CS3543 Lab Assignment for Jan 24th (Deadline: 23:59 on February 9th (SUN), 2020)

Member 1: SAI HARSHA KOTTAPALLI (CS17BTECH11036)
Member 2: SAGAR JAIN (CS17BTECH11034)

General Information

1. This assignment is a pair assignment. The same mark will be offered to the pair of students regardless of individual contributions.
2. The assignment is customized for Ubuntu + KVM environment. It is highly recommended for non-Ubuntu users to enable dual boot on your laptop computer and install Ubuntu. If you would like to work on another operating system and virtualization platform, you need to interpret the Ubuntu/KVM terminology to another environment's terminology.
3. Each pair can create a locally copy of this question file, give the answer to the local copy, and submit in a form of PDF file.
4. Only one submission is good enough as far as the student name and ID are properly mentioned.
5. Do not send any private comment to separately mention the buddy.

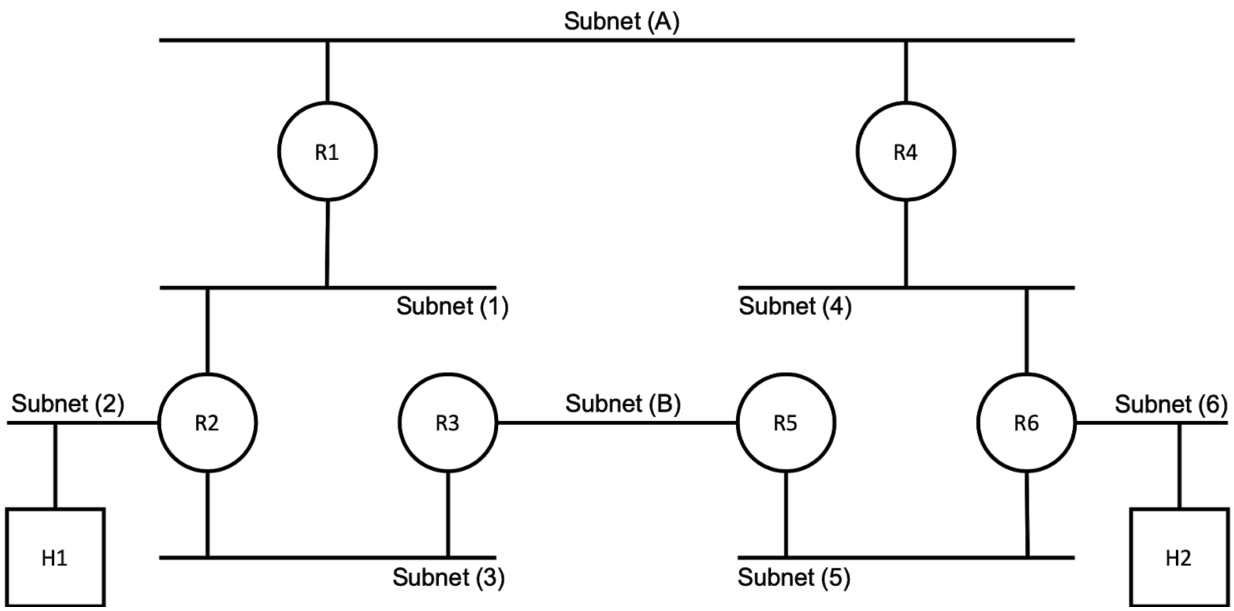


Fig.1. Blank Network Diagram

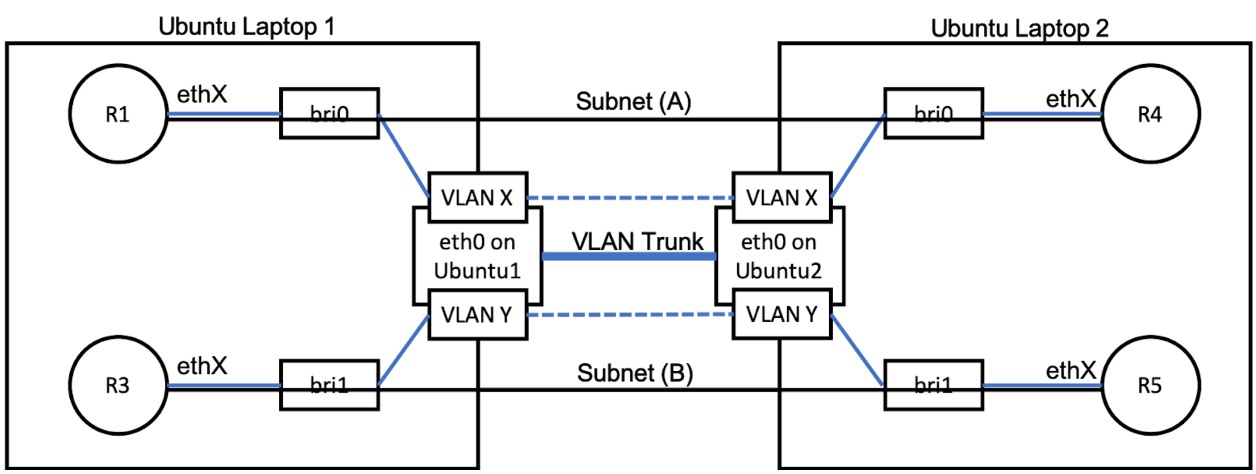


Fig.2. Conceptual Illustration of VLAN Configuration on Ubuntu Laptops 1 and 2

(Instruction)

This assignment requires to directly connect two Ubuntu laptops using a LAN cable to form a slightly bigger network than the previous assignment as shown in Fig. 1. In order to enable inter-router connections via Subnets (A) and (B) between the Ubuntu laptops, VLAN I/F (for VLAN Trunk) needs to be created on the physical LAN port of both Ubuntu laptops, and

VLAN I/F needs to be attached to the corresponding bridge I/F as illustrated in Fig.2. Explore the ubuntu configuration 1) to create VLANs on Ubuntu, 2) to configure VLAN I/F (   ), and 3) attach VLAN I/F to a bridge I/F, 4) to let the traffic go through a separate VLAN/Bridge between respective pairs of VMs {R1 and R4} and {R3 and R5}.

Question 0.
Complete the following table about the VLAN and Bridge configurations for Subnets (A) and (B). It is strongly recommended to unify the bridge name between Laptops 1 and 2 for each subnet to avoid confusion.

<table>
<tr><td></td><td>For Subnet (A)</td><td>For Subnet (B)</td></tr>
<tr><td>VLAN ID</td><td>10</td><td>20</td></tr>
<tr><td>Name of VLAN I/F on Laptop 1</td><td>eth1.10@eth1</td><td>eth1.20@eth1</td></tr>
<tr><td>Name of VLAN I/F on Laptop 2</td><td>eth2.10@eth2</td><td>eth2.20@eth2</td></tr>
<tr><td>Name of Bridge I/F on Laptop 1</td><td rowspan="2">bria_left, bria_right</td><td rowspan="2">brib_left, brib_right</td></tr>
<tr><td>Name of Bridge I/F on Laptop 2</td></tr>
</table>

Question 1.
Assign the necessary configuration (NIC and IPv4/v6 addresses) to implement the network illustrated in Fig. 1, note down the configuration in the network diagram, and insert the update network diagram as an image as the answer to the question.

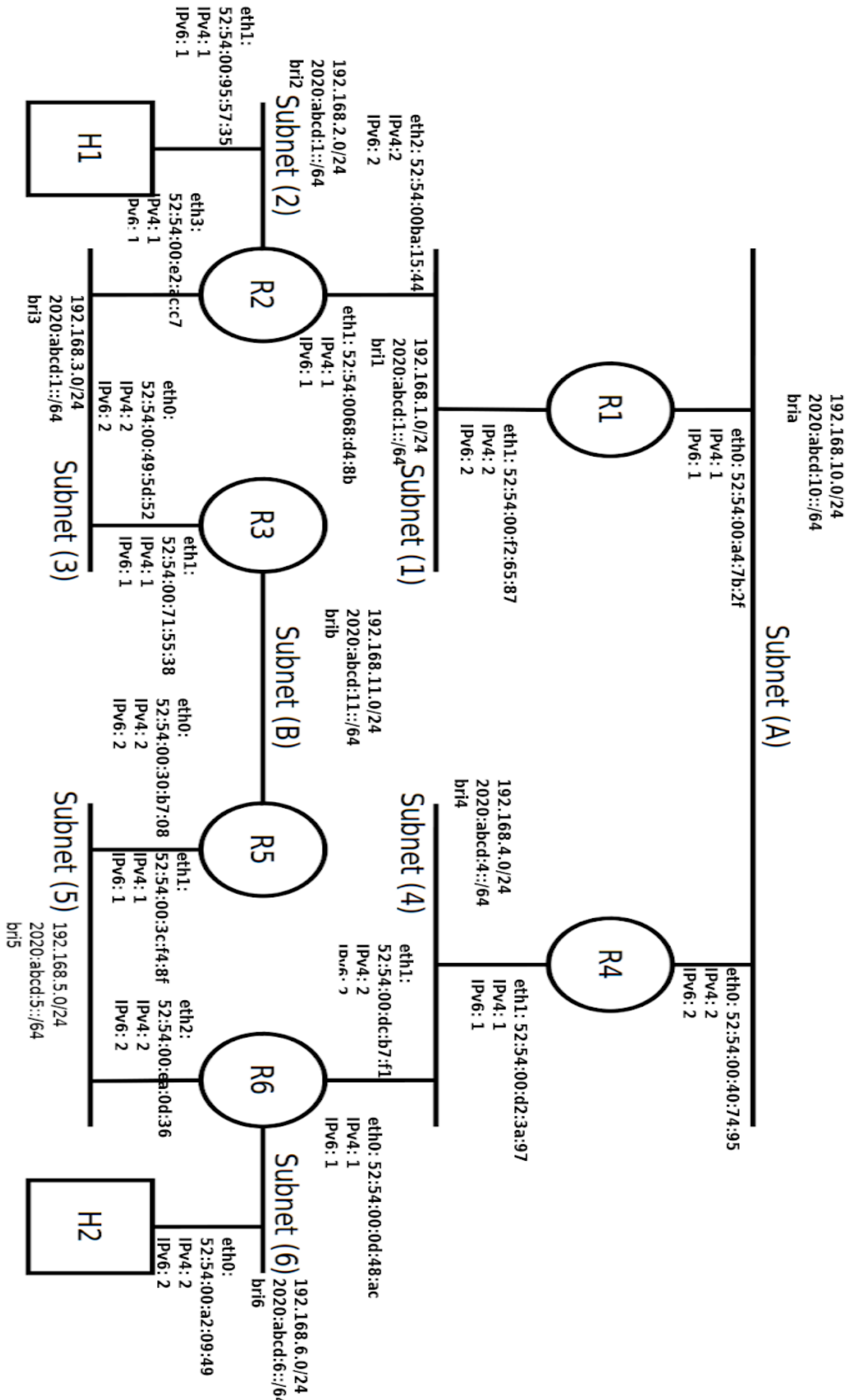
Supplemental Instructions for Question 1.

