Team:

|  |  |
| --- | --- |
| Khantil Choksi - khchoksi | Shubhankar Reddy - skatta2 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Staging Server IP |  |  | VM IP |  |  |
| 152.14.83.156 | ece792 | EcE792net! | 192.168.124.15 | ece792 | EcE792net! |

Helper Commands:

ip link set dev int\_name up/down

ipaddr add ip w mask dev int\_name

virsh attach-interface --domain vm\_name --type bridge --source bridge\_name

sudo tshark -i eth0 -T fields -e ip.src -e eth.src -e ip.dst -e eth.dst -e col.Protocol

sudo tshark -i eth1 -T fields -e ip.src -e ip.dst -e col.Protocol

sudo tshark -i eth0 -T fields -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -e col.Protocol

sudo tshark -i eth1 -T fields -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -e col.Protocol

ip route add 0.0.0.0/0 via 34.0.0.1 dev eth1

[**Problem 1:**](#_8qq9kht16x4j) **1**

[**Problem 2:**](#_gvo1x36vz6pl) **4**

[**Problem 3:**](#_l90ffv3e7wxh) **6**

[**Problem 4:**](#_vp0cir8a7tk) **15**

[**Problem 5:**](#_6stu92tpsyft) **16**

[**Problem 6:**](#_yq635ph0ku08) **33**

## Problem 1:

Questions:

* Can we extend the application by using Ansible, so that the script used for 1 & 2, can be executed on each hypervisor ?











After solving conflicts:



## Problem 2:

Creating 4 OVS bridges

swovs1

swovs2

swovs3

swovs4

To DO: Create image with installed packages for VM creation

Questions:

* Do we have to ask users for switch names?
* Do we also have to setup DHCP for ‘L3’ and ‘other’ networks?
* Why we need to create, define network first and then add bridge?

Refs:

<https://jamielinux.com/docs/libvirt-networking-handbook/routed-network.html>

## 

## Problem 3:

Design 1: All VMs from all tenants are connected to same bridge (in bridge mode).

Hypervisor 1:

sudo ip route add 192.168.12.0/24(Tenant subnet) via 192.168.123.66(Hypervisor’s IP)

sudo ip route add 192.168.11.0/24 () via 192.168.123.123 (Hypervisor’s IP)

In hypervisor 1:

sudo ip link add name vxlan0 type vxlan id 42 dev ens5 remote 192.168.123.66(hypervisor 2’s ip) dstport 4789

sudo ip link set dev vxlan0 up

sudo brctl addif virbr0 vxlan0

In hypervisor 2:

sudo ip link add name vxlan0 type vxlan id 42 dev ens5 remote 192.168.123.123(hypervisor 1’s ip) dstport 4789

sudo ip link set dev vxlan0 up

sudo brctl addif virbr0 vxlan0

Configuration:

Guest VM Config:



Hypervisor:

The ens4 interface is connected to OVS bridge swovs2.





1. **Design 1: All VMs from all tenants are connected to same bridge (in bridge mode).**
   1. **What are the disadvantages for tenants? Is a tenant’s traffic isolated from other tenants?**
      1. All the tenants’ VMs will be in the same subnet inside one hypervisor.
      2. It will not provide the security and isolated environment to the packets sent by one tenant. e.g. The VM of tenant1 will be able to ping and send packets to VM of tenant2 inside the same hypervisor. So, it will violate the isolated traffic promise for each tenant.
   2. **What, if anything, breaks if two tenants in the same hypervisor host use the same IP address?**
      1. Explanation:

We have configured other team’s VMs to be of same IP address 19.0.0.1

The ping is successful as one of the VMs on the other end responds to the ARP reply.

We lost connectivity to the VM with duplicate IP, unless we flush the ARP tables, and tried the ping again.

* Consider that we have following configuration:

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (19.0.0.2) | Tenant1 (19.0.0.1) |
| VM2 | Tenant2 (19.0.0.4) | Tenant2 (19.0.0.1) |

Below is the tcpdump capture when we tried to ping the other team (with duplicate IPs).



Screenshot for ifconfig on eth1, of tenant 1 on our side (Hypervisor 1)



Consider the above scenario, where the main issue over here is, the ping request from tenant2 - hypervisor1 - vm2 will not be able to decide whether it pinging to tenant2-hypervisor2-VM2 or tenant1-hypervisor2-VM1. So this is the drawback of Design 1 of attaching all VMs to single bridge.

