HP HELION OPENSTACK LAB GUIDE

ČÁST TŘETÍ – NETWORKING

HP Helion OpenStack 1.1

Červen 2015

Tomáš Kubica

Dokument verze 0.20

Obsah

1. OpenStack Neutron

Popis, funkce, části systému

2. Průvodce životem paketu

2.1. Znalost vs. troubleshooting

Častá otázka síťařů při seznamování s OpenStack je: „Co se tam děje a jak to troubleshootovat?“. Třeba se nechcete spokojit s tím, že to nějak magicky funguje. Nejsou to kouzla, vše je velmi chytré, ale uvnitř docela složité. Ještě, než začnete číst dál, chci navrhnout následující příměr. Uvnitř síťového prvku, který běžně používáte, se toho děje opravdu hodně – mnoho různých zdrojů a tabulek, velmi složitá pipeline, kominace hardwarových komponent s odlišnými vlasnostmi (hash tabulky, BCAM, TCAM, LPM, parsery, meta-data, modifikátory, replikátory, ...). Niz z toho vám ale výrobce nechce ukázat. Máte pro troubleshooting mnoho informací, ale do nejhlubší skutečnosti se podívat nemůžete.

OpenStack Neutron v Helion OpenStack tohle umožňuje. Podíváme se společně jak to funguje doopravdy. Nemějte to za nadměrně složité – jde jen o to, že skutečnost můžete vidět, u klasického prvku ne. Není nutné při troubleshootingu znát až takovou úroveň detailu. Bohatě postačí vědět jak zachytit pakety, přes jaké vSwitche má co procházet a jak se to zabaluje do VXLAN. Možná ale chcete znát víc, třeba jak fungují různé tabulky v pipeline... někdy se to může hodit.

2.2. Sběr údajů pro troubleshooting

Jak pro hraní si tak pro troubleshootig reálného problému je vždy dobré začít sesbíráním potřebných informací. Identifikátory problematických částí systému (instance, compute node), MAC a IP adresy, různá ID jako je tenant ID, instance ID a tak podobně.

Zjistíme si tenant ID pro náš projekt

```
keystone tenant-get mujprojekt
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |                                  |
|   enabled   |               True               |
|      id     | baa7096fe1d54571900c3758397e0939 |
|     name    |            mujprojekt            |
+-------------+----------------------------------+
```

Na jakých fyzických nodech běží instance tohoto projektu a jaké má lokální jméno?

```
nova list --all-tenants 1 --tenant baa7096fe1d54571900c3758397e0939 --fields name,OS-EXT-SRV-ATTR:host,OS-EXT-SRV-ATTR:instance_name
+--------------------------------------+------------+-------------------------------------+-------------------------------+
| ID                                   | Name       | OS-EXT-SRV-ATTR: Host               | OS-EXT-SRV-ATTR: Instance Name |
+--------------------------------------+------------+-------------------------------------+-------------------------------+
| eb347271-dc5a-46cf-9150-0a7defffc6d1 | instance-1 | overcloud-novacompute0-vli5de2egecg | instance-0000010d             |
| 70d0662f-9c69-4d0b-99e7-2dde4e0494e8 | instance-2 | overcloud-novacompute0-vli5de2egecg | instance-0000010e             |
| e1975422-a543-4ce4-be36-bce191816161 | instance-3 | overcloud-novacompute0-vli5de2egecg | instance-0000010f             |
+--------------------------------------+------------+-------------------------------------+-------------------------------+
```

Budeme se chtít podívat přímo dovnitř compute node, zjistěme si jeho IP adresu

```
nova hypervisor-list
+----+-------------------------------------------+
| ID | Hypervisor hostname                       |
+----+-------------------------------------------+
| 1  | overcloud-novacompute0-vli5de2egecg.novalocal |
```

```
| 2  | overcloud-novacompute1-c4ia2jfbd75d.novalocal |
+----+-----------------------------------------------+
nova hypervisor-show overcloud-novacompute0-vli5de2egecg.novalocal | grep host_ip
| host_ip                  | 10.0.10.14
```

Nalogujte se do tohoto compute node z místa, kde máte certifikát, tedy například ze seed VM.

```
root@hLinux:~# ssh heat-admin@10.0.10.14
Linux overcloud-novacompute0-vli5de2egecg 3.14.29-4-amd64-hlinux #hlinux1 SMP Mon Feb 9 20:32:22 UTC 2015 x86_64

The programs included with the hLinux system are free software; the exact
license terms for each program are described in the individual files in
/usr/share/doc/*/copyright.
Last login: Mon May  4 13:31:09 2015 from 10.0.10.2
$ sudo -i
root@overcloud-novacompute0-vli5de2egecg:~#
```

Můžeme se podívat na běžící VM.

```
root@overcloud-novacompute0-vli5de2egecg:~# virsh list
 Id    Name                           State
----------------------------------------------------
 5     instance-00000055              running
 6     instance-00000056              running
 74    instance-000000bd              running
 79    instance-000000c8              running
 96    instance-000000e2              running
 104   instance-000000eb              running
 105   instance-000000ed              running
 106   instance-000000ee              running
 107   instance-000000ef              running
 124   instance-00000108              running
 125   instance-00000109              running
 126   instance-0000010a              running
 127   instance-0000010b              running
 128   instance-0000010c              running
 129   instance-0000010d              running
 130   instance-0000010f              running
 131   instance-0000010e              running
```

Nás ale zajímá konkrétně první z našich VM, tedy jak jsme zjistili s lokálním názvem instance-0000010d. Vypíšeme si všechny informace – pro úsporu místa na tomto papíře použiji grep pro identifikaci té pasáže, která je pro nás podstatná.

```
root@overcloud-novacompute0-vli5de2egecg:~# virsh dumpxml instance-0000010d | grep -A 7 "<interface"
    <interface type='bridge'>
      <mac address='fa:16:3e:21:cf:75'/>
      <source bridge='qbr425fe781-d3'/>
      <target dev='tap425fe781-d3'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
```

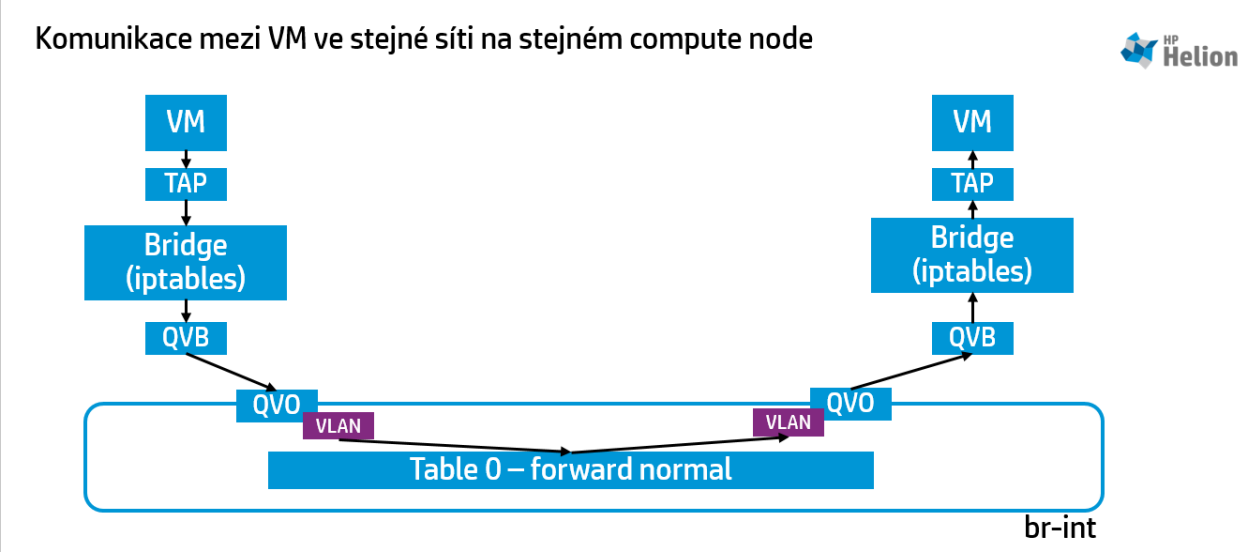
Zajímá nás především tap interface a také bridge.

Připojte se do naší instance-1, zkontrolujeme IP adresu a začneme ping na instance-2.

```
debian@instance-1:~$ ip a
1: 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
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1400 qdisc pfifo_fast state UP qlen 1000
    link/ether fa:16:3e:21:cf:75 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.8/24 brd 192.168.10.255 scope global eth0
    inet6 fe80::f816:3eff:fe21:cf75/64 scope link
       valid_lft forever preferred_lft forever
debian@instance-1:~$ ping 192.168.10.9
PING 192.168.10.9 (192.168.10.9) 56(84) bytes of data.
64 bytes from 192.168.10.9: icmp_req=1 ttl=64 time=1.58 ms
64 bytes from 192.168.10.9: icmp_req=2 ttl=64 time=0.713 ms
64 bytes from 192.168.10.9: icmp_req=3 ttl=64 time=0.616 ms
```

2.3. Komunikace mezi VM ve stejné síti na stejném compute node (east-west)



2.3.1. Pakety odcházející z VM

Informace máme posháněné, začneme kontrolovat cestu paketu. Nejprve se tedy podívejme přímo na tap interface naší VM zda vidíme ICMP provoz.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump icmp -e -i tap425fe781-d3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap425fe781-d3, link-type EN10MB (Ethernet), capture size 262144 bytes
09:30:34.286689 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2290, seq 102, length 64
09:30:34.287081 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2290, seq 102, length 64
09:30:35.286849 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2290, seq 103, length 64
09:30:35.287227 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2290, seq 103, length 64
09:30:36.286817 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2290, seq 104, length 64
09:30:36.287229 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2290, seq 104, length 64
09:30:37.286772 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2290, seq 105, length 64
```

Výborně, to se zdá být v pořádku.

2.3.2. Aplikace Security Group

Vypišme si všechna iptables chain týkající se našeho VM interface:

```
root@overcloud-novacompute0-vli5de2egecg:~# iptables --list-rules | grep tap425fe781-d3
-A neutron-openvswi-FORWARD -m physdev --physdev-out tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-sg-chain
-A neutron-openvswi-FORWARD -m physdev --physdev-in tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-sg-chain
-A neutron-openvswi-INPUT -m physdev --physdev-in tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-o425fe781-d
-A neutron-openvswi-sg-chain -m physdev --physdev-out tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-i425fe781-d
-A neutron-openvswi-sg-chain -m physdev --physdev-in tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-o425fe781-d
```

Tím jsme našli vstupní (neutron-openvswi-i425fe781-d) a výstupní (neutron-openvswi-o425fe781-d) řetězec pravidel. Můžeme si je vypsat:

```
root@overcloud-novacompute0-vli5de2egecg:~# iptables --list neutron-openvswi-i425fe781-d -v -n
Chain neutron-openvswi-i425fe781-d (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
15176 1012K RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 RETURN     udp  --  *      *       192.168.10.3         0.0.0.0/0           udp spt:67 dpt:68
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           match-set IPv4b9eaf0cf-e8b2-41f1-9 src
    0     0 RETURN     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 RETURN     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 RETURN     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           match-set IPv4ea62d680-0c24-4f60-9 src
    2   656 neutron-openvswi-sg-fallback  all  --  *      *       0.0.0.0/0            0.0.0.0/0

root@overcloud-novacompute0-vli5de2egecg:~# iptables --list neutron-openvswi-o425fe781-d -v -n
Chain neutron-openvswi-o425fe781-d (2 references)
 pkts bytes target     prot opt in     out     source               destination
 5733 1801K RETURN     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:68 dpt:67
 378K   42M neutron-openvswi-s425fe781-d  all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
 377K   42M RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
  278 19184 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0
  500 42000 neutron-openvswi-sg-fallback  all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

Všimněte si povoleného ICMP, SSH a web tak, jak to obsahuje naše security group v Helion OpenStack. Další zajímavost je explicitní povolení jen jednoho konkrétního DHCP serveru (to je ten, který patří k Neutron implementaci a zajišťuje distribuci adres – neoprávněné DHCP servery tak nebudou fungovat).

2.3.3. Provoz po aplikaci Security Group

Jméno bridge, do kterého vede tap interface už jsme zjistili dříve. Teď nás bude zajímat spojnice mezi tímto mostem a jeho napojením do OpenvSwitch. Tento interface najdeme snadno:

```
root@overcloud-novacompute0-vli5de2egecg:~# brctl show qbr425fe781-d3
bridge name     bridge id               STP enabled     interfaces
qbr425fe781-d3          8000.ca1a7962d69c       no              qvb425fe781-d3
                                                                tap425fe781-d3
```

Na qvb interface už uvidíte provoz po zpracování Security Group – pokud tato tedy provoz blokuje, na tap interface ho uvidíte, na qvb ne. V našem případě je ICMP povoleno (ale můžete experimentovat s různým nastavením, pokud chcete).

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump icmp -e -i qvb425fe781-d3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvb425fe781-d3, link-type EN10MB (Ethernet), capture size 262144 bytes
10:47:34.466519 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2296, seq 504, length 64
```

```
10:47:34.466804 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2296, seq 504, length 64
10:47:35.466540 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2296, seq 505, length 64
10:47:35.466782 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2296, seq 505, length 64
```

2.3.4. Vstup do vSwitch br-int

Zatím jsme došli k portu vystupujícímu z prvního bridge (který je tam kvůli implementaci security group). Tento port je ve spojnici směřující do vSwitche – druhý konec této spojnice poznáte snadno – má stejné jméno, ale místo „qvb“ obsahuje „qvo“. Do kterého vSwitch tedy vstupuje?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl port-to-br qvo425fe781-d3
br-int
```

To je dle očekávání – br-int je určen právě na spojení všech VM. Jak je lokálně zajištěna izolace jednotlivých sítí a tenantů (připomínám, že stále jsme na L2)? Je to řešeno přes VLAN tagy, které jsou automaticky přiřazovány Neutron komponentou v OpenStack. Mají pouze lokální význam, nikdy se nepublikují kamkoli do okolní světa, nikdy se neobjeví v paketu. Jaký tag je přiřazen našemu portu?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 qvo425fe781-d3
        Port "qvo425fe781-d3"
            tag: 69
            Interface "qvo425fe781-d3"
```

Je to VLAN 69 – VM je tedy napojena do portu typu access ve VLAN 69.

Teď můžeme zkoumat vSwitch tabulky. Každý port má nějaký identifikátor (podobně jako ve fyzickém prvku je description, která je informativní, rozhodující je identifikátor). Nejprve zjistíme číslo portu, do kterého vstupuje naše VM:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep qvo425fe781-d3
 211(qvo425fe781-d3): addr:da:04:37:a2:8a:f6
```

Je to tedy 211. Pokud přichází provoz z tohoto specifického portu, máme v prvku nějaká pravidla? Můžeme si je vypsat, ale v tuto chvíli nás zajímají jen první tři.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-int
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=2915201.181s, table=0, n_packets=2, n_bytes=220, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
 cookie=0x0, duration=2915201.371s, table=0, n_packets=1660, n_bytes=168645, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=2915201.488s, table=0, n_packets=42131982, n_bytes=5726840859, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
```

To co dělají první dva se týká směrování a to zatím neřešíme – z předchozího tcpdump je zřejmé, že cílová MAC adresa byla fa:16:3e:fd:7f:88. Ta se neshoduje s žádnou v prvních pravidlech, takže přijde ke slovu to třetí – a to říká, že se má prvek použít klasický switching. Jaká je tedy MAC forwarding tabulka v naší interní VLAN?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int
 port  VLAN  MAC                Age
...
  211    69  fa:16:3e:21:cf:75    0
  212    69  fa:16:3e:07:de:20    0
...
  215    69  fa:16:3e:fd:7f:88    0
```

Je tedy zřejmé, že cílová MAC našich paketů (tedy naše druhá VM) je pro switch známa a provoz odejde portem 215. Tímto jsme se dostali do poloviny cesty, nic dalšího už se dít nebude – routing není potřeba a cíl je přímo ve stejném compute node ve stejné síti, takže stačí jen doskákat do cílové VM.

2.3.5. Konec cesty

Obrátíme teď proces zkoumání a dojdeme do cíle.

Jaké je jméno odchozího portu 215?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep 215
 215(qvoaeee0c10-2e): addr:3a:0e:d4:c7:73:12
```

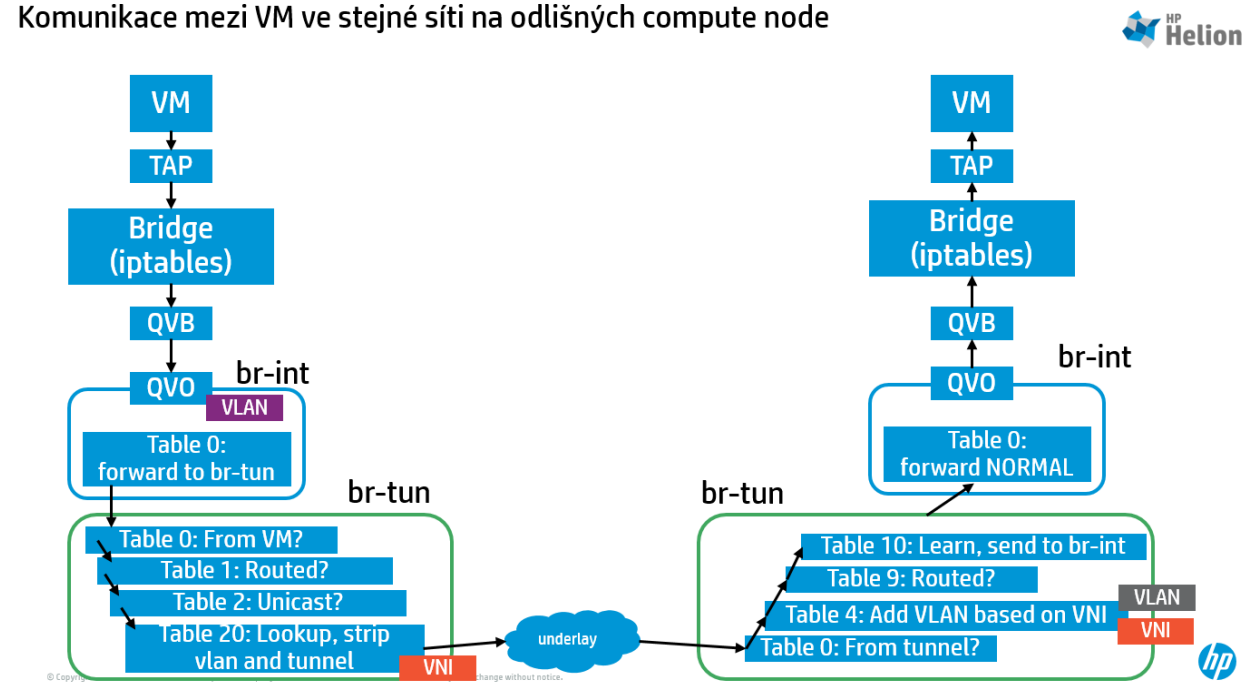
Protikus před vstupem do bridge pro implementaci Security Group je tedy qvbaeee0c10-2e. Můžeme se podívat na provoz.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump icmp -e -i qvbaeee0c10-2e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvbaeee0c10-2e, link-type EN10MB (Ethernet), capture size 262144 bytes
18:07:45.246766 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2341, seq 255, length 64
18:07:45.247053 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2341, seq 255, length 64
18:07:46.246789 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2341, seq 256, length 64
18:07:46.247032 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2341, seq 256, length 64
```

Na závěr si zjistíme ještě tap interface samotné cílové VM a uděláme další packet trace.

```
root@overcloud-novacompute0-vli5de2egecg:~# brctl show | grep -A1 qvbaeee0c10-2e
qbraeee0c10-2e          8000.6ec9629da982       no              qvbaeee0c10-2e
                                                                tapaeee0c10-2e
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump icmp -e -i tapaeee0c10-2e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tapaeee0c10-2e, link-type EN10MB (Ethernet), capture size 262144 bytes
18:10:23.246867 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2341, seq 413, length 64
18:10:23.247049 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2341, seq 413, length 64
18:10:24.246815 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.10.9: ICMP echo request, id 2341, seq 414, length 64
18:10:24.247005 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 2341, seq 414, length 64
```

2.4. Komunikace mezi VM ve stejné síti na odlišných compute node (east-west)



Vyjdeme z předchozího scénáře, kde už jsme se mnohé naučili a trochu situaci zkomplikujeme. Naše VM teď nebudou na stejném fyzickém serveru, ale na různých:

```
nova list --all-tenants 1 --tenant baa7096fe1d54571900c3758397e0939 --fields name,OS-EXT-SRV-ATTR:host,OS-EXT-SRV-ATTR:instance_name
+--------------------------------------+------------+-------------------------------------+-------------------------------+
| ID                                   | Name       | OS-EXT-SRV-ATTR: Host               | OS-EXT-SRV-ATTR: Instance Name |
+--------------------------------------+------------+-------------------------------------+-------------------------------+
| eb347271-dc5a-46cf-9150-0a7defffc6d1 | instance-1 | overcloud-novacompute0-vli5de2egecg | instance-0000010d             |
| 70d0662f-9c69-4d0b-99e7-2dde4e0494e8 | instance-2 | overcloud-novacompute1-c4ia2jfbd75d | instance-0000010e             |
| e1975422-a543-4ce4-be36-bce191816161 | instance-3 | overcloud-novacompute0-vli5de2egecg | instance-0000010f             |
+--------------------------------------+------------+-------------------------------------+-------------------------------+
```

2.4.1. Vstup do vSwitch br-int

V porovnání s předhozím scénářem je vše stejné až do okamžiku, kdy paket vstoupí do vSwitch a rozhodne se o jeho další dráze. Podívejme do forwardovacích tabulek kam tentokrát ukazuje destination MAC adresa.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int | grep fa:16:3e:fd:7f:88
   87     69   fa:16:3e:fd:7f:88    0
```

Co je interface číslo 87?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep 87
 ...
 87(patch-tun): addr:b2:5f:40:f0:2a:4f
 ...
```

Co to znamená? Destination MAC není na stejném compute node a paket máme poslat do speciálního interface, který propojuje integrační OVS bridge (br-int) s OVS, ve kterém se řeší práce s tunely (br-tun).

