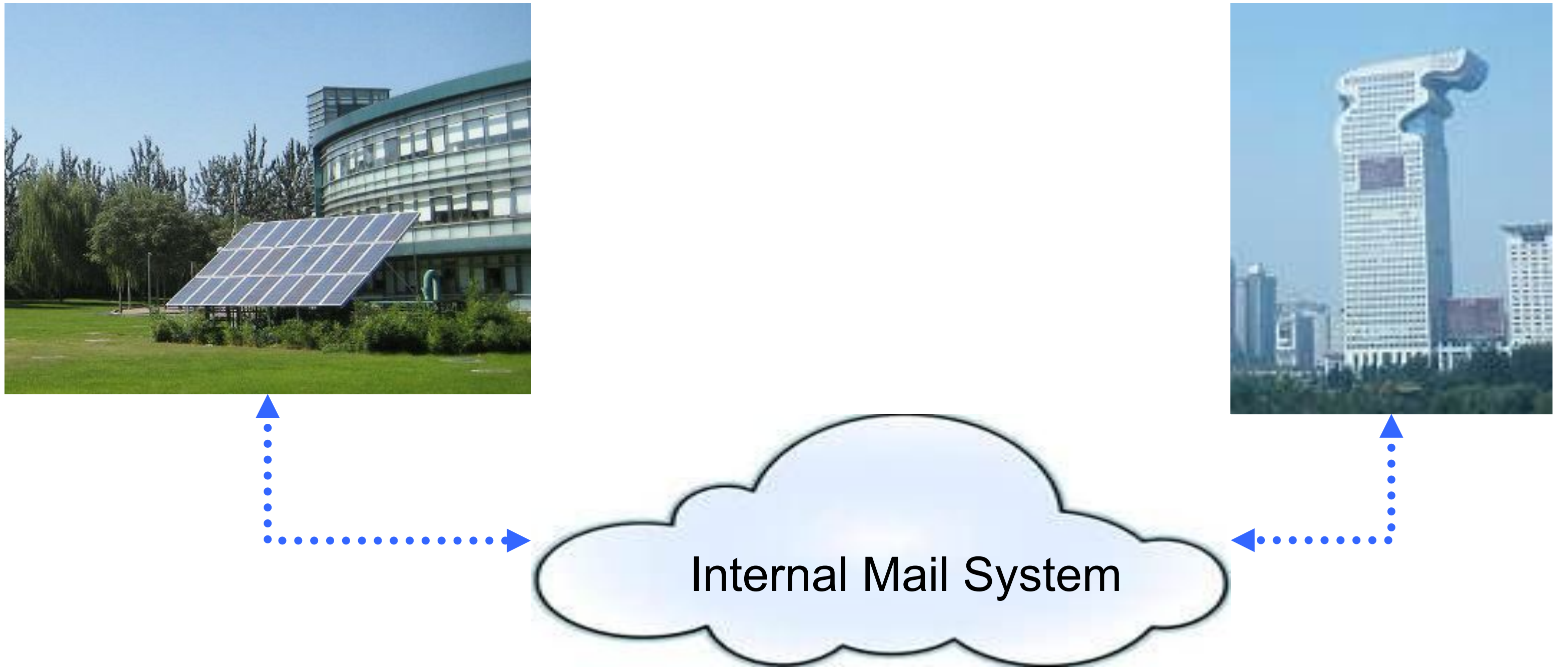
OpenStack Neutron
A system perspective

Baohua Yang
2014-04-11

Network as a Service



Internal Mail System

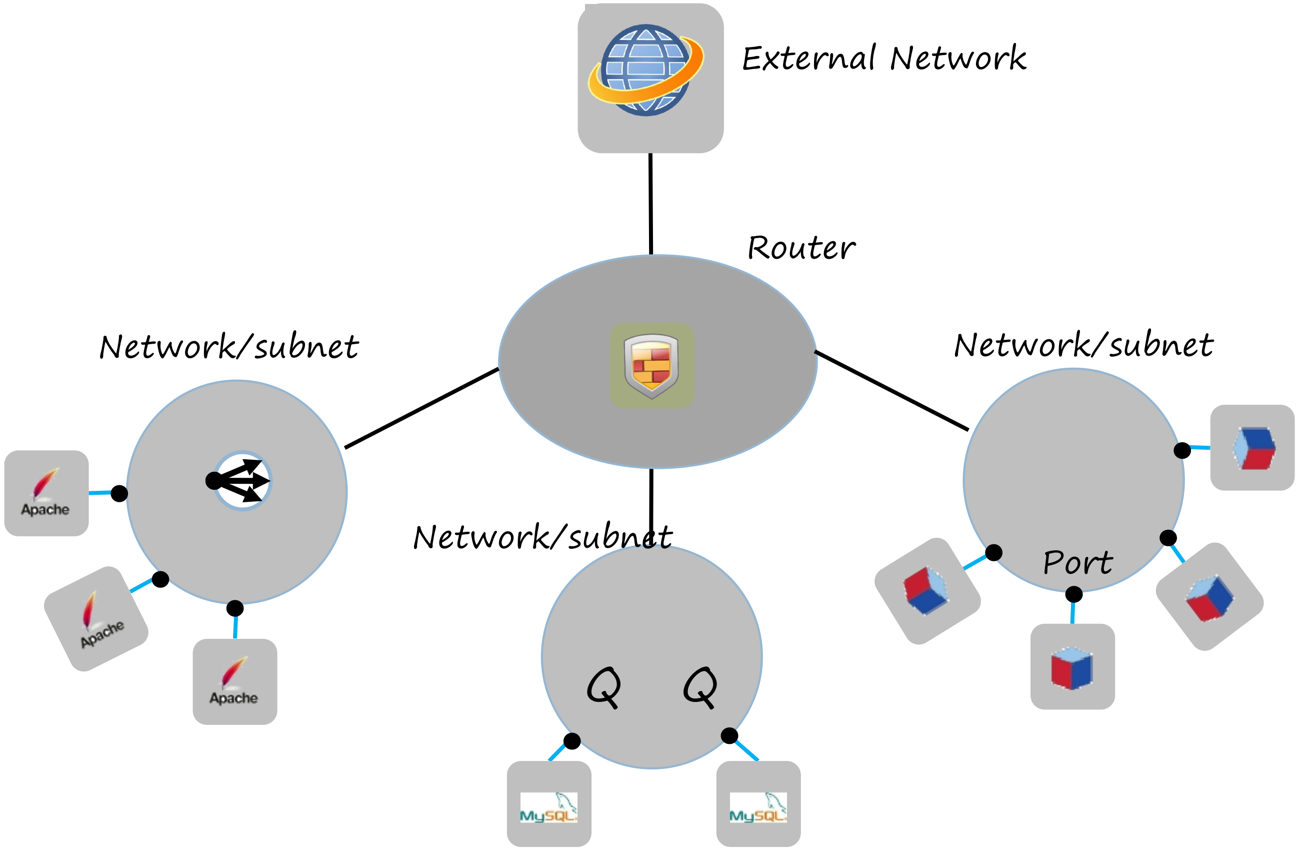
Neutron: Basic Concepts

□ Minimal set of interfaces required for setting up networks for users

Network	isolated layer-2 broadcast domain; *private/shared*
Subnet	CIDR IP address block associated with a `network`; optionally associated gateway, DNS/DHCP servers
Port	virtual switch port on a `network`; has MAC and IP address properties