* Now, we have configured as following:

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (19.0.0.2) | Tenant1 (19.0.0.1) |
| VM2 | Tenant2 (19.0.0.2) | Tenant2 (19.0.0.3) |

Below is the screenshot when we configured duplicate IPs on our (hypervisor 1) VMs.



Below is the wireshark (tshark) capture of the ping request and reply from our VM.



**Conclusion**: So datapath 2->1 breaks if the VMs with same IPs are on our hypervisor, datapath 9->10 breaks if they are on our peer hypervisor.

* 1. **What, if anything, breaks if two tenants in a different hypervisor host use the same IP address?**

We have configured as following:

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (19.0.0.3) | Tenant1 (19.0.0.1) |
| VM2 | Tenant2 (19.0.0.2) | Tenant2 (19.0.0.3) |

From hypervisor1 VM2 tenant2 is trying to ping to Hypervisor2 Tenant2 **(**his / her own VM)

$ ping 19.0.0.3

Datapath from (2) to onwards will not work as ARP finds the IP 19.0.0.3 in the same hypervisor.

* + 1. Connection to the tenant2’s VM on hypervisor 2 is lost.
    2. Below is the tcpdump of ping on the ens4 of hypervisor1, as we can observe, there are no ping packets from 19.0.0.2 exiting hypervisor1. As the switch has mapped its MAC address to tenant1’s VM1 which is connected to the same ovs switch.

Conclusion: So datapath 3->4 breaks as the ARP is resolved inside switch2 and the packets are tramsitted to the VMs of tenant2 from VM1 of tenant1.



* 1. **What, if anything, breaks if two tenants in the same hypervisor host use the same MAC address?**

Consider the following configuration:

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (19.0.0.2) (52:54:00:8e:f6:e4) | Tenant1 (19.0.0.1) |
| VM2 | Tenant2 (19.0.0.4) (52:54:00:8e:f6:e4) | Tenant2 (19.0.0.3) |



The above screenshot displays the configuration on VMs on Hypervisor1

Ping to 19.0.0.4 from VM1 of hypervisor1 will fail.

Datapath from (2) to VM2 fails.

* Now, consider the scenario that, having same MAC address on Hypervisor2 and VM from Hypervisor1 wants to ping to VMs on Hypervisor2. Following is the screenshot for that:

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (19.0.0.2) | Tenant1 (19.0.0.1)  (52:54:00:6c:17:ca) |
| VM2 | Tenant2 (19.0.0.4) | Tenant2 (19.0.0.3)  (52:54:00:6c:17:ca) |

****

We observed intermediate connectivity issues while trying to ping from vm on hypervisor1 to another vm on hypervisor2. Both had connectivity individually, but while switching the ping from one destination vm to another, there was a momentary loss of connectivity. We believe that the datapath from 9 to 10 breaks, as 2 same MAC addresses cannot be mapped to different ports on a switch.

**Conclusion**: Datapath from 9->10 breaks. (assuming we are trying to communicate with VM2 in hypervisor 2 from VM2 in hypervisor1, and VM1 and VM2 in hypervisor2 have the same MAC). VM2 in hypervisor 1 will be able to communicate with either of the VMs but not both simultaneously.

* 1. **What, if anything, breaks if two tenants in a different hypervisor host use the same MAC address?**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (19.0.0.2) | Tenant1 (19.0.0.1)  (52:54:00:6c:17:ca) |
| VM2 | Tenant2 (19.0.0.4)  (52:54:00:6c:17:ca) | Tenant2 (19.0.0.3) |

We observe that there is no connectivity between the VMs which have the same MAC addresses. Only one of the VMs with the same MAC addresses can participate in a conversation with the other VM.



Conclusion: If two VMs with same MAC address communicates, then ARP won’t be resolved. So datapath from 3 -> 4 will break.

* 1. **What about a VLAN based solution? Will it work to provide isolation? What are the limitations of this solution? No need to perform experiments for this question.**
     1. It will work, using VLANs we can effectively provide isolation between the VMs of two tenants. As the switch attaches the VLAN tag to the packet entering the switch from a particular port, it will work. We should still ensure that VMs of the same tenant do not have the same MAC/IP addresses.