2.4.2. Příprava tunelů a odeslání do fyzické sítě

Jaké porty najdeme na virtuálním prvku br-tun?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show
...
Bridge br-tun
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.23"}
        Port "vxlan-0a000a0a"
            Interface "vxlan-0a000a0a"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.10"}
    ovs_version: "2.3.0"
```

Kromě propojovacího portu tady vidíme porty pro VXLAN tunely. V našem případě jde o komunikaci do 10.0.10.23, což je druhý compute node a také do 10.0.10.10, což je network node (v našem případě je součástí controller node). Klíč, tedy VXLAN VNI, je určován OVS pravidly.

Vytáhneme si ještě interní čísla portů:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:00009e4ffab46e48
 1(patch-int): addr:7a:c7:3a:cf:90:5e
 2(vxlan-0a000a0a): addr:ba:0c:97:69:99:7f
 5(vxlan-0a000a17): addr:8a:30:a7:83:71:08
 LOCAL(br-tun): addr:9e:4f:fa:b4:6e:48
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

2.4.3. OpenFlow pravidla v br-tun

Jak vypadá forwardovací pipeline v br-tun? Začněme výpisem vstupní tabulky, tedy té s ID 0:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=2979679.382s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=2137891.215s, table=0, n_packets=1540418, n_bytes=159465418, idle_age=0, hard_age=65534,
priority=1,in_port=5 actions=resubmit(,4)
 cookie=0x0, duration=2979677.434s, table=0, n_packets=9757051, n_bytes=701535045, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=2979663.060s, table=0, n_packets=36460, n_bytes=2664832, idle_age=1, hard_age=65534,
priority=1,in_port=2 actions=resubmit(,4)
```

Zvýrazněná řádka je ta, která se týká našich paketů (určitě uvidíte i zvětšující se counter). Říká, že co vstupuje do OVS portem patch-int (tedy provoz lokálních VM směřujících mimo tento compute node), bude dále zkoumáno v tabulce 1. Pojďme se na ní tedy podívat.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=1
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3022441.256s, table=1, n_packets=9965883, n_bytes=717582013, idle_age=0, hard_age=65534,
priority=0 actions=resubmit(,2)
 cookie=0x0, duration=129202.321s, table=1, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=1,dl_vlan=70,dl_src=fa:16:3e:42:d7:50 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
 cookie=0x0, duration=1920675.743s, table=1, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=1,dl_vlan=49,dl_src=fa:16:3e:72:b4:78 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
...
```

V této tabulce je mnoho operací týkajících se provozu, který bude směrovaný. To ovšem není náš případ, takže se na nás vztahuje zvýrazněné pravidlo (s nejnižší prioritou) a pošle switchovaný provoz do tabulky 2. Co najdeme v ní?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=2
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3022697.188s, table=2, n_packets=2839090, n_bytes=250753467, idle_age=0, hard_age=65534,
priority=0,dl_dst=00:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,20)
 cookie=0x0, duration=3022697.094s, table=2, n_packets=7130151, n_bytes=467126446, idle_age=0, hard_age=65534,
priority=0,dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,22)
```

Velmi jednoduché – pokud je to unicast, pošli do tabulky 20, pokud multicast, pošli do tabulky 22. Můžeme se tedy soustředit na tabulku 20. Víme, že pakety nám z lokální VM přicházejí s lokálně významným tagem, v našem případě to bylo 69. Ukažme si tedy specificky pravidla pro takový případ:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=20,dl_vlan=69
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=129582.625s, table=20, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:1f:87:98 actions=strip_vlan,set_tunnel:0x3f2,output:2
 cookie=0x0, duration=129582.725s, table=20, n_packets=37, n_bytes=3140, idle_age=42689, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:b2:3d:19 actions=strip_vlan,set_tunnel:0x3f2,output:2
 cookie=0x0, duration=84689.592s, table=20, n_packets=27959, n_bytes=2694902, idle_age=17753, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:fd:7f:88 actions=strip_vlan,set_tunnel:0x3f2,output:5
```

Zvýrazněná destination MAC je vám asi povědomá – ano, je to MAC naší cílové VM. Co tedy uděláme? Odstřihneme lokálně významný VLAN tag a pak zabalíme paket do VXLAN tunelu s číslem 0x3f2 a odešleme do virtuálního portu 5, což jak už víme je tunel do jiného compute node – připomeňme si to:

```
Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.23"}
...
5(vxlan-0a000a17): addr:8a:30:a7:83:71:08
```

2.4.4. Posíláme ven z compute node

Odchytíme si provoz na fyzickém portu:

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump -e -i eth0 -c 100 | grep -B1 192.168.10.9
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:40:53.190877 fc:15:b4:84:12:98 (oui Unknown) > 14:58:d0:d3:00:ee (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute0-vli5de2egecg.33236 > overcloud-NovaCompute1-c4ia2jfbd75d.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 >
192.168.10.9: ICMP echo request, id 2569, seq 916, length 64
18:40:53.191267 14:58:d0:d3:00:ee (oui Unknown) > fc:15:b4:84:12:98 (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute1-c4ia2jfbd75d.46874 > overcloud-NovaCompute0-vli5de2egecg.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.9 >
192.168.10.8: ICMP echo reply, id 2569, seq 916, length 64
```

Všimněte si v paketu pár detailů. VNI, tedy číslo VXLAN, nám krásně sedí (dekadických 1010 je hexa 3F2). Vnější obálka jde z jednoho compute node na druhý a uvnitř se veze původní paket mezi vnitřními adresami. Vnitřní IP a MAC adresy tak nejsou ve fyzické síti vidět, ta pracuje jen na úrovni vnějších identifikátorů, tedy MAC a IP adresy fyzických serverů.

2.4.5. Přijímáme na vstupu druhého compute node

Dorazil nám paket přes fyzickou síť do druhého fyzického serveru?

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# tcpdump -e -i eth0 -c 100 | grep -B1 192.168.10.9
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
18:50:18.235754 fc:15:b4:84:12:98 (oui Unknown) > 14:58:d0:d3:00:ee (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute0-vli5de2egecg.33236 > overcloud-NovaCompute1-c4ia2jfbd75d.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 >
192.168.10.9: ICMP echo request, id 2569, seq 1481, length 64
18:50:18.236188 14:58:d0:d3:00:ee (oui Unknown) > fc:15:b4:84:12:98 (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute1-c4ia2jfbd75d.46874 > overcloud-NovaCompute0-vli5de2egecg.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.9 >
192.168.10.8: ICMP echo reply, id 2569, seq 1481, length 64
```

Přišel v pořádku a dle očekávání uvnitř VXLAN tunelu.

2.4.6. Vstup do vSwitch br-tun příjemce

V odchozím compute node jsme v br-tun mohli vidět odebírání interního tagu a příbírání VXLAN VNI a enkapsulaci paketu. U přijímacího compute node logicky očekáváme opak. Nejprve si zjistíme, jaké máme porty a jejich ID.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-vsctl show
c8132496-677e-4596-b869-db490bbde09a
    Bridge br-tun
        Port "vxlan-0a000a0e"
            Interface "vxlan-0a000a0e"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.23", out_key=flow, remote_ip="10.0.10.14"}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a0a"
            Interface "vxlan-0a000a0a"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.23", out_key=flow, remote_ip="10.0.10.10"}
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
...

root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000bea8033fc743
 1(patch-int): addr:12:bb:16:22:89:94
 9(vxlan-0a000a0a): addr:ae:d1:7b:25:4a:9f
 10(vxlan-0a000a0e): addr:86:5f:dd:f8:52:34
 LOCAL(br-tun): addr:be:a8:03:3f:c7:43
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Z VXLAN paketu víme, že přišel z compute node 10.0.10.14, takže nás zajímá virtuální port vxlan-0a000a0e s ID 10. Druhý zajímavý port je samozřejmě patch-int, tedy ID 1, který vede do integračního bridge (br-int), kde už jsou naše VM.

2.4.7. OpenFlow pravidla v br-tun příjemce

Pipeline vždy začíná v tabulce 0 – jak to tam vypadá?

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
```

```
 cookie=0x0, duration=3217627.895s, table=0, n_packets=1, n_bytes=70, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=2334908.443s, table=0, n_packets=3086659, n_bytes=273190971, idle_age=0, hard_age=65534,
priority=1,in_port=10 actions=resubmit(,4)
 cookie=0x0, duration=3217625.968s, table=0, n_packets=7881046, n_bytes=574586131, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=2338482.854s, table=0, n_packets=824, n_bytes=118581, idle_age=23628, hard_age=65534,
priority=1,in_port=9 actions=resubmit(,4)
```

Zvýrazněné pravidlo je to, které se vztahuje na náš paket. Skáčeme tedy do tabulky 4:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=4
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=2323822.652s, table=4, n_packets=194, n_bytes=30402, idle_age=41121, hard_age=65534,
priority=1,tun_id=0x3f8 actions=mod_vlan_vid:13,resubmit(,9)
 cookie=0x0, duration=2323818.137s, table=4, n_packets=43, n_bytes=2918, idle_age=1246, hard_age=65534,
priority=1,tun_id=0x3f7 actions=mod_vlan_vid:14,resubmit(,9)
 cookie=0x0, duration=283537.088s, table=4, n_packets=30573, n_bytes=2956761, idle_age=0, hard_age=65534,
priority=1,tun_id=0x3f2 actions=mod_vlan_vid:32,resubmit(,9)
 cookie=0x0, duration=1571703.627s, table=4, n_packets=2783687, n_bytes=243107744, idle_age=0, hard_age=65534,
priority=1,tun_id=0x3fb actions=mod_vlan_vid:22,resubmit(,9)
 cookie=0x0, duration=283531.497s, table=4, n_packets=24, n_bytes=2088, idle_age=647, hard_age=65534,
priority=1,tun_id=0x3f3 actions=mod_vlan_vid:33,resubmit(,9)
 cookie=0x0, duration=1578175.630s, table=4, n_packets=962, n_bytes=107040, idle_age=904, hard_age=65534,
priority=1,tun_id=0x3f6 actions=mod_vlan_vid:19,resubmit(,9)
 cookie=0x0, duration=3217700.707s, table=4, n_packets=152, n_bytes=10872, idle_age=65534, hard_age=65534, priority=0
actions=drop
```

Tato je vemi zajímavá. Na základě VXLAN VNI přiřazuje lokální tag – všimněte si, že ten (na rozdíl od VNI) má skutečně jen lokální význam uvnitř daného compute node. Přiřadili jsme tedy tag 32 a skáčeme do tabulky 9.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=9
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3217847.919s, table=9, n_packets=3087116, n_bytes=273274962, idle_age=0, hard_age=65534,
priority=0 actions=resubmit(,10)
 cookie=0x0, duration=3217848.293s, table=9, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=1,dl_src=fa:16:3f:5d:a5:3f actions=output:1
 cookie=0x0, duration=3217848.106s, table=9, n_packets=994, n_bytes=117609, idle_age=742, hard_age=65534,
priority=1,dl_src=fa:16:3f:9e:30:0c actions=output:1
```

Nic zásadního – odchytávají se pouze specifické MAC, ale ty nesouvisí s naším paketem (týkají spíše situací se směrováním). Jednoduše tedy skáčeme do tabulky 10.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=10
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3217976.164s, table=10, n_packets=3087488, n_bytes=273308532, idle_age=1, hard_age=65534,
priority=1
actions=learn(table=20,hard_timeout=300,priority=1,NXM_OF_VLAN_TCI[0..11],NXM_OF_ETH_DST[]=NXM_OF_ETH_SRC[],load:0-
>NXM_OF_VLAN_TCI[],load:NXM_NX_TUN_ID[]->NXM_NX_TUN_ID[],output:NXM_OF_IN_PORT[]),output:1
```

Vytáhneme si z paketu nějaké informace a ty si uložíme a posíláme paket ven portem 1, tedy do patch mezi br-tun a br-int.

2.4.8. Vstup do vSwitch br-int příjemce

Blížíme se k cílové VM a vstupujeme do br-int. Nejprve se podívejme na důležité porty. To bude jednak vstupní port (patch mezi br-tun a br-int) a také porty v cílové interní VLAN (z předchozího zkoumání víme, že je to ID 32).

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-vsctl show |  grep -A1 'tag: 32'
```

```
            tag: 32
            Interface "qvoaeee0c10-2e"
--
            tag: 32
            Interface "qr-9ab15d1e-3d"
```

Vidíme dva porty – zajímá nás QVO, tedy port směřující k VM. Zjistíme si ID portů:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl show br-int | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:000032cb98c22142
 1(int-br-svc): addr:2e:5b:31:74:67:44
 2(patch-tun): addr:36:64:57:50:fd:d1
 27(qvoba52a47a-2c): addr:3e:ae:5e:13:d9:9c
 28(qr-0d6c7979-4e): addr:00:00:00:00:00:00
 29(qr-2ce50678-96): addr:00:00:00:00:00:00
 30(qvo4de718f9-e1): addr:92:eb:02:e5:28:36
 42(qr-eb3e9a50-e0): addr:00:00:00:00:00:00
 43(qvo4882de2e-f0): addr:be:e5:e3:81:ca:3f
 47(qr-1da629e3-59): addr:00:00:00:00:00:00
 48(qvo24cb5e56-2e): addr:62:0d:de:77:41:ee
 49(qvo9f070d74-37): addr:9e:62:f8:31:71:f2
 69(qr-9ab15d1e-3d): addr:00:00:00:00:00:00
 70(qr-f01425f2-58): addr:00:00:00:00:00:00
 71(qvoaeee0c10-2e): addr:e6:be:fe:d4:c2:45
 LOCAL(br-int): addr:32:cb:98:c2:21:42
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Prohlédněme si tedy OpenFlow pravidla.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3235615.981s, table=0, n_packets=1016, n_bytes=121842, idle_age=608, hard_age=65534,
priority=2,in_port=2,dl_src=fa:16:3f:9e:30:0c actions=resubmit(,1)
 cookie=0x0, duration=3235616.167s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=2,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
 cookie=0x0, duration=3235616.280s, table=0, n_packets=12456333, n_bytes=1552895815, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2201144.250s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=31 actions=drop
 cookie=0x0, duration=1589613.591s, table=0, n_packets=3928327, n_bytes=730610264, idle_age=0, hard_age=65534,
priority=3,in_port=1,vlan_tci=0x0000 actions=mod_vlan_vid:23,NORMAL
```

Zvýrazněně je řádka, která se týká našeho paketu. Akce je NORMAL, tedy použije se běžné chování switche. Prohlédněme si tedy forwardovací tabulku.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-appctl fdb/show br-int
 port  VLAN  MAC                Age
    1    23  38:22:d6:e9:92:35    1
   49    23  fa:16:3e:b3:43:cd    0
    1    23  00:25:6e:0a:f6:07    0
    2    22  fa:16:3e:da:9d:c7    0
    2    32  fa:16:3e:21:cf:75    0
   48    22  fa:16:3e:d1:ab:81    0
   47    22  fa:16:3e:03:42:5e    0
   71    32  fa:16:3e:fd:7f:88    0
```

Máme jasno – paket odejde do portu 71, tedy qvoaeee0c10-2e.

2.4.9. Konec cesty

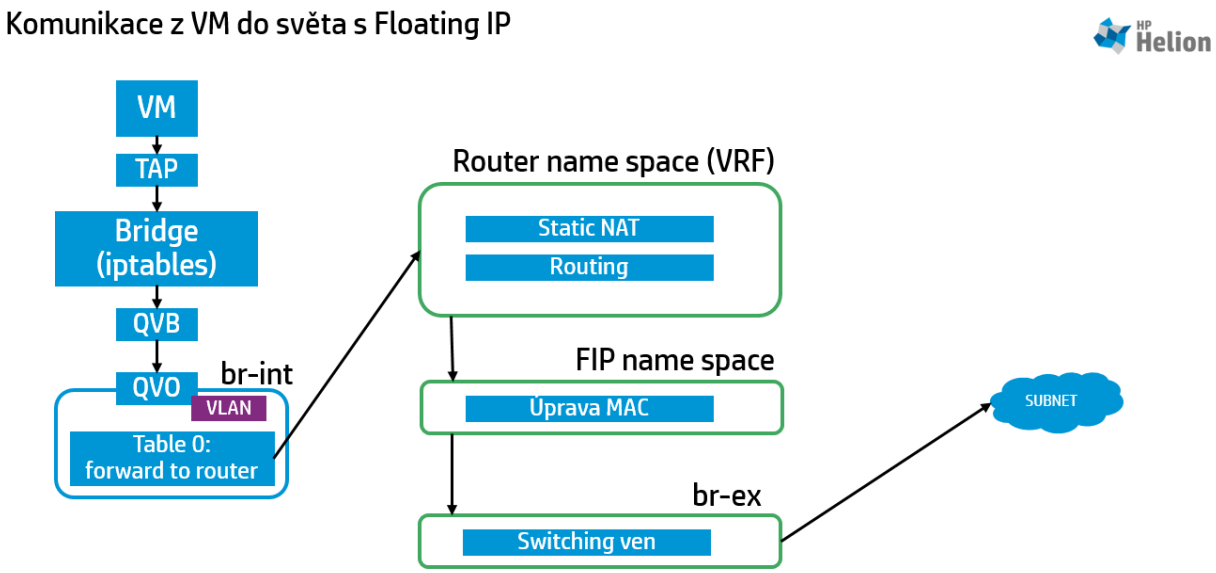
Z prvního kroku víme, že VM se v našem compute node jmenuje instance-0000010e a také už víme, že paket k ní prochází přes qvoaeee0c10-2e a qvbaeee0c10-2e. Pojďme kruh uzavřít.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# virsh dumpxml instance-0000010e | grep -A 7 "<interface"
    <interface type='bridge'>
      <mac address='fa:16:3e:fd:7f:88'/>
      <source bridge='qbraeee0c10-2e'/>
      <target dev='tapaeee0c10-2e'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>

root@overcloud-novacompute1-c4ia2jfbd75d:~# brctl show qbraeee0c10-2e
bridge name     bridge id               STP enabled     interfaces
qbraeee0c10-2e          8000.1268fea17497       no              qvbaeee0c10-2e
                                                                tapaeee0c10-2e
```

2.5. Komunikace ven s Floating IP (north-south)

V následujícím scénáři řešíme situaci, kdy VM komunikuje s reálnou sítí, například intranetem nebo Internetem a má přiřazenu floating IP, tedy externí identitu (scénář, kdy využívá SNAT, najdete v jiné kapitole). Zapneme ve VM ping ven a pustíme se do toho.



2.5.1. Pakety odcházející z VM

Stejně jako vždy na začátku se můžeme podívat na pakety tak, jak opouštějí VM.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump icmp -e -i tap425fe781-d3
listening on tap425fe781-d3, link-type EN10MB (Ethernet), capture size 262144 bytes
04:10:24.570674 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 172.16.2.1: ICMP echo request, id 3481, seq 99, length 64
04:10:24.571046 fa:16:3e:07:de:20 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.1 > 192.168.10.8: ICMP echo reply, id 3481, seq 99, length 64
04:10:25.570787 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 172.16.2.1: ICMP echo request, id 3481, seq 100, length 64
04:10:25.574141 fa:16:3e:07:de:20 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.1 > 192.168.10.8: ICMP echo reply, id 3481, seq 100, length 64
```

VM odesílá pakety na MAC adresu výchozí brány (jak později uvidíme tu pro ni představuje DVR, tedy distribuvaný router) a paket směřuje z vnitřní sítě do externí sítě.