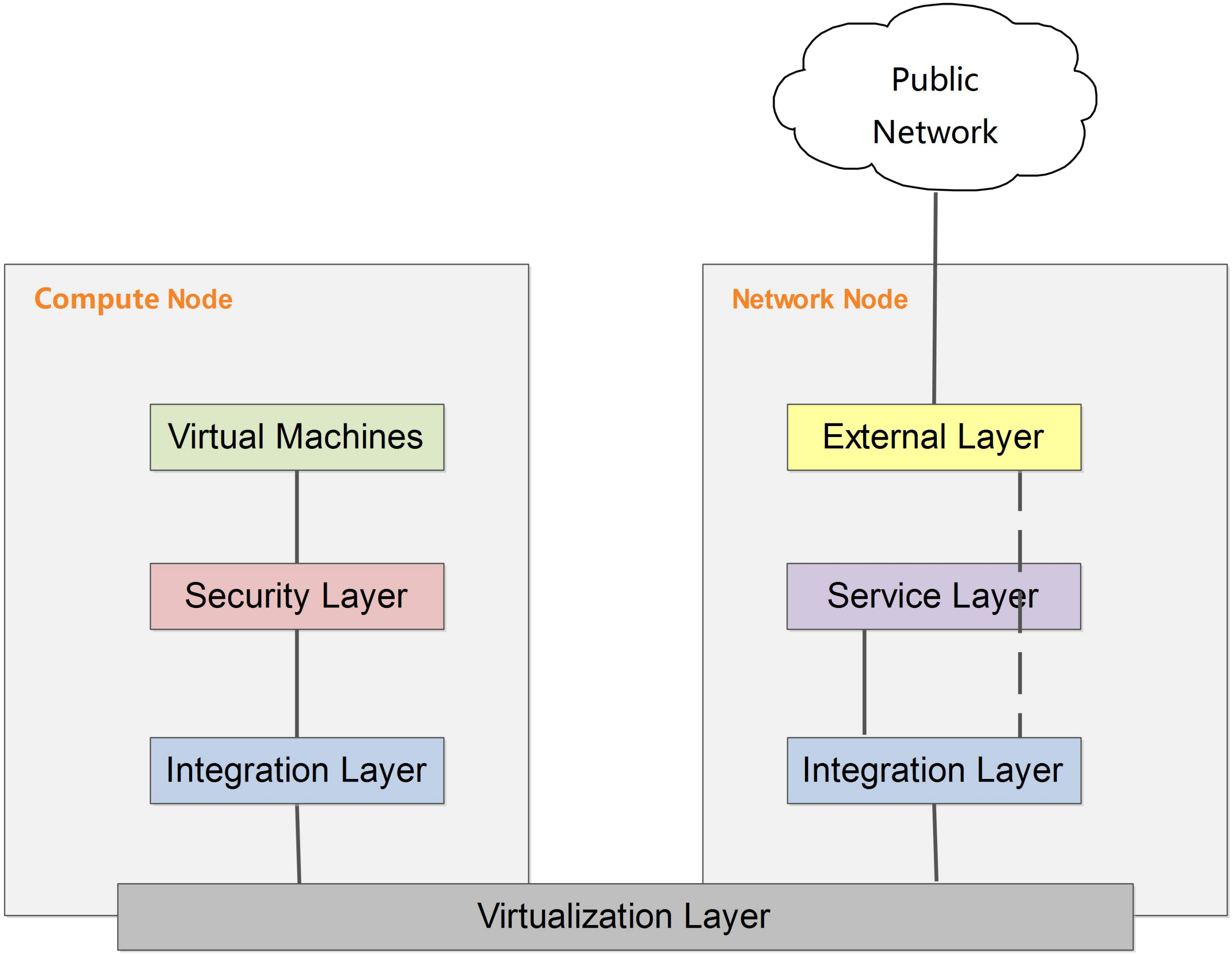
A 3-tier App Example

- One possible implementation using a single router



Architecture overview

Big Picture



Public Network

Compute Node

Virtual Machines

Security Layer

Integration Layer

Network Node

External Layer

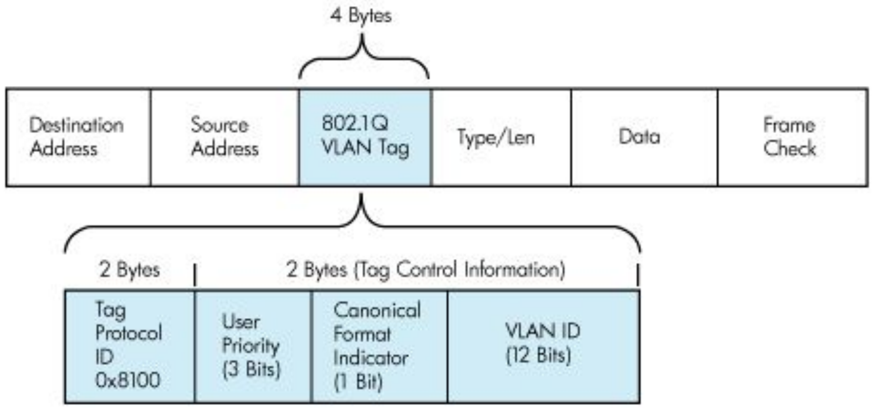
Service Layer

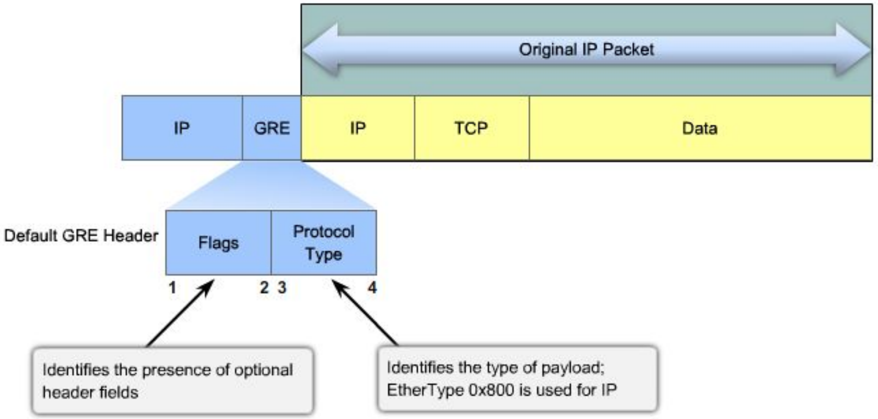
Integration Layer

Virtualization Layer

Prior Knowledge

- Vlan
  - 802.1Q
  - TPID : 16bit – 0x8100
  - TCI : 16bit
    - PCP : 3bit
    - DEI : 1bit
    - VID : 12bit(0 ~ 4095)
- GRE
  - 16 bytes header
  - IP header

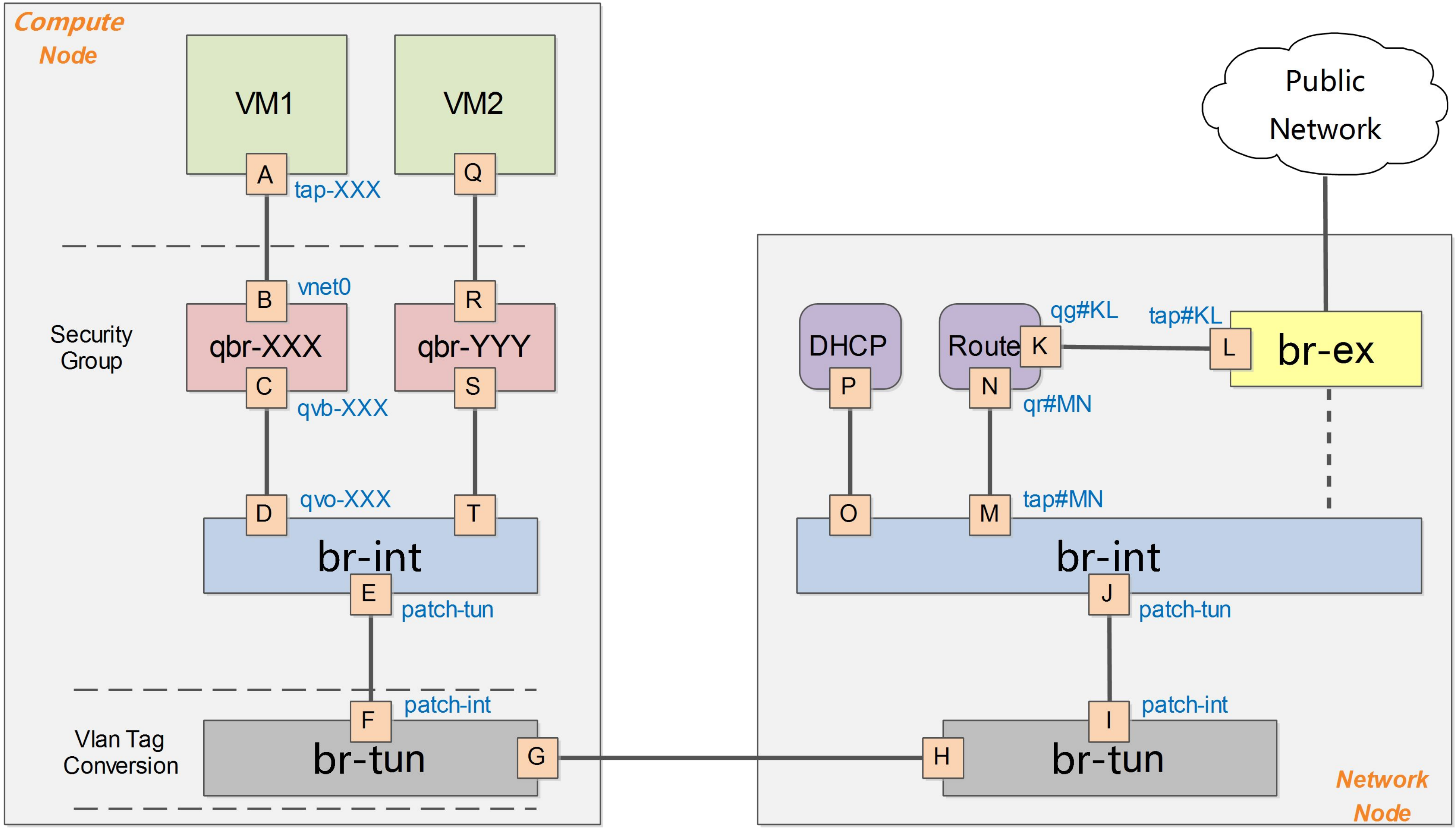




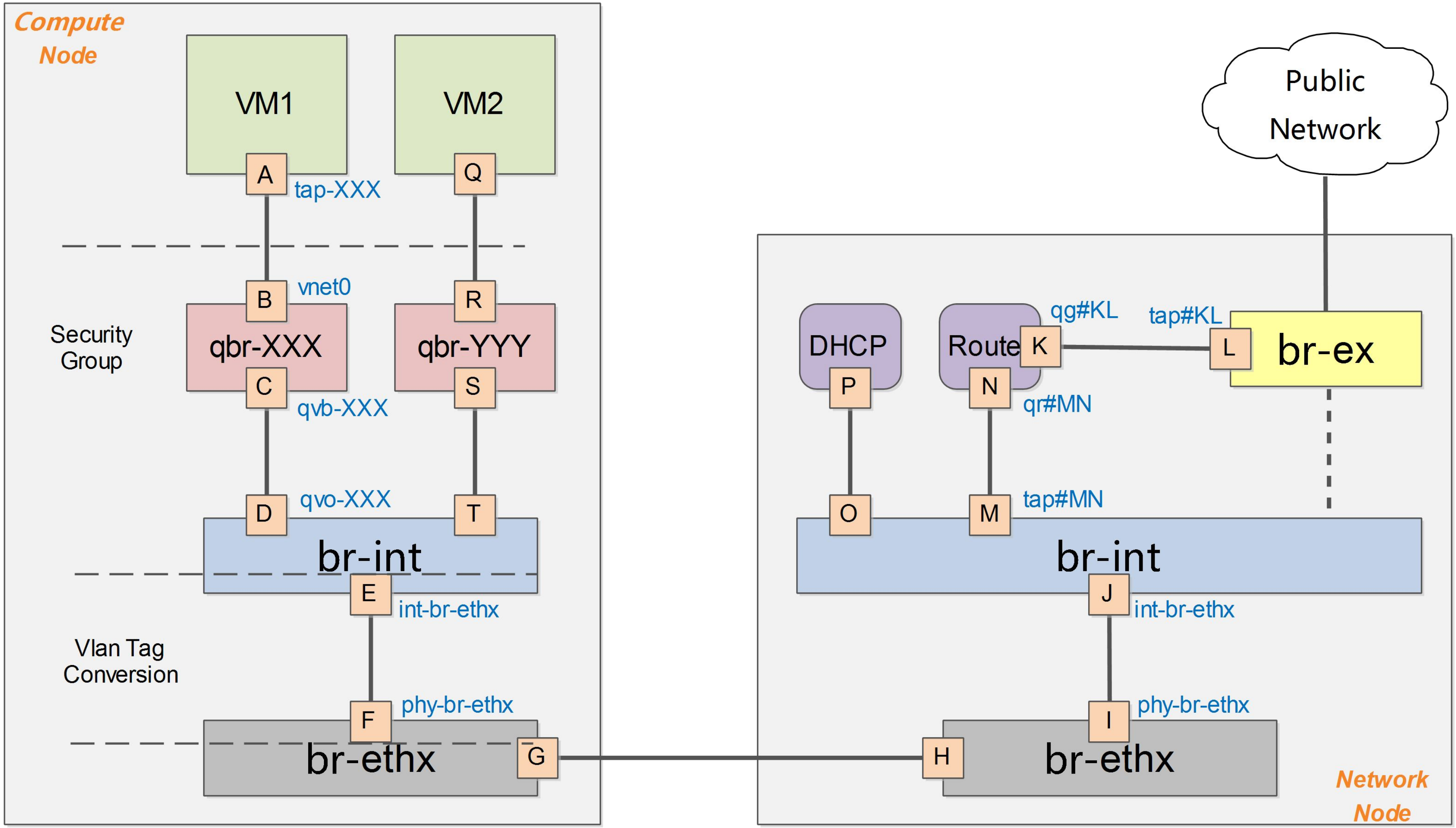
Identifies the presence of optional header fields

