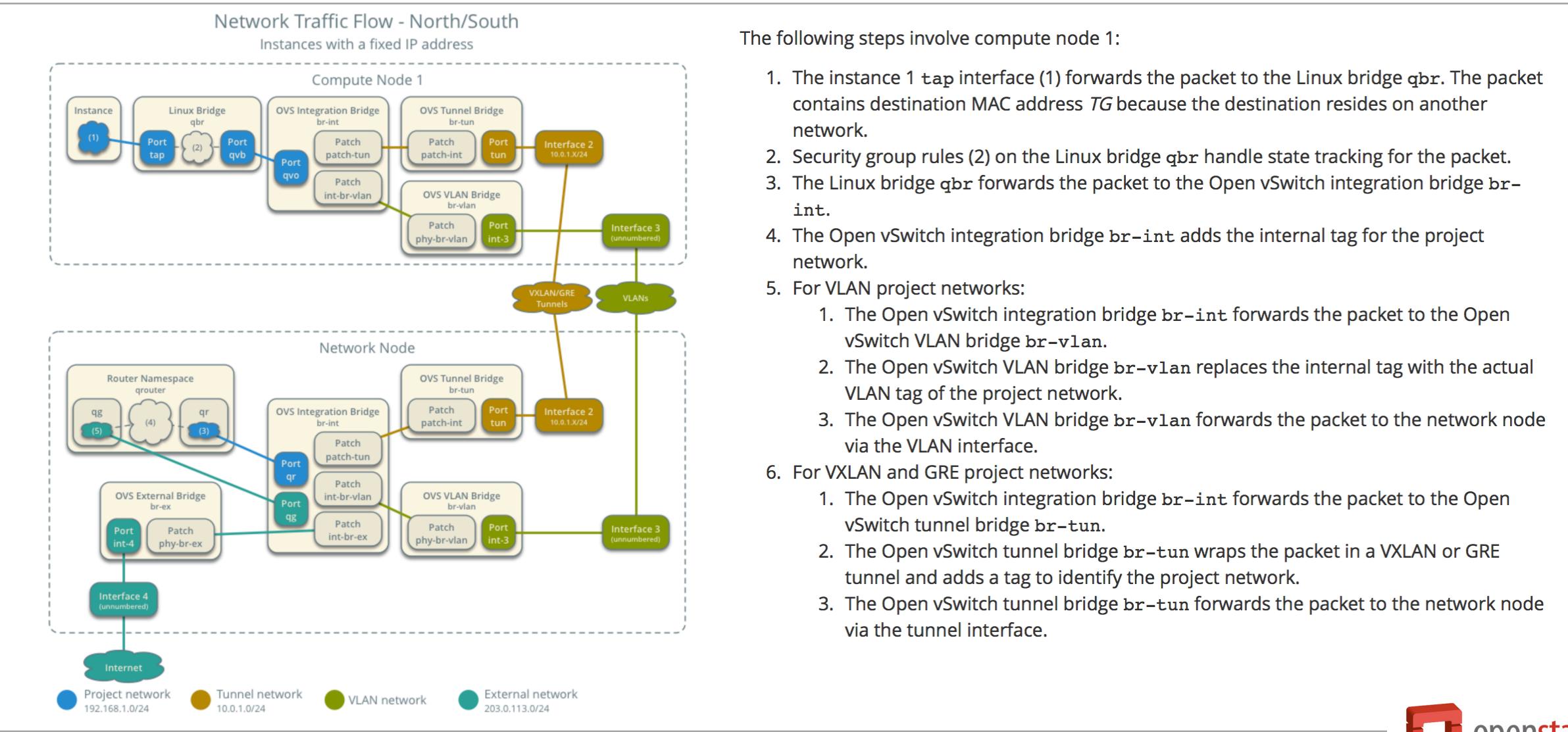
Project network
192.168.1.0/24

Tunnel network
10.0.1.0/24

VLAN network

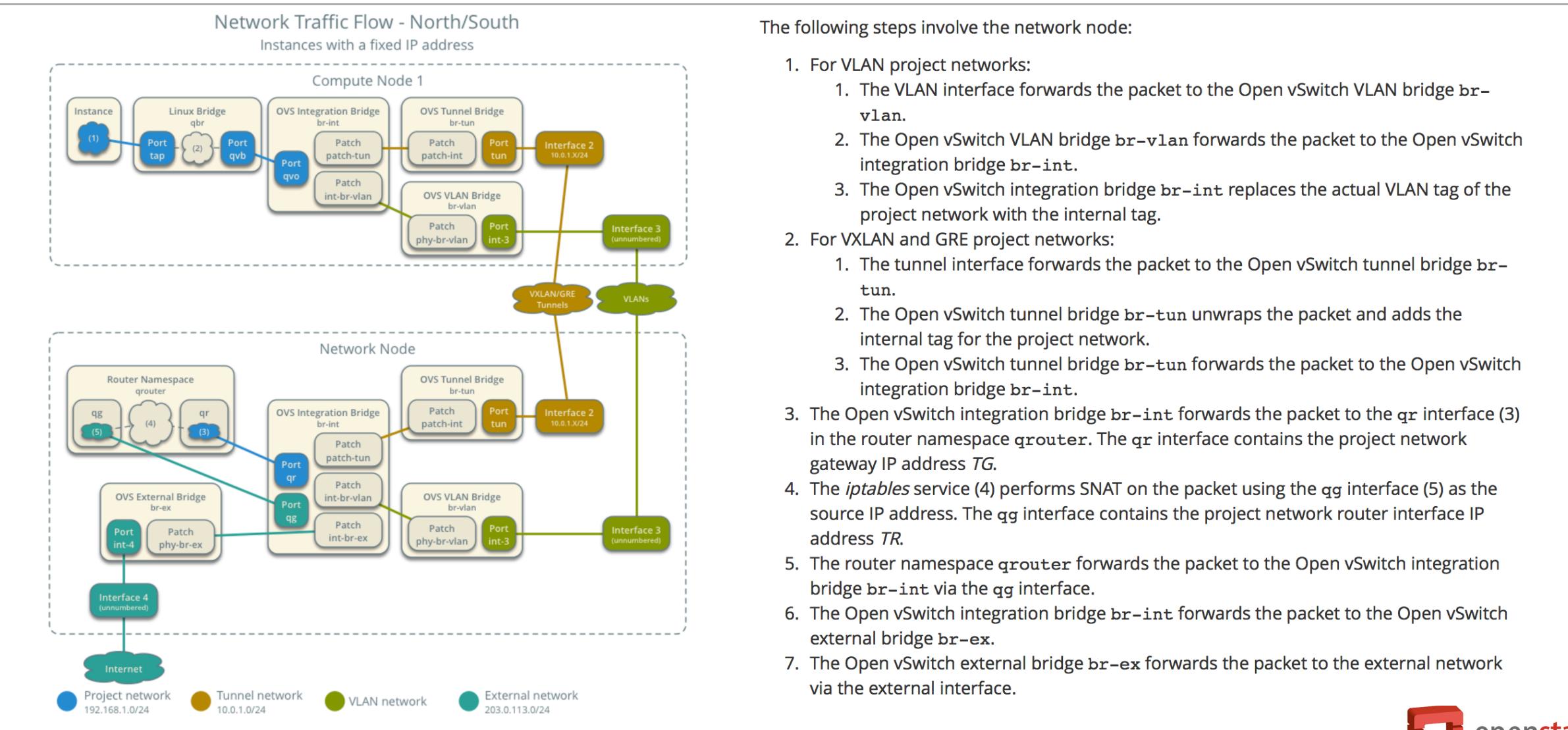
Open vSwitch

- 시나리오 : Scenario: Legacy with Open vSwitch



Open vSwitch

- 시나리오 : Scenario: Legacy with Open vSwitch



Questions?