2.5.2. Vstup do vSwitch br-int

V předchozích částech už jsme se naučili jak se aplikují iptables apod., takže přejděme rovnou ke vstupu paketu do vSwitche br-int. Stejně jako v předchozích případech si zjistíme tagy a porty.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 qvo425fe781-d3
        Port "qvo425fe781-d3"
            tag: 69
            Interface "qvo425fe781-d3"
```

V případě ID portů už nás ale budou zajímat i další typy portů.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000ce321315ab4c
 9(qr-0fd4fdbb-1f): addr:18:02:00:00:00:00
 25(qr-d6401d5a-66): addr:0c:02:00:00:00:00
 31(qvo418123be-b2): addr:52:4b:80:7d:58:8c
 32(qr-07443585-d6): addr:17:02:00:00:00:00
 33(qvof38bf3ec-47): addr:86:04:46:d0:2c:f0
 42(qr-c22cbd01-75): addr:17:02:00:00:00:00
 86(int-br-svc): addr:22:87:3d:89:2f:82
 87(patch-tun): addr:b2:5f:40:f0:2a:4f
 101(qvo455154db-6d): addr:42:aa:05:e1:5d:63
 102(qr-eb3e9a50-e0): addr:00:00:00:00:00:00
 103(qr-07f82580-15): addr:00:00:00:00:00:00
 112(qvo3175a674-3d): addr:be:23:5c:54:b5:a7
 113(qr-0d6c7979-4e): addr:00:00:00:00:00:00
 114(qr-2ce50678-96): addr:00:00:00:00:00:00
 151(qvo10089ae3-32): addr:0a:ba:41:10:72:7f
 152(qvoc3a85cae-52): addr:ce:a1:a7:9e:e5:cd
 153(qr-1e3dab95-19): addr:00:00:00:00:00:00
 166(qr-1da629e3-59): addr:00:00:00:00:00:00
 167(qvo78e94ad3-df): addr:76:27:60:47:45:8a
 168(qvo74546d7c-2c): addr:1a:3e:96:73:0a:ff
 169(qvo9154d28b-9d): addr:76:3d:cb:47:7c:85
 170(qvo6a03be8c-d2): addr:ee:62:38:97:6b:10
 171(qvof79487b0-ac): addr:aa:fb:9b:6a:da:05
 172(qvodc354bde-fd): addr:ce:fc:e9:ac:35:95
 173(qvo541b2834-37): addr:aa:b3:d4:a3:06:e6
 174(qvoc690710d-30): addr:6a:5c:9e:be:e1:e4
 198(qr-9c031a88-25): addr:00:00:00:00:00:00
 199(qr-c88ae80f-00): addr:00:00:00:00:00:00
 200(qvo0eada35b-39): addr:32:8d:ef:bd:16:c0
 201(qvo7b4b5a9a-4a): addr:82:6d:08:90:dd:6b
 202(qvoee91e31e-4a): addr:3a:3a:c0:68:8c:ab
 203(qvoced9aa79-fb): addr:52:3a:e9:18:60:1f
 204(qvodcb8ef62-ca): addr:26:7b:5f:f7:40:78
 205(qvoe33efd67-9a): addr:8e:c1:78:83:ef:24
 206(qvo8b4872d1-9f): addr:d6:7d:5b:85:5a:ad
 207(qvo314895e5-06): addr:02:5b:f4:44:dd:a2
 208(qvoc37e5970-63): addr:06:d4:42:b6:5a:88
 209(qvo8d7a5064-99): addr:12:a2:34:47:27:63
 210(qvoeea19d51-97): addr:fa:93:ae:ce:e6:47
 211(qvo425fe781-d3): addr:da:04:37:a2:8a:f6
 212(qr-9ab15d1e-3d): addr:00:00:00:00:00:00
 213(qr-f01425f2-58): addr:00:00:00:00:00:00
 214(qvob2018738-8a): addr:42:c5:bb:df:11:35
 216(qr-36fc3a6f-01): addr:00:00:00:00:00:00
 217(qvoa815917d-d3): addr:12:2f:86:63:df:bb
 218(qvof01a05ec-d2): addr:4e:14:99:95:de:4b
 LOCAL(br-int): addr:ce:32:13:15:ab:4c
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Označené jsou porty, které pro nás budou později významné.

2.5.3. OpenFlow pravidla v br-int

Začneme jako vždy tabulkou 0.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3404481.844s, table=0, n_packets=2, n_bytes=220, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
 cookie=0x0, duration=3404482.034s, table=0, n_packets=1661, n_bytes=168724, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=3404482.151s, table=0, n_packets=66559317, n_bytes=10311572170, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=1799411.686s, table=0, n_packets=11402766, n_bytes=2564330898, idle_age=0, hard_age=65534,
priority=3,in_port=86,vlan_tci=0x0000 actions=mod_vlan_vid:57,NORMAL
```

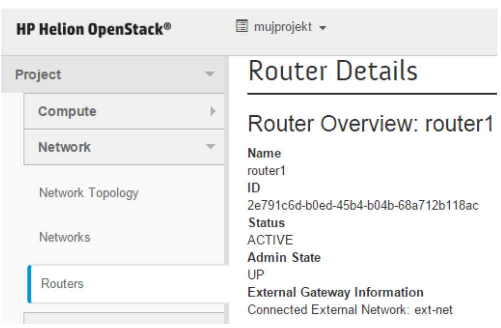
Stejně jako v předchozích případech je aplikováno pravidlo NORMAL, provedeme tedy běžný lookup na základě destination MAC. Hledejme cílovou MAC tak, jak jsme ji našli v zachycených ICMP paketech.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int | grep fa:16:3e:07:de:20
  212    69  fa:16:3e:07:de:20    1
```

Z předchozího výpisu tedy víme, že odchozím portem je qr-9ab15d1e-3d. Písmenko r označuje router – další zpracování tedy převezme virtuální router.

2.5.4. Router

Nejprve si zjistíme ID použitého routeru – například z GUI:



nebo z CLI

```
root@helion-ProLiant-DL380-Gen9:~# neutron router-list
+--------------------------------------+---------+----------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--+
| id                                   | name    | external_gateway_info
|
+--------------------------------------+---------+----------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--+
| 2e791c6d-b0ed-45b4-b04b-68a712b118ac | router1 | {"network_id": "3a5b5cd4-0c4b-4bc3-b44e-826c7b19556e",
"enable_snat": true, "external_fixed_ips": [{"subnet_id": "e3be37fb-1ced-432f-950c-99b887bb52c2", "ip_address":
"172.16.2.157"}]} |
+--------------------------------------+---------+----------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--+
```

Protože vnitřní adresy se mohou mezi projekty překrývat sedí každý router ve svém vlastním name space. Ten vyhledáme podle ID routeru:

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns | grep 2e791c6d-b0ed-45b4-b04b-68a712b118ac
qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac
```

Name space je tedy řetězec „qrouter-“ a za ním ID routeru. Jaké IP interfaces se v tomto name space nacházejí?

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: rfp-2e791c6d-b: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 8a:26:b3:8e:eb:68 brd ff:ff:ff:ff:ff:ff
    inet 169.254.30.210/31 scope global rfp-2e791c6d-b
       valid_lft forever preferred_lft forever
    inet 172.16.2.3/32 brd 172.16.2.3 scope global rfp-2e791c6d-b
       valid_lft forever preferred_lft forever
    inet6 fe80::8826:b3ff:fe8e:eb68/64 scope link
       valid_lft forever preferred_lft forever
632: qr-9ab15d1e-3d: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:07:de:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global qr-9ab15d1e-3d
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe07:de20/64 scope link
       valid_lft forever preferred_lft forever
634: qr-f01425f2-58: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:42:d7:50 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.1/24 brd 192.168.20.255 scope global qr-f01425f2-58
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe42:d750/64 scope link
       valid_lft forever preferred_lft forever
```

632 a 634 jsou IP rozhraní routeru pro směrování těchto dvou subnetů. 2 je pro floating IP a všimněte si, že má přiřazenu právě floating iP, které používáte. Směrování je uděláno tak, že výchozí brána vede do rfp-2e791c6d-b, tedy do floating IP name space. Opět je použitu vEth páru – z rfp* jdeme fo fpr*.

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip rule list
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
32797:  from 192.168.10.8 lookup 16
3232238081:     from 192.168.10.1/24 lookup 3232238081
3232238081:     from 192.168.10.1/24 lookup 3232238081
3232240641:     from 192.168.20.1/24 lookup 3232240641
3232240641:     from 192.168.20.1/24 lookup 3232240641
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip route show
table 16
default via 169.254.30.211 dev rfp-2e791c6d-b
```

Nicméně v rámci tohoto name space jsme ještě provedli NAT (Floating IP) a směrování. Podívejme se nejprve na NAT pravidla uvnitř name space:

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac iptables --
table nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
```

```
neutron-l3-agent-PREROUTING  all  --  anywhere             anywhere

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-OUTPUT  all  --  anywhere             anywhere

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-POSTROUTING  all  --  anywhere             anywhere
neutron-postrouting-bottom  all  --  anywhere             anywhere

Chain neutron-l3-agent-OUTPUT (1 references)
target     prot opt source               destination
DNAT       all  --  anywhere             172.16.2.3           to:192.168.10.8

Chain neutron-l3-agent-POSTROUTING (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ! ctstate DNAT

Chain neutron-l3-agent-PREROUTING (1 references)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             169.254.169.254      tcp dpt:http redir ports 9697
DNAT       all  --  anywhere             172.16.2.3           to:192.168.10.8

Chain neutron-l3-agent-float-snat (1 references)
target     prot opt source               destination
SNAT       all  --  192.168.10.8         anywhere             to:172.16.2.3

Chain neutron-l3-agent-snat (1 references)
target     prot opt source               destination
neutron-l3-agent-float-snat  all  --  anywhere             anywhere

Chain neutron-postrouting-bottom (1 references)
target     prot opt source               destination
neutron-l3-agent-snat  all  --  anywhere             anywhere
```

A jak teď vypadá komunikace?

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac tcpdump icmp -e -l -i rfp-2e791c6d-b
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on rfp-2e791c6d-b, link-type EN10MB (Ethernet), capture size 262144 bytes
07:14:35.522112 8a:26:b3:8e:eb:68 (oui Unknown) > 7a:91:ca:6c:2a:27 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.3 > 172.16.2.1: ICMP echo request, id 3595, seq 2429, length 64
07:14:35.522392 7a:91:ca:6c:2a:27 (oui Unknown) > 8a:26:b3:8e:eb:68 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.1 > 172.16.2.3: ICMP echo reply, id 3595, seq 2429, length 64
07:14:36.522920 8a:26:b3:8e:eb:68 (oui Unknown) > 7a:91:ca:6c:2a:27 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.3 > 172.16.2.1: ICMP echo request, id 3595, seq 2430, length 64
07:14:36.524196 7a:91:ca:6c:2a:27 (oui Unknown) > 8a:26:b3:8e:eb:68 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.1 > 172.16.2.3: ICMP echo reply, id 3595, seq 2430, length 64
```

2.5.5. Floating IP name space

V této tabulce dočistíme některé věci – tak například zdrojová MAC adresa bude pro compute node společná pro různé floating IP (jde z jednoho fyzického místa, tedy compute node).

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec fip-3a5b5cd4-0c4b-4bc3-b44e-826c7b19556e ip route
default via 172.16.0.1 dev fg-e19b2a1d-c6
169.254.30.12/31 dev fpr-fa6247f4-1  proto kernel  scope link  src 169.254.30.13
169.254.30.174/31 dev fpr-2bba3a8f-0  proto kernel  scope link  src 169.254.30.175
169.254.30.184/31 dev fpr-c8b81d14-7  proto kernel  scope link  src 169.254.30.185
```

```
169.254.30.210/31 dev fpr-2e791c6d-b  proto kernel  scope link  src 169.254.30.211
169.254.31.220/31 dev fpr-d9457e7c-7  proto kernel  scope link  src 169.254.31.221
172.16.0.0/16 dev fg-e19b2a1d-c6  proto kernel  scope link  src 172.16.2.80
172.16.2.3 via 169.254.30.210 dev fpr-2e791c6d-b
172.16.2.35 via 169.254.31.220 dev fpr-d9457e7c-7
172.16.2.69 via 169.254.30.12 dev fpr-fa6247f4-1
172.16.2.79 via 169.254.30.174 dev fpr-2bba3a8f-0
172.16.2.138 via 169.254.30.184 dev fpr-c8b81d14-7
172.16.2.139 via 169.254.30.184 dev fpr-c8b81d14-7
172.16.2.141 via 169.254.30.184 dev fpr-c8b81d14-7
172.16.2.142 via 169.254.30.184 dev fpr-c8b81d14-7
172.16.2.161 via 169.254.31.220 dev fpr-d9457e7c-7
```

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec fip-3a5b5cd4-0c4b-4bc3-b44e-826c7b19556e tcpdump icmp -e -l
-i fg-e19b2a1d-c6
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on fg-e19b2a1d-c6, link-type EN10MB (Ethernet), capture size 262144 bytes
07:17:20.711120 fa:16:3e:56:e4:37 (oui Unknown) > fa:16:3e:81:c5:ee (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.3 > 172.16.2.1: ICMP echo request, id 3595, seq 2594, length 64
07:17:20.728283 fa:16:3e:81:c5:ee (oui Unknown) > fa:16:3e:56:e4:37 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.1 > 172.16.2.3: ICMP echo reply, id 3595, seq 2594, length 64
07:17:21.713010 fa:16:3e:56:e4:37 (oui Unknown) > fa:16:3e:81:c5:ee (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.3 > 172.16.2.1: ICMP echo request, id 3595, seq 2595, length 64
07:17:21.713588 fa:16:3e:81:c5:ee (oui Unknown) > fa:16:3e:56:e4:37 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.1 > 172.16.2.3: ICMP echo reply, id 3595, seq 2595, length 64
```

Ze směrovací tabulky je vidět, že provoz dále poteče do portu fg-e19b2a1d-c6 a ten najdeme na vSwitch br-ext.

2.5.6. Posíláme ven z compute node

S paketem jsme provedli všechny potřebné manipulace a odesíláme ho do externí sítě. To je realizováno ve vSwitch br-ext. V našem případě je venkovní provoz na stejném fyzickém NIC, ale je označen VLAN 172 (externích subnetů a tedy i VLAN můžete mít víc, je možné i odesílání jiným fyzickým portem). Podívejme se na porty switche:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A100 br-ex
    Bridge br-ex
        Port "fg-e19b2a1d-c6"
            Interface "fg-e19b2a1d-c6"
                type: internal
        Port br-ex
            Interface br-ex
                type: internal
        Port "vlan172"
            Interface "vlan172"
```

Čísla portů:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-ex | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000fc15b4841298
 1(vlan172): addr:fc:15:b4:84:12:98
 4(fg-e19b2a1d-c6): addr:00:00:00:00:00:00
 LOCAL(br-ex): addr:fc:15:b4:84:12:98
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

A OpenFlow pravidla:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-ex
NXST_FLOW reply (xid=0x4):
```

```
 cookie=0x0, duration=3457032.656s, table=0, n_packets=1839631, n_bytes=1088169898, idle_age=0, hard_age=65534,
priority=0 actions=NORMAL
```

… tedy jinak řečeno nic neřešme a použijme normální switching. Jaká je tedy forwardovací tabulka?

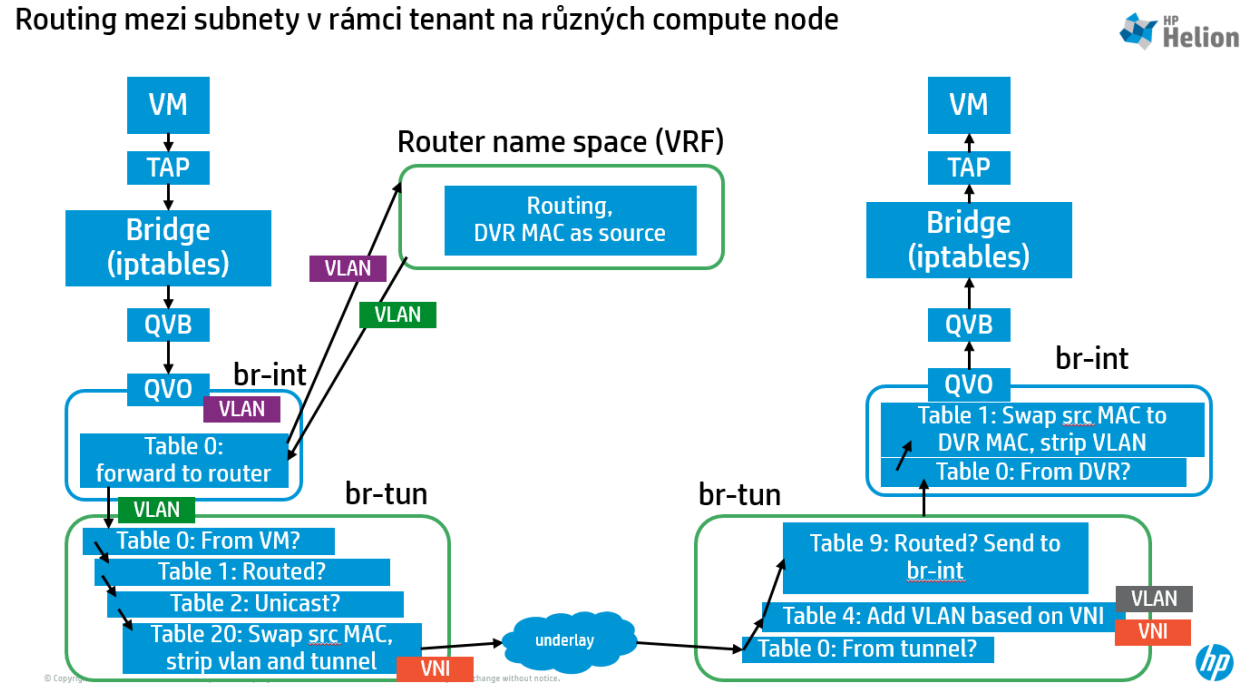
```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-ex
 port  VLAN  MAC                Age
    1     0  fa:16:3e:81:c5:ee    1
    4     0  fa:16:3e:56:e4:37    1
    1     0  38:22:d6:e9:92:23    1
```

Všimněte si, že MAC adresa VM není vidět v okolním světě a neodčerpává tedy zdroje síťových prostředků.

