# Openstack neutron packet analyse

## Topology：

Network-type: vxlan

One network node(controler)

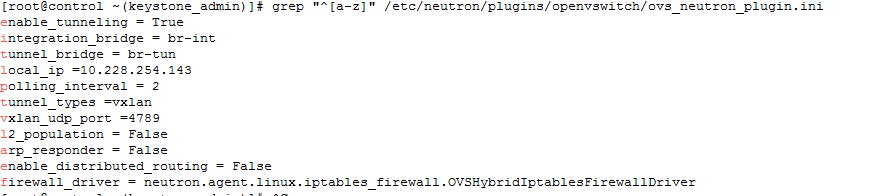
Two comute nodes, installed nova-compute and neutron-openvswitch-agent



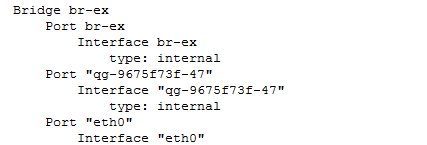
## Test one：compute2 vm10.10.10.10 ping compute1 vm10.10.20.100

1 Capture on network node

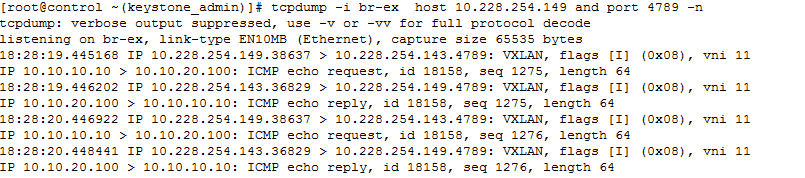
Port 4789,the config file “/etc/neutron/plugins/openvswitch/ovs\_neutron\_plugin.ini” defines the vxlan port.



Interface br-ex, as ovs switch “br-ex” bridges interface eth0.



The packets as below：



2 Analyse：

(1)The format of vxlan packet:

vxlan\_sip vxlan\_dip udp\_sport udp\_dport vm\_sip vm\_dip icmp\_date

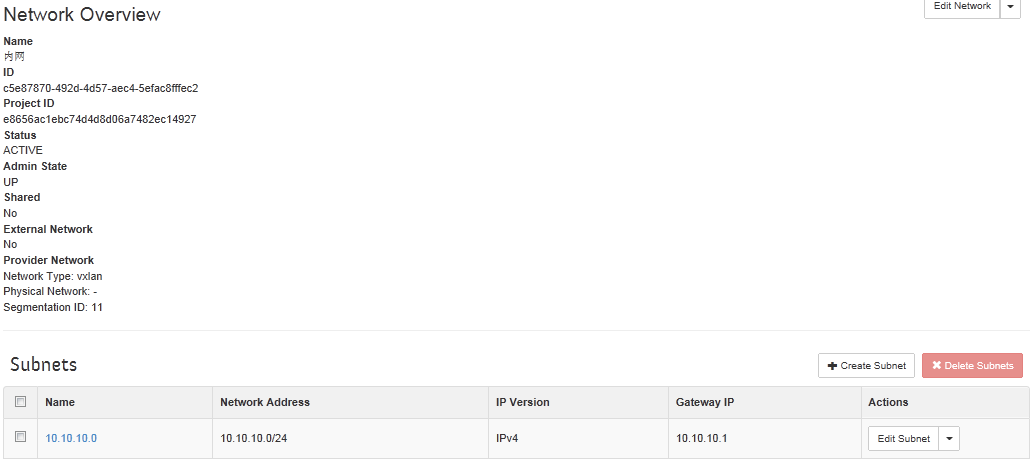
If no vxlan, the packet should be as below:

vm\_sip vm\_dip icmp\_date

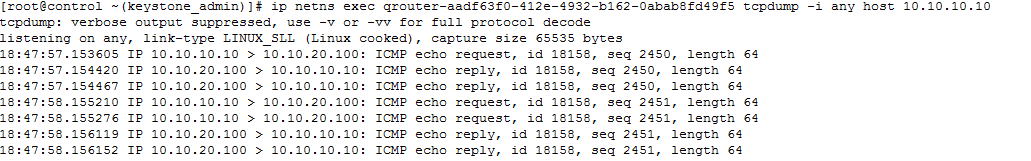
So the packets we captured, like this

10.228.254.149(compute2)—10.228.254.143(network)--（10.10.10.10---10.10.20.100）

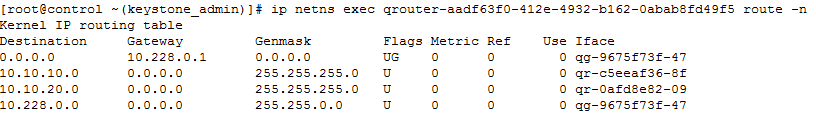
(2)The vni 11 shows which vlan the packets belong to. The vlan is automated assigned when the network creates.



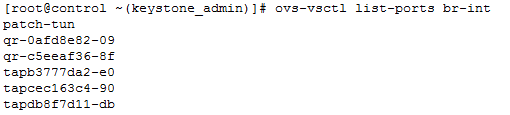
3.As compute2 vm10.10.10.10 and compute1 vm10.10.20.100 belong to different network, so to access each other that need have a route entry. On router namespace could receive icmp packets on network node. The icmp packets have de-encapsulated, so there has no udp header.