Identifies the type of payload; EtherType 0x800 is used for IP

GRE Mode

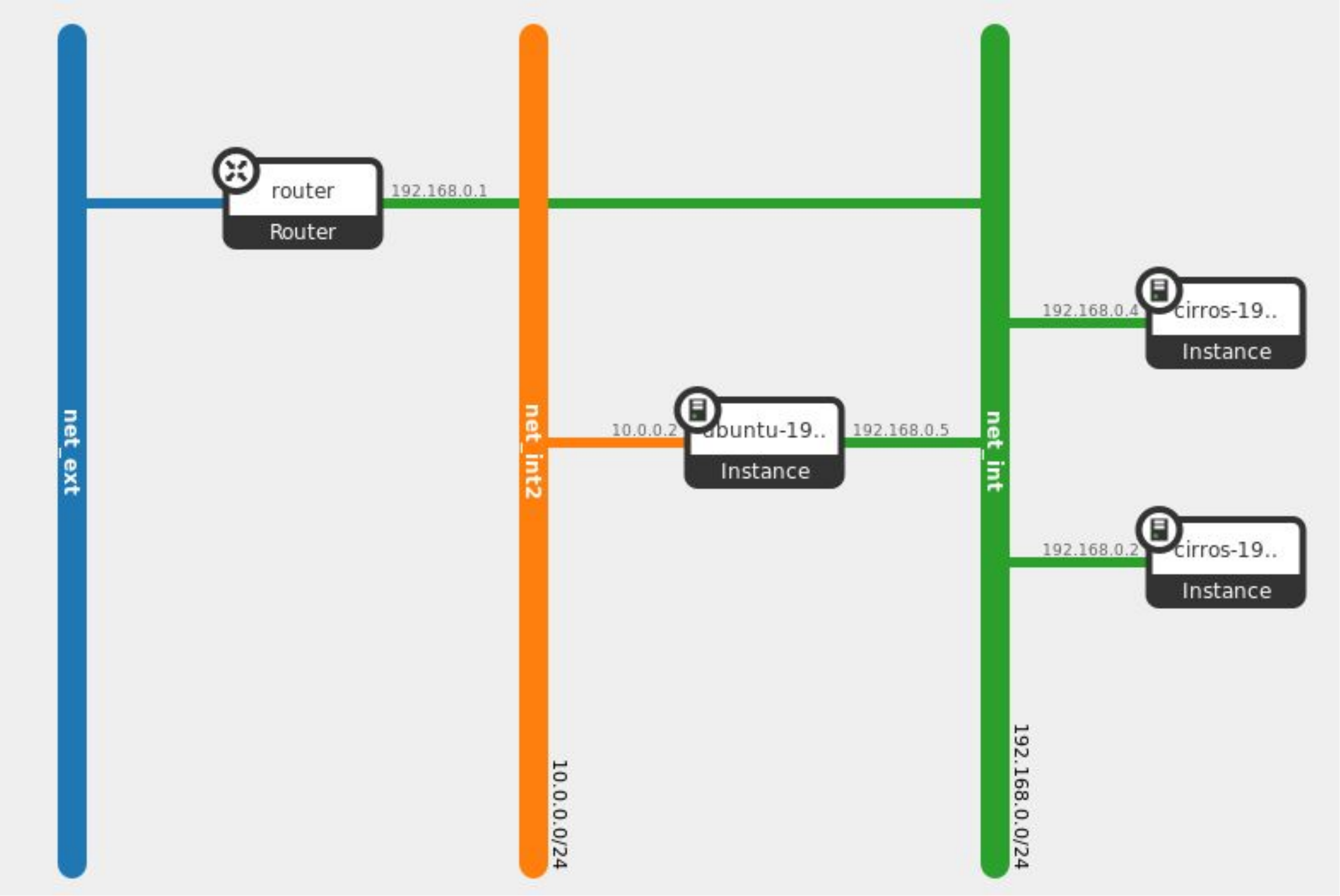

Compute Node
VM1
VM2
A
tap-XXX
Q
B
vnet0
R
Security Group
qbr-XXX
qbr-YYY
C
qvb-XXX
S
D
qvo-XXX
T
br-int
E
patch-tun
F
patch-int
Vlan Tag Conversion
br-tun
G
Public Network
DHCP
Route
K
qg#KL
tap#KL
L
br-ex
P
N
qr#MN
O
M
tap#MN
br-int
J
patch-tun
I
patch-int
H
br-tun
Network Node

Vlan Mode

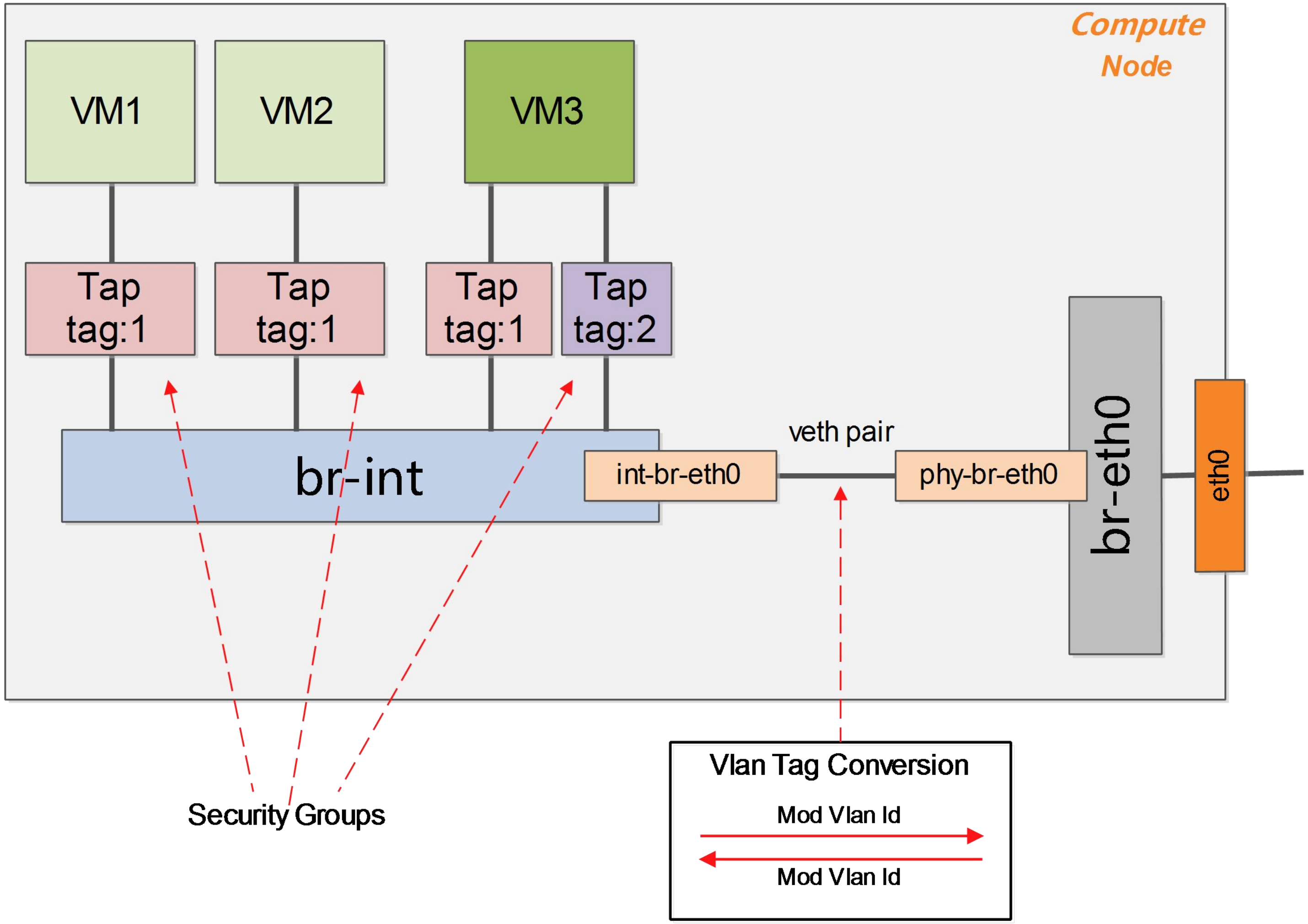

Compute Node
VM1
VM2
A
Q
tap-XXX
B
R
vnet0
Security Group
qbr-XXX
qbr-YYY
C
S
qvb-XXX
D
T
qvo-XXX
br-int
E
int-br-ethx
Vlan Tag Conversion
F
phy-br-ethx
br-ethx
G
Public Network
DHCP
Route
K
qg#KL
tap#KL
L
br-ex
P
N
qr#MN
O
M
tap#MN
br-int
J
int-br-ethx
I
phy-br-ethx
H
br-ethx
Network Node

Walkthrough of Vlan Mode

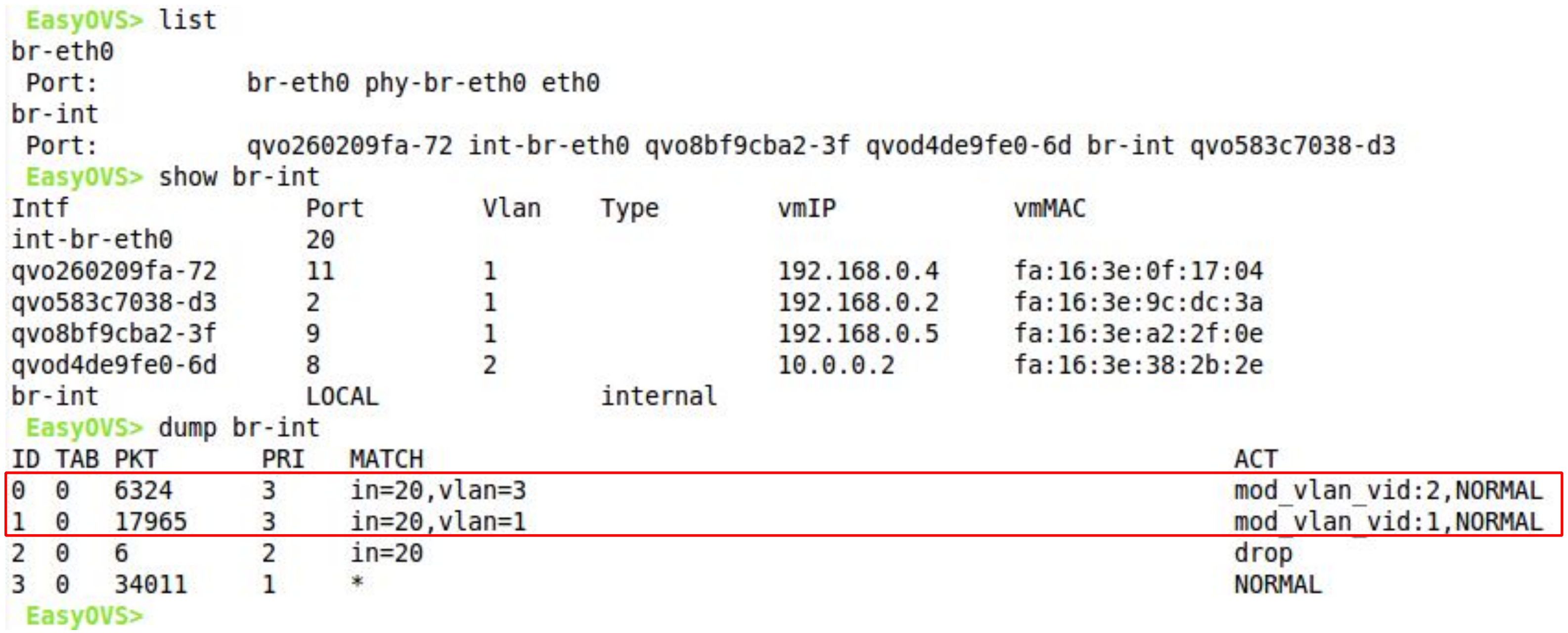
Topology


net_ext
router
Router
192.168.0.1
net_int2
10.0.0.2
ubuntu-19..
Instance
192.168.0.5
10.0.0.0/24
net_int
192.168.0.4
cirros-19..
Instance
192.168.0.2
cirros-19..
Instance
192.168.0.0/24