2.6. Routing mezi subnety (east-west)

Následující scénář se týká distribuovaného routingu mezi dvěma subnety a dvěma VM v rámci jednoho tenanta. V případě, že se nachází na stejném fyzickém serveru (compute node) proběhne tento proces lokálně. My se zaměříme na případ, kdy jsou VM na různých compute node (první situace je tak podmnožinou). Na jaké VM se tedy zaměříme ve zkoumání dráhy paketu?

```
root@helion-ProLiant-DL380-Gen9:~# nova list --all-tenants 1 --tenant baa7096fe1d54571900c3758397e0939 --fields name,OS-EXT-SRV-ATTR:host,OS-EXT-SRV-
ATTR:instance_name,Networks
+--------------------------------------+------------+-------------------------------+-------------------------------+-----------------------------------------+
| ID                                   | Name       | OS-EXT-SRV-ATTR: Host         | OS-EXT-SRV-ATTR: Instance Name | Networks                                |
+--------------------------------------+------------+-------------------------------+-------------------------------+-----------------------------------------+
| eb347271-dc5a-46cf-9150-0a7defffc6d1 | instance-1 | overcloud-novacompute0-vli5de2egecg | instance-0000010d             | net1=192.168.10.8, 172.16.2.3           |
| 70d0662f-9c69-4d0b-99e7-2dde4e0494e8 | instance-2 | overcloud-novacompute0-vli5de2egecg | instance-0000010e             | net1=192.168.10.9                       |
| e1975422-a543-4ce4-be36-bce191816161 | instance-3 | overcloud-novacompute1-c4ia2jfbd75d | instance-0000010f             | net2=192.168.20.3                       |
+--------------------------------------+------------+-------------------------------+-------------------------------+-----------------------------------------+
```



2.6.1. Pakety odcházející z VM

Stejně jako v předchozích kapitolách si najděte správný interface a podívejme se na pakety.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump icmp -e -i tap425fe781-d3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap425fe781-d3, link-type EN10MB (Ethernet), capture size 262144 bytes
```

```
07:09:31.252917 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4636, seq 262, length 64
07:09:31.253447 fa:16:3e:07:de:20 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.20.3 > 192.168.10.8: ICMP echo reply, id 4636, seq 262, length 64
07:09:32.254094 fa:16:3e:21:cf:75 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4636, seq 263, length 64
07:09:32.254585 fa:16:3e:07:de:20 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.20.3 > 192.168.10.8: ICMP echo reply, id 4636, seq 263, length 64
```

2.6.2. Vstup do vSwitch br-int

Nejprve si zjistíme, tak jako v předchozích případech, vstupní port:

```
root@overcloud-novacompute0-vli5de2egecg:~# brctl show | grep -B1 tap425fe781-d3
qbr425fe781-d3          8000.ca1a7962d69c       no              qvb425fe781-d3
                                                                tap425fe781-d3
```

Z bridge (pro iptables) tedy odcházíme portem qvb425fe781-d3, takže druhé ústí veth páru, tedy vstupní bod do br-int, je qvo425fe781-d3.

Jaký je aplikován vnitřní tag?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 qvo425fe781-d3
        Port "qvo425fe781-d3"
            tag: 69
            Interface "qvo425fe781-d3"
```

Zjistíme si ID portu:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep qvo425fe781-d3
 211(qvo425fe781-d3): addr:da:04:37:a2:8a:f6
```

Vypíšeme si OpenFlow pravidle v tabulce 0:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3847209.754s, table=0, n_packets=2, n_bytes=220, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
 cookie=0x0, duration=3847209.944s, table=0, n_packets=2341, n_bytes=235364, idle_age=0, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=3847210.061s, table=0, n_packets=88100169, n_bytes=13792305674, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2242139.596s, table=0, n_packets=16314557, n_bytes=4001739751, idle_age=0, hard_age=65534,
priority=3,in_port=86,vlan_tci=0x0000 actions=mod_vlan_vid:57,NORMAL
```

Podívejme se do forwarding tabulky a hledejmé cílovou MAC adresu:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int | grep fa:16:3e:07:de:20
  212     69   fa:16:3e:07:de:20    1
```

Co je to za port?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep '212('
 212(qr-9ab15d1e-3d): addr:00:00:00:00:00:00
```

Cílová MAC je tedy ve skutečnosti MAC výchozí brány, což je router – ten je distribuovaný, takže je součástí přímo našeho compute node. Pro směrování tedy není třeba posílat provoz někam jinam.

2.6.3. Router

Zjistímě si, stejně jako v předchozím případě, ID našeho routeru – třeba z CLI.

```
root@helion-ProLiant-DL380-Gen9:~# neutron router-list
```

```
+--------------------------------------+----------+-----------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--+
| id                                   | name     | external_gateway_info
|
+--------------------------------------+----------+-----------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--+
| 2e791c6d-b0ed-45b4-b04b-68a712b118ac | router1 | {"network_id": "3a5b5cd4-0c4b-4bc3-b44e-826c7b19556e",
"enable_snat": true, "external_fixed_ips": [{"subnet_id": "e3be37fb-1ced-432f-950c-99b887bb52c2", "ip_address":
"172.16.2.157"}]} |
+--------------------------------------+----------+-----------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--+
```

Protože vnitřní adresy se mohou mezi projekty překrývat sedí každý router ve svém vlastním name space. Ten vyhledáme podle ID routeru:

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns | grep 2e791c6d-b0ed-45b4-b04b-68a712b118ac
qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac
```

Name space je tedy řetězec „qrouter-" a za ním ID routeru. Jaké IP interfaces se v tomto name space nacházejí?

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: rfp-2e791c6d-b: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 8a:26:b3:8e:eb:68 brd ff:ff:ff:ff:ff:ff
    inet 169.254.30.210/31 scope global rfp-2e791c6d-b
       valid_lft forever preferred_lft forever
    inet 172.16.2.3/32 brd 172.16.2.3 scope global rfp-2e791c6d-b
       valid_lft forever preferred_lft forever
    inet6 fe80::8826:b3ff:fe8e:eb68/64 scope link
       valid_lft forever preferred_lft forever
632: qr-9ab15d1e-3d: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:07:de:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global qr-9ab15d1e-3d
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe07:de20/64 scope link
       valid_lft forever preferred_lft forever
634: qr-f01425f2-58: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:42:d7:50 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.1/24 brd 192.168.20.255 scope global qr-f01425f2-58
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe42:d750/64 scope link
       valid_lft forever preferred_lft forever
```

632 a 634 jsou IP rozhraní routeru pro směrování těchto dvou subnetů.

Jak vypadá směrovací tabulka tohoto namespace (tedy našeho routeru, v zásadě ekvivalentu síťového VRF)?

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip route
169.254.30.210/31 dev rfp-2e791c6d-b  proto kernel  scope link  src 169.254.30.210
192.168.10.0/24 dev qr-9ab15d1e-3d  proto kernel  scope link  src 192.168.10.1
192.168.20.0/24 dev qr-f01425f2-58  proto kernel  scope link  src 192.168.20.1
```

Cílová IP 192.168.20.3 tedy bude směrována na port qr-f01425f2-58.

Pojďme si prohlédnout provoz na tomto portu:

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac tcpdump icmp -e -l -i qr-f01425f2-58
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qr-f01425f2-58, link-type EN10MB (Ethernet), capture size 262144 bytes
07:24:59.917411 fa:16:3e:42:d7:50 (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4636, seq 1190, length 64
07:25:00.918608 fa:16:3e:42:d7:50 (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4636, seq 1191, length 64
07:25:01.919649 fa:16:3e:42:d7:50 (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4636, seq 1192, length 64
07:25:02.918974 fa:16:3e:42:d7:50 (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4636, seq 1193, length 64
```

Všimněte si, že došlo k routingu – tzn. IP vrstva se nezměnila, ale MAC adresy ano. Zdrojová MAC teď odpovídá předchozí destination MAC (tedy adrese routeru).

2.6.4. Vracíme se z routeru

Po směrování jsme vrátili paket do br-int, ale v jiné VLAN. V jaké?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 qr-f01425f2-58
        Port "qr-f01425f2-58"
            tag: 70
            Interface "qr-f01425f2-58"
                type: internal
```

Podívejme se tedy na switching tabulku v této VLAN.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int | grep '\s70\s'
  213    70  fa:16:3e:42:d7:50    1
```

Paket nemá žádnou vnitřní destinaci (cíl není na stejném compute node), odchází tedy portem patch-tun do vSwitche br-tun, který má na starost přípravu komunikace do vnějšího světa. Nicméně důležité je, že máme interní VLAN 70 – podle toho tento provoz poznáme v br-tun.

2.6.5. Posíláme ven z compute node

Jaké porty na br-tun najdeme a kam směřují?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A100 br-tun
    Bridge br-tun
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.23"}
        Port "vxlan-0a000a0a"
            Interface "vxlan-0a000a0a"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.10"}
    ovs_version: "2.3.0"
```

Zvýrazněný je vstupní port z br-int a odchozí VXLAN port směřující do druhého compute node. Jaká mají interní ID?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:00009e4ffab46e48
 1(patch-int): addr:7a:c7:3a:cf:90:5e
 2(vxlan-0a000a0a): addr:ba:0c:97:69:99:7f
 5(vxlan-0a000a17): addr:8a:30:a7:83:71:08
 LOCAL(br-tun): addr:9e:4f:fa:b4:6e:48
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Podívejme se na OpenFlow pravidla v tabulce 0.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3850104.326s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=3008316.159s, table=0, n_packets=2304460, n_bytes=244396605, idle_age=0, hard_age=65534,
priority=1,in_port=5 actions=resubmit(,4)
 cookie=0x0, duration=3850102.378s, table=0, n_packets=13380000, n_bytes=968571340, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=3850088.004s, table=0, n_packets=417011, n_bytes=85765290, idle_age=17, hard_age=65534,
priority=1,in_port=2 actions=resubmit(,4)
```

Podle pravidel skáčeme do tabulky 1 – tam je pravidel hodně, můžeme si odfiltrovat rovnou ta, která jsou specificky zaměřena na naši VLAN 70.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=1,dl_vlan=70
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=956990.900s, table=1, n_packets=3698, n_bytes=362404, idle_age=1, hard_age=65534,
priority=1,dl_vlan=70,dl_src=fa:16:3e:42:d7:50 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
 cookie=0x0, duration=956990.998s, table=1, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,dl_vlan=70,dl_dst=fa:16:3e:42:d7:50 actions=drop
 cookie=0x0, duration=956991.110s, table=1, n_packets=1, n_bytes=42, idle_age=65534, hard_age=65534,
priority=3,arp,dl_vlan=70,arp_tpa=192.168.20.1 actions=drop
```

Pravidla řeší ARP provoz a také filtrují to, co nebylo správně přeroutované (pokus o spoofing), tedy nemá správnou MAC adresu. Zvýrazněná je řádka, která odpovídá našemu paketu. Posílá další zpracování do tabulky 2 a současně mění zdrojovou MAC adresu. Proč? Aktuální zdrojová MAC odpovídá adrese distribuovaného routeru – a ta je všude stejná, tedy v různých lokalitách. V případě, že by paket nebyl zabalen do VXLAN (tedy při použití VLAN režimu), tak je to samozřejmě zásadní problém pro fyzickou síť. Nicméně i při izolaci VXLAN by se jednalo o nepříjemnou situaci pro vSwitche. Z toho důvodu DVR MAC adresu brány před odesláním změníme.

Posouváme se tedy do tabulky 2.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=2
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3863937.845s, table=2, n_packets=4589811, n_bytes=398514290, idle_age=0, hard_age=65534,
priority=0,dl_dst=00:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,20)
 cookie=0x0, duration=3863937.751s, table=2, n_packets=8852811, n_bytes=574861246, idle_age=0, hard_age=65534,
priority=0,dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,22)
```

Rozlišujeme unicast a broadcast/multicast. Náš unicast paket tedy bude dále zpracován v tabulce 20. Ta má ocela dost záznamů, tak se podívejme specificky na naší VLAN 70.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=20,dl_vlan=70
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=970765.333s, table=20, n_packets=107, n_bytes=11290, idle_age=32616, hard_age=65534,
priority=2,dl_vlan=70,dl_dst=fa:16:3e:af:39:1a actions=strip_vlan,set_tunnel:0x3f3,output:2
 cookie=0x0, duration=27175.460s, table=20, n_packets=8387, n_bytes=821926, idle_age=0,
priority=2,dl_vlan=70,dl_dst=fa:16:3e:e4:71:f4 actions=strip_vlan,set_tunnel:0x3f3,output:5
```

```
 cookie=0x0, duration=970779.898s, table=20, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,dl_vlan=70,dl_dst=fa:16:3e:c7:6e:85 actions=strip_vlan,set_tunnel:0x3f3,output:2
```

Co se děje? Interní VLAN odstřihováme a nasazujeme VXLAN VNI a odcházíme portem číslo 5, což jak už jsme zjistili je virtuální tunel port vxlan-0a000a17 do druhého compute node.

Prohlédněme si pakety opouštějící compute node.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump -e -i eth0 -c 200 | grep -B1 192.168.20.3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:01:21.853324 fc:15:b4:84:12:98 (oui Unknown) > 14:58:d0:d3:00:ee (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute0-vli5de2egecg.54153 > overcloud-NovaCompute1-c4ia2jfbd75d.4789: VXLAN, flags [I] (0x08),
vni 1011
fa:16:3f:9e:30:0c (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.8 >
192.168.20.3: ICMP echo request, id 4668, seq 890, length 64
12:01:21.853682 14:58:d0:d3:00:ee (oui Unknown) > fc:15:b4:84:12:98 (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute1-c4ia2jfbd75d.54618 > overcloud-NovaCompute0-vli5de2egecg.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3f:4d:1f:fb (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.20.3 >
192.168.10.8: ICMP echo reply, id 4668, seq 890, length 64
```

Přeroutovaný paket navíc s vyměněnou gateway MAC je zabalen do VXLAN a jde do druhého compute node.

2.6.6. Přijímáme na vstupu druhého compute node

V odchozím compute node jsme v br-tun mohli vidět odebírání interního tagu a příbírání VXLAN VNI a enkapsulaci paketu. O přijímacího compute node logicky očekáváme opak. Nejprve si zjistíme, jaké máme porty a jejich ID.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-vsctl show
c8132496-677e-4596-b869-db490bbde09a
    Bridge br-tun
        Port "vxlan-0a000a0e"
            Interface "vxlan-0a000a0e"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.23", out_key=flow, remote_ip="10.0.10.14"}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a0a"
            Interface "vxlan-0a000a0a"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.23", out_key=flow, remote_ip="10.0.10.10"}
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
...

root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000bea8033fc743
 1(patch-int): addr:12:bb:16:22:89:94
 9(vxlan-0a000a0a): addr:ae:d1:7b:25:4a:9f
 10(vxlan-0a000a0e): addr:86:5f:dd:f8:52:34
 LOCAL(br-tun): addr:be:a8:03:3f:c7:43
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Z VXLAN paketu víme, že přišel z compute node 10.0.10.14, takže nás zajímá virtuální port vxlan-0a000a0e s ID 10. Druhý zajímavý port je samozřejmě patch-int, tedy ID 1, který vede do integračního bridge (br-int), kde už jsou naše VM.

2.6.7. OpenFlow pravidla v přijímacím br-tun vSwitch

Jako vždy začneme tabulkou 0:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3905679.872s, table=0, n_packets=1, n_bytes=70, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=3022960.420s, table=0, n_packets=4189206, n_bytes=370256941, idle_age=0, hard_age=65534,
priority=1,in_port=10 actions=resubmit(,4)
 cookie=0x0, duration=3905677.945s, table=0, n_packets=9860505, n_bytes=726198632, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=3026534.831s, table=0, n_packets=15498, n_bytes=19925328, idle_age=12019, hard_age=65534,
priority=1,in_port=9 actions=resubmit(,4)
```

Paket z prvního compute node bude dále zpracován v tabulce 4:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=4
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3011937.845s, table=4, n_packets=254, n_bytes=39522, idle_age=26364, hard_age=65534,
priority=1,tun_id=0x3f8 actions=mod_vlan_vid:13,resubmit(,9)
 cookie=0x0, duration=3011933.330s, table=4, n_packets=52, n_bytes=3296, idle_age=63028, hard_age=65534,
priority=1,tun_id=0x3f7 actions=mod_vlan_vid:14,resubmit(,9)
 cookie=0x0, duration=971652.281s, table=4, n_packets=43563, n_bytes=4220670, idle_age=1478, hard_age=65534,
priority=1,tun_id=0x3f2 actions=mod_vlan_vid:32,resubmit(,9)
 cookie=0x0, duration=2259818.820s, table=4, n_packets=3864108, n_bytes=338015766, idle_age=0, hard_age=65534,
priority=1,tun_id=0x3fb actions=mod_vlan_vid:22,resubmit(,9)
 cookie=0x0, duration=971646.690s, table=4, n_packets=9300, n_bytes=909908, idle_age=0, hard_age=65534,
priority=1,tun_id=0x3f3 actions=mod_vlan_vid:33,resubmit(,9)
 cookie=0x0, duration=2266290.823s, table=4, n_packets=1025, n_bytes=115121, idle_age=12136, hard_age=65534,
priority=1,tun_id=0x3f6 actions=mod_vlan_vid:19,resubmit(,9)
 cookie=0x0, duration=519682.468s, table=4, n_packets=13, n_bytes=2062, idle_age=65534, hard_age=65534,
priority=1,tun_id=0x3e9 actions=mod_vlan_vid:37,resubmit(,9)
 cookie=0x0, duration=3905815.900s, table=4, n_packets=202, n_bytes=14684, idle_age=65534, hard_age=65534, priority=0
actions=drop
```

Zvýrazněné je naše pravidlo reagující na naší VXLAN VNI. Paketu bude přiřazeno interní VLAN ID 33 a bude zpracováno v tabulce 9.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-tun table=9
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3905918.292s, table=9, n_packets=4194896, n_bytes=389212645, idle_age=1, hard_age=65534,
priority=0 actions=resubmit(,10)
 cookie=0x0, duration=3905918.666s, table=9, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=1,dl_src=fa:16:3f:5d:a5:3f actions=output:1
 cookie=0x0, duration=3905918.479s, table=9, n_packets=10433, n_bytes=1053441, idle_age=0, hard_age=65534,
priority=1,dl_src=fa:16:3f:9e:30:0c actions=output:1
```

Zdrojová adresa je zachycena a proces zpracování v br-tun je u konce. Provoz posíláme do portu 1, tedy portu patch-int, který směřuje do vSwitche br-int.

2.6.8. OpenFlow pravidla v br-int

Jaké porty najdeme v br-int?

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl show br-int | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:000032cb98c22142
 1(int-br-svc): addr:2e:5b:31:74:67:44
 2(patch-tun): addr:36:64:57:50:fd:d1
 27(qvoba52a47a-2c): addr:3e:ae:5e:13:d9:9c
 28(qr-0d6c7979-4e): addr:00:00:00:00:00:00
 29(qr-2ce50678-96): addr:00:00:00:00:00:00
 30(qvo4de718f9-e1): addr:92:eb:02:e5:28:36
```

```
 42(qr-eb3e9a50-e0): addr:00:00:00:00:00:00
 43(qvo4882de2e-f0): addr:be:e5:e3:81:ca:3f
 47(qr-1da629e3-59): addr:00:00:00:00:00:00
 48(qvo24cb5e56-2e): addr:62:0d:de:77:41:ee
 49(qvo9f070d74-37): addr:9e:62:f8:31:71:f2
 69(qr-9ab15d1e-3d): addr:00:00:00:00:00:00
 70(qr-f01425f2-58): addr:00:00:00:00:00:00
 80(qr-07f82580-15): addr:00:00:00:00:00:00
 95(qvob2018738-8a): addr:c2:2a:b0:08:ab:16
 LOCAL(br-int): addr:32:cb:98:c2:21:42
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Podívejme se na pravidla při přijímání paketu. Začneme v tabulce 0:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3906220.429s, table=0, n_packets=10735, n_bytes=1083037, idle_age=0, hard_age=65534,
priority=2,in_port=2,dl_src=fa:16:3f:9e:30:0c actions=resubmit(,1)
 cookie=0x0, duration=3906220.615s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=2,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
 cookie=0x0, duration=3906220.728s, table=0, n_packets=15780439, n_bytes=2040720540, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2871748.698s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=31 actions=drop
 cookie=0x0, duration=2260218.039s, table=0, n_packets=5577670, n_bytes=1031770844, idle_age=0, hard_age=65534,
priority=3,in_port=1,vlan_tci=0x0000 actions=mod_vlan_vid:23,NORMAL
```

Zachytilo nás pravidlo zdrojové MAC adresy a dále bude zpracování probíhat v tabulce 1. Podíváme se specificky na záznamy pro interní VLAN ID 33:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# ovs-ofctl dump-flows br-int table=1,dl_vlan=33
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=28563.096s, table=1, n_packets=0, n_bytes=0, idle_age=28585,
priority=2,ip,dl_vlan=33,nw_dst=192.168.20.0/24 actions=strip_vlan,mod_dl_src:fa:16:3e:42:d7:50,output:95
 cookie=0x0, duration=28563.193s, table=1, n_packets=9773, n_bytes=957754, idle_age=0,
priority=4,dl_vlan=33,dl_dst=fa:16:3e:e4:71:f4 actions=strip_vlan,mod_dl_src:fa:16:3e:42:d7:50,output:95
```

Co se děje? Všimněte si, že modifikujeme zdrojovou MAC adresu zpět na adresu distribuovaného routeru! Vnitřek VM tak nepozná, že v průběhu transportu do pracovalo s jinou, než router MAC (z důvodů, které už jsme probrali).

2.6.9. Konec cesty

Můžeme se podívat v jakém stavu paket opouští br-int:

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# tcpdump icmp -e -l -i qvob2018738-8a
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvob2018738-8a, link-type EN10MB (Ethernet), capture size 262144 bytes
12:24:39.789423 fa:16:3e:42:d7:50 (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4668, seq 2287, length 64
12:24:39.789678 fa:16:3e:e4:71:f4 (oui Unknown) > fa:16:3e:42:d7:50 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.20.3 > 192.168.10.8: ICMP echo reply, id 4668, seq 2287, length 64
12:24:40.790656 fa:16:3e:42:d7:50 (oui Unknown) > fa:16:3e:e4:71:f4 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.8 > 192.168.20.3: ICMP echo request, id 4668, seq 2288, length 64
12:24:40.790895 fa:16:3e:e4:71:f4 (oui Unknown) > fa:16:3e:42:d7:50 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.20.3 > 192.168.10.8: ICMP echo reply, id 4668, seq 2288, length 64
```

Teď už jen projdeme bridgem pro iptables a dostáváme se docíle. Vstup do iptables bridge je tedy druhý konec qvob2018738-8a, čili qvbb2018738-8a.

```
root@overcloud-novacompute1-c4ia2jfbd75d:~# brctl show | grep -A1 qvbb2018738-8a
qbrb2018738-8a          8000.a6fb3300d5cd       no              qvbb2018738-8a
```

```
tapb2018738-8a
```

Do VM tedy vede port tapb2018738-8a. Pojďme kruh uzavřít – na začátku kapitoly jsme zjistili, že lokální název cílove VM je instance-0000010f. Ujistěme se, že je k ní přiřazen právě tento tap port.

