Neutron enhancement

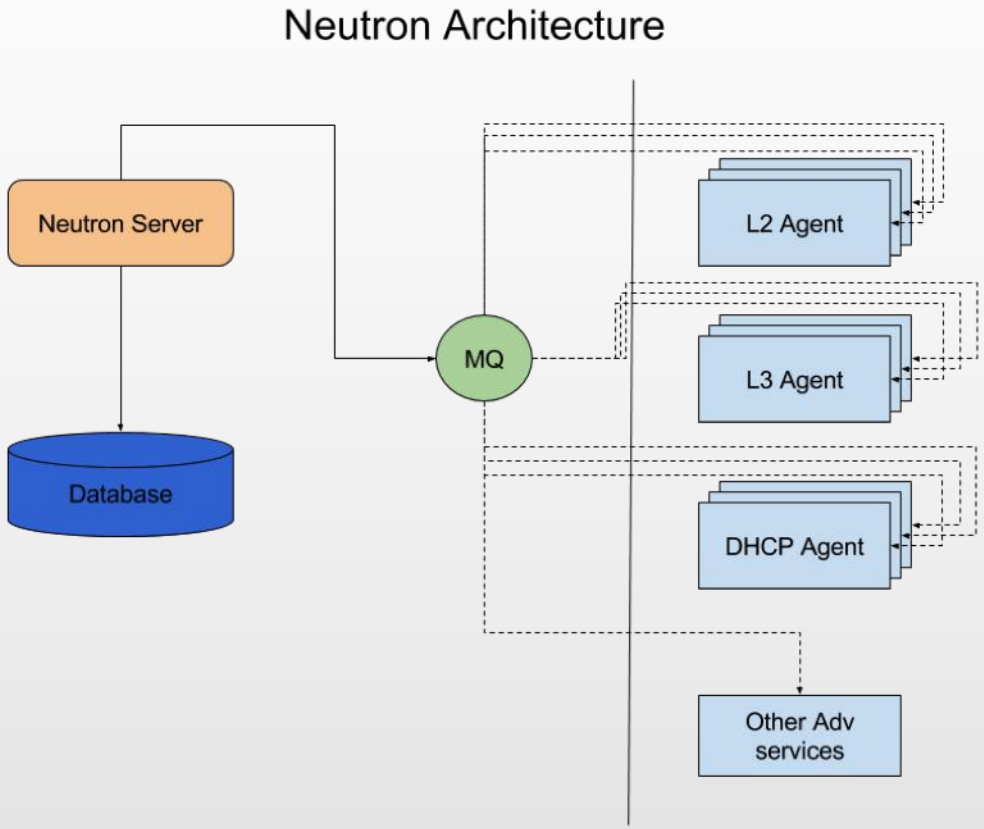
LIU Yulong

1. DHCP and ml2 openvswitch agent

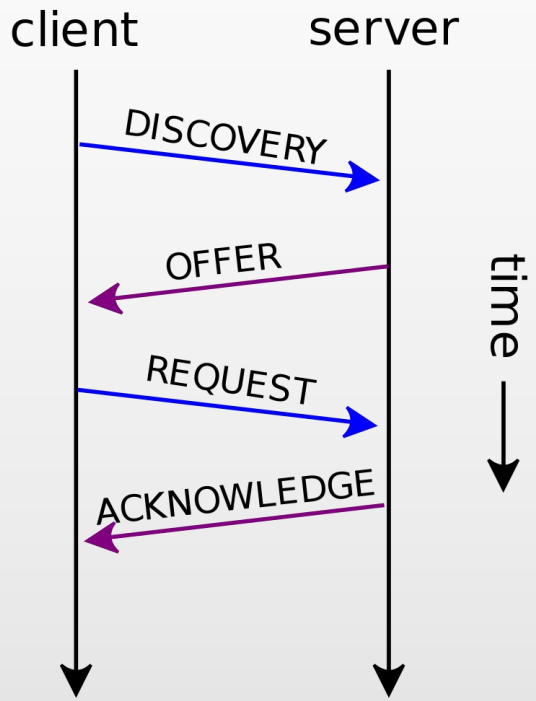
- DHCP agent backgrouds
  - one more RPC S/C, one more persure on MQ
  - extra processes
    - dnsmasq
    - haproxy (for metadata, if needed)
  - namespace
    - DHCP namespace
    - tons of DVR local router namespace
      - host_dvr_for_dhcp = False (Train)
  - provisioning blocks
  - retry and fullsync
  - loops
  - locks
  - caches
  - ...