This assignment does not provide any addressing information. It must be determined and noted down in the network diagram by yourself. Make sure that the following minimum information are clearly visible.

a. To each subnet: Bridge Name, IPv4 Prefix, IPv6 Prefix
b. To each NIC: I/F Name, MAC Address, IPv4 Address, IPv6 Address

You may use the base network diagram given in the supplemental power-point file or your hand illustration. The example of subnet and

NIC information is also available in the same power-point file. If you don't have a better idea, follow the example.



Question 2.
Configure static routes so that the traffic between H1 and H2 goes through Subnet (A) for both directions. Answer by inserting the screen captures of routing table of R2 and R6, and traceroute results between H1 and H2.

R6 Screen Capture:

```
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
S>* 192.168.2.0/24 [1/0] via 192.168.4.1, eth1
C>* 192.168.4.0/24 is directly connected, eth1
C>* 192.168.5.0/24 is directly connected, eth2
C>* 192.168.6.0/24 is directly connected, eth0
vyos@vyos:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
S>* 2020:abcd:2::/64 [1/0] via 2020:abcd:4::1, eth1
C>* 2020:abcd:4::/64 is directly connected, eth1
C>* 2020:abcd:5::/64 is directly connected, eth2
C>* 2020:abcd:6::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth2
vyos@vyos:~$
```

R2 Screen Capture:

```
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, eth1
C>* 192.168.2.0/24 is directly connected, eth2
C>* 192.168.3.0/24 is directly connected, eth3
S>* 192.168.6.0/24 [1/0] via 192.168.1.2, eth1
vyos@vyos:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
C>* 2020:abcd:1::/64 is directly connected, eth1
C>* 2020:abcd:2::/64 is directly connected, eth2
C>* 2020:abcd:3::/64 is directly connected, eth2
S>* 2020:abcd:6::/64 [1/0] via 2020:abcd:1::2, eth1
C * fe80::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth3
vyos@vyos:~$ _
```

Traceroute Screen capture:

H1

```
vyos@vyos:~$ traceroute 192.168.6.2
traceroute to 192.168.6.2 (192.168.6.2), 30 hops max, 60 byte packets
 1  192.168.2.2 (192.168.2.2)  0.376 ms  0.344 ms  0.335 ms
 2  192.168.1.2 (192.168.1.2)  0.696 ms  0.695 ms  0.691 ms
 3  192.168.10.2 (192.168.10.2)  1.457 ms  1.455 ms  1.454 ms
 4  192.168.4.2 (192.168.4.2)  1.871 ms  1.864 ms  1.855 ms
 5  192.168.6.2 (192.168.6.2)  1.844 ms  1.823 ms  1.812 ms
vyos@vyos:~$ traceroute 2020:abcd:6::2
traceroute to 2020:abcd:6::2 (2020:abcd:6::2), 30 hops max, 80 byte packets
 1  2020:abcd:2::2 (2020:abcd:2::2)  0.388 ms  0.382 ms  0.378 ms
 2  2020:abcd:1::2 (2020:abcd:1::2)  0.801 ms  0.806 ms  0.819 ms
 3  2020:abcd:10::2 (2020:abcd:10::2)  1.689 ms  1.698 ms  1.695 ms
 4  2020:abcd:4::2 (2020:abcd:4::2)  1.829 ms  1.829 ms  1.907 ms
 5  2020:abcd:6::2 (2020:abcd:6::2)  2.092 ms  2.104 ms  2.102 ms
vyos@vyos:~$ _
```

H2

```
vyos@vyos:~$ traceroute 192.168.2.1
traceroute to 192.168.2.1 (192.168.2.1), 30 hops max, 60 byte packets
 1  192.168.6.1 (192.168.6.1)  0.319 ms  0.303 ms  0.298 ms
 2  192.168.4.1 (192.168.4.1)  0.589 ms  0.589 ms  0.587 ms
 3  192.168.10.1 (192.168.10.1)  1.190 ms  1.189 ms  1.190 ms
 4  192.168.1.1 (192.168.1.1)  1.450 ms  1.451 ms  1.444 ms
 5  192.168.2.1 (192.168.2.1)  1.646 ms  1.647 ms  1.641 ms
vyos@vyos:~$ traceroute 2020:abcd:2::1
traceroute to 2020:abcd:2::1 (2020:abcd:2::1), 30 hops max, 80 byte packets
 1  2020:abcd:6::1 (2020:abcd:6::1)  0.432 ms  0.415 ms  0.421 ms
 2  2020:abcd:4::1 (2020:abcd:4::1)  0.626 ms  0.624 ms  0.625 ms
 3  2020:abcd:10::1 (2020:abcd:10::1)  1.601 ms  1.602 ms  1.594 ms
 4  2020:abcd:1::1 (2020:abcd:1::1)  1.836 ms  1.824 ms  1.805 ms
 5  2020:abcd:2::1 (2020:abcd:2::1)  2.016 ms  2.018 ms  2.014 ms
vyos@vyos:~$
```

Question 3. (For Static Routing)

Execute ping (only IPv4 is OK) from H1 to H2, and disconnect Subnet (A) by unplugging the LAN cable between the laptops, and explain what happens in the network and how a router react when the next hop router becomes unreachable. Insert the screen captures of tcpdump on both NICs of R1, and provide additional explanation of what you observe as part of the answer.

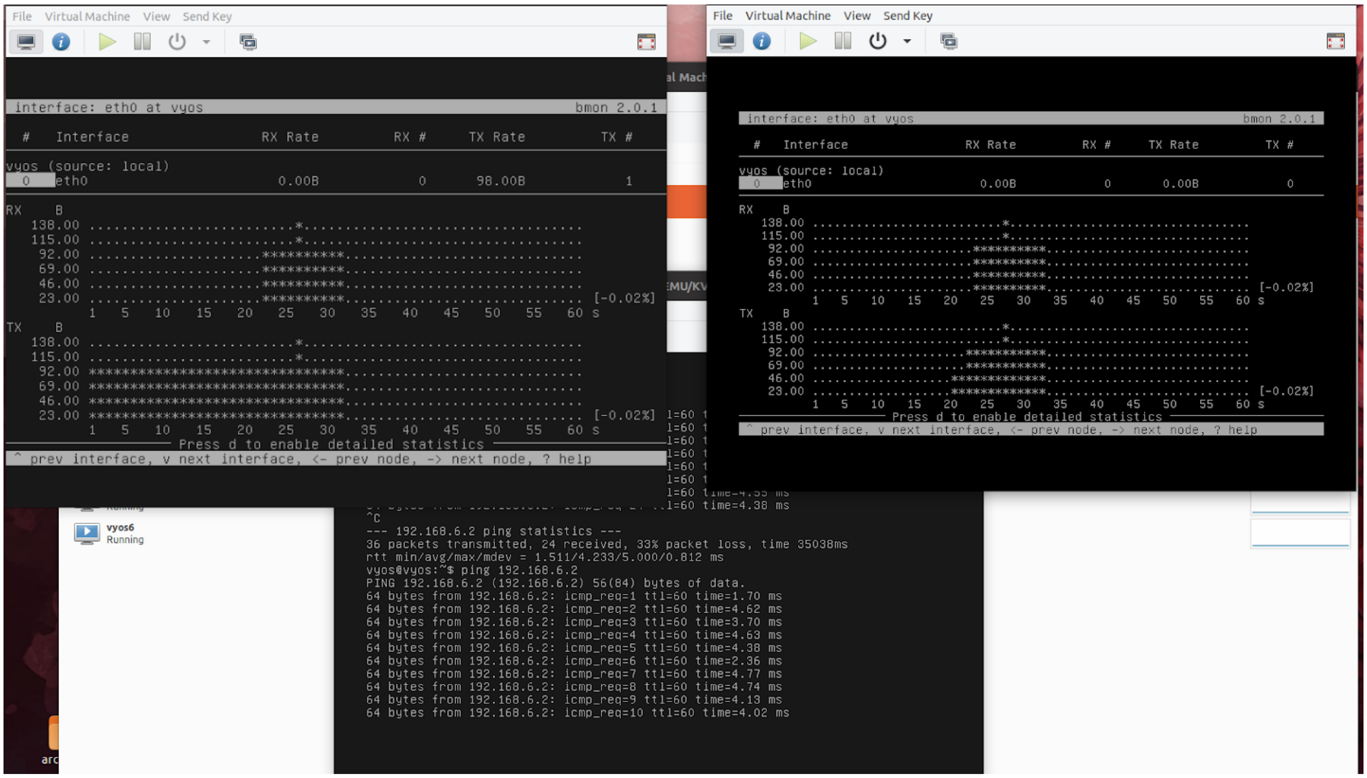
Observation:

When I pinged from H1 to H2 initially I was getting replies back. Once, the connection between laptops is removed I notice that there are no more replies. I wanted for few more seconds and then terminated ping cmd. I noticed packet loss (refer screen capture below).

```
64 bytes from 192.168.6.2: icmp_req=17 ttl=60 time=4.56 ms
64 bytes from 192.168.6.2: icmp_req=18 ttl=60 time=5.00 ms
64 bytes from 192.168.6.2: icmp_req=19 ttl=60 time=4.65 ms
64 bytes from 192.168.6.2: icmp_req=20 ttl=60 time=4.87 ms
64 bytes from 192.168.6.2: icmp_req=21 ttl=60 time=4.64 ms
64 bytes from 192.168.6.2: icmp_req=22 ttl=60 time=4.51 ms
64 bytes from 192.168.6.2: icmp_req=23 ttl=60 time=4.55 ms
64 bytes from 192.168.6.2: icmp_req=24 ttl=60 time=4.38 ms
^C
--- 192.168.6.2 ping statistics ---
36 packets transmitted, 24 received, 33% packet loss, time 35038ms
rtt min/avg/max/mdev = 1.511/4.233/5.000/0.812 ms
vyos@vyos:~$
```

Ping from H1 to H2

The transmit rate of R1 is still unchanged but as for R4 it becomes 0. I monitored traffic of R1 (left)and R4(right) to show so.



Right after the connection is cut, the router doesn’t know that the next-hop is unreachable .

After some more time, the router gets destination host unreachable and hence, routers starts sending arp messages.(shown below initiated from R1’s eth0)

H1 SCREEN CAPTURE

```
PING 192.168.6.2 (192.168.6.2) 56(84) bytes of data.
64 bytes from 192.168.6.2: icmp_req=1 ttl=60 time=2.57 ms
64 bytes from 192.168.6.2: icmp_req=2 ttl=60 time=4.55 ms
64 bytes from 192.168.6.2: icmp_req=3 ttl=60 time=4.94 ms
64 bytes from 192.168.6.2: icmp_req=4 ttl=60 time=4.55 ms
64 bytes from 192.168.6.2: icmp_req=5 ttl=60 time=4.81 ms
64 bytes from 192.168.6.2: icmp_req=6 ttl=60 time=3.13 ms
64 bytes from 192.168.6.2: icmp_req=7 ttl=60 time=4.69 ms
64 bytes from 192.168.6.2: icmp_req=8 ttl=60 time=4.73 ms
64 bytes from 192.168.6.2: icmp_req=9 ttl=60 time=3.75 ms
64 bytes from 192.168.6.2: icmp_req=10 ttl=60 time=4.76 ms
64 bytes from 192.168.6.2: icmp_req=11 ttl=60 time=4.62 ms
64 bytes from 192.168.6.2: icmp_req=12 ttl=60 time=1.95 ms
64 bytes from 192.168.6.2: icmp_req=13 ttl=60 time=4.20 ms
From 192.168.1.2 icmp_seq=36 Destination Host Unreachable
From 192.168.1.2 icmp_seq=37 Destination Host Unreachable
From 192.168.1.2 icmp_seq=38 Destination Host Unreachable
From 192.168.1.2 icmp_seq=39 Destination Host Unreachable
From 192.168.1.2 icmp_seq=40 Destination Host Unreachable
From 192.168.1.2 icmp_seq=41 Destination Host Unreachable
^C
--- 192.168.6.2 ping statistics ---
42 packets transmitted, 13 received, +6 errors, 69% packet loss, time 41036ms
rtt min/avg/max/mdev = 1.951/4.099/4.941/0.928 ms, pipe 3
vyos@vyos:~$ _
```

tcpdump captures of both interfaces still recorded the messages being sent(**R1**).

```
12:21:06.038283 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2903, seq 30
, length 64
12:21:07.038848 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2903, seq 31
, length 64
12:21:08.038764 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2903, seq 32
, length 64
12:21:08.044539 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
12:21:09.038451 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2903, seq 33
, length 64
12:21:09.044567 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
12:21:10.038795 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2903, seq 34
, length 64
12:21:10.044503 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
12:21:11.038664 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2903, seq 35
, length 64
12:21:11.556306 IP6 fe80::b47d:3dff:fe89:49a3 > ip6-allrouters: ICMP6, router so
licitation, length 16
12:21:12.038225 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
12:21:13.034648 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
12:21:14.034568 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
^C
32 packets captured
32 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$
```

```
 length 92
12:25:21.944720 IP 192.168.1.2 > 192.168.2.1: ICMP host 192.168.6.2 unreachable,
 length 92
12:25:21.944729 IP 192.168.1.2 > 192.168.2.1: ICMP host 192.168.6.2 unreachable,
 length 92
12:25:21.954427 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2938, seq 35
, length 64
12:25:21.956222 ARP, Request who-has 192.168.1.2 tell 192.168.1.1, length 28
12:25:21.956244 ARP, Reply 192.168.1.2 is-at 52:54:00:f2:65:87 (oui Unknown), le
ngth 28
12:25:22.958509 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2938, seq 36
, length 64
12:25:23.958520 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 2938, seq 37
, length 64
12:25:24.954745 IP 192.168.1.2 > 192.168.2.1: ICMP host 192.168.6.2 unreachable,
 length 92
12:25:24.954801 IP 192.168.1.2 > 192.168.2.1: ICMP host 192.168.6.2 unreachable,
 length 92
12:25:24.954811 IP 192.168.1.2 > 192.168.2.1: ICMP host 192.168.6.2 unreachable,
 length 92
^C
42 packets captured
42 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$
```

Question 4. (For Static Routing)
Configure static routes on the routers so that 1) ping traffic from H1 to H2 goes through Subnet (A), and 2) that from H2 to H1 goes through Subnet (B) using both IPv4 and IPv6. Answer by inserting the screen captures of the routing table on R2 and R6, and the tcpdump result on R1 and R5 on those you should observe the traffic is one way.

R2 routing table:

```
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
C>* 192.168.1.0/24 is directly connected, eth1
C>* 192.168.2.0/24 is directly connected, eth2
C>* 192.168.3.0/24 is directly connected, eth3
S>* 192.168.6.0/24 [1/0] via 192.168.1.2, eth1
vyos@vyos:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
C>* 2020:abcd:1::/64 is directly connected, eth1
C>* 2020:abcd:2::/64 is directly connected, eth2
C>* 2020:abcd:3::/64 is directly connected, eth2
S>* 2020:abcd:6::/64 [1/0] via 2020:abcd:1::2, eth1
C * fe80::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth3
vyos@vyos:~$
```

R6 routing table:

```
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
S>* 192.168.2.0/24 [1/0] via 192.168.5.1, eth2
C>* 192.168.4.0/24 is directly connected, eth1
C>* 192.168.5.0/24 is directly connected, eth2
C>* 192.168.6.0/24 is directly connected, eth0
vyos@vyos:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
S>* 2020:abcd:2::/64 [1/0] via 2020:abcd:5::1, eth2
C>* 2020:abcd:4::/64 is directly connected, eth1
C>* 2020:abcd:5::/64 is directly connected, eth2
C>* 2020:abcd:6::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth2
vyos@vyos:~$
```