1. **Design 2: Each tenant has its own bridge (in bridge mode)**

According to design 2, we need to connect the different tenant’s VMs to the tenant’s own bridge. But, on our hypervisor only ens4 interface is provided. So, we are not able to connect two tenant’s bridge to ens4 a the same time. (Datapath from 3 -> 4 for one tenant will break while for other tenant is working)

Diagram

* 1. **What are the disadvantages for the provider? Which resource in the hypervisor hosts will be a bottleneck?**
     1. The design is not feasible. As the number of switches may scale in each hypervisor, it is not possible for the provider to provision as many interfaces as the switches to each hypervisor.
  2. **What, if anything, breaks if two tenants in the same hypervisor host use the same IP address?**
     1. If they are connected to different switches beneath the hypervisor by using 2 different network interfaces, then nothing breaks, as both the tenant’s VMs are completely isolated from each other.
  3. **What, if anything, breaks if two tenants in the different hypervisor host use the same IP address?**
     1. As having different bridges for different tenants, it will provide network isolation between two tenants. So, everything will work and two tenants will be able to have same IP subnets across different hypervisor.
  4. **What, if anything, breaks if two tenants in the same hypervisor host use the same MAC address?**
     1. Even if two tenants use the same MAC address, nothing breaks as the ARP broadcast packets are isolated from each other by the two switches. There won’t be any MAC address conflicts between the tenants.
  5. **What, if anything, breaks if two tenants in a different hypervisor host use the same MAC address?**
     1. As having different bridges for different tenants, it will provide network isolation between two tenants. So, everything will work even if two tenants in different hypervisor host use the same MAC address. The ARP broadcast packets are still isolated from two tenants’ different bridges. Therefore, there won’t be any conflict.
  6. **Do we need VLANs in the hypervisor bridge or do VLANs in Physical L2 network suffice? No need to perform experiments for this question.**
     1. If there is a single L2 switch which is capable of detecting and transmitting tagged frames, then VLANs in the hypervisor will suffice.

1. **Design 1 vs. Design 2:**
   1. **Admin Hat: List trade-offs with Design 1 and Design 2.**
      1. **Design 1:**

* The admin has to make sure that, they don’t use same subnet address with other tenants inside the hypervisor, as well as across the hypervisor.
* The admin has to make sure that, the ping packets from their own tenant VM is reaching to their own other VMs. So, they have to make sure that, they are not sending packets to other tenant or receiving packets from other tenant.
* The admin also has to configure unique MAC address or resolve MAC address conflicts inside and across the hypervisor VMs.

ii. Design 2:

* The admin doesn’t have to worry about the subnet address conflicts, MAC address conflicts across different tenants.
  1. **Provider hat (hypervisor host's configuration point of view): List trade-offs with Design 1 and Design 2.**
     1. **Design 1:**
* Easier for the provider to configure, but doesn’t provide isolation between tenants. A lot of the bandwidth will be wasted on ARP broadcasts, as there is no L2 isolation.
  + 1. **Design 2:**
* More expensive to configure, in terms of providing different interfaces. But provides isolation and more available bandwidth compared to design 1 to each tenant.

**Reference:**

We have performed experiment with other Team (Staging Server IP:152.14.18.157 and VM ip: 192.168.122.38) by setting the infrastructure as explained above.

## 

## Problem 4:

Tshark capture of our VM pinging the neighbor VM in neighbor hypervisor



## 

## 

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.50) |
| VM2 | Tenant2 (119.0.0.3) | Tenant2 (117.0.0.51) |

**1. Design 1:**

**(a) What are the disadvantages for tenants? Is a tenant's traffic isolated from other tenants?**

* The tenant’s traffic isn’t isolated from other tenants.
* And as they use a bridge, the tenants should use the same subnet IP addresses within the same hypervisor.
* All the tenants’ VMs will be in the same subnet inside one hypervisor.