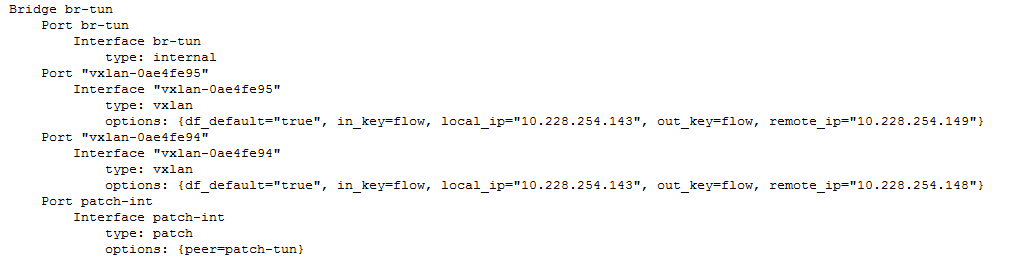
4.After router received icmp packets, searched that there has a route entry to 10.10.20.0 or not. The router sent icmp  packets to interface “qr-0afd8e82-09”



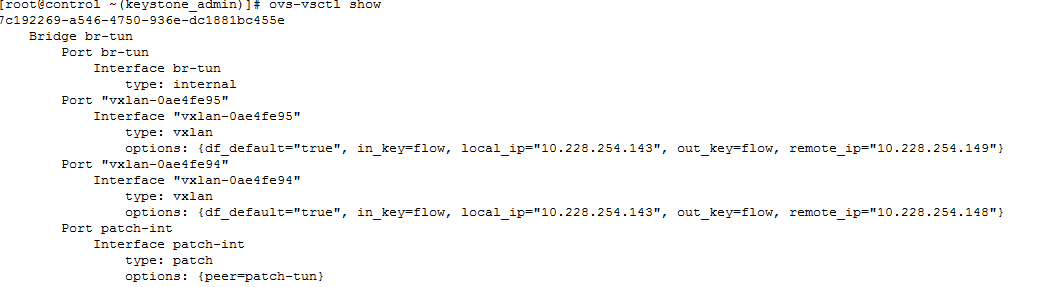
5. Interface “qr-0afd8e82-09” belongs to ovs switch “br-int”. When switch “br-int” received those packets, it sent the packets to “patch-tun”.



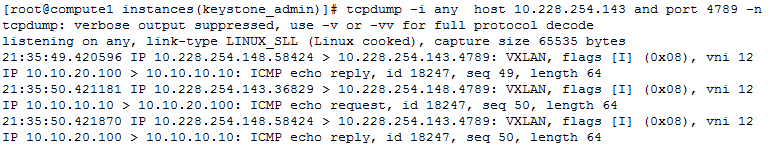
6. Interface “patch-tun” and ovs switch “br-tun” interface “patch-int” are veth pair.(Packets received one of veth pair, and then sent to another interface)



7. The ovs switch “br-tun” received the packets on network node。It find that the packets should be sent to port “vxlan-0ae4fe94”, so the packets encapsulated to vxlan format, the destination ip address is 10.228.254.148.



8. Captured on compute1 node



9.The packets transfer as below:

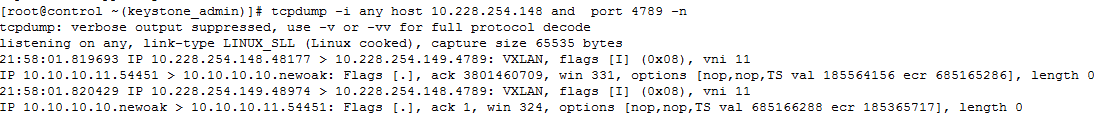
compute2 vm 10.10.10.10----compute2 ovs br-int----- compute2 ovs br-tun----- compute2 eth0----- network eth0----- network ovs br-tun----- network ovs br-int--------- network router----- network ovs br-int--------- network ovs br-tun----- network eth0---- compute1 eth0----- compute1 ovs br-tun----- compute2 ovs br-int---- Compute1 vm 10.10.20.100

## Test two：compute1 vm10.10.10.101 ping compute1 vm10.10.10.11

These VMs belong to same compute node and same vlan id, so the packets directly interactive on ovs switch “br-int” .

## Test three：compute1 vm10.10.10.11 ping compute2 vm10.10.10.10

These VMs belong to different compute nodes , but have the same vlan id, so the packets should be sent to network node, and then sent to destination compute node. Because of belong to same vlan, so the packets don’t need to route by router namespace. The other difference from test one is that the ip addresses are compute node’s ip address. The test one, the ip addresses are network node and compute node’s ip address.



## Summary：

1. If the VMs belong to the same vlans and compute node, the packets directly interactive on ovs switch “br-int” .
2. If the VMs belong to the same vlans, but belong to different compute nodes, the packets should be sent to network node, but the network node don’t modify the packets.
3. If the VMs belong to the different vlans and compute nodes , the packets should be sent to network node, and then network node’s router handled the packets.
4. If the VMs belong to the different vlans , but belong to same compute nodes, the packets should be sent to network node, and then network node’s router handled the packets. As ovs switch does not support route.