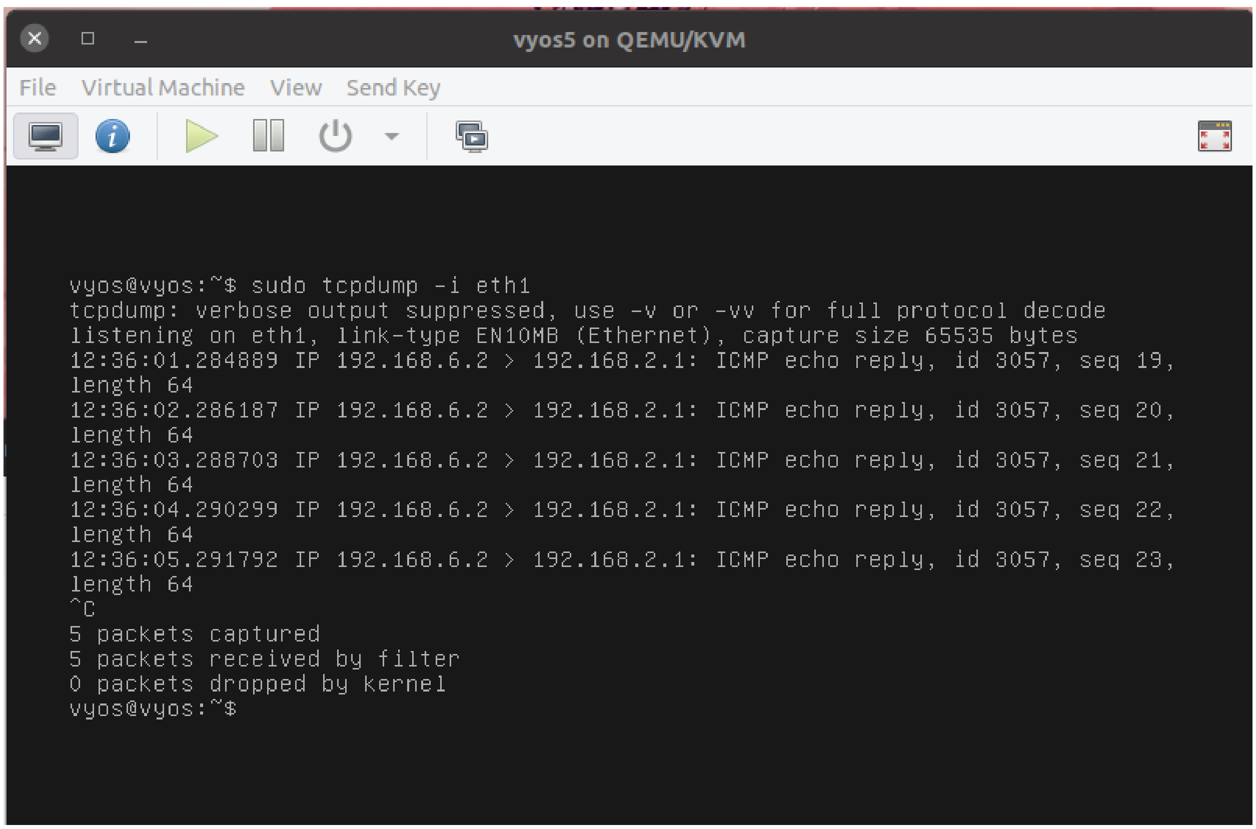
tcpdump of **R1**

```
vyos@vyos:~$ sudo tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
12:34:19.497386 ARP, Request who-has 192.168.1.2 tell 192.168.1.1, length 28
12:34:19.497423 ARP, Reply 192.168.1.2 is-at 52:54:00:f2:65:87 (oui Unknown), le
ngth 28
12:34:19.502555 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 21
, length 64
12:34:20.503959 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 22
, length 64
12:34:21.505196 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 23
, length 64
12:34:22.506556 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 24
, length 64
12:34:23.507627 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 25
, length 64
12:34:24.509736 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 26
, length 64
12:34:25.511146 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 27
, length 64
^C
9 packets captured
9 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$
```

```
vyos@vyos:~$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:34:03.479332 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 5,
 length 64
12:34:04.475328 ARP, Request who-has 192.168.10.2 tell 192.168.10.1, length 28
12:34:04.477301 ARP, Reply 192.168.10.2 is-at 52:54:00:40:74:95 (oui Unknown), l
ength 28
12:34:04.480688 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 6,
 length 64
12:34:05.481476 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 7,
 length 64
12:34:06.483393 IP 192.168.2.1 > 192.168.6.2: ICMP echo request, id 3040, seq 8,
 length 64
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _
```

tcpdump of **R5**

```
vyos@vyos:~$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
12:35:46.261128 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 4, l
ength 64
12:35:47.262869 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 5, l
ength 64
12:35:48.256514 ARP, Request who-has 192.168.11.1 tell 192.168.11.2, length 28
12:35:48.258338 ARP, Reply 192.168.11.1 is-at 52:54:00:71:55:38 (oui Unknown), l
ength 28
12:35:48.264305 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 6, l
ength 64
12:35:49.265634 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 7, l
ength 64
12:35:50.267204 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 8, l
ength 64
^C
7 packets captured
7 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _
```




```
vyos@vyos:~$ sudo tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
12:36:01.284889 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 19,
length 64
12:36:02.286187 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 20,
length 64
12:36:03.288703 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 21,
length 64
12:36:04.290299 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 22,
length 64
12:36:05.291792 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 3057, seq 23,
length 64
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$
```

Question 5.
Create a routing loop among R1, R2, … R6, and explain what happens in the network. Insert the screen captures of tcpdump on H1 and IPv4/v6

traceroute performed from H1 to H2. And explain 1) the traceroute results, and 2) what kind of message H1 receives when routing loop happens.

traceroute

```
vyos@vyos:~$ traceroute 192.168.20.2
traceroute to 192.168.20.2 (192.168.20.2), 30 hops max, 60 byte packets
 1  192.168.2.2 (192.168.2.2)  0.296 ms  0.256 ms  0.253 ms
 2  192.168.1.2 (192.168.1.2)  0.492 ms  0.497 ms  0.493 ms
 3  192.168.10.2 (192.168.10.2)  1.022 ms  1.014 ms  1.012 ms
 4  192.168.4.2 (192.168.4.2)  1.128 ms  1.102 ms  1.083 ms
 5  192.168.5.1 (192.168.5.1)  1.077 ms  1.047 ms  1.023 ms
 6  192.168.11.1 (192.168.11.1)  0.998 ms  0.887 ms  0.927 ms
 7  192.168.3.1 (192.168.3.1)  0.919 ms  1.714 ms  1.716 ms
 8  192.168.1.2 (192.168.1.2)  1.869 ms  1.870 ms  1.873 ms
 9  192.168.10.2 (192.168.10.2)  2.311 ms  2.378 ms  2.379 ms
10  192.168.4.2 (192.168.4.2)  2.306 ms  2.287 ms  2.279 ms
11  192.168.5.1 (192.168.5.1)  2.275 ms  2.274 ms  2.272 ms
12  192.168.11.1 (192.168.11.1)  1.634 ms  1.626 ms  0.863 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * 192.168.11.1 (192.168.11.1)  8.315 ms  8.281 ms
19  192.168.3.1 (192.168.3.1)  8.276 ms  8.293 ms  8.288 ms
```


```
 8  192.168.1.2 (192.168.1.2)  1.869 ms  1.870 ms  1.873 ms
 9  192.168.10.2 (192.168.10.2)  2.311 ms  2.378 ms  2.379 ms
10  192.168.4.2 (192.168.4.2)  2.306 ms  2.287 ms  2.279 ms
11  192.168.5.1 (192.168.5.1)  2.275 ms  2.274 ms  2.272 ms
12  192.168.11.1 (192.168.11.1)  1.634 ms  1.626 ms  0.863 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * 192.168.11.1 (192.168.11.1)  8.315 ms  8.281 ms
19  192.168.3.1 (192.168.3.1)  8.276 ms  8.293 ms  8.288 ms
```


```
20  192.168.1.2 (192.168.1.2)  8.832 ms  8.824 ms  9.140 ms
21  192.168.10.2 (192.168.10.2)  10.495 ms  10.486 ms  10.464 ms
22  192.168.4.2 (192.168.4.2)  11.908 ms  11.906 ms  11.891 ms
23  192.168.5.1 (192.168.5.1)  11.884 ms  12.016 ms  10.400 ms
24  192.168.11.1 (192.168.11.1)  10.353 ms  10.300 ms  10.272 ms
25  192.168.3.1 (192.168.3.1)  10.244 ms  5.113 ms *
26  192.168.1.2 (192.168.1.2)  5.480 ms  5.461 ms *
27  192.168.10.2 (192.168.10.2)  6.011 ms  6.008 ms *
28  192.168.4.2 (192.168.4.2)  6.383 ms  6.373 ms *
29  192.168.5.1 (192.168.5.1)  4.163 ms  4.122 ms *
30  * * *
vyos@vyos:~$
vyos@vyos:~$ _
```