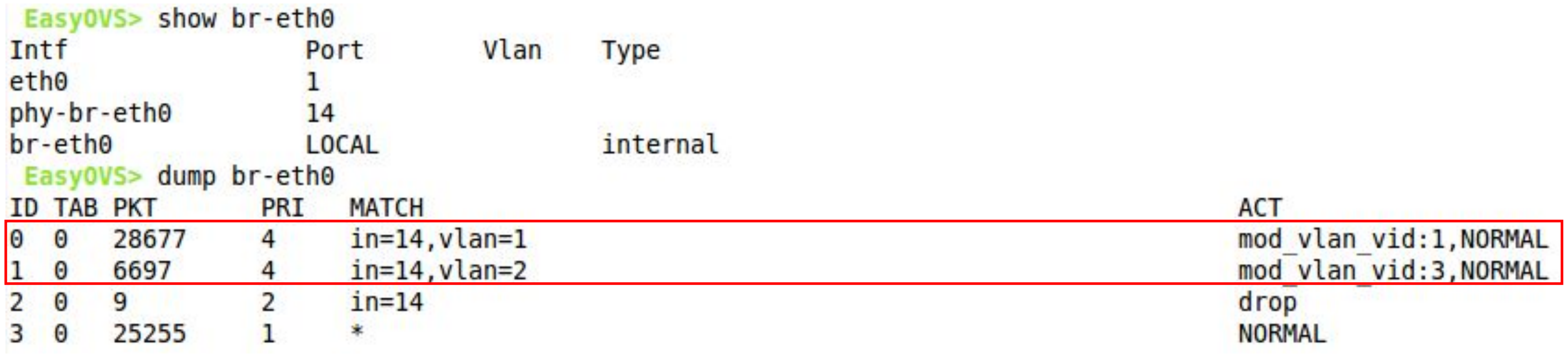
Compute Node



Compute Node

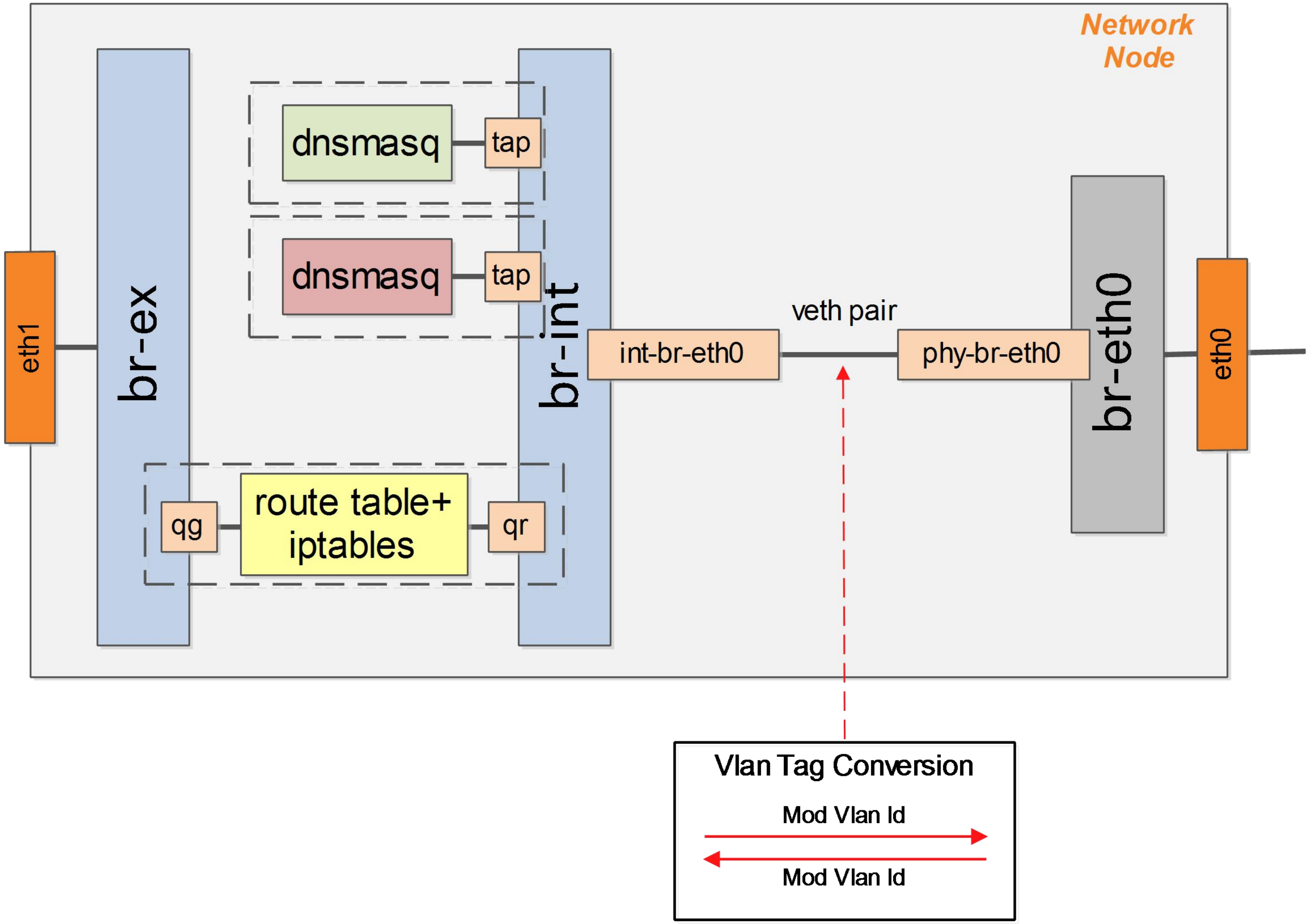


```
 EasyOVS> list
br-eth0
 Port:          br-eth0 phy-br-eth0 eth0
br-int
 Port:          qvo260209fa-72 int-br-eth0 qvo8bf9cba2-3f qvod4de9fe0-6d br-int qvo583c7038-d3
 EasyOVS> show br-int
Intf                Port        Vlan    Type        vmIP            vmMAC
int-br-eth0         20
qvo260209fa-72      11          1                   192.168.0.4     fa:16:3e:0f:17:04
qvo583c7038-d3      2           1                   192.168.0.2     fa:16:3e:9c:dc:3a
qvo8bf9cba2-3f      9           1                   192.168.0.5     fa:16:3e:a2:2f:0e
qvod4de9fe0-6d      8           2                   10.0.0.2        fa:16:3e:38:2b:2e
br-int              LOCAL               internal
 EasyOVS> dump br-int
ID TAB PKT       PRI   MATCH                                                    ACT
0  0   6324      3     in=20,vlan=3                                             mod_vlan_vid:2,NORMAL
1  0   17965     3     in=20,vlan=1                                             mod_vlan_vid:1,NORMAL
2  0   6         2     in=20                                                    drop
3  0   34011     1     *                                                        NORMAL
 EasyOVS>
```

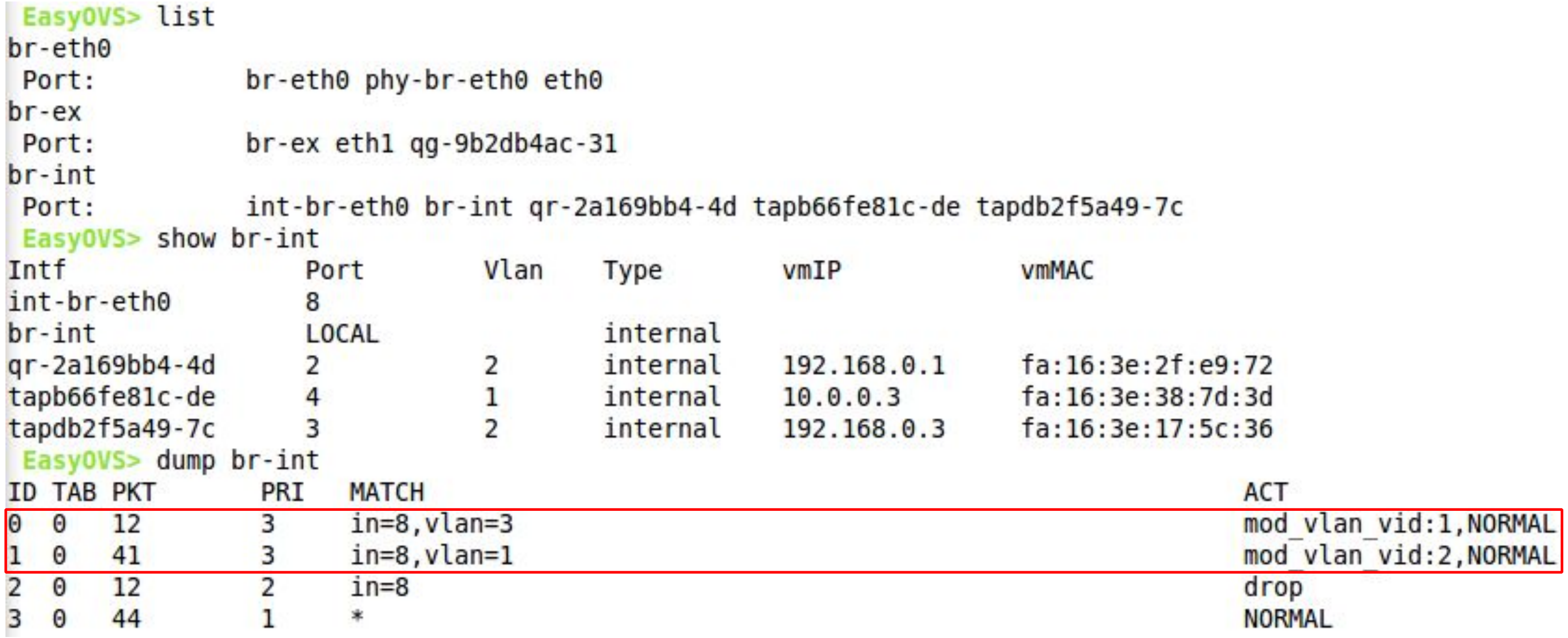


```
 EasyOVS> show br-eth0
Intf                Port        Vlan    Type
eth0                1
phy-br-eth0         14
br-eth0             LOCAL               internal
 EasyOVS> dump br-eth0
ID TAB PKT       PRI   MATCH                                                    ACT
0  0   28677     4     in=14,vlan=1                                             mod_vlan_vid:1,NORMAL
1  0   6697      4     in=14,vlan=2                                             mod_vlan_vid:3,NORMAL
2  0   9         2     in=14                                                    drop
3  0   25255     1     *                                                        NORMAL
```