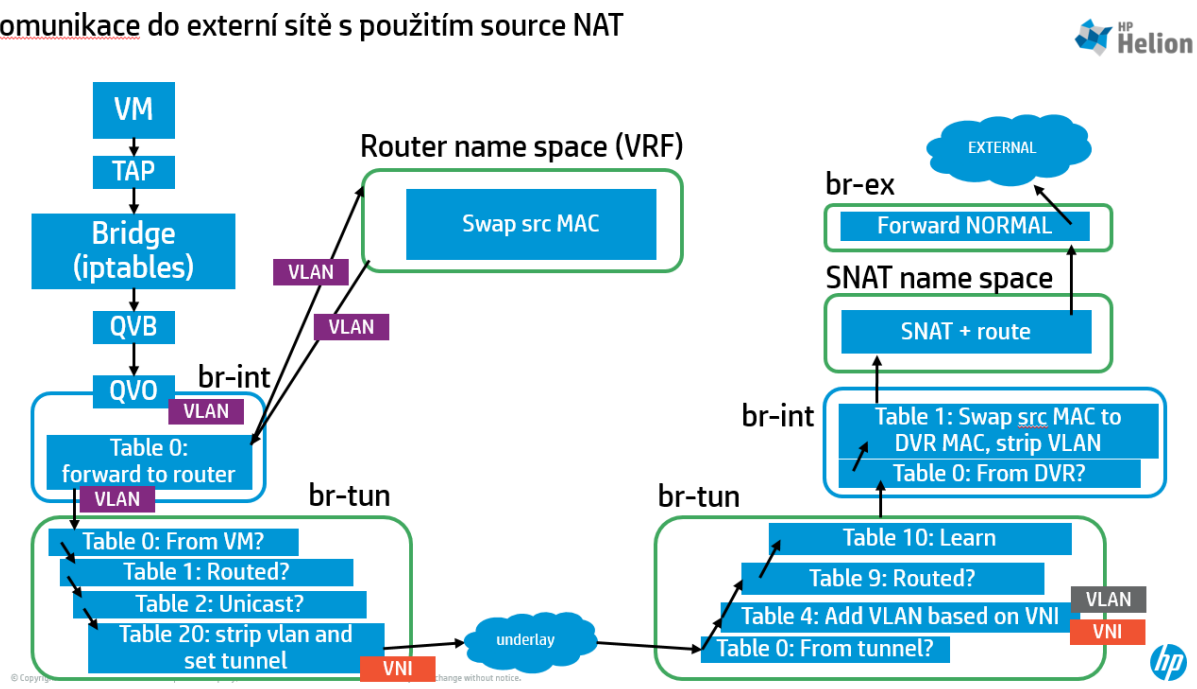
```
root@overcloud-novacompute1-c4ia2jfbd75d:~# virsh dumpxml instance-0000010f | grep -A 7 "<interface"
    <interface type='bridge'>
      <mac address='fa:16:3e:e4:71:f4'/>
      <source bridge='qbrb2018738-8a'/>
      <target dev='tapb2018738-8a'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
```

2.6.10. A cesta zpět?

Ta bude to samé obráceně – můžete tedy opakovat kroky v této kapitole pro opačný směr. Replay paket tedy bude zaroutován přímo tímto compute node, tedy tam, kde paket opouští VM. To je vlastnost distribuovaného routingu a přináší vysoký výkon a škálovatelnost.

2.7. Routing do externí sítě s využitím dynamické source NAT (north-south)

Instance, které mají být dostupné uživatelům, se vybavují Floating IP a mají tak externí identitu. Ta je směrována a NATována distribuovaným způsobem přímo v compute node, kde se VM nachází. Helion OpenStack ale také umožňuje, aby instance sdílely jednu externí adresu a docházelo k dynamickému NAT (PAT) – tedy tak jak připojujete svoje domácí počítače do Internetu. Smyslem je umožnit VM přístup na Internet nebo interní repozitář software za účelem stažení aktualizací apod. Tento typ provozu aktuálně nevyužívá distribuvaný router a je tedy centralizovaný (což má své výkonnostní limity, takže pro primární komunikaci na front end vždy využívejte Floating IP). Centrální router se v základním nastavení nachází na Helion OpenStack overcloud kontroleru.



S jakou instancí tedy budeme v této části pracovat?

```
root@helion-ProLiant-DL380-Gen9:~# nova list --all-tenants 1 --tenant baa7096fe1d54571900c3758397e0939 --fields name,OS-EXT-SRV-ATTR:host,OS-EXT-SRV-ATTR:instance_name,Networks
+--------------------------------------+------------+-------------------------------+-------------------------+-------------------------+
| ID                                   | Name       | OS-EXT-SRV-ATTR: Host         | OS-EXT-SRV-ATTR: Instance Name | Networks                |
+--------------------------------------+------------+-------------------------------+-------------------------+-------------------------+
| eb347271-dc5a-46cf-9150-0a7defffc6d1 | instance-1 | overcloud-novacompute0-vli5de2egecg | instance-0000010d       | net1=192.168.10.8, 172.16.2.3 |
| 70d0662f-9c69-4d0b-99e7-2dde4e0494e8 | instance-2 | overcloud-novacompute0-vli5de2egecg | instance-0000010e       | net1=192.168.10.9       |
| e1975422-a543-4ce4-be36-bce191816161 | instance-3 | overcloud-novacompute1-c4ia2jfbd75d | instance-0000010f       | net2=192.168.20.3       |
+--------------------------------------+------------+-------------------------------+-------------------------+-------------------------+
```

2.7.1. Pakety odcházející z VM

Zjistíme si tap interface a poslechneme provoz

```
root@overcloud-novacompute0-vli5de2egecg:~# virsh dumpxml instance-0000010e | grep -A 7 "<interface"
    <interface type='bridge'>
      <mac address='fa:16:3e:fd:7f:88'/>
      <source bridge='qbraeee0c10-2e'/>
      <target dev='tapaeee0c10-2e'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tapaeee0c10-2e, link-type EN10MB (Ethernet), capture size 262144 bytes
03:52:23.952227 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 10.0.98.1: ICMP echo request, id 5015, seq 70, length 64
03:52:23.972099 fa:16:3e:1f:87:98 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 10.0.98.1 > 192.168.10.9: ICMP echo reply, id 5015, seq 70, length 64
03:52:24.953261 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 10.0.98.1: ICMP echo request, id 5015, seq 71, length 64
03:52:24.960822 fa:16:3e:1f:87:98 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length
98: 10.0.98.1 > 192.168.10.9: ICMP echo reply, id 5015, seq 71, length 64
```

Máme tady tedy provoz směřující do externí sítě.

2.7.2. Vstup do vSwitch br-int

Nejprve si zjistíme, tak jako v předchozích případech, vstupní port:

```
root@overcloud-novacompute0-vli5de2egecg:~# brctl show | grep -B1 tapaeee0c10-2e
qbraeee0c10-2e          8000.9efc47859395       no              qvbaeee0c10-2e
                                                                tapaeee0c10-2e
```

Z bridge (pro iptables) tedy odcházíme portem qvbaeee0c10-2e, takže druhé ústí veth páru, tedy vstupní bod do br-int, je qvbaeee0c10-2e.

Jaký je aplikován vnitřní tag?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 qvoaeee0c10-2e
        Port "qvoaeee0c10-2e"
            tag: 69
            Interface "qvoaeee0c10-2e"
```

Zjistíme si ID portu:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep qvoaeee0c10-2e
 235(qvoaeee0c10-2e): addr:4a:2b:f6:9f:8d:19
```

Vypíšeme si OpenFlow pravidle v tabulce 0:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3847209.754s, table=0, n_packets=2, n_bytes=220, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
```

```
 cookie=0x0, duration=3847209.944s, table=0, n_packets=2341, n_bytes=235364, idle_age=0, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=3847210.061s, table=0, n_packets=88100169, n_bytes=13792305674, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2242139.596s, table=0, n_packets=16314557, n_bytes=4001739751, idle_age=0, hard_age=65534,
priority=3,in_port=86,vlan_tci=0x0000 actions=mod_vlan_vid:57,NORMAL
```

Podívejme se do forwarding tabulky a hledejmé cílovou MAC adresu:

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int | grep fa:16:3e:07:de:20
  212     69  fa:16:3e:07:de:20    0
```

Co je to za port?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep '212('
 212(qr-9ab15d1e-3d): addr:00:00:00:00:00:00
```

Odcházíme tedy do routeru

2.7.3. Router

Zjistíme si, stejně jako v předchozím případě, ID našeho routeru – třeba z CLI.

```
root@helion-ProLiant-DL380-Gen9:~# neutron router-list
+--------------------------------------+---------+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
--+
| id                                   | name    | external_gateway_info
|
+--------------------------------------+---------+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
--+
| 2e791c6d-b0ed-45b4-b04b-68a712b118ac | router1 | {"network_id": "3a5b5cd4-0c4b-4bc3-b44e-826c7b19556e",
"enable_snat": true, "external_fixed_ips": [{"subnet_id": "e3be37fb-1ced-432f-950c-99b887bb52c2", "ip_address":
"172.16.2.157"}]} |
+--------------------------------------+---------+-----------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------
--+
```

Protože vnitřní adresy se mohou mezi projekty překrývat sedí každý router ve svém vlastním name space. Ten vyhledáme podle ID routeru:

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns | grep 2e791c6d-b0ed-45b4-b04b-68a712b118ac
qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac
```

Name space je tedy řetězec „qrouter-“ a za ním ID routeru. Jaké IP interfaces se v tomto name space nacházejí?

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: rfp-2e791c6d-b: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 8a:26:b3:8e:eb:68 brd ff:ff:ff:ff:ff:ff
    inet 169.254.30.210/31 scope global rfp-2e791c6d-b
       valid_lft forever preferred_lft forever
    inet 172.16.2.3/32 brd 172.16.2.3 scope global rfp-2e791c6d-b
       valid_lft forever preferred_lft forever
    inet6 fe80::8826:b3ff:fe8e:eb68/64 scope link
       valid_lft forever preferred_lft forever
```

```
632: qr-9ab15d1e-3d: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:07:de:20 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global qr-9ab15d1e-3d
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe07:de20/64 scope link
       valid_lft forever preferred_lft forever
634: qr-f01425f2-58: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:42:d7:50 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.1/24 brd 192.168.20.255 scope global qr-f01425f2-58
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe42:d750/64 scope link
       valid_lft forever preferred_lft forever
```

V případě SNAT ovšem router nesměruje, takže se paket vrací na stejném interface, z kterého přišel – to je výchozí cesta.

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip rule list
0:      from all lookup local
32766:  from all lookup main
32767:  from all lookup default
32797:  from 192.168.10.8 lookup 16
3232238081:     from 192.168.10.1/24 lookup 3232238081
3232238081:     from 192.168.10.1/24 lookup 3232238081
3232240641:     from 192.168.20.1/24 lookup 3232240641
3232240641:     from 192.168.20.1/24 lookup 3232240641
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip route show
table 3232238081
default via 192.168.10.7 dev qr-9ab15d1e-3d
```

Přesto jednu práci pro router máme – vyměnit cílovou MAC adresu ze sdílené DVR MAC na jinou.

```
root@overcloud-novacompute0-vli5de2egecg:~# ip netns exec qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac tcpdump icmp -
e -l -i qr-9ab15d1e-3d
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qr-9ab15d1e-3d, link-type EN10MB (Ethernet), capture size 262144 bytes
03:53:31.052919 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 10.0.98.1: ICMP echo request, id 5015, seq 137, length 64
03:53:31.052963 fa:16:3e:07:de:20 (oui Unknown) > fa:16:3e:1f:87:98 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 10.0.98.1: ICMP echo request, id 5015, seq 137, length 64
03:53:32.054817 fa:16:3e:fd:7f:88 (oui Unknown) > fa:16:3e:07:de:20 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 10.0.98.1: ICMP echo request, id 5015, seq 138, length 64
03:53:32.054846 fa:16:3e:07:de:20 (oui Unknown) > fa:16:3e:1f:87:98 (oui Unknown), ethertype IPv4 (0x0800), length
98: 192.168.10.9 > 10.0.98.1: ICMP echo request, id 5015, seq 138, length 64
```

2.7.4. Vracíme se z routeru

Po směrování jsme vrátili paket do br-int a to stále ve stejné VLAN.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 qr-9ab15d1e-3d
        Port "qr-9ab15d1e-3d"
            tag: 69
            Interface "qr-9ab15d1e-3d"
                type: internal
```

Tentokrát už ale máme jinou destination MAC, takže provedeme lookup znovu.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-appctl fdb/show br-int | grep fa:16:3e:1f:87:98
   87    69  fa:16:3e:1f:87:98    0
```

A co je port 87?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep '87('
 87(patch-tun): addr:b2:5f:40:f0:2a:4f
```

Opouštíme tedy br-int a jdeme do br-tun.

2.7.5. Posíláme ven z compute node

Jaké porty na br-tun najdeme a kam směřují?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A100 br-tun
    Bridge br-tun
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.23"}
        Port "vxlan-0a000a0a"
            Interface "vxlan-0a000a0a"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.10"}
    ovs_version: "2.3.0"
```

Tentokrát nás bude zajímat zvýrazněný port, který směřuje do kontroleru, respektive do network node. Zjistíme si ID.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:00009e4ffab46e48
 1(patch-int): addr:7a:c7:3a:cf:90:5e
 2(vxlan-0a000a0a): addr:ba:0c:97:69:99:7f
 5(vxlan-0a000a17): addr:8a:30:a7:83:71:08
 LOCAL(br-tun): addr:9e:4f:fa:b4:6e:48
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Můžeme začít zkoumat OpenFlow pravidla – jako vždy od tabulky 0.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3923903.494s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=3082115.327s, table=0, n_packets=2385586, n_bytes=253280753, idle_age=0, hard_age=65534,
priority=1,in_port=5 actions=resubmit(,4)
 cookie=0x0, duration=3923901.546s, table=0, n_packets=13713693, n_bytes=993529751, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=3923887.172s, table=0, n_packets=446801, n_bytes=87876656, idle_age=1, hard_age=65534,
priority=1,in_port=2 actions=resubmit(,4)
```

Jdeme do tabulky 1 a prohlédneme si nejprve pravidla specifická pro VLAN 69.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=1,dl_vlan=69
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1030733.343s, table=1, n_packets=2527, n_bytes=247390, idle_age=1, hard_age=65534,
priority=1,dl_vlan=69,dl_src=fa:16:3e:07:de:20 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
 cookie=0x0, duration=1030733.441s, table=1, n_packets=2, n_bytes=276, idle_age=65534, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:07:de:20 actions=drop
 cookie=0x0, duration=1030733.539s, table=1, n_packets=15, n_bytes=630, idle_age=2352, hard_age=65534,
priority=3,arp,dl_vlan=69,arp_tpa=192.168.10.1 actions=drop
```

Nic z uvedeného se našeho paketu netýká. Musíme tedy na generičtější pravidlo.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=1
NXST_FLOW reply (xid=0x4):
```

```
 cookie=0x0, duration=3924083.843s, table=1, n_packets=13673850, n_bytes=990360128, idle_age=0, hard_age=65534,
priority=0 actions=resubmit(,2)
 cookie=0x0, duration=1030844.908s, table=1, n_packets=12959, n_bytes=1269982, idle_age=55489, hard_age=65534,
priority=1,dl_vlan=70,dl_src=fa:16:3e:42:d7:50 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
...
```

Pokračujeme tedy v tabulce 2.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=2
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3924131.804s, table=2, n_packets=4736846, n_bytes=410971859, idle_age=1, hard_age=65534,
priority=0,dl_dst=00:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,20)
 cookie=0x0, duration=3924131.710s, table=2, n_packets=8976032, n_bytes=582566690, idle_age=0, hard_age=65534,
priority=0,dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,22)
```

Náš paket je unicast, takže pokračujeme v tabulce 20.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=20,dl_vlan=69
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1030945.736s, table=20, n_packets=2805, n_bytes=270522, idle_age=0, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:1f:87:98 actions=strip_vlan,set_tunnel:0x3f2,output:2
 cookie=0x0, duration=1030945.836s, table=20, n_packets=170, n_bytes=16275, idle_age=2565, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:b2:3d:19 actions=strip_vlan,set_tunnel:0x3f2,output:2
 cookie=0x0, duration=986052.703s, table=20, n_packets=43248, n_bytes=4170488, idle_age=65534, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:fd:7f:88 actions=strip_vlan,set_tunnel:0x3f2,output:5
```

Máme shodu s odchozí MAC adresou, takže ustřihneme VLAN, přidáme VXLAN VNI 3F2 a odesíláme do portu 2, tedy do VXLAN směřující do compute node.

Jak nás provoz opouští?

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump -e -i eth0 -c 200 | grep -B1 192.168.10.9
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
03:55:02.198288 fc:15:b4:84:12:98 (oui Unknown) > 00:25:6e:0a:f6:07 (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-NovaCompute0-vli5de2egecg.35509 > overcloud-controller0-sujhw52cufku.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3f:9e:30:0c (oui Unknown) > fa:16:3e:1f:87:98 (oui Unknown), ethertype IPv4 (0x0800), length 98: 192.168.10.9 >
10.0.98.1: ICMP echo request, id 5015, seq 228, length 64
--
03:55:02.202161 00:25:6e:0a:f6:07 (oui Unknown) > fc:15:b4:84:12:98 (oui Unknown), ethertype IPv4 (0x0800), length
148: overcloud-controller0-sujhw52cufku.46717 > overcloud-NovaCompute0-vli5de2egecg.4789: VXLAN, flags [I] (0x08),
vni 1010
fa:16:3e:1f:87:98 (oui Unknown) > fa:16:3e:fd:7f:88 (oui Unknown), ethertype IPv4 (0x0800), length 98: 10.0.98.1 >
192.168.10.9: ICMP echo reply, id 5015, seq 228, length 64
```