Traceroute ipv6

```
vyos@vyos:~$ traceroute 2020:abcd:20::2
traceroute to 2020:abcd:20::2 (2020:abcd:20::2), 30 hops max, 80 byte packets
 1  2020:abcd:2::2 (2020:abcd:2::2)  0.416 ms  0.395 ms  0.389 ms
 2  2020:abcd:1::2 (2020:abcd:1::2)  0.834 ms  0.834 ms  0.830 ms
 3  2020:abcd:10::2 (2020:abcd:10::2)  1.822 ms  1.824 ms  1.816 ms
 4  2020:abcd:5::2 (2020:abcd:5::2)  2.091 ms  2.086 ms  2.066 ms
 5  2020:abcd:11::2 (2020:abcd:11::2)  2.049 ms  2.044 ms  2.041 ms
 6  2020:abcd:3::2 (2020:abcd:3::2)  2.037 ms  1.287 ms  1.269 ms
 7  2020:abcd:2::2 (2020:abcd:2::2)  1.261 ms  1.332 ms  1.322 ms
 8  2020:abcd:1::2 (2020:abcd:1::2)  1.509 ms  1.512 ms  1.510 ms
 9  2020:abcd:10::2 (2020:abcd:10::2)  1.838 ms  1.833 ms  1.823 ms
10  2020:abcd:5::2 (2020:abcd:5::2)  1.991 ms  1.994 ms  1.987 ms
11  2020:abcd:11::2 (2020:abcd:11::2)  1.979 ms  1.966 ms  1.136 ms
12  2020:abcd:3::2 (2020:abcd:3::2)  1.119 ms  1.101 ms  0.967 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * 2020:abcd:3::2 (2020:abcd:3::2)  9.624 ms  9.582 ms
19  2020:abcd:2::2 (2020:abcd:2::2)  9.571 ms  9.558 ms  9.550 ms
```

```
 8  2020:abcd:1::2 (2020:abcd:1::2)  1.509 ms  1.512 ms  1.510 ms
 9  2020:abcd:10::2 (2020:abcd:10::2)  1.838 ms  1.833 ms  1.823 ms
10  2020:abcd:5::2 (2020:abcd:5::2)  1.991 ms  1.994 ms  1.987 ms
11  2020:abcd:11::2 (2020:abcd:11::2)  1.979 ms  1.966 ms  1.136 ms
12  2020:abcd:3::2 (2020:abcd:3::2)  1.119 ms  1.101 ms  0.967 ms
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * 2020:abcd:3::2 (2020:abcd:3::2)  9.624 ms  9.582 ms
19  2020:abcd:2::2 (2020:abcd:2::2)  9.571 ms  9.558 ms  9.550 ms
20  2020:abcd:1::2 (2020:abcd:1::2)  10.294 ms  10.282 ms  10.256 ms
21  2020:abcd:10::2 (2020:abcd:10::2)  12.618 ms  12.581 ms  12.534 ms
22  2020:abcd:5::2 (2020:abcd:5::2)  12.563 ms  12.528 ms  12.496 ms
23  2020:abcd:11::2 (2020:abcd:11::2)  12.465 ms  12.418 ms  12.305 ms
24  2020:abcd:3::2 (2020:abcd:3::2)  12.240 ms  12.187 ms  12.063 ms
25  2020:abcd:2::2 (2020:abcd:2::2)  12.048 ms  6.369 ms *
26  2020:abcd:1::2 (2020:abcd:1::2)  6.585 ms  6.570 ms *
27  2020:abcd:10::2 (2020:abcd:10::2)  7.295 ms  7.274 ms *
28  2020:abcd:5::2 (2020:abcd:5::2)  7.575 ms  7.550 ms *
29  2020:abcd:11::2 (2020:abcd:11::2)  4.314 ms  4.279 ms *
30  * * *
vyos@vyos:~$
vyos@vyos:~$
```

1. First 12 hops itself we can see the loop. 2nd entry is repeated in 8th entry and so on. This pattern goes on since each router checks its routing table to forward the packet. There is no way for it to know that a loop exists(for a router) based on how traceroute is

implemented. Hence, the packets are forwarded until TTL limit is exceeded. Based on observation, we can also see that traceroute by default only records 30 hops.

2. It shows “Time to live exceeded”

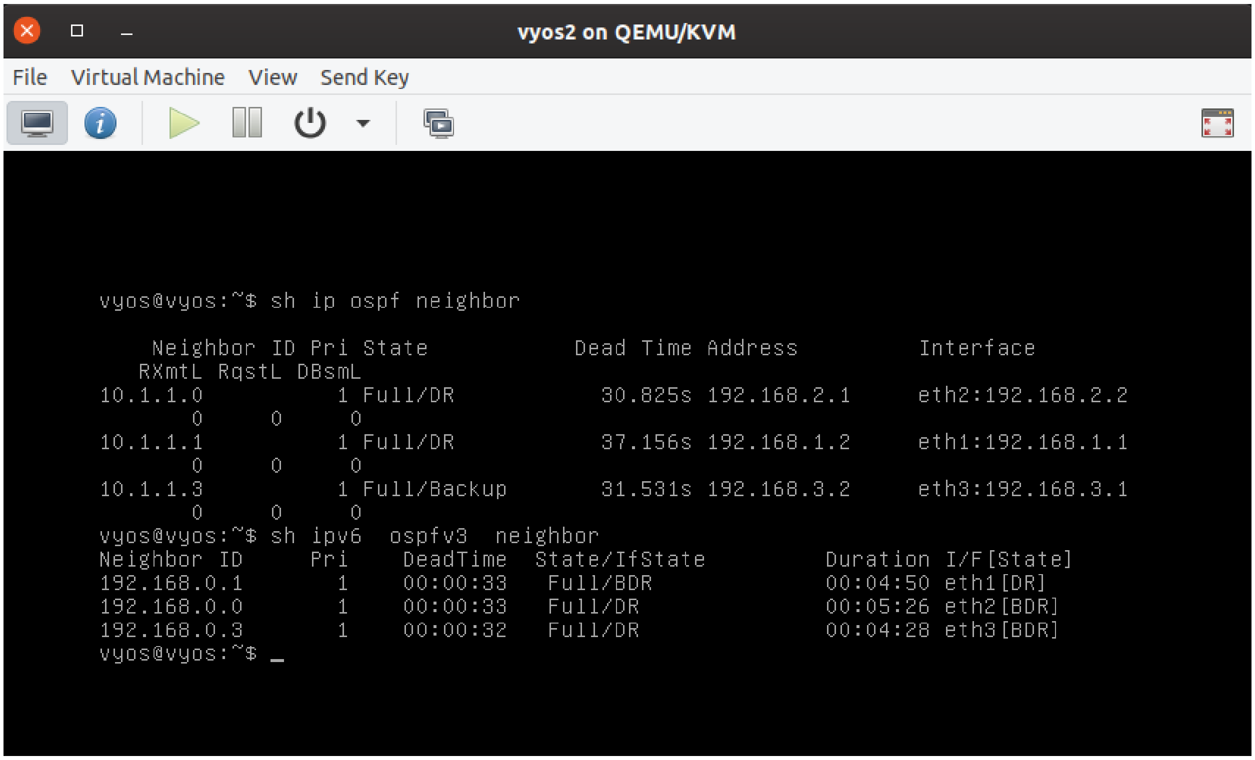
Tcpdump of H1

```
vyos@vyos:~$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
15:34:14.628676 IP 192.168.2.1 > 192.168.20.2: ICMP echo request, id 5317, seq 2
3, length 64
15:34:14.658160 IP 192.168.4.2 > 192.168.2.1: ICMP time exceeded in-transit, len
gth 92
15:34:15.630654 IP 192.168.2.1 > 192.168.20.2: ICMP echo request, id 5317, seq 2
4, length 64
15:34:15.661554 IP 192.168.4.2 > 192.168.2.1: ICMP time exceeded in-transit, len
gth 92
15:34:16.631883 IP 192.168.2.1 > 192.168.20.2: ICMP echo request, id 5317, seq 2
5, length 64
15:34:16.660410 IP 192.168.4.2 > 192.168.2.1: ICMP time exceeded in-transit, len
gth 92
15:34:17.633899 IP 192.168.2.1 > 192.168.20.2: ICMP echo request, id 5317, seq 2
6, length 64
15:34:17.661827 IP 192.168.4.2 > 192.168.2.1: ICMP time exceeded in-transit, len
gth 92
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _
```

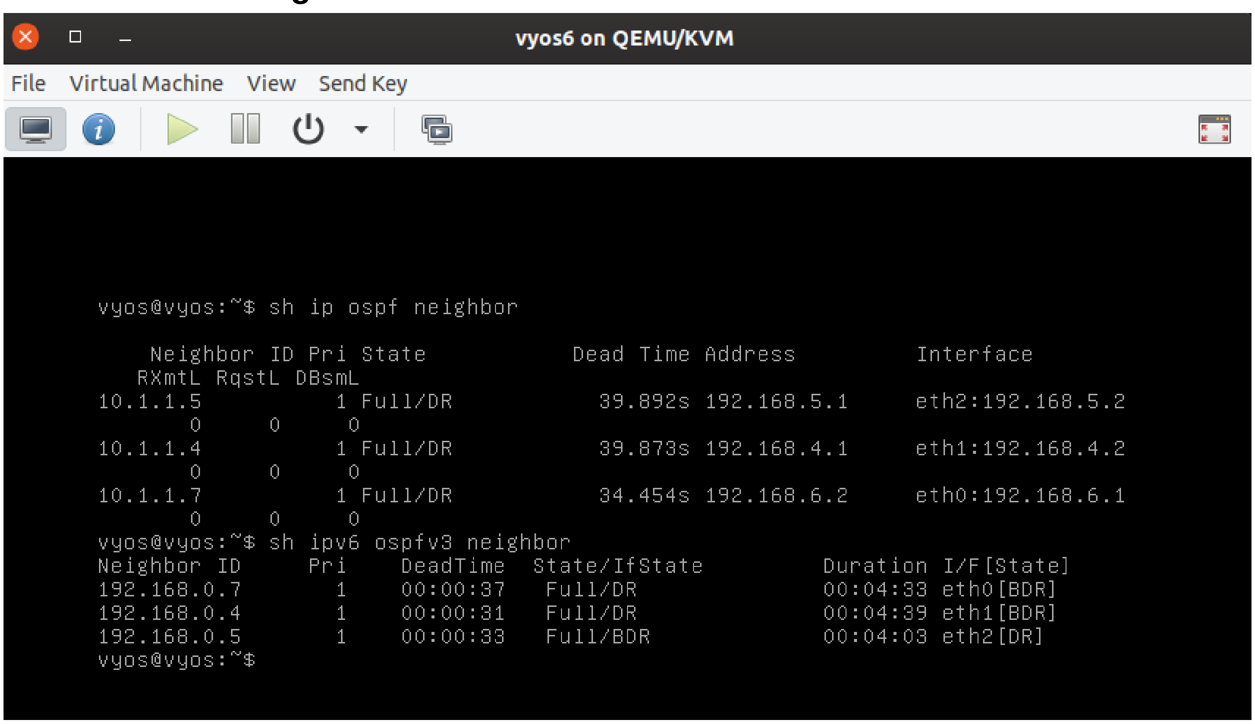
Question 6.1.