**(b) What, if anything, breaks if two tenants in the same hypervisor host use the same IP address?**

* L3 breaks from 2->1, as only one of the VMs is reachable.
* Case 1:

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.50) |
| VM2 | Tenant2 (119.0.0.2) | Tenant2 (117.0.0.51) |



* In the above we can observe that only one of the VMs is connected to the network at a point in time. When the ARP reply comes for 119.0.0.2, the last machine replying to ARP takes preference and the gateway uses that MAC address to send the reply. L3 datapath breaks from the gateway to the VM(with the duplicate IP).
* Case 2: Take another scenario

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.50) |
| VM2 | Tenant2 (119.0.0.3) | Tenant2 (117.0.0.50) |

**(c) What, if anything, breaks if two tenants in a different hypervisor host use the same IP address?**

****

Connectivity is lost to the VM on the other hypervisor having the same IP as the ARP broadcast never extends beyond the OVS switch. I.e. L3 datapath breaks at the gateway (3 -> 4), as the gateway is in the subnet, it doesn’t transmit the packet out of it. Hence L2 encap decap doesnt take place at the gateway.

The other VM on the hypervisor, which has the same IP, is out of the default subnet and doesn’t transmit packets to the gateway as the gateway doesn’t respond to the arp requests. L2 breaks from 2 -> 3.

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.50) |
| VM2 | Tenant2 (119.0.0.3) | Tenant2 (119.0.0.2) |

**(d) What, if anything, breaks if two tenants in the same hypervisor host use the same MAC address?**

* There is no communication between 2 IPs having the same MAC within the same hypervisor. The MAC is resolved to the port of the sending IP and the packet is sent back on the sending VM’s output interface. L2 breaks inside the OVS bridge.

****

This above screenshot displays the scenario when we try to ping the peer VM having duplicate IPs. Only one of the 2 peer devices is reachable at a point in time, if any other VM tries to communicate the original connection is lost. The data path from switch on peer hypervisor to the peer VM i.e. the L2 breaks (8) -> (9) datapath will break.

**(e) What, if anything, breaks if two tenants in a different hypervisor host use the same MAC address?**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2)  (52:54:00:b0:68:a7) | Tenant1 (117.0.0.50)  (52:54:00:b0:68:a7) |
| VM2 | Tenant2 (119.0.0.3) | Tenant2 (119.0.0.2) |

Ping Successful of two VMs having same mac address

****

Wireshark capture at ens4 of hypervisor

****

Conclusion: Nothing breaks.

**(f) What about a VLAN based solution for providing L3 connectivity to each VM? Will it work?**

**What are the limitations of this solution?**

**2. Design 2:**

**Topology Setup:**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.50) |
| VM2 | Tenant2 (121.0.0.2) | Tenant2 (118.0.0.51) |

**At our hypervisor:**

**Tenant1 VM:**

****

**Tenant2 VM:**



Hypervisor Configuration and Route

  
**(a) What are the disadvantages for the provider? Which resource in the hypervisor hosts will be a  
Bottleneck?**

* The provider needs to ensure that the subnets provisioned are mutually exclusive.
* The same tenant can not have the same subnet across different hypervisors.
* The physical interface ens4 of the hypervisor will be the bottleneck as all the tenants will share the bandwidth on it.

**(b) What, if anything, breaks if two tenants in the same hypervisor host use the same IP address?**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.50) |
| VM2 | Tenant2 (119.0.0.2) | Tenant2 (118.0.0.51) |

* Case 1: (Gateway’s ip is not changed)

The L2 datapath from 1 -> 2 breaks as VM(119.0.0.2) is in a different subnet(121.0.0.1/24). As they are in different subnets, the gateway doesn’t respond to the ARP requests from the VM.

* Case 2: When the gateway is also assigned an IP of 119.0.0.1, the L2 datapath between the VM(with the duplicate IP) and the gateway(119.0.0.1) breaks.

Screenshot for case 2



* Case 3: When we assign the gateway of other tenant as different IP of 119.0.0.10, the peer hypervisor loses connectivity to one of the subnets completely. The screenshot below displays the tcpdump when our peer hypervisor tries to reach us.

Case 3 Screenshot:

Screenshot: If we try to ping



Screenshot:



* In this below we can observe, if the gateway IPs are different then the VM with duplicate IP is able to ping both the gateways.