1. DHCP and ml2 openvswitch agent

- Basic Framework



Graphic1. neutron architecture



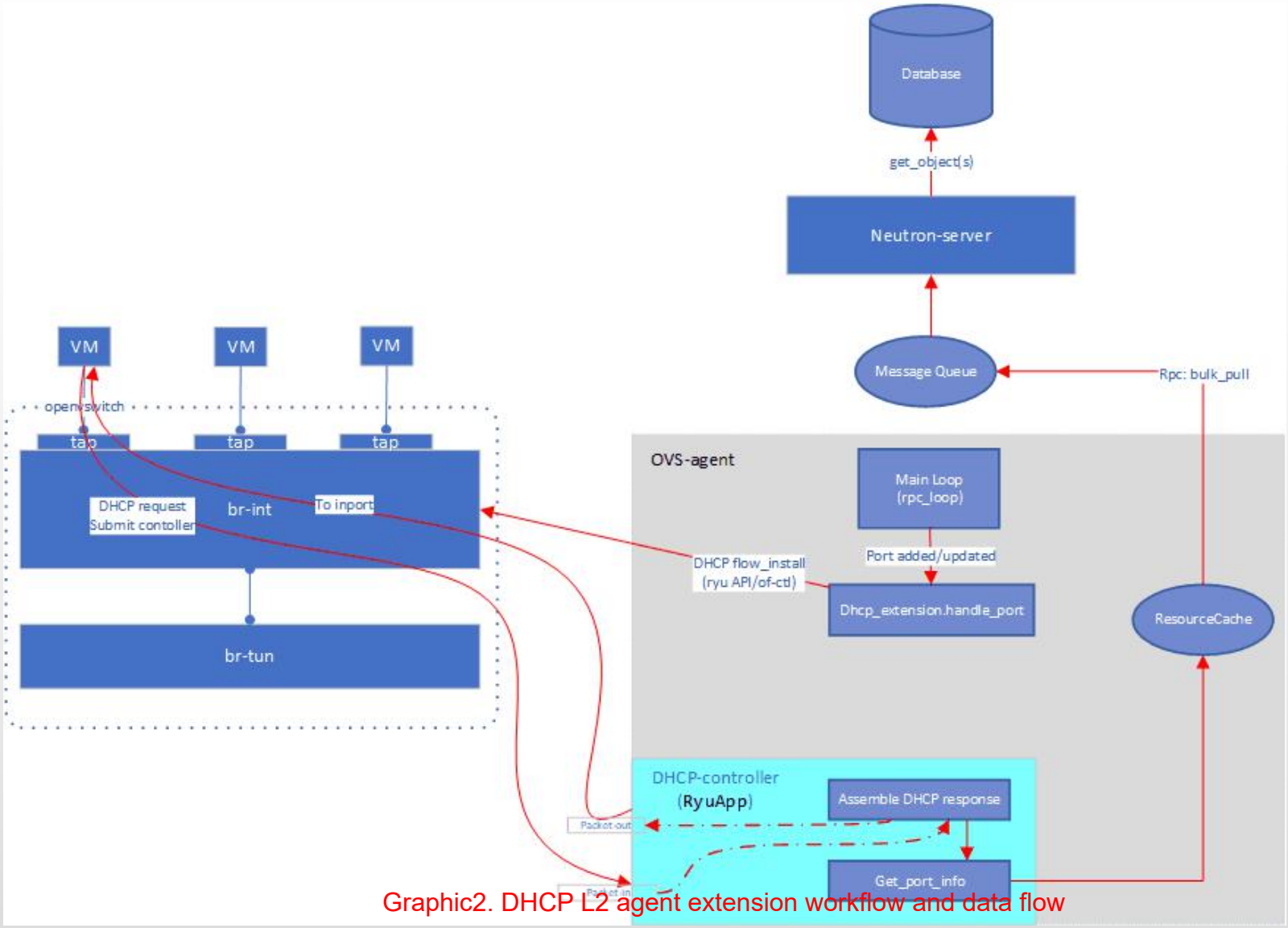
Graphic1. DHCPv4

1. DHCP and ml2 openvswitch agent

- New approach
  - Retire DHCP-agent
    - No more extra processes, no DNS
    - For large scale
  - Add L2 agent extension to replace the DHCP-agent
  - DHCPv4 and DHCPv6
  - Flows for DHCP to send request to the local controller
  - A DHCP server based on RYU (os-key) application
  - Directly packet-out to the INPORT
  - Upgrading without any side effect
  - Fully distributed
    - no sigle point of failure
    - every compute node will do DHCP R/R locally

1. DHCP and ml2 openvswitch agent

- New Framework



Graphic2. DHCP L2 agent extension workflow and data flow

1. DHCP and ml2 openvswitch agent

- Basic Flows

DHCP_IPV4_TABLE = 77
DHCP_IPV6_TABLE = 78

table=60, n_packets=0, n_bytes=0, priority=101,udp,nw_dst=255.255.255.255,tp_src=68,tp_dst=67 actions=resubmit(,77)
table=60, n_packets=0, n_bytes=0, priority=101,udp6,ipv6_dst=ff02::1:2,tp_src=546,tp_dst=547 actions=resubmit(,78)

table=77, n_packets=0, n_bytes=0, priority=100,udp,in_port=58,dl_src=fa:16:3e:f0:2a:c3,tp_src=68,tp_dst=67
actions=controller(userdata=fd.eb.08.bf.db.81.46.4b.8e.a8.2a.bb.41.1c.30.a9)

table=78, n_packets=0, n_bytes=0, priority=100,udp6,in_port=58,dl_src=fa:16:3e:f0:2a:c3,tp_src=546,tp_dst=547
actions=controller(userdata=fd.eb.08.bf.db.81.46.4b.8e.a8.2a.bb.41.1c.30.a9)

2. Egress packets are flooding on br-int

- Original problem
  - https://bugs.launchpad.net/neutron/+bug/1732067
  - openvswitch firewall flows cause flooding on integration bridge
    - ingress output -> port