Delete the static routes from all the routers, and enable OSPF for IPv4 and OSPFv3 for IPv6 on them so that ping and ping6 are successful between H1 and H2. Insert screen captures of the OSPF/OSPFv3 neighbor tables, IPv4/v6 routing tables on R2 and R6, successful ping/ping6 results on H1 or H2.

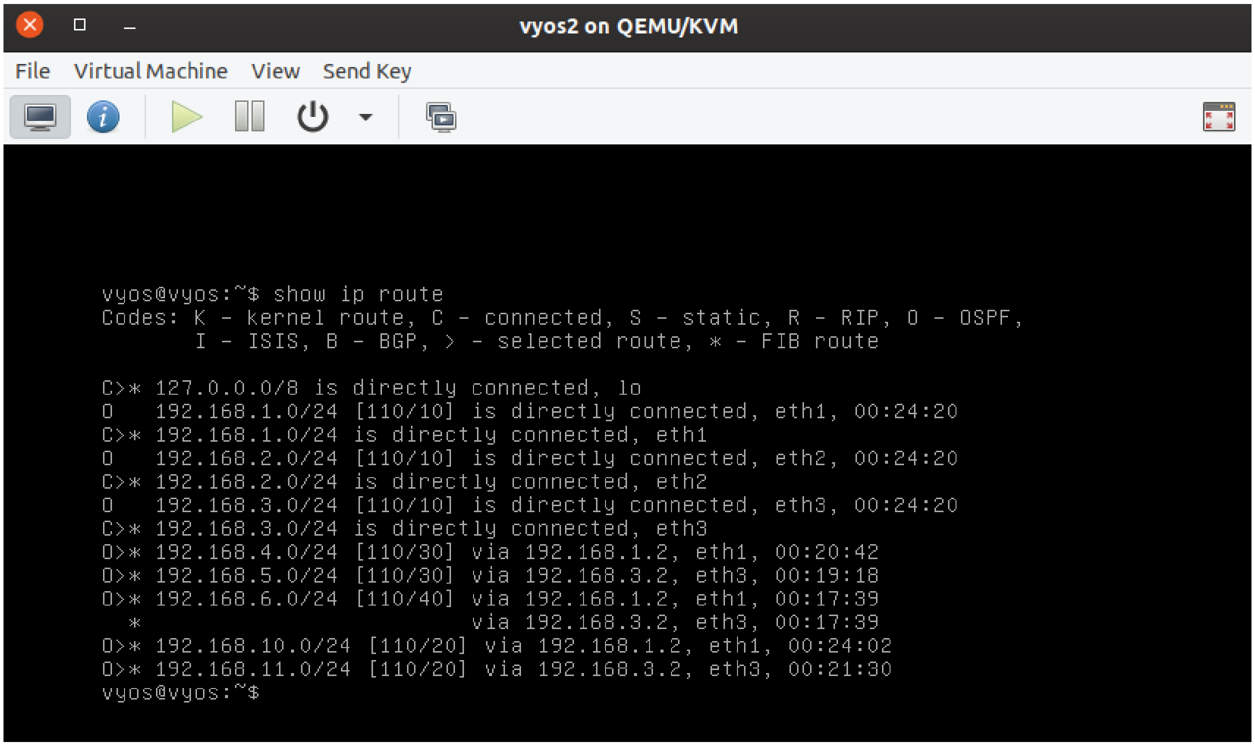
OSPF/OSPFv3 neighbor tables R2

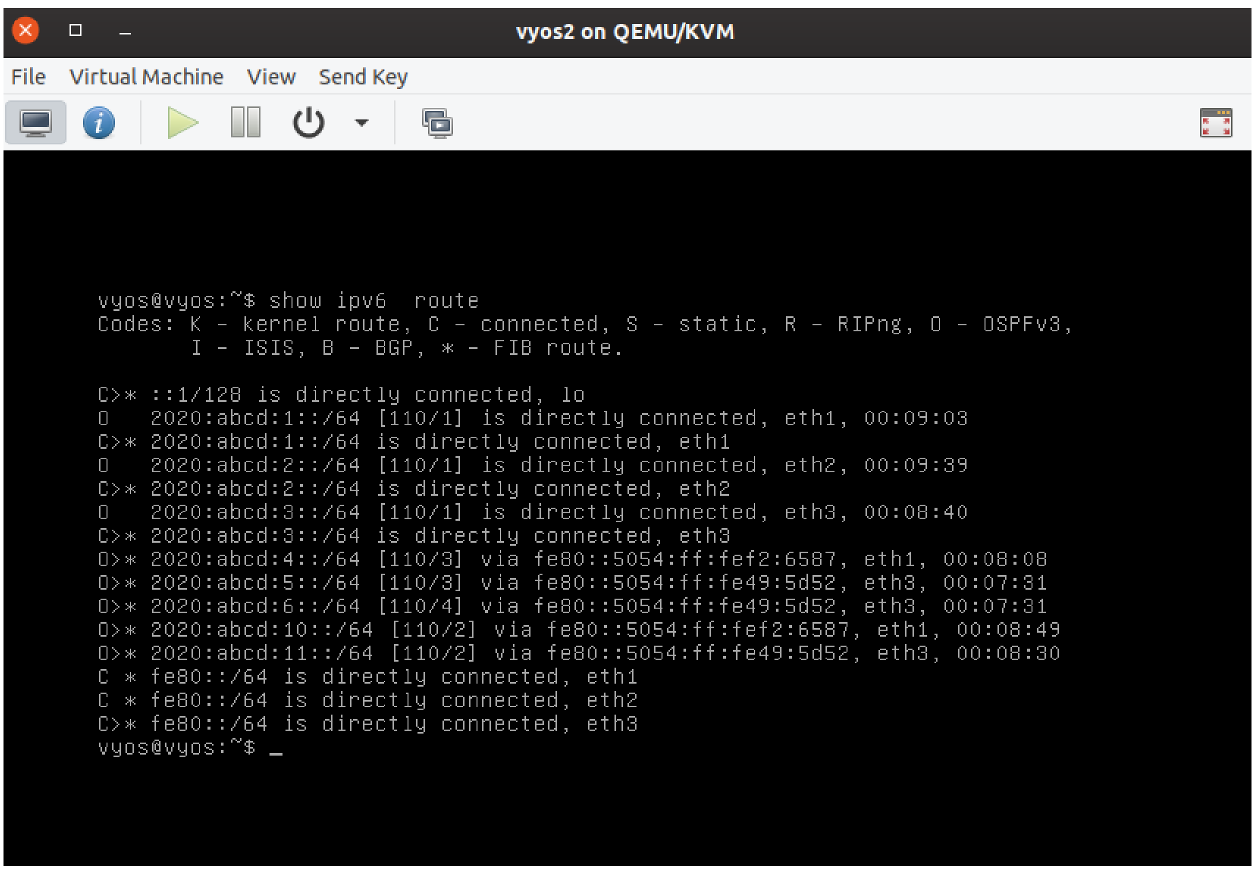


OSPF/OSPFv3 neighbor tables R6

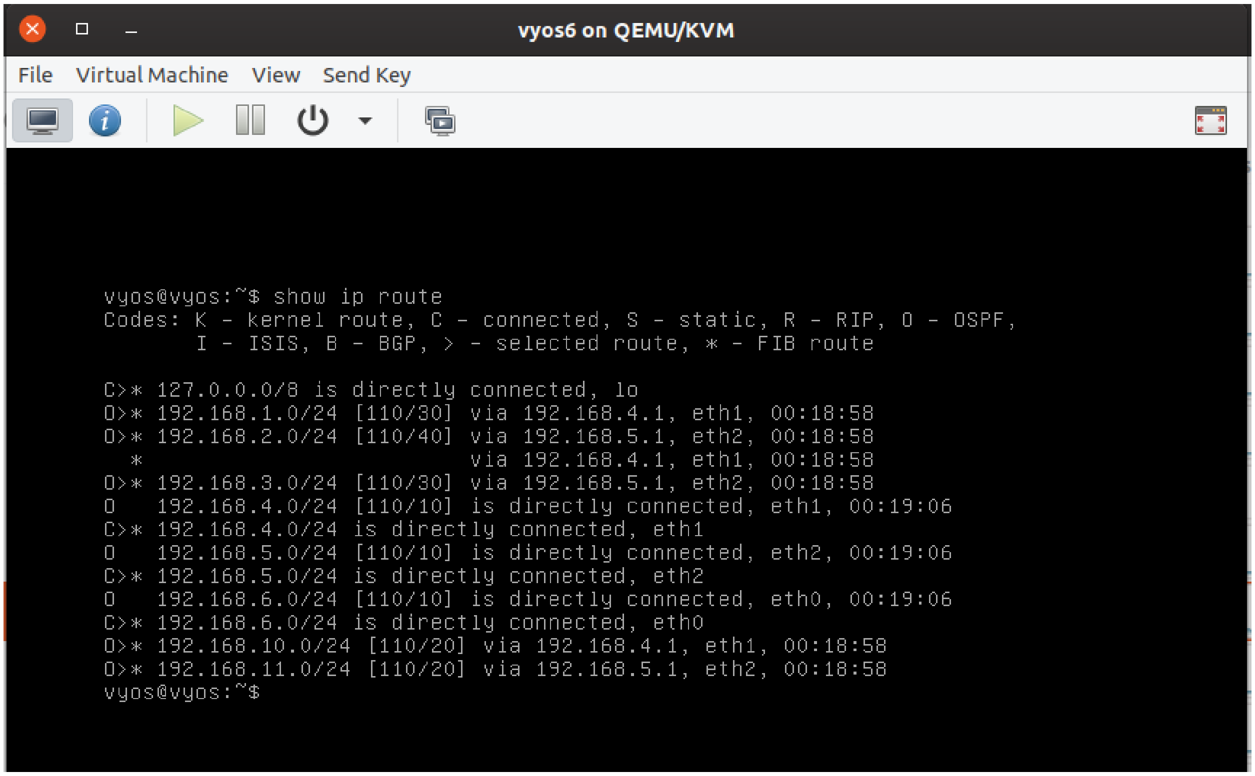


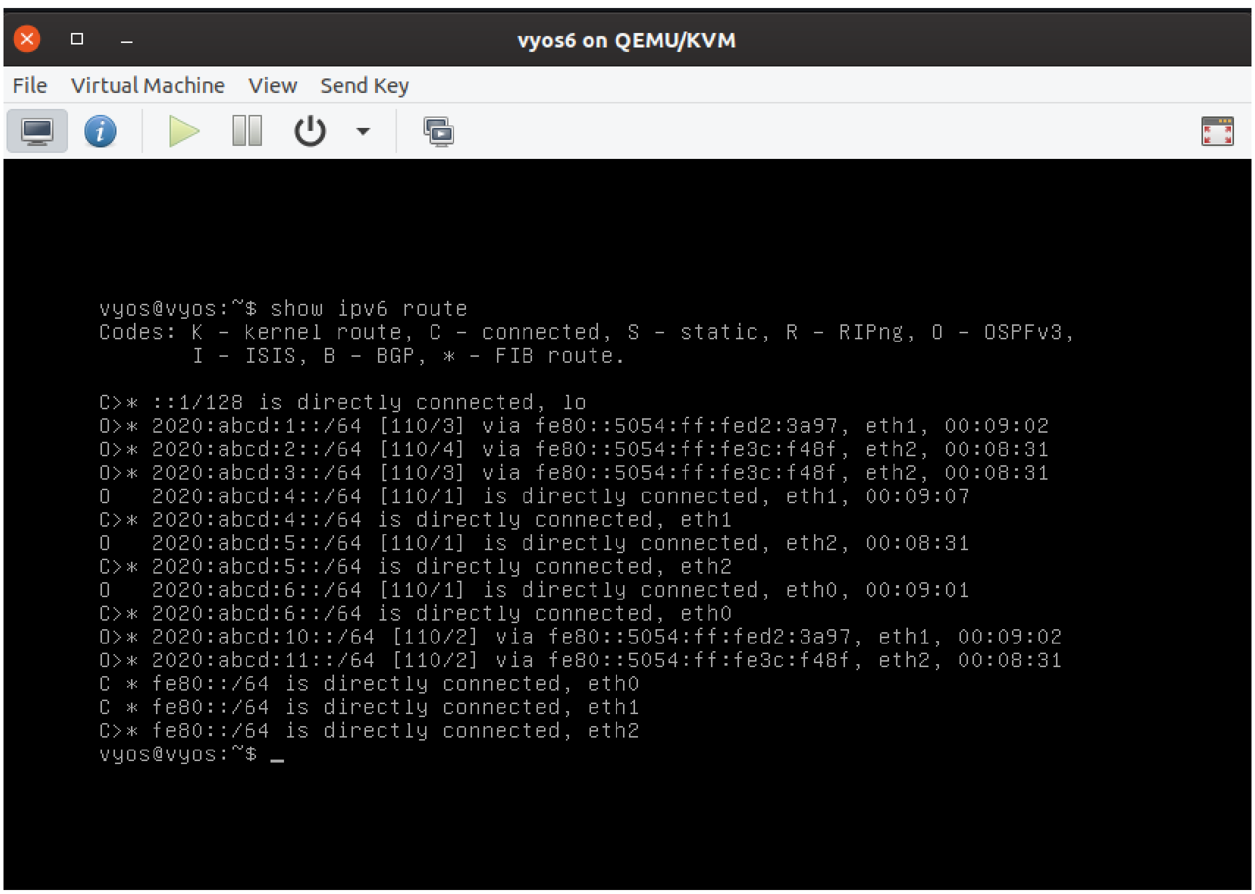
ipv4/v6 route R2



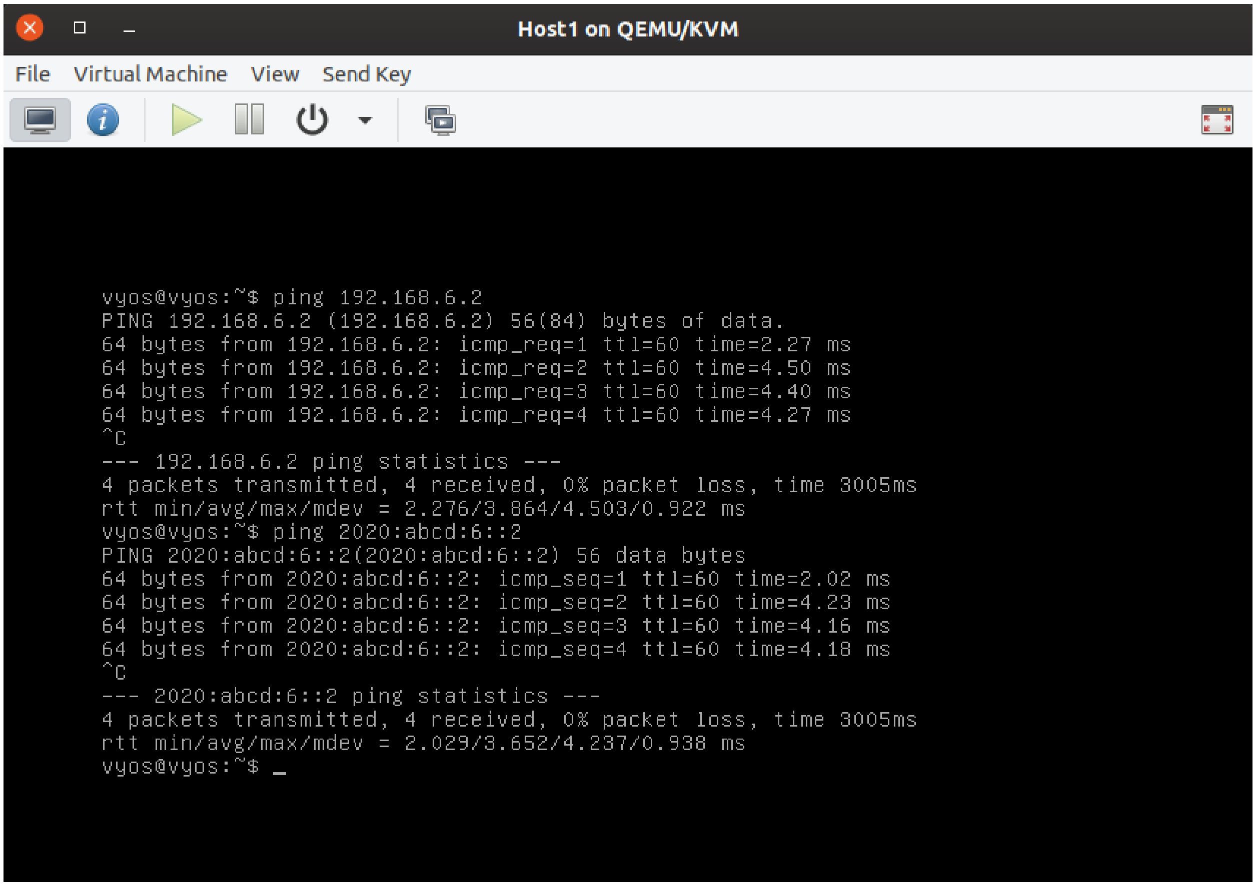


ipv4/v6 route R6

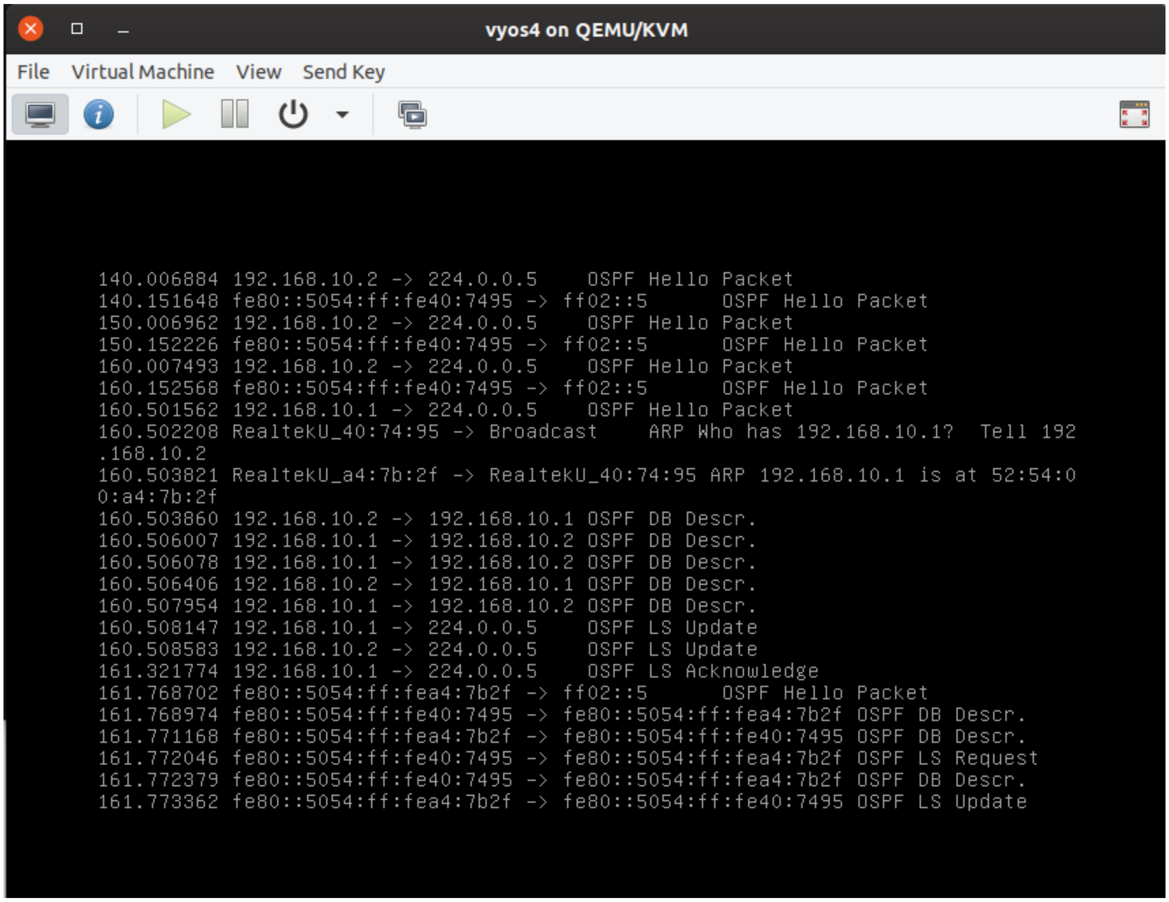


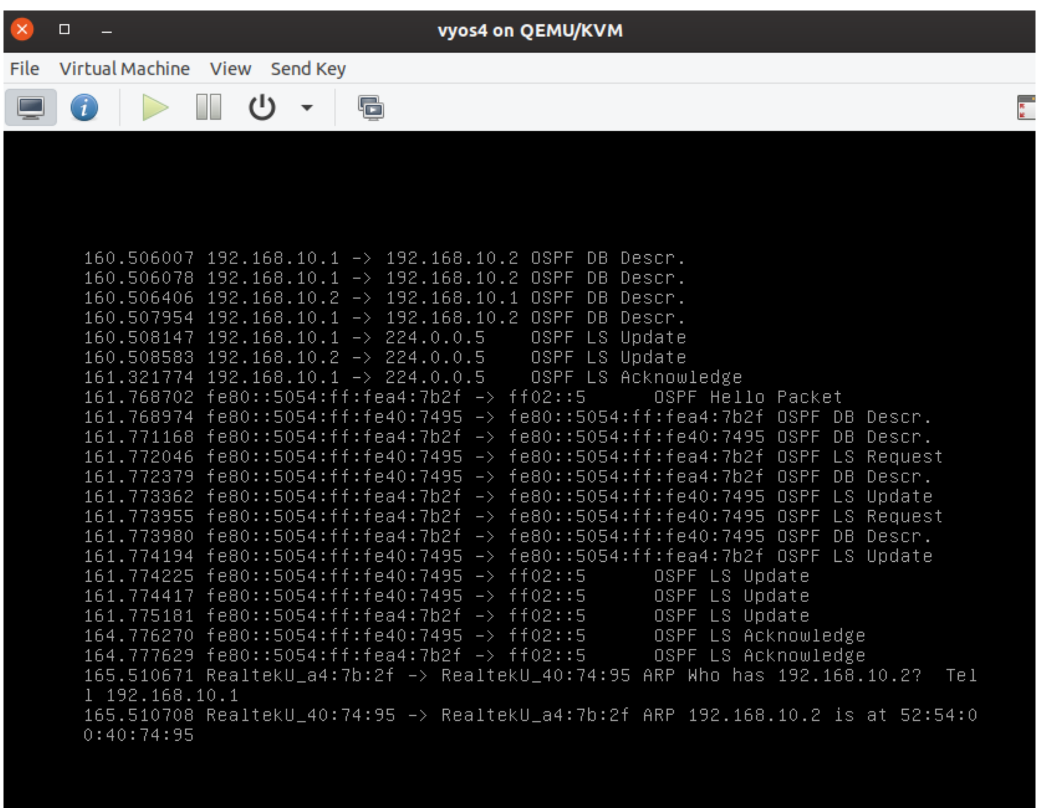


Ping from H1 to H2 (ipv4/v6)



Question 6.2.
Select two different types of OSPF-related messages that you observe in the experiment, and explain what kind of information they carry and what their role is respectively.


vyos4 on QEMU/KVM
File Virtual Machine View Send Key
140.006884 192.168.10.2 -> 224.0.0.5 OSPF Hello Packet
140.151648 fe80::5054:ff:fe40:7495 -> ff02::5 OSPF Hello Packet
150.006962 192.168.10.2 -> 224.0.0.5 OSPF Hello Packet
150.152226 fe80::5054:ff:fe40:7495 -> ff02::5 OSPF Hello Packet
160.007493 192.168.10.2 -> 224.0.0.5 OSPF Hello Packet