**(c) What, if anything, breaks if two tenants in a different hypervisor host use the same IP address?**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2) | Tenant1 (117.0.0.2) |
| VM2 | Tenant2 (117.0.0.2) | Tenant2 (118.0.0.51) |

* The L3 datapath breaks from 3 -> 4, as the IP subnet is connected locally the ping packet never crosses the gateway in the hypervisor and ends up in the local subnet.
* So basically Tenant1 wants the packet to send out to Tenant1’s other VM running on different hypervisor. But, the ICMP packet reaches to Tenant2’s same hypervisor.



**(d) What, if anything, breaks if two tenants in the same hypervisor host use the same MAC address?**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2)  (MAC Address: 52:54:00:b0:68:a7) | Tenant1 (117.0.0.2) |
| VM2 | Tenant2 (121.0.0.2)  (MAC Address: 52:54:00:b0:68:a7) | Tenant2 (118.0.0.51) |

Nothing breaks, as both the VMs are connected to different switches(in routed mode). Even when they have the same MAC address there is no conflict in the MAC table.

Screenshot shows that both the tenants are able to ping to their respective VMs on different hypervisor.

****

**(e) What, if anything, breaks if two tenants in a different hypervisor host use the same MAC address?**

|  |  |  |
| --- | --- | --- |
|  | Hypervisor1 | Hypervisor2 |
| VM1 | Tenant1 (119.0.0.2)  (MAC Address: 52:54:00:b0:68:a7) | Tenant1 (117.0.0.2) |
| VM2 | Tenant2 (121.0.0.2) | Tenant2 (118.0.0.51)  (MAC Address: 52:54:00:b0:68:a7) |

Nothing breaks, as both the VMs are connected to different switches(in routed mode). Even when they have the same MAC address there is no conflict in the MAC table.

And here we have L3 datapath so MAC conflicts will not break any datapaths.

Tenant1 across different hypervisors are able to ping each other:



Tenant2 across different hypervisors are able to ping each other:



**(f) Does a VLAN based solution in this design overcome any limitations of the VLAN based solution  
used in design 1?**

## 

## Problem 5:



Prerequisite: To enable IP forwarding in Router VM:

In the file /etc/sysctl.conf, change the line:

net.ipv4.ip\_forward = 0

to

net.ipv4.ip\_forward = 1

1. Set the topology as given in Figure 5. List IP/subnet plan for each L3 interfaces and provide Forwarding Table output for each VM.
   1. For router VM (khchoksilab2VM2)



* 1. For client VM (khchoksi)



c. For ServerVM1(khchoksi4)

d. For ServerVM2 (khchoksi5)



1. Ping from server to client

|  |
| --- |
| sudo iptables -t nat -I PREROUTING 1 -p icmp -s 20.0.0.2 -j DNAT --to-destination 10.0.0.3 sudo iptables -t nat -I PREROUTING 1 -p icmp -s 20.0.0.2 -j DNAT --to-destination 10.0.0.4 sudo iptables -t nat -I PREROUTING 1 -p icmp -d 20.0.0.2 -j SNAT --to 20.0.0.1 |



Following are wireshark captures if do ping from Server VM1 (10.0.0.3) to Client (20.0.0.2):

* Wireshark at Client VM



* Wireshark at RouterVM eth1 (connected to public net)



* Wireshark at RouterVM eth2 (connected to private net)



* Wireshark at ServerVM (10.0.0.3)





1. Configure NAT / PAT proxy settings

iptables -t nat -I PREROUTING 1 -p tcp --dport 2000 -j DNAT --to 10.0.0.3:22

iptables -t nat -I PREROUTING 1 -p tcp --dport 2001 -j DNAT --to 10.0.0.4:22

iptables -I FORWARD 1 -i eth2 -o eth1 -p tcp --dport 22 -j ACCEPT

iptables -t nat -I POSTROUTING 1 -p tcp -d 10.0.0.3 --dport 22 -j SNAT --to-source 10.0.0.1

iptables -t nat -I POSTROUTING 1 -p tcp -d 10.0.0.4 --dport 22 -j SNAT --to-source 10.0.0.1

|  |
| --- |
| iptables -t nat -I PREROUTING 1 -p tcp -s 20.0.0.2 --dport 2000 -j DNAT --to 10.0.0.3:22 iptables -t nat -I PREROUTING 1 -p tcp -s 20.0.0.2 --dport 2001 -j DNAT --to 10.0.0.4:22 iptables -I FORWARD 1 -i eth2 -o eth1 -p tcp -s 20.0.0.2 --dport 22 -j ACCEPT iptables -t nat -I POSTROUTING 1 -p tcp -d 10.0.0.3 --dport 22 -j SNAT --to-source 10.0.0.1 iptables -t nat -I POSTROUTING 1 -p tcp -d 10.0.0.4 --dport 22 -j SNAT --to-source 10.0.0.1 |