    - egress normal
      - fdb unknow/unlearned -> flood
  - For shared network, it is an security issue.
    - VMs among different tenants should not receive packets from others.

2. Egress packets are flooding on br-int

- New approach
  - [L2][OVS] add accepted egress fdb flows
  - https://bugs.launchpad.net/neutron/+bug/1841622
- To solve the egress traffic flood in br-int
- New config option will be needed for those deployments which require the fix
- Openflow firewall enabled
  - 1. the "dest mac" is handled in this ovs-agent, direct "output" to that of port
  - 2. "ARP request" with enabled L2 pop, packets will still be sent to patch port to tunnel bridge by NORMAL action
  - 3. "dest mac" is not in this host, vlan or tunnel (gre/vxlan/geneve) unicast will be sent (output) to corresponding patch port of tunnel/physical bridge.
  - 4. other traffic still match the original NORMAL flow

2. Egress packets are flooding on br-int

- A new table=61 will be used to do egress traffic classification when not enable openflow firewall (for HA routers this should not be enabled):
  - 1. egress packets will be send to table 61, match rule will be of-port which be handled by ovs-agent "in_port=<some_local_of_port>"
  - 2. the "dest mac" is handled this ovs-agent, direct "output" to that port
  - 3. "ARP request" with enabled L2 pop, packets will still be sent to patch port to tunnel bridge by the original NORMAL
  - 4. "dest mac" not in this host, vlan or tunnel (gre/vxlan/geneve) unicast will be sent (output) to corresponding patch port of tunnel/physical bridge.
  - 5. other traffic still match the original NORMAL flow