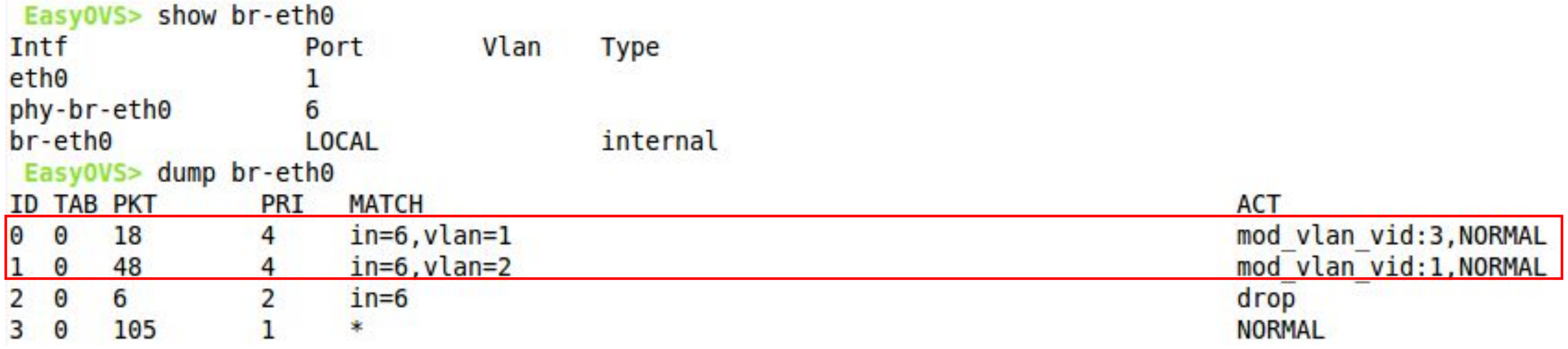
Network Node



Network Node



```
 EasyOVS> list
br-eth0
 Port:          br-eth0 phy-br-eth0 eth0
br-ex
 Port:          br-ex eth1 qg-9b2db4ac-31
br-int
 Port:          int-br-eth0 br-int qr-2a169bb4-4d tapb66fe81c-de tapdb2f5a49-7c
 EasyOVS> show br-int
Intf                Port        Vlan    Type        vmIP            vmMAC
int-br-eth0         8
br-int              LOCAL               internal
qr-2a169bb4-4d      2           2       internal    192.168.0.1     fa:16:3e:2f:e9:72
tapb66fe81c-de      4           1       internal    10.0.0.3        fa:16:3e:38:7d:3d
tapdb2f5a49-7c      3           2       internal    192.168.0.3     fa:16:3e:17:5c:36
 EasyOVS> dump br-int
ID TAB PKT       PRI   MATCH                                                      ACT
0  0   12        3     in=8,vlan=3                                                mod_vlan_vid:1,NORMAL
1  0   41        3     in=8,vlan=1                                                mod_vlan_vid:2,NORMAL
2  0   12        2     in=8                                                       drop
3  0   44        1     *                                                          NORMAL
```



```
 EasyOVS> show br-eth0
Intf                Port        Vlan    Type
eth0                1
phy-br-eth0         6
br-eth0             LOCAL               internal
 EasyOVS> dump br-eth0
ID TAB PKT       PRI   MATCH                                                      ACT
0  0   18        4     in=6,vlan=1                                                mod_vlan_vid:3,NORMAL
1  0   48        4     in=6,vlan=2                                                mod_vlan_vid:1,NORMAL
2  0   6         2     in=6                                                       drop
3  0   105       1     *                                                          NORMAL
```

Advanced Topics

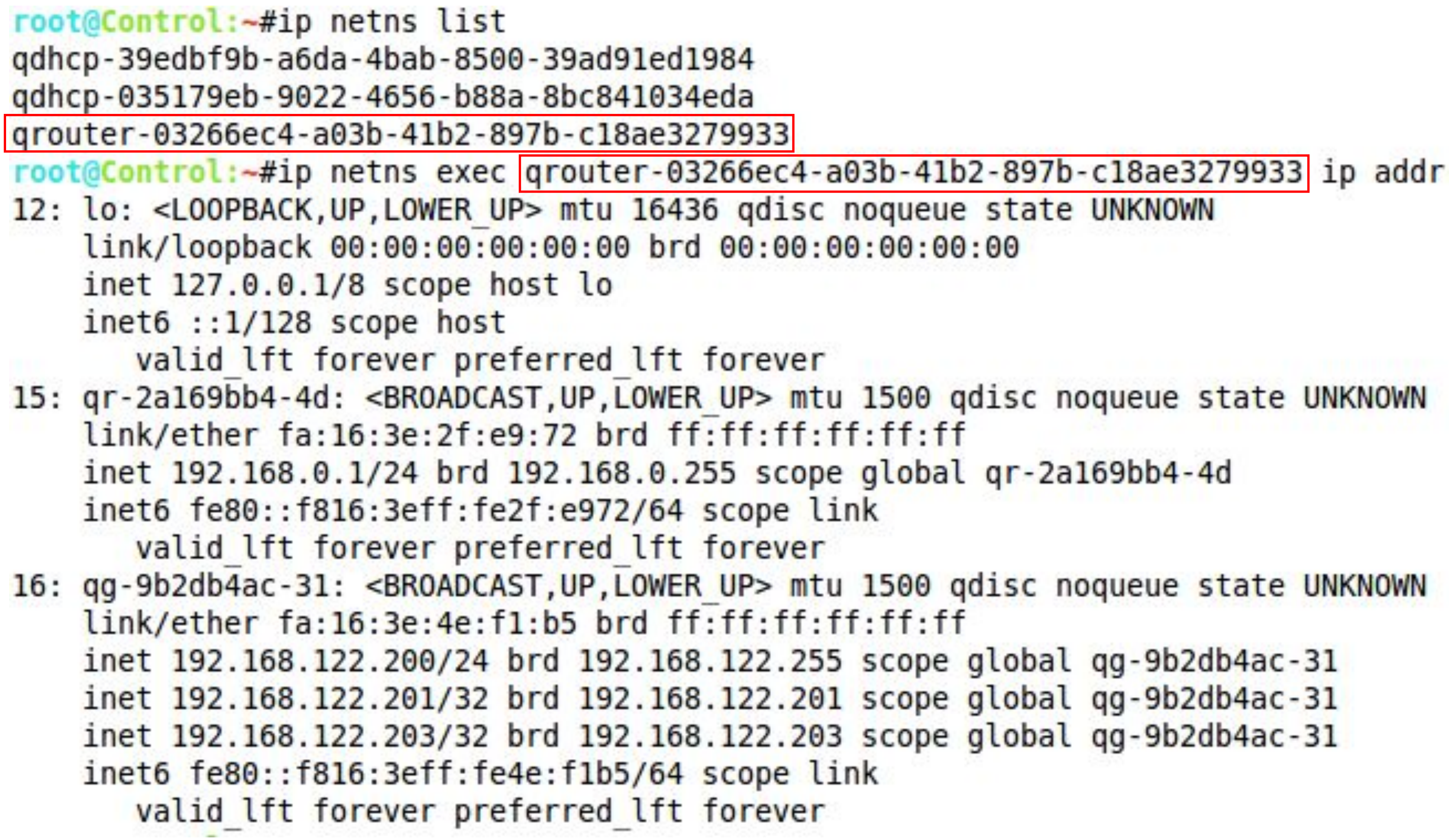
- Network Namespace
- Floating IP
- Security Group
- VXLAN
- ML2
- Multihost

Network Namespace

- □ Network namespace isolates the network interface controllers (physical or virtual), iptables firewall rules, routing tables etc.
- □ Network namespaces can be connected with each other using the "veth" virtual Ethernet device.

```
ip netns list
ip netns add new_ns
ip link add veth0 type veth peer name veth1
ip link set veth1 netns new_ns
ip netns exec new_ns <commands>
```

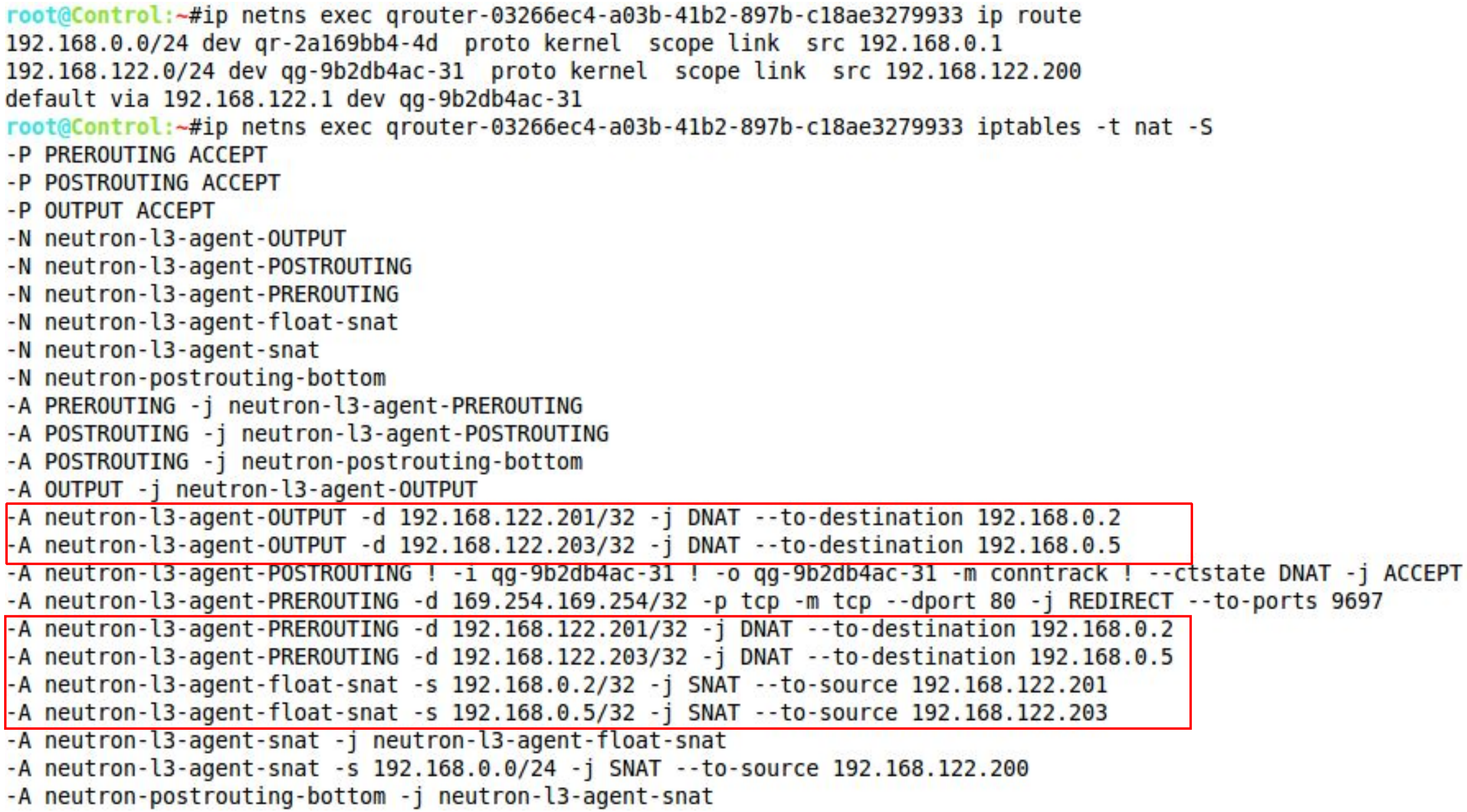
Network Namespace



```
root@Control:~#ip netns list
qdhcp-39edbf9b-a6da-4bab-8500-39ad91ed1984
qdhcp-035179eb-9022-4656-b88a-8bc841034eda
qrouter-03266ec4-a03b-41b2-897b-c18ae3279933
root@Control:~#ip netns exec qrouter-03266ec4-a03b-41b2-897b-c18ae3279933 ip addr
12: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
15: qr-2a169bb4-4d: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:2f:e9:72 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global qr-2a169bb4-4d
    inet6 fe80::f816:3eff:fe2f:e972/64 scope link
       valid_lft forever preferred_lft forever
16: qg-9b2db4ac-31: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
    link/ether fa:16:3e:4e:f1:b5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.200/24 brd 192.168.122.255 scope global qg-9b2db4ac-31
    inet 192.168.122.201/32 brd 192.168.122.201 scope global qg-9b2db4ac-31
    inet 192.168.122.203/32 brd 192.168.122.203 scope global qg-9b2db4ac-31
    inet6 fe80::f816:3eff:fe4e:f1b5/64 scope link
       valid_lft forever preferred_lft forever
```