SSH from Client to Server 1



SSH from Client to Server 2



Wireshark Captures for ssh from client vm to Server 2:

i.e. ssh root@20.0.0.1 -p 2001

Following are screenshots for the same:

Wireshark capture at eth2 (connected to publicnet) of RouterVM



Wireshark capture at eth1 (connected to privatenet) of RouterVM



Wireshark capture at eth1 (connected to privatenet) of Server2VM (10.0.0.4)



**4.**

1. **Defining Load:** Send TCP packets for port range 2000:2499 to Server 1(10.0.0.3) and 2500:3000 to Server 2(10.0.0.4). (e.g. in a real-life scenario, let's say we have many clients who wants to send TCP packets to our servers’ 22 port. Also, assume that server 1,2 are running same application on port 22. So we will load balance based on port number ranges from clients. But client will not know from which server their data is being served.)
2. **Configure the balancing knob at the RouterVM.**

|  |
| --- |
| **iptables -t nat -I PREROUTING 1 -p tcp -s 20.0.0.0/24 --dport 2000:2499 -j DNAT --to 10.0.0.3:22 iptables -t nat -I PREROUTING 1 -p tcp -s 20.0.0.0/24 --dport 2500:3000 -j DNAT --to 10.0.0.4:22 iptables -I FORWARD 1 -i eth2 -o eth1 -p tcp -s 20.0.0.0/24 --dport 22 -j ACCEPT iptables -t nat -I POSTROUTING 1 -p tcp -d 10.0.0.3 --dport 22 -j SNAT --to-source 10.0.0.1 iptables -t nat -I POSTROUTING 1 -p tcp -d 10.0.0.4 --dport 22 -j SNAT --to-source 10.0.0.1** |

1. **Verify that the load balancing mechanism is working.**

From client machine:

|  |
| --- |
| $ssh root@20.0.0.1 -p 2100 (so it will redirect to Server1) |



|  |
| --- |
| $ssh root@20.0.0.1 -p 2600 (so it will redirect to Server2) |



**IP Tables configuration:**

****

****

**Wireshark Captures:**

**Ex1: Traffic forwarded to Server1**

$ssh root@20.0.0.1 -p 2100 (so it will redirect to Server1)

Wireshark capture at eth0 (connected to privatenet) of Server2VM (10.0.0.3)

****

Wireshark packet capture at eth1 (connected to privatenet) of the router VM.

****

Wireshark packet capture at eth2 (connected to publicnet) of the router VM.



**Ex2: Traffic forwarded to Server2**

$ssh root@20.0.0.1 -p 2600 (so it will redirect to Server2)

Wireshark capture at eth1 (connected to privatenet) of Server2VM (10.0.0.4)

****

Wireshark packet capture at eth1 (connected to privatenet) of the router VM.



Wireshark packet capture at eth2 (connected to publicnet) of the router VM.



## 

## Problem 6:

Diagram:



Setup commands:

At Tenant1 Subnet 1

|  |
| --- |
| sudo iptables -t nat -A POSTROUTING -s 30.0.0.0/24 ! -d 30.0.0.0/24 -j MASQUERADE |

For Tenant1 Subnet 2

|  |
| --- |
| sudo iptables -t nat -A POSTROUTING -s 32.0.0.0/24 ! -d 32.0.0.0/24 -j MASQUERADE |

**1. Demonstrate the L2 isolation between two subnets of the same tenant. (Hint: Broadcast should be restricted and VMs can have same MAC addresses).**

To prove L2 isolation, we show that the ARP broadcasts from one subnet of tenant1 are not transmitted on the other subnet of tenant1, and having same MAC addresses doesn’t break anything.

Ex. If T1VM2 (34.0.0.2) tries to ARP for 34.0.0.3 which not actually present in the subnet2 (t1br1), it will not send any of the ARP broadcasts to subnet1 (t1br0).