2.7.6. Přijímáme v network node

Podívejme se nejprve na porty br-tun a jejich čísla.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-vsctl show
1718fb09-77ba-4171-80a8-86b1dcdfe4bb
    Bridge br-tun
        Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.10", out_key=flow, remote_ip="10.0.10.23"}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a0e"
            Interface "vxlan-0a000a0e"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.10", out_key=flow, remote_ip="10.0.10.14"}
        Port patch-int
```

```
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
```

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:00003602a443274e
 1(patch-int): addr:b2:84:3b:2c:07:d6
 2(vxlan-0a000a17): addr:16:32:ca:3a:50:06
 3(vxlan-0a000a0e): addr:a2:6d:15:1d:b2:cb
 LOCAL(br-tun): addr:36:02:a4:43:27:4e
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

2.7.7. OpenFlow pravidla v Network Node br-tun vSwitch

Začneme v tabulce 0.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101202.673s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=3101194.652s, table=0, n_packets=864713, n_bytes=49231013, idle_age=0, hard_age=65534,
priority=1,in_port=3 actions=resubmit(,4)
 cookie=0x0, duration=3101200.743s, table=0, n_packets=7468912, n_bytes=1230567666, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=3101195.474s, table=0, n_packets=10997, n_bytes=892718, idle_age=1043, hard_age=65534,
priority=1,in_port=2 actions=resubmit(,4)
```

Přicházíme z portu 3, pokračujeme tedy v tabulce 4. Na Network Node bývá hodně pravidel, budeme tedy hledat specificky naše číslo tunelu, tedy VXLAN VNI.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=4,tun_id=0x3f2
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101233.824s, table=4, n_packets=11274, n_bytes=1107027, idle_age=0, hard_age=65534,
priority=1,tun_id=0x3f2 actions=mod_vlan_vid:14,resubmit(,9)
```

Paketu přiřazujeme lokální VLAN 14 a pokračujeme do tabulky 9.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=9
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101343.824s, table=9, n_packets=833201, n_bytes=46768187, idle_age=1, hard_age=65534,
priority=0 actions=resubmit(,10)
 cookie=0x0, duration=3101344.201s, table=9, n_packets=9411, n_bytes=700956, idle_age=1186, hard_age=65534,
priority=1,dl_src=fa:16:3f:4d:1f:fb actions=output:1
 cookie=0x0, duration=3101344.022s, table=9, n_packets=33285, n_bytes=2671458, idle_age=1, hard_age=65534,
priority=1,dl_src=fa:16:3f:9e:30:0c actions=output:1
```

Dále do tabulky 10.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=10
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101409.350s, table=10, n_packets=833226, n_bytes=46769741, idle_age=0, hard_age=65534,
priority=1
actions=learn(table=20,hard_timeout=300,priority=1,NXM_OF_VLAN_TCI[0..11],NXM_OF_ETH_DST[]=NXM_OF_ETH_SRC[],load:0-
>NXM_OF_VLAN_TCI[],load:NXM_NX_TUN_ID[]->NXM_NX_TUN_ID[],output:NXM_OF_IN_PORT[]),output:1
```

Paket prozkoumáme a naučíme se jeho hlavičky, což zapíšeme do tabulky 20. Odcházíme z br-tun do patch-int portu, tedy do vSwitch br-int.

2.7.8. OpenFlow pravidla v Network Node br-int vSwitch

V Network Node bude hodně portů, tak si pro začátek zjistíme jen jaké je číslo patch mezi br-tun a br-int.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl show br-int | grep patch
 127(patch-tun): addr:2a:75:6e:b7:0e:10
```

Podívejme se na pravidla v tabulce 0.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3106124.326s, table=0, n_packets=9411, n_bytes=700956, idle_age=5966, hard_age=65534,
priority=2,in_port=127,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=3106124.168s, table=0, n_packets=38061, n_bytes=3139506, idle_age=0, hard_age=65534,
priority=2,in_port=127,dl_src=fa:16:3f:9e:30:0c actions=resubmit(,1)
 cookie=0x0, duration=3106124.440s, table=0, n_packets=1614948, n_bytes=176983979, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2406992.930s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=172 actions=drop
 cookie=0x0, duration=1908738.959s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=183 actions=drop
 cookie=0x0, duration=2941285.074s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=160 actions=drop
 cookie=0x0, duration=1908743.232s, table=0, n_packets=1, n_bytes=42, idle_age=65534, hard_age=65534,
priority=2,in_port=182 actions=drop
 cookie=0x0, duration=3106113.776s, table=0, n_packets=7341436, n_bytes=1148437924, idle_age=1, hard_age=65534,
priority=3,in_port=126,vlan_tci=0x0000 actions=mod_vlan_vid:6,NORMAL
```

Máme match na pakety z br-tun a zdrojovou MAC, jdeme tedy to tabulky 1 a hledejme specificky naší VLAN 14.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-int table=1,dl_vlan=14
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1164110.409s, table=1, n_packets=154, n_bytes=42546, idle_age=401, hard_age=65534,
priority=2,ip,dl_vlan=14,nw_dst=192.168.10.0/24 actions=strip_vlan,mod_dl_src:fa:16:3e:07:de:20,output:227,output:121
 cookie=0x0, duration=1164114.182s, table=1, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=4,dl_vlan=14,dl_dst=fa:16:3e:b2:3d:19 actions=strip_vlan,mod_dl_src:fa:16:3e:07:de:20,output:121
 cookie=0x0, duration=1164110.550s, table=1, n_packets=21661, n_bytes=2122778, idle_age=0, hard_age=65534,
priority=4,dl_vlan=14,dl_dst=fa:16:3e:1f:87:98 actions=strip_vlan,mod_dl_src:fa:16:3e:07:de:20,output:227
```

Máme match na cílovou MAC adresu. Odstřihneme VLAN tak a zmodifikujeme zdrojovou MAC adresu, kde dosadíme MAC našeho distribuovaného routeru (důvody už jsme popsali – před odesláním na drát DVR MAC nahrazujeme jinou a po příjmu vracíme do původního stavu). Odcházíme do portu 227 – co tam je?

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl show br-int | grep '227('
 227(sg-ff1a1932-74): addr:00:00:00:00:00:00
```

2.7.9. SNAT namespace

Port sg-ff1a1932-74 se nachází v SNAT name space, které opět najdeme podle router ID.

```
root@overcloud-controller0-sujhw52cufku:~# ip netns | grep 2e791c6d-b0ed-45b4-b04b-68a712b118ac
snat-2e791c6d-b0ed-45b4-b04b-68a712b118ac
qrouter-2e791c6d-b0ed-45b4-b04b-68a712b118ac
```

Podívejme se na interfacy

```
root@overcloud-controller0-sujhw52cufku:~# ip netns exec snat-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
407: qg-b2712c4a-2b: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:c3:f1:8d brd ff:ff:ff:ff:ff:ff
    inet 172.16.2.157/16 brd 172.16.255.255 scope global qg-b2712c4a-2b
       valid_lft forever preferred_lft forever
```

```
    inet6 fe80::f816:3eff:fec3:f18d/64 scope link
       valid_lft forever preferred_lft forever
409: sg-ff1a1932-74: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:1f:87:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.7/24 brd 192.168.10.255 scope global sg-ff1a1932-74
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fe1f:8798/64 scope link
       valid_lft forever preferred_lft forever
411: sg-f9e28eef-bc: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:c7:6e:85 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.2/24 brd 192.168.20.255 scope global sg-f9e28eef-bc
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:fec7:6e85/64 scope link
       valid_lft forever preferred_lft forever
```

Kudy bude paket odcházet?

```
root@overcloud-controller0-sujhw52cufku:~# ip netns exec snat-2e791c6d-b0ed-45b4-b04b-68a712b118ac ip route
default via 172.16.0.1 dev qg-b2712c4a-2b
172.16.0.0/16 dev qg-b2712c4a-2b  proto kernel  scope link  src 172.16.2.157
192.168.10.0/24 dev sg-ff1a1932-74  proto kernel  scope link  src 192.168.10.7
192.168.20.0/24 dev sg-f9e28eef-bc  proto kernel  scope link  src 192.168.20.2
```

Před tím ale budeme provádět SNAT, podívejme se do iptables.

```
root@overcloud-controller0-sujhw52cufku:~# ip netns exec snat-2e791c6d-b0ed-45b4-b04b-68a712b118ac iptables --table
nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-PREROUTING  all  --  anywhere             anywhere

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-OUTPUT  all  --  anywhere             anywhere

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
neutron-l3-agent-POSTROUTING  all  --  anywhere             anywhere
neutron-postrouting-bottom  all  --  anywhere             anywhere

Chain neutron-l3-agent-OUTPUT (1 references)
target     prot opt source               destination

Chain neutron-l3-agent-POSTROUTING (1 references)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ! ctstate DNAT

Chain neutron-l3-agent-PREROUTING (1 references)
target     prot opt source               destination

Chain neutron-l3-agent-float-snat (0 references)
target     prot opt source               destination

Chain neutron-l3-agent-snat (1 references)
target     prot opt source               destination
SNAT       all  --  192.168.10.0/24      anywhere             to:172.16.2.157
SNAT       all  --  192.168.20.0/24      anywhere             to:172.16.2.157

Chain neutron-postrouting-bottom (1 references)
```

```
target     prot opt source               destination
neutron-l3-agent-snat  all  --  anywhere             anywhere
```

Podívejme se na provoz opouštějící SNAT name space.

```
root@overcloud-controller0-sujhw52cufku:~# ip netns exec snat-2e791c6d-b0ed-45b4-b04b-68a712b118ac tcpdump icmp -e -l
-i qg-b2712c4a-2b
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qg-b2712c4a-2b, link-type EN10MB (Ethernet), capture size 262144 bytes
04:09:17.521252 fa:16:3e:c3:f1:8d (oui Unknown) > 38:22:d6:e9:92:23 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.157 > 10.0.98.1: ICMP echo request, id 5015, seq 1082, length 64
04:09:17.526868 38:22:d6:e9:92:23 (oui Unknown) > fa:16:3e:c3:f1:8d (oui Unknown), ethertype IPv4 (0x0800), length
98: 10.0.98.1 > 172.16.2.157: ICMP echo reply, id 5015, seq 1082, length 64
04:09:18.522571 fa:16:3e:c3:f1:8d (oui Unknown) > 38:22:d6:e9:92:23 (oui Unknown), ethertype IPv4 (0x0800), length
98: 172.16.2.157 > 10.0.98.1: ICMP echo request, id 5015, seq 1083, length 64
04:09:18.528884 38:22:d6:e9:92:23 (oui Unknown) > fa:16:3e:c3:f1:8d (oui Unknown), ethertype IPv4 (0x0800), length
98: 10.0.98.1 > 172.16.2.157: ICMP echo reply, id 5015, seq 1083, length 64
```

NAT i routing byl proveden

2.7.10. Konec cesty

Máme zaNATováno a paket vstupuje do br-ext – má nějaký tag? Jaké ID má tento port?

```
root@overcloud-controller0-sujhw52cufku:~# ovs-vsctl show | grep -A3 qg-b2712c4a-2b
        Port "qg-b2712c4a-2b"
            Interface "qg-b2712c4a-2b"
                type: internal

root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl show br-ex | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:000000256e0af607
 1(vlan172): addr:00:25:6e:0a:f6:07
 10(qg-e9c2c154-9c): addr:26:01:00:00:00:00
 13(qg-92be0f9c-e8): addr:26:01:00:00:00:00
 14(qg-ef8abb09-76): addr:26:01:00:00:00:00
 15(qg-cf26a386-18): addr:26:01:00:00:00:00
 19(qg-15fc384e-a5): addr:00:00:00:00:00:00
 22(qg-da9fa75a-7b): addr:00:00:00:00:00:00
 25(qg-ddb6b311-95): addr:00:00:00:00:00:00
 31(qg-6a140e68-f6): addr:00:00:00:00:00:00
 34(qg-88e27db1-6b): addr:00:00:00:00:00:00
 42(qg-27387ea9-a9): addr:00:00:00:00:00:00
 43(qg-b89620d8-3a): addr:00:00:00:00:00:00
 44(qg-b2712c4a-2b): addr:00:00:00:00:00:00
 49(qg-184f9c03-d9): addr:00:00:00:00:00:00
 LOCAL(br-ex): addr:00:25:6e:0a:f6:07
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Zvýrazněný je také port 1, který směřuje do externí sítě (v našem případě stejným fyzickém portem, ale tagována v externí VLAN 172).

Jaká má br-ex pravidla?

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-ex table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3722047.411s, table=0, n_packets=199158, n_bytes=88736522, idle_age=0, hard_age=65534,
priority=0 actions=NORMAL
```

Jde tedy o běžný switching – podívejme se proto do forwarding tabulky.

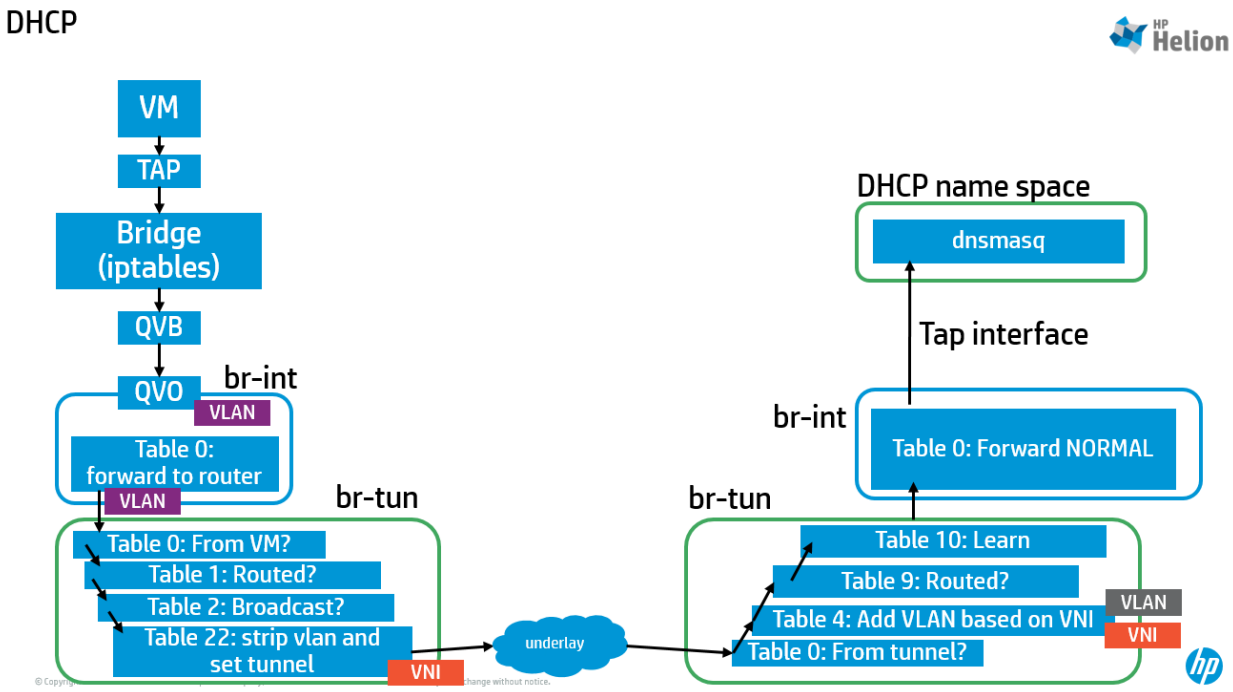
```
root@overcloud-controller0-sujhw52cufku:~# ovs-appctl fdb/show br-ex
 port  VLAN  MAC                Age
   49     0  fa:16:3e:03:44:e7  127
```

```
 1      0  38:22:d6:e9:92:23  103
34      0  fa:16:3e:3b:0b:76  103
15      0  fa:16:3e:81:c5:ee    0
44      0  fa:16:3e:c3:f1:8d    0
```

Máme match na destination MAC paketu a posíláme provoz na port 1, tedy vlan172, což je interface do našeho externího subnetu ve skutečném světě.

2.8. DHCP provoz

V tomto scénaři se zaměříme na způsob jakým dojde k přiřazení adresy DHCP protokolem.



Na jedné z instancí spustíme generování DHCP dotazu každou vteřinu. Nainstalujte si udhcpc – jednoduchého DHCP klienta (sudo apt-get install udhcpc) a necháme ho poslat dotaz každou vteřinu:

```
root@instance-1:~# while true; do udhcpc; sleep 1; done
udhcpc (v1.20.2) started
Sending discover...
Sending select for 192.168.10.8...
Lease of 192.168.10.8 obtained, lease time 172800
/etc/udhcpc/default.script: Resetting default routes
SIOCDELRT: No such process
/etc/udhcpc/default.script: Adding DNS 192.168.10.3
udhcpc (v1.20.2) started
Sending discover...
Sending select for 192.168.10.8...
Lease of 192.168.10.8 obtained, lease time 172800
/etc/udhcpc/default.script: Resetting default routes
SIOCDELRT: No such process
/etc/udhcpc/default.script: Adding DNS 192.168.10.3
```

2.8.1. Pakety odcházející z VM

Podívejme se na provoz na tap interface.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump port 67 or port 68 -e -i tap425fe781-d3
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tap425fe781-d3, link-type EN10MB (Ethernet), capture size 262144 bytes
06:11:32.238031 fa:16:3e:21:cf:75 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 322: 0.0.0.0.bootpc >
255.255.255.255.bootps: BOOTP/DHCP, Request from fa:16:3e:21:cf:75 (oui Unknown), length 280
06:11:32.243728 fa:16:3e:b2:3d:19 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
373: 192.168.10.3.bootps > 192.168.10.8.bootpc: BOOTP/DHCP, Reply, length 331
06:11:32.244108 fa:16:3e:21:cf:75 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 334: 0.0.0.0.bootpc >
255.255.255.255.bootps: BOOTP/DHCP, Request from fa:16:3e:21:cf:75 (oui Unknown), length 292
06:11:32.244506 fa:16:3e:b2:3d:19 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
373: 192.168.10.3.bootps > 192.168.10.8.bootpc: BOOTP/DHCP, Reply, length 331
```

2.8.2. Ochrana před neoprávněnýi nabídkami

Připomeňme si, že z tap interface prochází do bridge kvůli aplikaci Security Group pravidel, tedy do stavového mini-firewallu. Pro prevenci neoprávněného DHCP serveru tu existuje několik pravidel:

```
root@overcloud-novacompute0-vli5de2egecg:~# iptables --list-rules | grep tap425fe781-d3
-A neutron-openvswi-FORWARD -m physdev --physdev-out tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-sg-chain
-A neutron-openvswi-FORWARD -m physdev --physdev-in tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-sg-chain
-A neutron-openvswi-INPUT -m physdev --physdev-in tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-o425fe781-d
-A neutron-openvswi-sg-chain -m physdev --physdev-out tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-
i425fe781-d
-A neutron-openvswi-sg-chain -m physdev --physdev-in tap425fe781-d3 --physdev-is-bridged -j neutron-openvswi-
o425fe781-d

root@overcloud-novacompute0-vli5de2egecg:~# iptables --list neutron-openvswi-i425fe781-d -v -n
Chain neutron-openvswi-i425fe781-d (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
 328K   44M RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
 5701 2047K RETURN     udp  --  *      *       192.168.10.3         0.0.0.0/0           udp spt:67 dpt:68
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           match-set IPv4b9eaf0cf-e8b2-
41f1-9 src
    2   120 RETURN     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
   11   924 RETURN     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
   28  1680 RETURN     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           match-set IPv4ea62d680-0c24-
4f60-9 src
  137  9565 neutron-openvswi-sg-fallback  all  --  *      *       0.0.0.0/0            0.0.0.0/0

root@overcloud-novacompute0-vli5de2egecg:~# iptables --list neutron-openvswi-o425fe781-d -v -n
Chain neutron-openvswi-o425fe781-d (2 references)
 pkts bytes target     prot opt in     out     source               destination
 5733 1801K RETURN     udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:68 dpt:67
 378K   42M neutron-openvswi-s425fe781-d  all  --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       udp  --  *      *       0.0.0.0/0            0.0.0.0/0           udp spt:67 dpt:68
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
 377K   42M RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
  278 19184 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0
  500 42000 neutron-openvswi-sg-fallback  all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

2.8.3. Vstup do vSwitch br-int

Portu tap425fe781-d3 jak už víme bude odpovídat port qvo425fe781-d3 na vstupu do br-int. Jaký VLAN tag bude provozu přiřazen?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A3 425fe781-d3
        Port "qvo425fe781-d3"
            tag: 69
            Interface "qvo425fe781-d3"
```

A číslo portu?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-int | grep 425fe781-d3
 211(qvo425fe781-d3): addr:da:04:37:a2:8a:f6
```

Projděme si OpenFlow pravidla – nejprve v tabulce 0.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-int table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=4448449.957s, table=0, n_packets=2, n_bytes=220, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:5d:a5:3f actions=resubmit(,1)
 cookie=0x0, duration=4448450.147s, table=0, n_packets=14620, n_bytes=1438706, idle_age=65534, hard_age=65534,
priority=2,in_port=87,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=4448450.264s, table=0, n_packets=117836242, n_bytes=18512592587, idle_age=0, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2843379.799s, table=0, n_packets=23002022, n_bytes=5960432596, idle_age=0, hard_age=65534,
priority=3,in_port=86,vlan_tci=0x0000 actions=mod_vlan_vid:57,NORMAL
```

Náš paket je broadcast, takže se dostane o do patch-int, tedz odejde do br-tun.