160.152568 fe80::5054:ff:fe40:7495 -> ff02::5 OSPF Hello Packet
160.501562 192.168.10.1 -> 224.0.0.5 OSPF Hello Packet
160.502208 RealtekU_40:74:95 -> Broadcast ARP Who has 192.168.10.1? Tell 192
.168.10.2
160.503821 RealtekU_a4:7b:2f -> RealtekU_40:74:95 ARP 192.168.10.1 is at 52:54:0
0:a4:7b:2f
160.503860 192.168.10.2 -> 192.168.10.1 OSPF DB Descr.
160.506007 192.168.10.1 -> 192.168.10.2 OSPF DB Descr.
160.506078 192.168.10.1 -> 192.168.10.2 OSPF DB Descr.
160.506406 192.168.10.2 -> 192.168.10.1 OSPF DB Descr.
160.507954 192.168.10.1 -> 192.168.10.2 OSPF DB Descr.
160.508147 192.168.10.1 -> 224.0.0.5 OSPF LS Update
160.508583 192.168.10.2 -> 224.0.0.5 OSPF LS Update
161.321774 192.168.10.1 -> 224.0.0.5 OSPF LS Acknowledge
161.768702 fe80::5054:ff:fea4:7b2f -> ff02::5 OSPF Hello Packet
161.768974 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF DB Descr.
161.771168 fe80::5054:ff:fea4:7b2f -> fe80::5054:ff:fe40:7495 OSPF DB Descr.
161.772046 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF LS Request
161.772379 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF DB Descr.
161.773362 fe80::5054:ff:fea4:7b2f -> fe80::5054:ff:fe40:7495 OSPF LS Update


vyos4 on QEMU/KVM
File Virtual Machine View Send Key
160.506007 192.168.10.1 -> 192.168.10.2 OSPF DB Descr.
160.506078 192.168.10.1 -> 192.168.10.2 OSPF DB Descr.