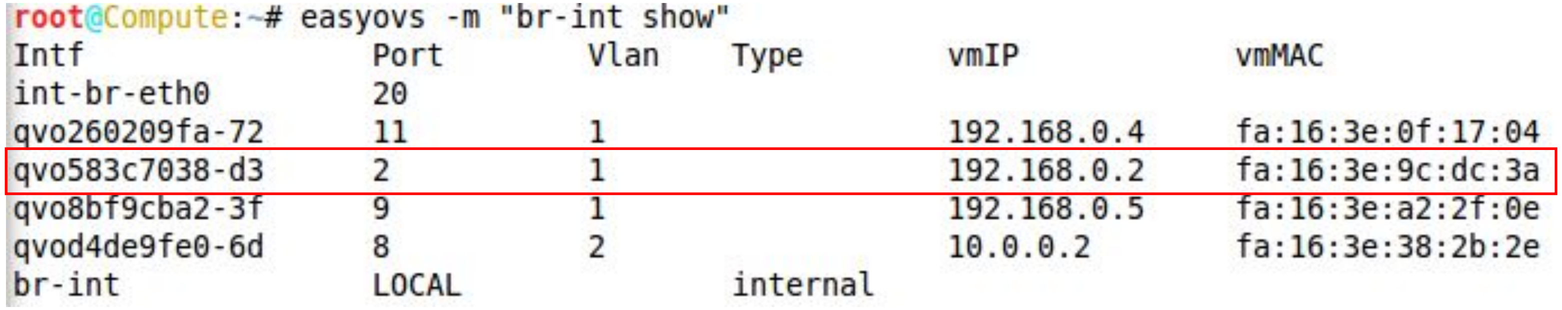
Floating IP

- □ Route table + NAT

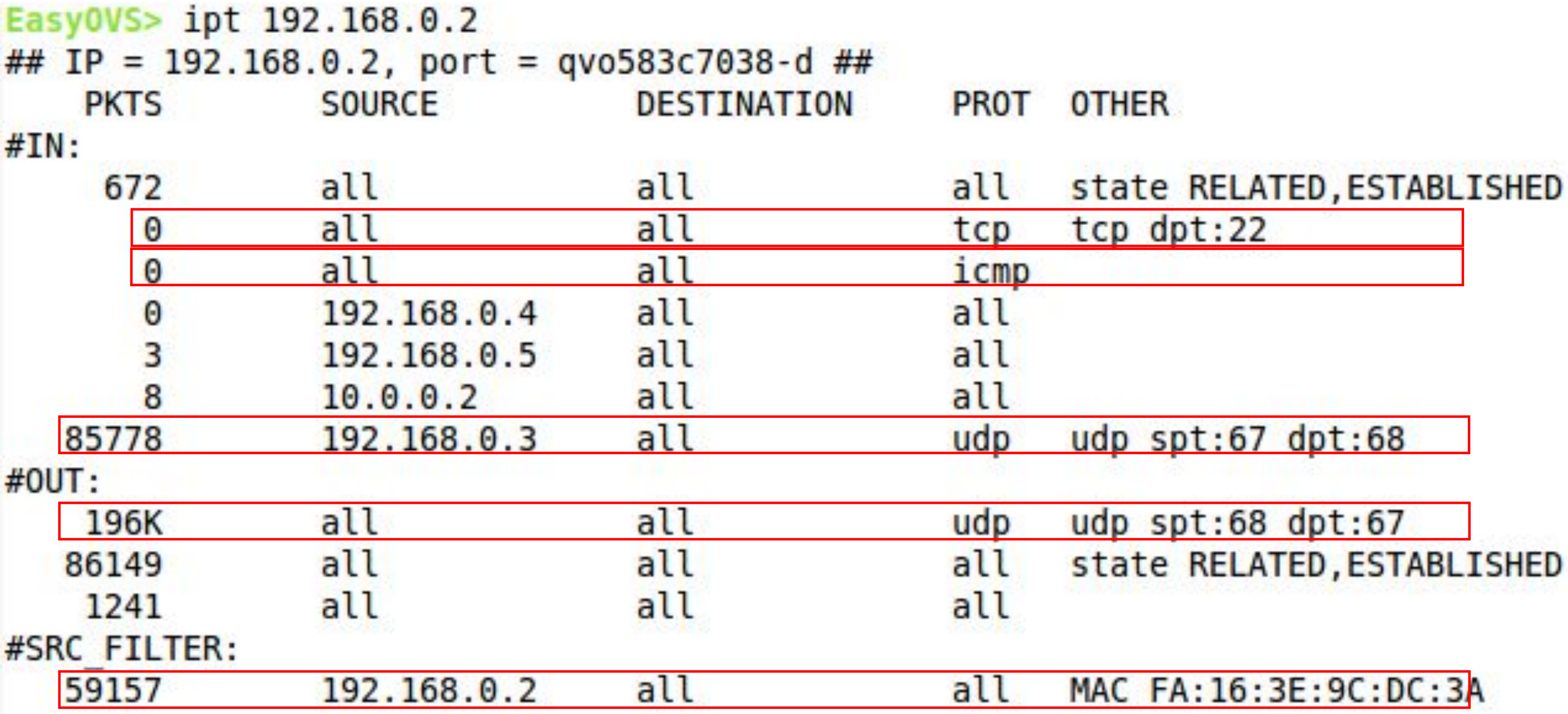


```
root@Control:~#ip netns exec qrouter-03266ec4-a03b-41b2-897b-c18ae3279933 ip route
192.168.0.0/24 dev qr-2a169bb4-4d  proto kernel  scope link  src 192.168.0.1
192.168.122.0/24 dev qg-9b2db4ac-31  proto kernel  scope link  src 192.168.122.200
default via 192.168.122.1 dev qg-9b2db4ac-31
root@Control:~#ip netns exec qrouter-03266ec4-a03b-41b2-897b-c18ae3279933 iptables -t nat -S
-P PREROUTING ACCEPT
-P POSTROUTING ACCEPT
-P OUTPUT ACCEPT
-N neutron-l3-agent-OUTPUT
-N neutron-l3-agent-POSTROUTING
-N neutron-l3-agent-PREROUTING
-N neutron-l3-agent-float-snat
-N neutron-l3-agent-snat
-N neutron-postrouting-bottom
-A PREROUTING -j neutron-l3-agent-PREROUTING
-A POSTROUTING -j neutron-l3-agent-POSTROUTING
-A POSTROUTING -j neutron-postrouting-bottom
-A OUTPUT -j neutron-l3-agent-OUTPUT
-A neutron-l3-agent-OUTPUT -d 192.168.122.201/32 -j DNAT --to-destination 192.168.0.2
-A neutron-l3-agent-OUTPUT -d 192.168.122.203/32 -j DNAT --to-destination 192.168.0.5
-A neutron-l3-agent-POSTROUTING ! -i qg-9b2db4ac-31 ! -o qg-9b2db4ac-31 -m conntrack ! --ctstate DNAT -j ACCEPT
-A neutron-l3-agent-PREROUTING -d 169.254.169.254/32 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 9697
-A neutron-l3-agent-PREROUTING -d 192.168.122.201/32 -j DNAT --to-destination 192.168.0.2
-A neutron-l3-agent-PREROUTING -d 192.168.122.203/32 -j DNAT --to-destination 192.168.0.5
-A neutron-l3-agent-float-snat -s 192.168.0.2/32 -j SNAT --to-source 192.168.122.201
-A neutron-l3-agent-float-snat -s 192.168.0.5/32 -j SNAT --to-source 192.168.122.203
-A neutron-l3-agent-snat -j neutron-l3-agent-float-snat
-A neutron-l3-agent-snat -s 192.168.0.0/24 -j SNAT --to-source 192.168.122.200
-A neutron-postrouting-bottom -j neutron-l3-agent-snat
```