2. Egress packets are flooding on br-int

- table and flows

table=94, priority=12,reg6=0x2,dl_dst=fa:16:3e:f0:2a:c3 actions=output:58
table=94, priority=10,reg6=0x2,dl_src=fa:16:3e:f0:2a:c3,dl_dst=00:00:00:00:00:00/01:00:00:00:00:00 actions=mod_vlan_vid:2,output:3
table=94, priority=1 actions=NORMAL

LOCAL_SWITCHING = 0
DVR_TO_SRC_MAC = 1
DVR_TO_SRC_MAC_VLAN = 2
CANARY_TABLE = 23
ARP_SPOOF_TABLE = 24
MAC_SPOOF_TABLE = 25
TRANSIENT_TABLE = 60
BASE_EGRESS_TABLE = 71
RULES_EGRESS_TABLE = 72
ACCEPT_OR_INGRESS_TABLE = 73
BASE_INGRESS_TABLE = 81
RULES_INGRESS_TABLE = 82
ACCEPTED_EGRESS_TRAFFIC_TABLE = 91
ACCEPTED_INGRESS_TRAFFIC_TABLE = 92
DROPPED_TRAFFIC_TABLE = 93
ACCEPTED_EGRESS_TRAFFIC_NORMAL_TABLE = 94

TRANSIENT_EGRESS_TABLE = 61

table=61, priority=12,dl_dst=fa:16:3e:39:42:e4 actions=output:"tapd2df8572-62"
table=61, priority=10,in_port="tapd2df8572-62",dl_src=fa:16:3e:39:42:e4,dl_dst=00:00:00:00:00:00/01:00:00:00:00:00 actions=mod_vlan_vid:1,output:"patch-tun"
table=61, priority=3 actions=NORMA

3. Performance of ml2 openvswitch agent

- Local resource cache
  - dump cache to local file
  - load cache from the local file path duing the agent restart
- Local flows cache and batch/defer updating
  - '--bundle' has been used in openflow security group flows.
- Async-processing large set flows after ports basic flows installed
  - [L2] update the port DB status directly in agent-side
  - https://bugs.launchpad.net/neutron/+bug/1840979

3. Performance of ml2 openvswitch agent

- [L2] stop processing ports twice in ovs-agent (Sapna Jadhav is looking into this now.)
- https://bugs.launchpad.net/neutron/+bug/1841865
- Increase port processing time linearly
  - rpc_loop X (10 added, 0 updated)
  - rpc_loop X +1 (20 added, 10 updated)
  - rpc_loop X + 2 (30 added, 20 updated)

4. Health check for ml2 openvswitch agent

- A new API and small (one or two long-term) agent for L2 traffic health check
  - https://bugs.launchpad.net/neutron/+bug/1830014
  - https://review.opendev.org/#/c/662541/
  - Why introduce a new agent?
    - For now, we have no choice.
    - We just want one smiple agent which can save time for operators.

Thank You

To be continue...

L3 is coming...

neutron L3 agent

- Retire metering-agent
- https://bugs.launchpad.net/neutron/+bug/1817881
- https://review.opendev.org/#/c/658511/
- https://review.opendev.org/#/c/675654/

neutron L3 agent

- Lazy-load agent side router resources when no related service port (compute ports or baremental ports)
- Router (HA) will be processed at least 4 times after a user create a new router and bind floating IP under it.
    - create
    - add subnet
    - set external gateway
    - bind floating ip

neutron L3 agent

- [RFE][L3] l3-agent should have its capacity
- https://bugs.launchpad.net/neutron/+bug/1828494

neutron L3 agent

- Centralized DNAT traffic (floating IP) Scale-out
- Using some protocols to extend the theoretical maximum bandwidth (one host's max bandwidth) for single IP
- Active-Active model for a single router in DNAT nodes
- each connection to the destenation IP will be hashed to different nodes

neutron L3 agent

- Router agent side health check
  - namespace
  - iptables
  - route rules
  - arp entries
  - dvr related flows
  - HA related processes
  - Extension related
    - floating IP tc rules
    - gateway IP tc rules