2.8.4. Posíláme ven z compute node

DHCP server je usazen v kontroleru, konkrétně v network node. Očekáváme tedy, že DHCP request bude zabalen do VXLAN tunelu a odcestuje do network node. Je to tak?

Jaké porty na br-tun najdeme a kam směřují?

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl show | grep -A100 br-tun
    Bridge br-tun
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.23"}
        Port "vxlan-0a000a0a"
            Interface "vxlan-0a000a0a"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.14", out_key=flow, remote_ip="10.0.10.10"}
    ovs_version: "2.3.0"
```

Tentokrát nás bude zajímat zvýrazněný port, který směřuje do kontroleru, respektive do network node. Zjistíme si ID.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl show br-tun | grep '('
OFPT_FEATURES_REPLY (xid=0x2): dpid:00009e4ffab46e48
 1(patch-int): addr:7a:c7:3a:cf:90:5e
 2(vxlan-0a000a0a): addr:ba:0c:97:69:99:7f
 5(vxlan-0a000a17): addr:8a:30:a7:83:71:08
 LOCAL(br-tun): addr:9e:4f:fa:b4:6e:48
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Můžeme začít zkoumat OpenFlow pravidla – jako vždy od tabulky 0.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3923903.494s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=3082115.327s, table=0, n_packets=2385586, n_bytes=253280753, idle_age=0, hard_age=65534,
priority=1,in_port=5 actions=resubmit(,4)
 cookie=0x0, duration=3923901.546s, table=0, n_packets=13713693, n_bytes=993529751, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
```

```
 cookie=0x0, duration=3923887.172s, table=0, n_packets=446801, n_bytes=87876656, idle_age=1, hard_age=65534,
priority=1,in_port=2 actions=resubmit(,4)
```

Jdeme do tabulky 1 a prohlédneme si nejprve pravidla specifická pro VLAN 69.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=1,dl_vlan=69
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1030733.343s, table=1, n_packets=2527, n_bytes=247390, idle_age=1, hard_age=65534,
priority=1,dl_vlan=69,dl_src=fa:16:3e:07:de:20 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
 cookie=0x0, duration=1030733.441s, table=1, n_packets=2, n_bytes=276, idle_age=65534, hard_age=65534,
priority=2,dl_vlan=69,dl_dst=fa:16:3e:07:de:20 actions=drop
 cookie=0x0, duration=1030733.539s, table=1, n_packets=15, n_bytes=630, idle_age=2352, hard_age=65534,
priority=3,arp,dl_vlan=69,arp_tpa=192.168.10.1 actions=drop
```

Nic z uvedeného se našeho paketu netýká. Musíme tedy na generičtější pravidlo.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=1
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3924083.843s, table=1, n_packets=13673850, n_bytes=990360128, idle_age=0, hard_age=65534,
priority=0 actions=resubmit(,2)
 cookie=0x0, duration=1030844.908s, table=1, n_packets=12959, n_bytes=1269982, idle_age=55489, hard_age=65534,
priority=1,dl_vlan=70,dl_src=fa:16:3e:42:d7:50 actions=mod_dl_src:fa:16:3f:9e:30:0c,resubmit(,2)
...
```

Pokračujeme tedy v tabulce 2.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=2
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3924131.804s, table=2, n_packets=4736846, n_bytes=410971859, idle_age=1, hard_age=65534,
priority=0,dl_dst=00:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,20)
 cookie=0x0, duration=3924131.710s, table=2, n_packets=8976032, n_bytes=582566690, idle_age=0, hard_age=65534,
priority=0,dl_dst=01:00:00:00:00:00/01:00:00:00:00:00 actions=resubmit(,22)
```

Náš paket je broadcast, takže pokračujeme v tabulce 22 a najdeme specifickz pravidlo pro VLAN 69.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-ofctl dump-flows br-tun table=22,dl_vlan=69
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=1562253.088s, table=22, n_packets=8302, n_bytes=2676568, idle_age=0, hard_age=65534, dl_vlan=69
actions=strip_vlan,set_tunnel:0x3f2,output:2,output:5
```

Odstřihneme VLAN, prodáme VXLAN VNI 3F2 a odešleme do tunelu do network node.

2.8.5. Přijímáme v network node

Podívejme se nejprve na porty br-tun a jejich čísla.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-vsctl show
1718fb09-77ba-4171-80a8-86b1dcdfe4bb
    Bridge br-tun
        Port "vxlan-0a000a17"
            Interface "vxlan-0a000a17"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.10", out_key=flow, remote_ip="10.0.10.23"}
        Port br-tun
            Interface br-tun
                type: internal
        Port "vxlan-0a000a0e"
            Interface "vxlan-0a000a0e"
                type: vxlan
                options: {df_default="false", in_key=flow, local_ip="10.0.10.10", out_key=flow, remote_ip="10.0.10.14"}
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}

root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl show br-tun | grep '('
```

```
OFPT_FEATURES_REPLY (xid=0x2): dpid:00003602a443274e
 1(patch-int): addr:b2:84:3b:2c:07:d6
 2(vxlan-0a000a17): addr:16:32:ca:3a:50:06
 3(vxlan-0a000a0e): addr:a2:6d:15:1d:b2:cb
 LOCAL(br-tun): addr:36:02:a4:43:27:4e
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

2.8.6. OpenFlow pravidla v Network Node br-tun vSwitch

Začneme v tabulce 0.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=0
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101202.673s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534, priority=0
actions=drop
 cookie=0x0, duration=3101194.652s, table=0, n_packets=864713, n_bytes=49231013, idle_age=0, hard_age=65534,
priority=1,in_port=3 actions=resubmit(,4)
 cookie=0x0, duration=3101200.743s, table=0, n_packets=7468912, n_bytes=1230567666, idle_age=0, hard_age=65534,
priority=1,in_port=1 actions=resubmit(,1)
 cookie=0x0, duration=3101195.474s, table=0, n_packets=10997, n_bytes=892718, idle_age=1043, hard_age=65534,
priority=1,in_port=2 actions=resubmit(,4)
```

Přicházíme z portu 3, pokračujeme tedy v tabulce 4. Na Network Node bývá hodně pravidel, budeme tedy hledat specificky naše číslo tunelu, tedy VXLAN VNI.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=4,tun_id=0x3f2
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101233.824s, table=4, n_packets=11274, n_bytes=1107027, idle_age=0, hard_age=65534,
priority=1,tun_id=0x3f2 actions=mod_vlan_vid:14,resubmit(,9)
```

Paketu přiřazujeme lokální VLAN 14 a pokračujeme do tabulky 9.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=9
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101343.824s, table=9, n_packets=833201, n_bytes=46768187, idle_age=1, hard_age=65534,
priority=0 actions=resubmit(,10)
 cookie=0x0, duration=3101344.201s, table=9, n_packets=9411, n_bytes=700956, idle_age=1186, hard_age=65534,
priority=1,dl_src=fa:16:3f:4d:1f:fb actions=output:1
 cookie=0x0, duration=3101344.022s, table=9, n_packets=33285, n_bytes=2671458, idle_age=1, hard_age=65534,
priority=1,dl_src=fa:16:3f:9e:30:0c actions=output:1
```

Dále do tabulky 10.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-tun table=10
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3101409.350s, table=10, n_packets=833226, n_bytes=46769741, idle_age=0, hard_age=65534,
priority=1
actions=learn(table=20,hard_timeout=300,priority=1,NXM_OF_VLAN_TCI[0..11],NXM_OF_ETH_DST[]=NXM_OF_ETH_SRC[],load:0-
>NXM_OF_VLAN_TCI[],load:NXM_NX_TUN_ID[]->NXM_NX_TUN_ID[],output:NXM_OF_IN_PORT[]),output:1
```

Paket prozkoumáme a naučíme se jeho hlavičky, což zapíšeme do tabulky 20. Odcházíme z br-tun do patch-int portu, tedy do vSwitch br-int.

2.8.7. OpenFlow pravidla v Network Node br-int vSwitch

V Network Node bude hodně portů, tak si pro začátek zjistíme jen jaké je číslo patch mezi br-tun a br-int.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl show br-int | grep patch
 127(patch-tun): addr:2a:75:6e:b7:0e
```

Vypišme pravidle v tabulce 0.

```
root@overcloud-controller0-sujhw52cufku:~# ovs-ofctl dump-flows br-int table=0
```

```
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=3626153.575s, table=0, n_packets=9459, n_bytes=704412, idle_age=7595, hard_age=65534,
priority=2,in_port=127,dl_src=fa:16:3f:4d:1f:fb actions=resubmit(,1)
 cookie=0x0, duration=3626153.417s, table=0, n_packets=52177, n_bytes=4538666, idle_age=2605, hard_age=65534,
priority=2,in_port=127,dl_src=fa:16:3f:9e:30:0c actions=resubmit(,1)
 cookie=0x0, duration=3626153.689s, table=0, n_packets=2189084, n_bytes=222490233, idle_age=1, hard_age=65534,
priority=1 actions=NORMAL
 cookie=0x0, duration=2927022.179s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=172 actions=drop
 cookie=0x0, duration=2428768.208s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=183 actions=drop
 cookie=0x0, duration=3461314.323s, table=0, n_packets=0, n_bytes=0, idle_age=65534, hard_age=65534,
priority=2,in_port=160 actions=drop
 cookie=0x0, duration=2428772.481s, table=0, n_packets=1, n_bytes=42, idle_age=65534, hard_age=65534,
priority=2,in_port=182 actions=drop
 cookie=0x0, duration=3626143.025s, table=0, n_packets=8683279, n_bytes=1378612476, idle_age=0, hard_age=65534,
priority=3,in_port=126,vlan_tci=0x0000 actions=mod_vlan_vid:6,NORMAL
```

Použijeme tedy NORMAL forwarding. Protože náš paket je broadcast, půjde na všechny porty. Co tedy všechno je ve VLAN 14?

```
root@overcloud-controller0-sujhw52cufku:~# ovs-vsctl show | grep -B1 'tag: 14'
        Port "tape8a769e3-6e"
            tag: 14
--
        Port "sg-ff1a1932-74"
            tag: 14
--
        Port "qr-9ab15d1e-3d"
            tag: 14
```

DHCP provoz je řešen ve svém vlastním name space, protože adresy jednotlivých tenantů se mohou překrývat. Správný name space najdeme podle network ID:

```
root@helion-ProLiant-DL380-Gen9:~# neutron net-list
+--------------------------------------+------------+-------------------------------------------------------+
| id                                   | name       | subnets                                               |
+--------------------------------------+------------+-------------------------------------------------------+
| 3a5b5cd4-0c4b-4bc3-b44e-826c7b19556e | ext-net    | e3be37fb-1ced-432f-950c-99b887bb52c2                  |
| 41778abb-b994-4ccb-a9ab-0d60a77cc1f8 | net1       | a62d865f-e87f-4ebd-b3e6-10b806299582 192.168.10.0/24  |
| 7590c21a-4878-48ae-b957-7562e4dc1d0d | default-net | 2c223f18-79f9-41c0-b19a-e5bdfa294895 192.168.1.0/24   |
| ac163954-4b86-439c-8617-522c17467c95 | net2       | 8833c0ac-260f-4c32-a971-c6b31e3f8b9e 192.168.20.0/24  |
+--------------------------------------+------------+-------------------------------------------------------+

root@overcloud-controller0-sujhw52cufku:~# ip netns | grep 41778abb-b994-4ccb-a9ab-0d60a77cc1f8
qdhcp-41778abb-b994-4ccb-a9ab-0d60a77cc1f8
```

Najdeme tam náš interface tape8a769e3-6e.

```
root@overcloud-controller0-sujhw52cufku:~# ip netns exec qdhcp-41778abb-b994-4ccb-a9ab-0d60a77cc1f8 ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
280: tape8a769e3-6e: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether fa:16:3e:b2:3d:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.3/24 brd 192.168.10.255 scope global tape8a769e3-6e
       valid_lft forever preferred_lft forever
    inet6 fe80::f816:3eff:feb2:3d19/64 scope link
       valid_lft forever preferred_lft forever
```

2.8.8. DHCP server

Poslechněme si provoz na tap interface v našem DHCP name space.

```
root@overcloud-controller0-sujhw52cufku:~# ip netns exec qdhcp-41778abb-b994-4ccb-a9ab-0d60a77cc1f8 tcpdump port 67
or port 68 -e -l -i tape8a769e3-6e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on tape8a769e3-6e, link-type EN10MB (Ethernet), capture size 262144 bytes
09:35:30.699417 fa:16:3e:21:cf:75 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 322: 0.0.0.0.bootpc >
255.255.255.255.bootps: BOOTP/DHCP, Request from fa:16:3e:21:cf:75 (oui Unknown), length 280
09:35:30.699661 fa:16:3e:b2:3d:19 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
373: 192.168.10.3.bootps > 192.168.10.8.bootpc: BOOTP/DHCP, Reply, length 331
09:35:30.945597 fa:16:3e:21:cf:75 (oui Unknown) > Broadcast, ethertype IPv4 (0x0800), length 334: 0.0.0.0.bootpc >
255.255.255.255.bootps: BOOTP/DHCP, Request from fa:16:3e:21:cf:75 (oui Unknown), length 292
09:35:30.945841 fa:16:3e:b2:3d:19 (oui Unknown) > fa:16:3e:21:cf:75 (oui Unknown), ethertype IPv4 (0x0800), length
373: 192.168.10.3.bootps > 192.168.10.8.bootpc: BOOTP/DHCP, Reply, length 331
```

Podívejme se přímo na proces DHCP serveru.

```
root@overcloud-controller0-sujhw52cufku:~# ps -efa | grep tape8a769e3-6e
root     21291  5679  0 09:37 pts/28   00:00:00 grep tape8a769e3-6e
nobody   24114     1  0 May14 ?        00:00:03 dnsmasq --no-hosts --no-resolv --strict-order --bind-interfaces --
interface=tape8a769e3-6e --except-interface=lo --pid-file=/var/run/neutron/dhcp/41778abb-b994-4ccb-a9ab-
0d60a77cc1f8/pid --dhcp-hostsfile=/var/run/neutron/dhcp/41778abb-b994-4ccb-a9ab-0d60a77cc1f8/host --addn-
hosts=/var/run/neutron/dhcp/41778abb-b994-4ccb-a9ab-0d60a77cc1f8/addn_hosts --dhcp-
optsfile=/var/run/neutron/dhcp/41778abb-b994-4ccb-a9ab-0d60a77cc1f8/opts --leasefile-ro --dhcp-
range=set:tag0,192.168.10.0,static,172800s --dhcp-lease-max=256 --conf-file=/etc/neutron/dnsmasq/dnsmasq-neutron.conf
--domain=openstacklocal
```

Podívejme se, jestli paket doputuje až do našeho procesu.

```
root@overcloud-controller0-sujhw52cufku:~# strace -p 24114 -e network,write -s 4096
Process 24114 attached
recvmsg(3, {msg_name(16)={sa_family=AF_INET, sin_port=htons(68), sin_addr=inet_addr("0.0.0.0")},
msg_iov(1)=[{"\1\1\6\0\224\223\5k\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\372\26>!\317u\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0c\202Sc5\1\1=\7\1\372\26>!\317u9\2\2@7\7\1\3\6\f\17\34*<\fudhcp 1.20.2\377", 548}],
msg_controllen=32, {cmsg_len=28, cmsg_level=SOL_IP, cmsg_type=, ...}, msg_flags=0}, MSG_PEEK|MSG_TRUNC) = 280
recvmsg(3, {msg_name(16)={sa_family=AF_INET, sin_port=htons(68), sin_addr=inet_addr("0.0.0.0")},
msg_iov(1)=[{"\1\1\6\0\224\223\5k\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\372\26>!\317u\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0
\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\
0\0\0\0\0\0\0\0\0\0\0\0c\202Sc5\1\1=\7\1\372\26>!\317u9\2\2@7\7\1\3\6\f\17\34*<\fudhcp 1.20.2\377", 548}],
msg_controllen=32, {cmsg_len=28, cmsg_level=SOL_IP, cmsg_type=, ...}, msg_flags=0}, 0) = 280
sendto(4, "\24\0\0\0\26\0\5\3i(\0\0\0\0\0\0\2ape", 20, 0, {sa_family=AF_NETLINK, pid=0, groups=00000000}, 12) = 20
recvmsg(4, {msg_name(12)={sa_family=AF_NETLINK, pid=0, groups=00000000},
msg_iov(1)=[{"L\0\0\0\24\0\2\0i(\0\0000^\0\0\2\10\200\376\1\0\0\0\10\0\1\0\177\0\0\1\10\0\2\0\177\0\0\1\7\0\3\01o\0\0
\10\0\10\0\200\0\0\0\24\0\6\0\377\377\377\377\377\377\377\377\263\0!\4\263\0!\4`\0\0\0\24\0\2\0i(\0\0000^\0\0\2\30\20
0\0\30\1\0\0\10\0\1\0\300\250\n\3\10\0\2\0\300\250\n\3\10\0\4\0\300\250\n\377\23\0\3\0tape8a769e3-
6e\0\0\10\0\10\0\200\0\0\0\24\0\6\0\377\377\377\377\377\377\377\377\2\1!\4\2\1!\4", 172}], msg_controllen=0,
msg_flags=0}, MSG_PEEK|MSG_TRUNC) = 172
```

3. Monitoring a troubleshooting

3.1. Přeposílání paketů do centrálního analyzátoru (RSPAN)

V předchozích kapitolách jsme se naučili zjistit, kde se nachází konkrétní instance (na jakém compute node) a také jakým rozhraním je připojena vNIC. Například v CLI takto:

```
root@helion-ProLiant-DL380-Gen9:~# nova list --all-tenants 1 --tenant baa7096fe1d54571900c3758397e0939 --fields name,OS-EXT-SRV-
ATTR:hypervisor_hostname,OS-EXT-SRV-ATTR:instance_name
+--------------------------------------+------------+-------------------------------------------+-------------------------------
+
| ID                                   | Name       | OS-EXT-SRV-ATTR: Hypervisor Hostname      | OS-EXT-SRV-ATTR: Instance Name
|
+--------------------------------------+------------+-------------------------------------------+-------------------------------
+
| eb347271-dc5a-46cf-9150-0a7defffc6d1 | instance-1 | overcloud-novacompute0-vli5de2egecg.novalocal | instance-0000010d
|
| 70d0662f-9c69-4d0b-99e7-2dde4e0494e8 | instance-2 | overcloud-novacompute0-vli5de2egecg.novalocal | instance-0000010e
|
| e1975422-a543-4ce4-be36-bce191816161 | instance-3 | overcloud-novacompute1-c4ia2jfbd75d.novalocal | instance-0000010f
|
+--------------------------------------+------------+-------------------------------------------+-------------------------------
+

root@helion-ProLiant-DL380-Gen9:~# nova hypervisor-show overcloud-novacompute0-vli5de2egecg.novalocal | grep host_ip
| host_ip                  | 10.0.10.14
```

Přihlašte se do nalezeho compute node ze seed VM pod účtem heat-admin (ssh heat-admin@10.0.10.14) a zjistěte jméno tap interface.