Security Group

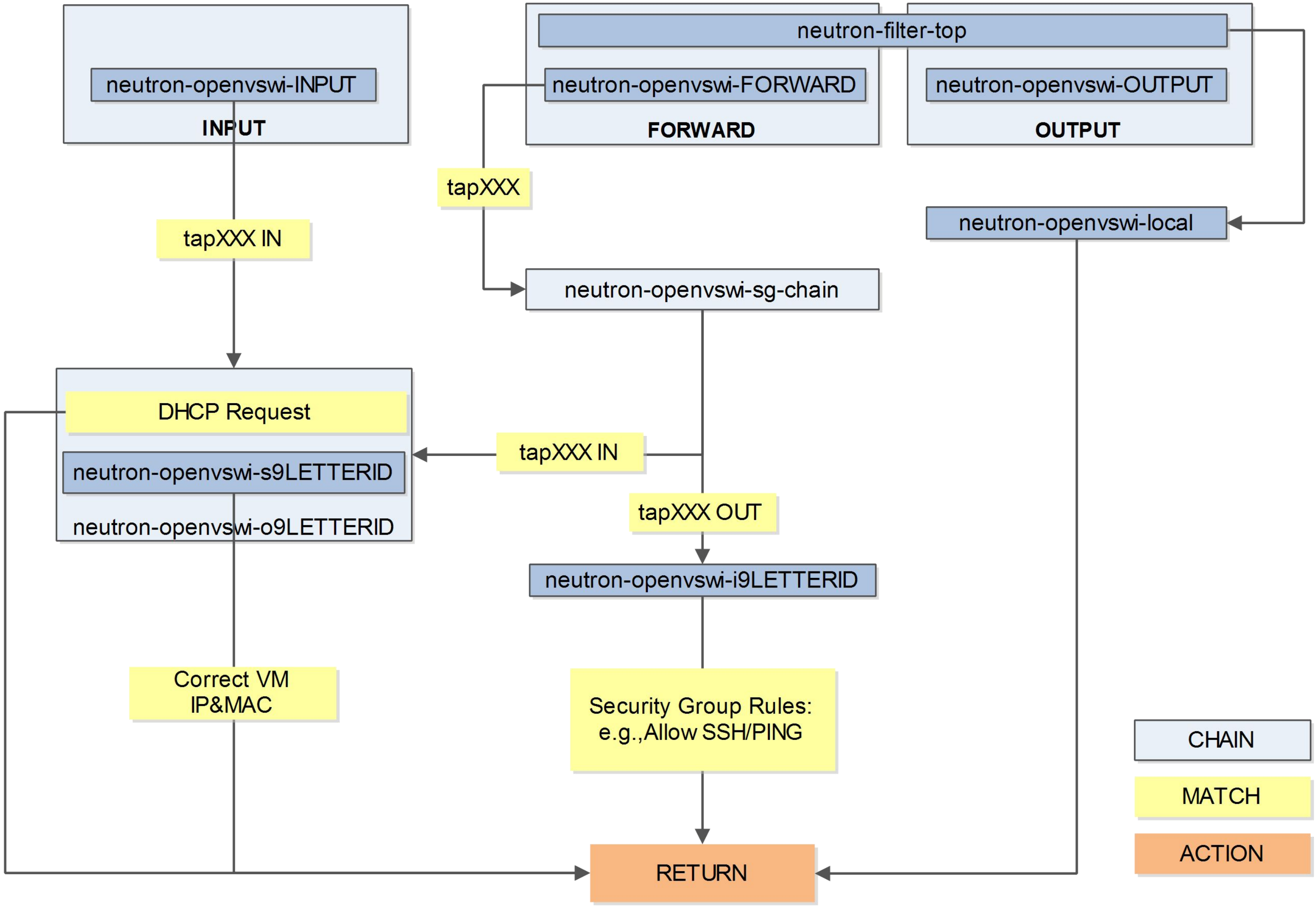


```
root@Compute:~# easyovs -m "br-int show"
Intf                Port        Vlan     Type         vmIP             vmMAC
int-br-eth0         20
qvo260209fa-72      11          1                     192.168.0.4      fa:16:3e:0f:17:04
qvo583c7038-d3      2           1                     192.168.0.2      fa:16:3e:9c:dc:3a
qvo8bf9cba2-3f      9           1                     192.168.0.5      fa:16:3e:a2:2f:0e
qvod4de9fe0-6d      8           2                     10.0.0.2         fa:16:3e:38:2b:2e
br-int              LOCAL                internal
```



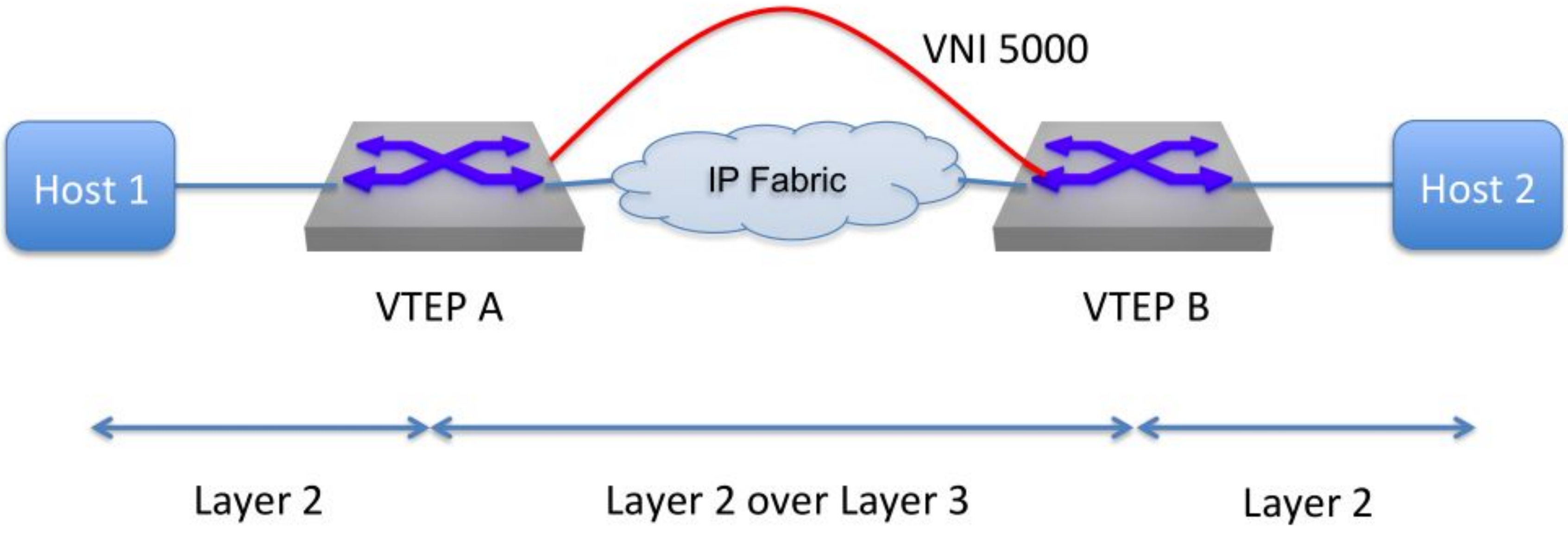
```
EasyOVS> ipt 192.168.0.2
## IP = 192.168.0.2, port = qvo583c7038-d ##
    PKTS        SOURCE          DESTINATION      PROT  OTHER
#IN:
     672        all             all              all   state RELATED,ESTABLISHED
       0        all             all              tcp   tcp dpt:22
       0        all             all              icmp
       0        192.168.0.4     all              all
       3        192.168.0.5     all              all
       8        10.0.0.2        all              all
   85778        192.168.0.3     all              udp   udp spt:67 dpt:68
#OUT:
    196K        all             all              udp   udp spt:68 dpt:67
   86149        all             all              all   state RELATED,ESTABLISHED
    1241        all             all              all
#SRC_FILTER:
   59157        192.168.0.2     all              all   MAC FA:16:3E:9C:DC:3A
```

Security Group

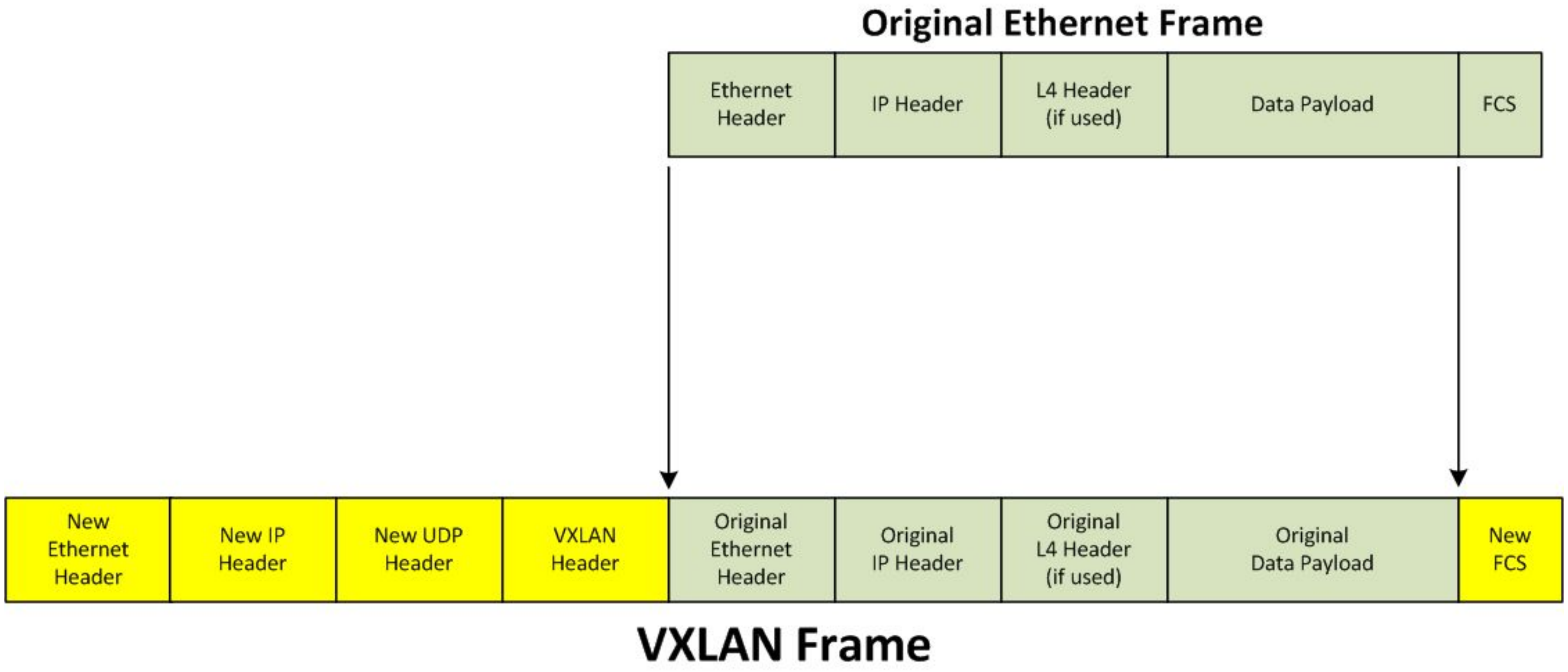


VXLAN

- □ Standardized overlay technology for encapsulaIn layer 2 traffic on top of an IP fabric