Wireshark on T1VM1 port eth2



**Ex. Now, we set the same MAC address for both T1VM1 and T1VM2.**

T1VM1



T1VM2



Ping from Tenant1’s one VM to another VM is successful even though tenant1 has same MAC address in his VMs.



IP tables configuration at hypervisor: (please see the last two POSTROUTING rules)



**2. Demonstrate the L3 isolation between two tenants.**

|  |
| --- |
|  |

inside blue\_ns

sudo iptables -t nat -A POSTROUTING -s 30.0.0.0/24 ! -d 30.0.0.0/24 -j MASQUERADE

sudo iptables -t nat -A POSTROUTING -s 34.0.0.0/24 ! -d 34.0.0.0/24 -j MASQUERADE

inside red\_ns

sudo iptables -t nat -A POSTROUTING -s 40.0.0.0/24 ! -d 40.0.0.0/24 -j MASQUERADE

sudo iptables -t nat -A POSTROUTING -s 34.0.0.0/24 ! -d 34.0.0.0/24 -j MASQUERADE

inside hypervisor

sudo iptables -t nat -A POSTROUTING -s 32.0.0.0/24 ! -d 32.0.0.0/24 -j MASQUERADE

sudo iptables -t nat -A POSTROUTING -s 42.0.0.0/24 ! -d 42.0.0.0/24 -j MASQUERADE

routes in hypervisor

ip route add 32.0.0.0/24 via 32.0.0.2 dev veth0

ip route add 42.0.0.0/24 via 42.0.0.2 dev veth8



Iptabels in hypervisor:



* Screenshot of iptables of blue\_ns
* Screenshot of iptables of red\_ns

• VMs of tenant 1 should not be able to ping to tenant 2 VMs whether they have same or different  
IP subnets.

Screenshot of Tenant1 VM1 (30.0.0.2) tries to ping to Tenant2 VM1(40.0.0.2):



Screenshot of Tenant1 VM2 (34.0.0.2) tries to ping to Tenant2 VM2(34.0.0.3) so they are in the same subnet but won’t be able to ping each other:



**• In another experiment, both tenants use one subnet that is common (e.g., 10.0.0.0/8) and one  
that is different. The hosts in the common subnet for tenant red and tenant blue should be able  
to ping the internet.**

Tenant1VM2 (34.0.0.2) tries to ping to internet (google.com) and at the same time Tenant2VM2 (34.0.0.3) tries to ping to internet (google.com).



**3. Forwarding table and IP tables in Host hypervisor. In this experiment, make sure not to use the  
hypervisor host’s default forwarding table/IP tables. Configure a new network namespaces (call it  
provider ns ) in the hypervisor. Implement and verify the following policies in the provider network  
namespace.**

We assume the internet to be 54.0.0.2 which is present in a separate namespace called the provider\_ns

The topology we use for the experiment is shown below:

Routes:

blue\_ns

sudo ip route add 30.0.0.0/24 dev veth3

sudo ip route add 34.0.0.0/24 dev veth5

sudo ip route add 0.0.0.0/0 via 32.0.0.1 dev veth1



red\_ns

sudo ip route add 0.0.0.0/0 via 42.0.0.1 dev veth9



provider\_ns

sudo ip route add 0.0.0.0/0 via 54.0.0.1 dev veth12



(a) Internet policy. Allow ICMP traffic for both tenants. Allow SSH traffic for only the blue tenant.

blue\_ns

sudo iptables -t nat -A POSTROUTING -s 30.0.0.0/24 ! -d 30.0.0.0/24 -j MASQUERADE

sudo iptables -t nat -A POSTROUTING -s 34.0.0.0/24 ! -d 34.0.0.0/24 -j MASQUERADE

red\_ns

sudo iptables -t nat -A POSTROUTING -s 40.0.0.0/24 ! -d 40.0.0.0/24 -j MASQUERADE

sudo iptables -t nat -A POSTROUTING -s 34.0.0.0/24 ! -d 34.0.0.0/24 -j MASQUERADE

indise provider\_ns

provider\_ns

sudo iptables -t nat -A POSTROUTING -p ICMP -s 32.0.0.0/24 ! -d 32.0.0.0/24 -j MASQUERADE

sudo iptables -t nat -A POSTROUTING -p ICMP -s 42.0.0.0/24 ! -d 42.0.0.0/24 -j MASQUERADE