```
root@overcloud-novacompute0-vli5de2egecg:~# virsh dumpxml instance-0000010d | grep "target dev='tap"
      <target dev='tap425fe781-d3'/>
```

Ve VM máme puštěný ping. Na compute si cvičně zachytíme pakety přes lokální tcmpdump na portu qvo, tedy řetězec tap nahradíme za qvo.

```
root@overcloud-novacompute0-vli5de2egecg:~# tcpdump -i qvo425fe781-d3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on qvo425fe781-d3, link-type EN10MB (Ethernet), capture size 262144 bytes
09:53:06.674946 IP 192.168.10.8 > 192.168.10.9: ICMP echo request, id 5439, seq 13906, length 64
09:53:06.675227 IP 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 5439, seq 13906, length 64
09:53:06.675497 IP 192.168.10.8.ssh > 10.0.10.254.36952: Flags [P.], seq 3109336920:3109337032, ack 982140678, win
3862, options [nop,nop,TS val 327160865 ecr 1358118284], length 112
09:53:06.675568 IP 192.168.10.8.ssh > 10.0.10.254.36952: Flags [P.], seq 112:160, ack 1, win 3862, options
[nop,nop,TS val 327160865 ecr 1358118284], length 48
09:53:06.675781 IP 10.0.10.254.36952 > 192.168.10.8.ssh: Flags [.], ack 160, win 550, options [nop,nop,TS val
1358118534 ecr 327160865], length 0
```

Nyní zapneme RSPAN, tedy budeme chtít provoz této VM zrcadlit GRE tunelem do vzdáleného analyzátoru, například do našeho notebooku s běžícím Wireshark nebo v případě našeho labu na jeden ze serverů, kde budeme kolektovat provoz přes tcpdump. Cílový analyzátor bude na IP 10.0.10.53.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl add-port br-int gre0 \
>           -- set interface gre0 type=gre options:remote_ip=10.0.10.53 \
>           -- --id=@p get port gre0 \
>           -- --id=@vm get port qvo425fe781-d3 \
>           -- --id=@m create mirror name=m0 select-src-port=@vm select-dst-port=@vm output-port=@p \
>           -- set bridge br-int mirrors=@m
7b6e9c49-e335-4e77-b259-01f0cb0ba71d
root@overcloud-novacompute0-vli5de2egecg:~# ¨
```

Otevřete váš analyzátor – uvidíte pakety zabelené v GRE a uvnitř provoz ze sledované VM.

```
root@LabServer:~# tcpdump -i eth0 | grep GRE
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:10:58.696424 IP 10.0.10.14 > LabServer.helion.demo: GREv0, length 106: IP 192.168.10.8 > 192.168.10.9: ICMP echo
request, id 5591, seq 95, length 64
```

```
20:10:58.696638 IP 10.0.10.14 > LabServer.helion.demo: GREv0, length 106: IP 192.168.10.9 > 192.168.10.8: ICMP echo reply, id 5591, seq 95, length 64
20:10:58.697056 IP 10.0.10.14 > LabServer.helion.demo: GREv0, length 170: IP 192.168.10.8.ssh > 10.0.10.254.53007: Flags [P.], seq 1784770515:1784770611, ack 1561286070, win 3525, options [nop,nop,TS val 334628841 ecr 1365586260], length 96
20:10:58.697108 IP 10.0.10.14 > LabServer.helion.demo: GREv0, length 122: IP 192.168.10.8.ssh > 10.0.10.254.53007: Flags [P.], seq 96:144, ack 1, win 3525, options [nop,nop,TS val 334628841 ecr 1365586260], length 48
```

Vraťte se do compute node a mirorring zrušte.

```
ovs-vsctl clear bridge br-int mirrors
ovs-vsctl del-port br-int gre0
```

3.2. Flow monitoring

OpenvSwitch použitý v rámci Helion OpenStack vám může poskytnout vizibilitu do datových toků generovaných jednotlivými VM, nicméně je důležité počítat s tím, že to samo o sobě nerozlišuje jednotlivé tenanty a vhledem možnosti překrývajících se adres to může některé analyzátory mást. Pokud vám jde o víc, než jen přehled o tom co se ve virtuální síti děje, bude potřeba striktně rozlišovat reportované interface, podle kterých poznáte konkrétní VM.

OVS podporuje jak flow metody (NetFlow a IPFIX), které mají výhodu ve velké přesnosti, tak také sampling sFlow, kde sice je nižší přesnost, za to ale analyzátor dostává ve vzorcích více informací (část payload), což zase přináší jiné možnosti v analýze. Většina analyzátorů jako je HP iMC NTA podporuje obě metody.

3.2.1. NetFlow

OVS umožňuje získávat informace o datových tocích a exportovat je ve formátu NetFlow nebo IPFIX stejně, jako to dokáže třeba váš router. Připojte se do vybraného compute node a zapneme export NetFlow záznamů do kolektoru, který máme spuštěný na adrese 10.0.10.53 a portu 9995.

```
root@overcloud-novacompute0-vli5de2egecg:~# sudo ovs-vsctl -- set Bridge br-int netflow=@nf --   --id=@nf  \
>   create   NetFlow   targets=\"10.0.10.53:2055\" \
>  active-timeout=20
```

Pro vyzkoušení použijeme jednoduchý open source NetFlow collector nfcapd na labovém serveru 10.0.10.53:

```
root@LabServer:~# nfcapd -p 9995 -E -l .

Flow Record:
  Flags        =              0x00 Unsampled
  export sysid =                 1
  size         =                52
  first        =        1432977759 [2015-05-30 11:22:39]
  last         =        1432977759 [2015-05-30 11:22:39]
  msec_first   =               400
  msec_last    =               400
  src addr     =     192.168.44.15
  dst addr     =     192.168.44.14
  src port     =              9200
  dst port     =             39705
  fwd status   =                 0
  tcp flags    =              0x00 ......
  proto        =                 6
  (src)tos     =                 0
  (in)packets  =                 3
  (in)bytes    =               322
```

```
  input        =              167
  output       =              173
  src as       =                0
  dst as       =                0


Flow Record:
  Flags        =             0x00 Unsampled
  export sysid =                1
  size         =               52
  first        =       1432977759 [2015-05-30 11:22:39]
  last         =       1432977759 [2015-05-30 11:22:39]
  msec_first   =              404
  msec_last    =              404
  src addr     =    192.168.44.15
  dst addr     =    192.168.44.13
  src port     =             9200
  dst port     =            41506
  fwd status   =                0
  tcp flags    =             0x00 ......
  proto        =                6
  (src)tos     =                0
  (in)packets  =                3
  (in)bytes    =              322
  input        =              167
  output       =              169
  src as       =                0
  dst as       =                0
```

Kolektor zachytil surová data do souboru, nad kterým můžeme s nfdump dělat další výpiy, filtrace a exporty.

```
root@LabServer:~# nfdump -r nfcapd.201505301122
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets     Bytes Flows
2015-05-30 11:22:39.400     0.000 TCP      192.168.44.15:9200   ->     192.168.44.14:39705         3       322     1
2015-05-30 11:22:39.404     0.000 TCP      192.168.44.15:9200   ->     192.168.44.13:41506         3       322     1
2015-05-30 11:22:39.400     0.000 TCP      192.168.44.14:39705  ->     192.168.44.15:9200          5       340     1
2015-05-30 11:22:39.404     0.000 TCP      192.168.44.13:41506  ->     192.168.44.15:9200          5       340     1
2015-05-30 11:22:39.471     0.000 ICMP      192.168.10.8:0      ->      192.168.10.9:8.0            1        98     1
2015-05-30 11:22:39.471     0.000 TCP        10.0.10.254:44556  ->      192.168.10.8:22             1        66     1
2015-05-30 11:22:39.471     0.000 ICMP      192.168.10.9:0      ->      192.168.10.8:0.0            1        98     1
2015-05-30 11:22:39.471     0.000 TCP       192.168.10.8:22     ->       10.0.10.254:44556          2       276     1
2015-05-30 11:22:40.181     0.004 ICMP        10.0.30.90:0      ->         10.0.10.3:0.0            5       550     1
2015-05-30 11:22:40.192     0.005 ICMP        10.0.30.92:0      ->         10.0.10.3:0.0            5       550     1
2015-05-30 11:22:40.187     0.006 ICMP        10.0.30.91:0      ->         10.0.10.3:0.0            5       550     1
2015-05-30 11:22:40.246     0.015 ICMP        10.0.30.93:0      ->         10.0.10.3:0.0            5       550     1
2015-05-30 11:22:40.176     0.005 ICMP        10.0.30.89:0      ->         10.0.10.3:0.0            5       550     1
2015-05-30 11:22:41.098     0.000 TCP         172.17.0.14:3306  ->       172.17.0.12:38818          1        74     1
2015-05-30 11:22:40.872     0.000 TCP         172.17.0.14:3306  ->       172.17.0.10:50064          1        74     1
2015-05-30 11:22:41.100     0.000 TCP         172.17.0.14:3306  ->       172.17.0.15:42048          1        74     1
2015-05-30 11:22:40.871     0.001 TCP         172.17.0.10:50064 ->       172.17.0.14:3306           2       140     1
2015-05-30 11:22:41.100     0.000 TCP         172.17.0.15:42048 ->       172.17.0.14:3306           2       140     1
2015-05-30 11:22:41.098     0.002 TCP         172.17.0.12:38818 ->       172.17.0.14:3306           2       140     1
2015-05-30 11:22:41.182     0.000 TCP          10.0.30.86:9001  ->       10.0.30.200:52251          1        74     1
2015-05-30 11:22:41.403     0.030 TCP      192.168.44.13:41507  ->     192.168.44.15:9200          6       414     1
2015-05-30 11:22:41.402     0.035 TCP      192.168.44.14:39707  ->     192.168.44.15:9200          6       414     1
2015-05-30 11:22:41.402     0.035 TCP      192.168.44.15:9200   ->     192.168.44.14:39707         4       396     1
2015-05-30 11:22:41.404     0.029 TCP      192.168.44.15:9200   ->     192.168.44.13:41507         4       396     1
2015-05-30 11:22:43.101     0.000 TCP         172.17.0.14:3306  ->       172.17.0.12:38819          1        74     1
2015-05-30 11:22:43.102     0.000 TCP         172.17.0.14:3306  ->       172.17.0.15:42049          1        74     1
2015-05-30 11:22:42.874     0.000 TCP         172.17.0.14:3306  ->       172.17.0.10:50065          1        74     1
2015-05-30 11:22:43.102     0.002 TCP         172.17.0.15:42049 ->       172.17.0.14:3306           2       140     1
```

```
2015-05-30 11:22:43.100     0.000 TCP        172.17.0.12:38819 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:42.873     0.003 TCP        172.17.0.10:50065 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:43.187     0.000 TCP         10.0.30.86:9001  ->      10.0.30.200:52353         1        74     1
2015-05-30 11:22:43.434     0.031 TCP      192.168.44.14:39708 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:43.433     0.032 TCP      192.168.44.13:41508 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:43.434     0.031 TCP      192.168.44.15:9200  ->    192.168.44.13:41508         4       396     1
2015-05-30 11:22:43.435     0.030 TCP      192.168.44.15:9200  ->    192.168.44.14:39708         4       396     1
2015-05-30 11:22:44.082     0.000 UDP      192.168.21.2:53     ->      10.0.10.54:60873          1        82     1
2015-05-30 11:22:44.082     0.000 UDP       10.0.10.54:60873 ->      192.168.21.2:53           1        82     1
2015-05-30 11:22:39.690     4.764 TCP      192.168.40.5:46733 ->    169.254.169.254:80          26      6750     1
2015-05-30 11:22:39.690     4.764 TCP    169.254.169.254:80     ->    192.168.40.5:46733          45      7210     1
2015-05-30 11:22:44.874     0.001 TCP        172.17.0.10:50066 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:45.102     0.001 TCP        172.17.0.15:42050 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:45.100     0.003 TCP        172.17.0.12:38820 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:44.875     0.000 TCP        172.17.0.14:3306  ->      172.17.0.10:50066         1        74     1
2015-05-30 11:22:45.103     0.000 TCP        172.17.0.14:3306  ->      172.17.0.15:42050         1        74     1
2015-05-30 11:22:45.101     0.000 TCP        172.17.0.14:3306  ->      172.17.0.12:38820         1        74     1
2015-05-30 11:22:45.463     0.032 TCP      192.168.44.15:9200  ->    192.168.44.14:39710         4       396     1
2015-05-30 11:22:40.380     4.827 TCP      192.168.44.14:41082 ->    169.254.169.254:80          29      7184     1
2015-05-30 11:22:45.465     0.026 TCP      192.168.44.13:41509 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:45.465     0.026 TCP      192.168.44.15:9200  ->    192.168.44.13:41509         4       396     1
2015-05-30 11:22:45.463     0.032 TCP      192.168.44.14:39710 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:45.309     0.002 TCP        10.0.30.84:55541 ->       10.0.10.3:5672           6      1228     1
2015-05-30 11:22:45.209     0.002 TCP        10.0.30.84:43893 ->      10.0.10.13:8000           4       272     1
2015-05-30 11:22:40.380     4.827 TCP    169.254.169.254:80     ->    192.168.44.14:41082         44      7216     1
2015-05-30 11:22:45.195     0.000 TCP         10.0.30.86:9001  ->      10.0.30.200:52451         1        74     1
2015-05-30 11:22:47.103     0.000 TCP        172.17.0.14:3306  ->      172.17.0.12:38821         1        74     1
2015-05-30 11:22:47.105     0.000 TCP        172.17.0.14:3306  ->      172.17.0.15:42051         1        74     1
2015-05-30 11:22:46.903     0.040 TCP      192.168.44.14:54894 ->    192.168.44.15:3306          4       429     1
2015-05-30 11:22:47.103     0.000 TCP        172.17.0.12:38821 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:47.104     0.003 TCP        172.17.0.15:42051 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:46.876     0.003 TCP        172.17.0.10:50067 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:46.903     0.000 TCP      192.168.44.15:3306  ->    192.168.44.14:54894         2       840     1
2015-05-30 11:22:46.877     0.000 TCP        172.17.0.14:3306  ->      172.17.0.10:50067         1        74     1
2015-05-30 11:22:47.493     0.034 TCP      192.168.44.13:41510 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:47.495     0.028 TCP      192.168.44.14:39711 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:47.493     0.034 TCP      192.168.44.15:9200  ->    192.168.44.13:41510         4       396     1
2015-05-30 11:22:47.496     0.027 TCP      192.168.44.15:9200  ->    192.168.44.14:39711         4       396     1
2015-05-30 11:22:47.201     0.000 TCP         10.0.30.86:9001  ->      10.0.30.200:52497         1        74     1
2015-05-30 11:22:40.470     9.000 ICMP     192.168.10.9:0      ->    192.168.10.8:0.0            10       980     1
2015-05-30 11:22:49.105     0.002 TCP        172.17.0.12:38822 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:48.879     0.000 TCP        172.17.0.10:50068 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:49.106     0.001 TCP        172.17.0.15:42052 ->      172.17.0.14:3306          2       140     1
2015-05-30 11:22:49.105     0.000 TCP        172.17.0.14:3306  ->      172.17.0.12:38822         1        74     1
2015-05-30 11:22:49.107     0.000 TCP        172.17.0.14:3306  ->      172.17.0.15:42052         1        74     1
2015-05-30 11:22:48.879     0.000 TCP        172.17.0.14:3306  ->      172.17.0.10:50068         1        74     1
2015-05-30 11:22:49.528     0.027 TCP      192.168.44.13:41511 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:49.528     0.027 TCP      192.168.44.15:9200  ->    192.168.44.13:41511         4       396     1
2015-05-30 11:22:49.523     0.032 TCP      192.168.44.14:39712 ->    192.168.44.15:9200          6       414     1
2015-05-30 11:22:49.524     0.031 TCP      192.168.44.15:9200  ->    192.168.44.14:39712         4       396     1
Summary: total flows: 78, total bytes: 48491, total packets: 367, avg bps: 38200, avg pps: 36, avg bpp: 132
Time window: 2015-05-30 11:22:39 - 2015-05-30 11:22:49
Total flows processed: 78, Blocks skipped: 0, Bytes read: 4176
Sys: 0.003s flows/second: 19622.6    Wall: 0.006s flows/second: 12117.4
```

Následně NetFlow export zase zrušíme.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl clear Bridge br-int netflow
```

3.2.2. sFlow

Pro nastavení sFlow musíme kromě definice kolektoru ještě zadat další údaje. Sampling rate (sFlow zachytí každý X-tý paket), Polling rate (jak často sebere countery) a header size (jak velkou část vybraného paketu chceme poslat do analyzátoru). Připojte se do compute node a sFlow zapněte.

```
root@overcloud-novacompute0-vli5de2egecg:~#  ovs-vsctl -- --id=@sflow create sflow agent=eth0  \
> target=\"10.0.10.53:6343\" header=128 \
> sampling=100 polling=5 \
> -- set bridge br-int sflow=@sflow
```

Na cílové stanici použijeme jednoduchý nástroj sflowtool pro rozbalení a přečtení jednotlivých vzorků (nejedná se o jejich analázu, jen rozbalení – pro reálné použití použijte například HP iMC NTA).

```
root@LabServer:/opt/sflow/sflowtool-3.35# sflowtool -t | tcpdump -r -
reading from file -, link-type EN10MB (Ethernet)
11:45:13.000000 ARP, Request who-has 192.168.10.8 tell 192.168.10.9, length 32
11:45:14.000000 IP 10.0.10.54.58437 > 192.168.21.2.domain: 11761+ A? stun.client.akadns.net. (40)
11:45:17.000000 IP 172.17.0.15.33479 > 169.254.169.254.http: Flags [P.], seq 2512264303:2512264532, ack 957230571,
win 255, options [nop,nop,TS val 350003318 ecr 1095937596], length 229
11:45:21.000000 IP 192.168.44.14.40452 > 192.168.44.15.9200: Flags [R], seq 2256554003, win 0, length 0
11:45:21.000000 IP 10.0.30.92.42276 > 10.0.10.3.amqp: Flags [P.], seq 1354347799:1354347812, ack 489773881, win 221,
options [nop,nop,TS val 350117352 ecr 1375614141], length 13
11:45:21.000000 IP 10.0.30.86.56435 > 10.0.10.3.amqp: Flags [.], ack 3581118546, win 221, options [nop,nop,TS val
670627113 ecr 1375614150], length 0
11:45:21.000000 IP 169.254.169.254.http > 172.17.0.14.53610: Flags [P.], seq 1403557118:1403557271, ack 2979092076,
win 243, options [nop,nop,TS val 1095938856 ecr 350121398], length 153
11:45:22.000000 IP 192.168.44.15.mysql > 192.168.44.14.49158: Flags [P.], seq 4224772569:4224773062, ack 1949650615,
win 1250, options [nop,nop,TS val 670683767 ecr 670587642], length 493
11:45:22.000000 IP 172.17.0.14.53610 > 169.254.169.254.http: Flags [P.], seq 222:444, ack 294, win 238, options
[nop,nop,TS val 350121610 ecr 1095938933], length 222
11:45:25.000000 IP 192.168.44.13.42254 > 192.168.44.15.9200: Flags [S], seq 18764925, win 27200, options [mss
1360,sackOK,TS val 670628849 ecr 0,nop,wscale 7], length 0
11:45:32.000000 IP 192.168.44.15.50674 > 169.254.169.254.http: Flags [P.], seq 2466163253:2466163483, ack 80110350,
win 272, options [nop,nop,TS val 670686344 ecr 1095941486], length 230
11:45:33.000000 IP 169.254.169.254.http > 172.17.0.12.58790: Flags [P.], seq 947768543:947768694, ack 2509488609, win
419, options [nop,nop,TS val 1095941631 ecr 350004711], length 151
```

Vrátíme se do compute node a sFlow zase vypneme.

```
root@overcloud-novacompute0-vli5de2egecg:~# ovs-vsctl -- clear Bridge br-int sflow
```