VXLAN



Original Ethernet Frame

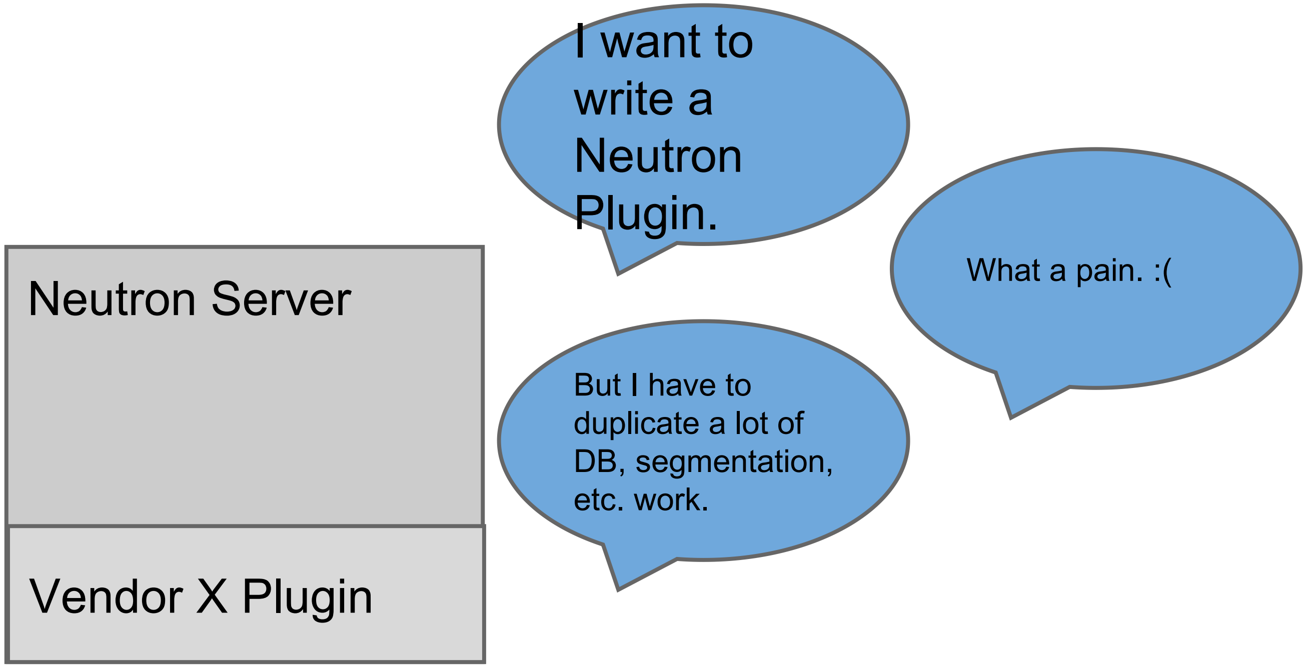
Ethernet Header	IP Header	L4 Header (if used)	Data Payload	FCS

New Ethernet Header	New IP Header	New UDP Header	VXLAN Header	Original Ethernet Header	Original IP Header	Original L4 Header (if used)	Original Data Payload	New FCS

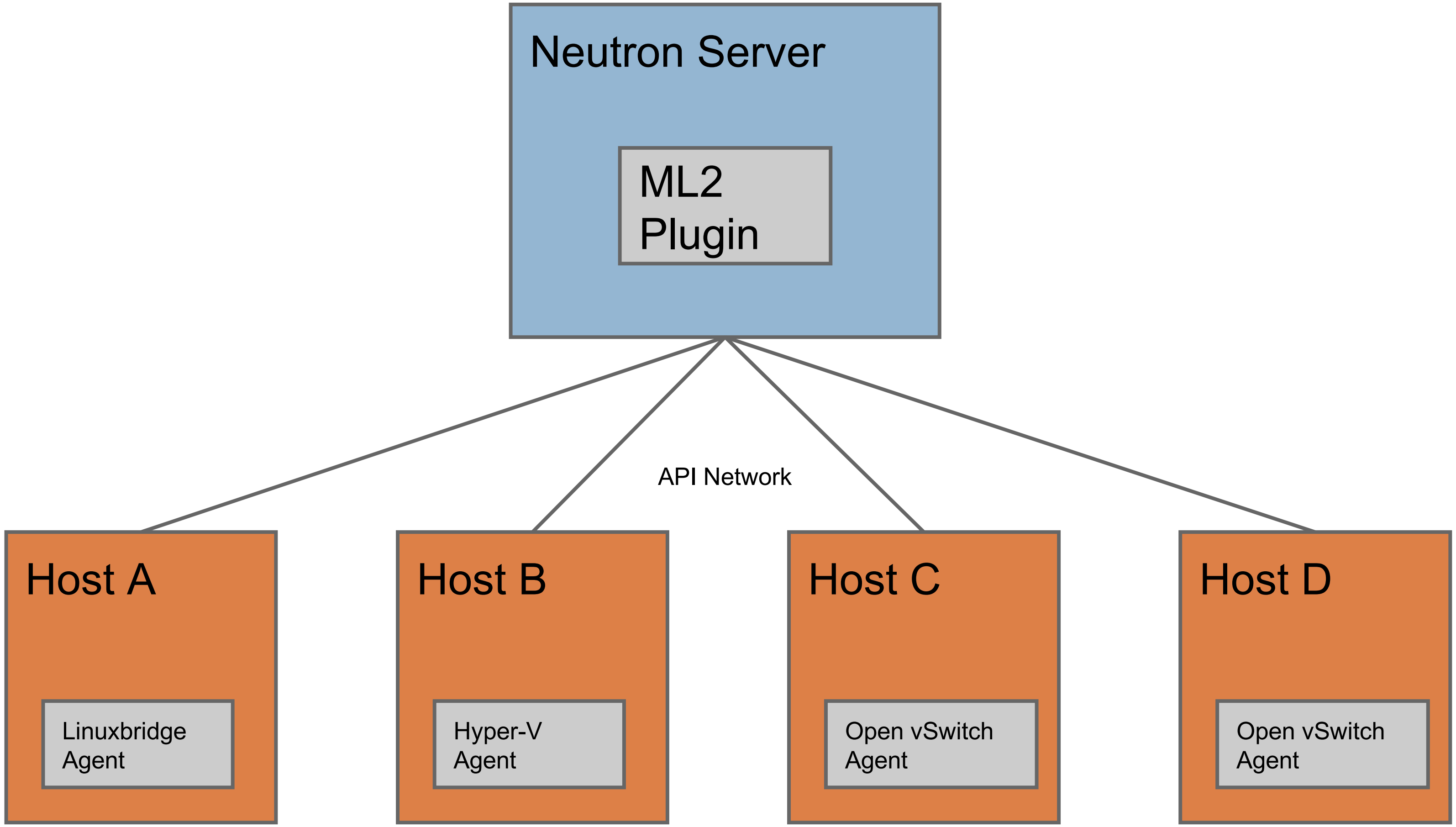
VXLAN Frame

ML2

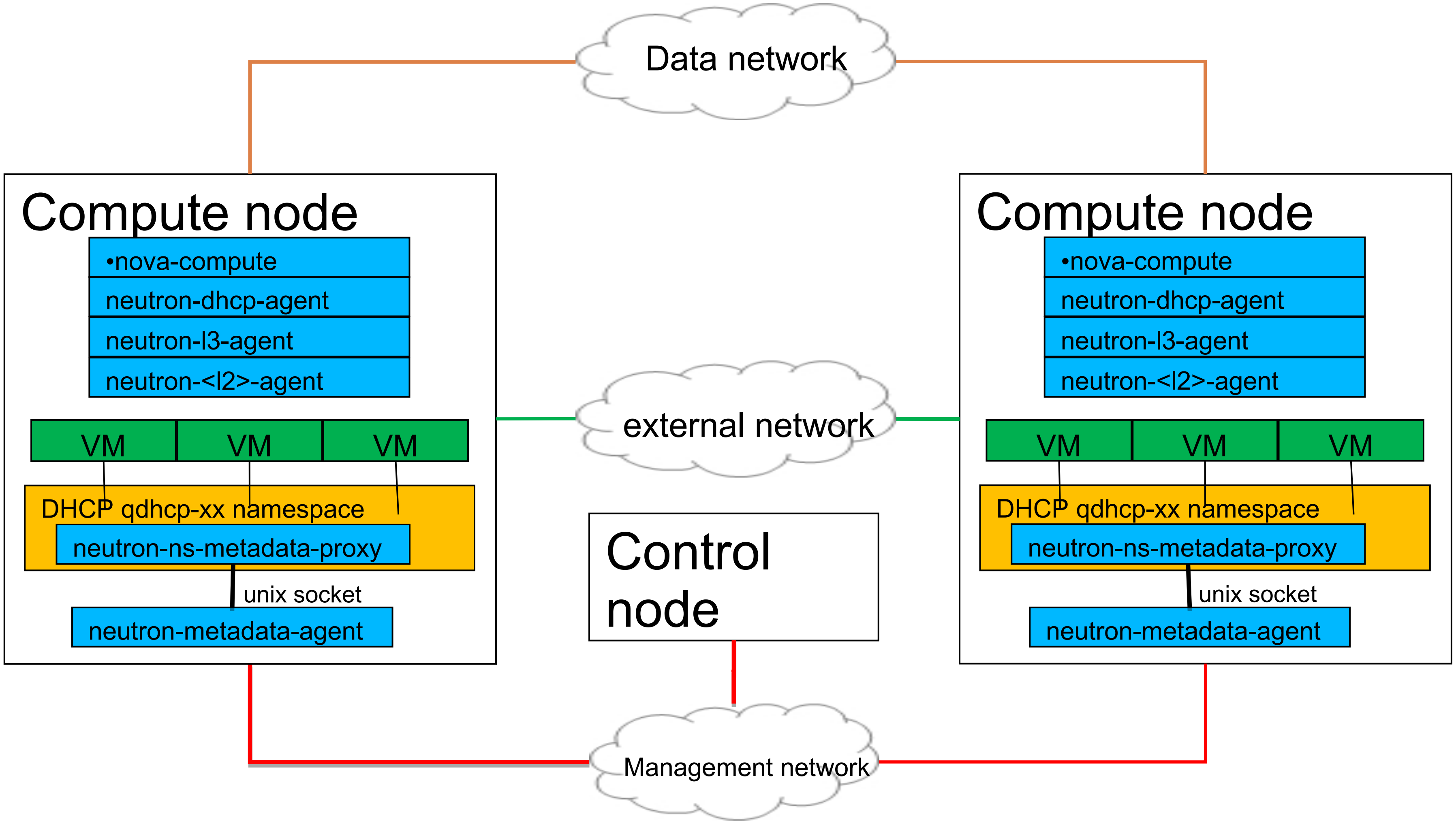
- Before Modular Layer 2


Neutron Server
Vendor X Plugin
I want to write a Neutron Plugin.
But I have to duplicate a lot of DB, segmentation, etc. work.
What a pain. :(

ML2


Neutron Server
ML2
Plugin
API Network
Host A
Linuxbridge
Agent
Host B
Hyper-V
Agent
Host C
Open vSwitch
Agent
Host D
Open vSwitch
Agent

Multihost


Data network
Compute node
•nova-compute
neutron-dhcp-agent
neutron-l3-agent
neutron-<l2>-agent
VM
VM
VM
DHCP qdhcp-xx namespace
neutron-ns-metadata-proxy
unix socket
neutron-metadata-agent
external network
Control node
Compute node
•nova-compute
neutron-dhcp-agent
neutron-l3-agent
neutron-<l2>-agent
VM
VM
VM
DHCP qdhcp-xx namespace
neutron-ns-metadata-proxy
unix socket
neutron-metadata-agent
Management network

Thank You

Backup

- □ 深入理解OpenStack中的网络实现
- □ https://github.com/yeasy/easyOVS