Hypervisor

ip route add 54.0.0.0/24 dev veth13

sudo iptables -t nat -A POSTROUTING -p ICMP -s 54.0.0.0/24 ! -d 54.0.0.0/24 -j MASQUERADE

Screenshot of iptables inside the provider\_ns

Screenshot of iptables of hypervisor

SSH traffic:

provider\_ns

sudo iptables -t nat -D POSTROUTING -p TCP -s 32.0.0.0/24 --d-port 22 ! -d 32.0.0.0/24 -j MASQUERADE

hypervisor:

sudo iptables -t nat -A POSTROUTING -p TCP -s 54.0.0.0/24 --d-port 22 ! -d 54.0.0.0/24 -j MASQUERADE

Screenshot of ICMP traffic for both tenants:



For SSH Forwarding for blue tenant:

We have to Forwarding iptables rule in blue\_ns and provider\_ns

blue\_ns  
iptables -I FORWARD 1 -i veth3 -o veth1 -p tcp --dport 22 -j ACCEPT

block ssh traffic from red\_ns

iptables -I FORWARD 1 -p TCP --dport 22 ! -d 30.0.0.0/24 -j DROP



provider\_ns

iptables -I FORWARD 1 -i veth0 -o veth12 -p tcp --dport 22 -j ACCEPT

sudo iptables -t nat -A POSTROUTING -p TCP -s 32.0.0.0/24 --dport 22 ! -d 32.0.0.0/24 -j MASQUERADE

Screenshot of iptables rules inside provider\_ns



Screenshot of Tenant1 (blue subnet) who is able to ssh to 54.0.0.2 (provider\_ns ip)







The above displays the screenshot of a VM in red\_ns, which can ping an IP in the internet, but can't ssh to the host VM or any other VM of a different tenant.

The following screenshot shows that, the SSH packet is being received at veth7 but it is dropped and not able to capture at veth9.



**(b) Local L3 policy. Allow red tenant and blue tenant to ssh each other’s VM, provided the subnets  
are different.**





The above screenshot displays that the ssh traffic from 40.0.0.2 doesn’t reach the internet(54.0.0.2)

provider\_ns

sudo ip route add 40.0.0.0/24 via 42.0.0.2 dev veth8

sudo ip route add 30.0.0.0/24 via 32.0.0.2 dev veth0

provider\_ns

iptables -I FORWARD 1 -i veth0 -o veth8 -p tcp --dport 22 -j ACCEPT

iptables -I FORWARD 1 -i veth8 -o veth0 -p tcp --dport 22 -j ACCEPT

New ones

iptables -I FORWARD 1 -p TCP -i veth0 -o veth8 -j ACCEPT

iptables -I FORWARD 1 -p TCP -i veth8 -o veth0 -j ACCEPT

red\_ns

iptables -I FORWARD 1 -i veth7 -o veth9 -p tcp --dport 22 -j ACCEPT

iptables -I FORWARD 1 -i veth9 -o veth7 -p tcp --dport 22 -j ACCEPT

iptables -I FORWARD 1 -p TCP -d 54.0.0.0/24 --dport 22 -j DROP

blue\_ns

iptables -I FORWARD 1 -i veth1 -o veth3 -p tcp --dport 22 -j ACCEPT

iptables -D FORWARD -i veth3 -o veth1 -p tcp --dport 22 -j ACCEPT

iptables -I FORWARD 1 -p TCP -s 34.0.0.0/24 --dport 22 -d 34.0.0.0/24 -p tcp --dport 22 -j DROP

sudo iptables -t nat -A POSTROUTING -p TCP -s 32.0.0.0/24 --dport 22 -d 54.0.0.0/24 -j MASQUERADE

Reference: <https://askubuntu.com/questions/466445/what-is-masquerade-in-the-context-of-iptables>

Reference:

* <https://www.crybit.com/how-to-save-current-iptables-rules/>
* <https://serverfault.com/questions/201186/iptables-forwarding-ssh-ports>
* <https://blog.scottlowe.org/2013/09/04/introducing-linux-network-namespaces/>