160.506406 192.168.10.2 -> 192.168.10.1 OSPF DB Descr.
160.507954 192.168.10.1 -> 192.168.10.2 OSPF DB Descr.
160.508147 192.168.10.1 -> 224.0.0.5 OSPF LS Update
160.508583 192.168.10.2 -> 224.0.0.5 OSPF LS Update
161.321774 192.168.10.1 -> 224.0.0.5 OSPF LS Acknowledge
161.768702 fe80::5054:ff:fea4:7b2f -> ff02::5 OSPF Hello Packet
161.768974 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF DB Descr.
161.771168 fe80::5054:ff:fea4:7b2f -> fe80::5054:ff:fe40:7495 OSPF DB Descr.
161.772046 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF LS Request
161.772379 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF DB Descr.
161.773362 fe80::5054:ff:fea4:7b2f -> fe80::5054:ff:fe40:7495 OSPF LS Update
161.773955 fe80::5054:ff:fea4:7b2f -> fe80::5054:ff:fe40:7495 OSPF LS Request
161.773980 fe80::5054:ff:fea4:7b2f -> fe80::5054:ff:fe40:7495 OSPF DB Descr.
161.774194 fe80::5054:ff:fe40:7495 -> fe80::5054:ff:fea4:7b2f OSPF LS Update
161.774225 fe80::5054:ff:fe40:7495 -> ff02::5 OSPF LS Update
161.774417 fe80::5054:ff:fe40:7495 -> ff02::5 OSPF LS Update
161.775181 fe80::5054:ff:fea4:7b2f -> ff02::5 OSPF LS Update
164.776270 fe80::5054:ff:fe40:7495 -> ff02::5 OSPF LS Acknowledge
164.777629 fe80::5054:ff:fea4:7b2f -> ff02::5 OSPF LS Acknowledge
165.510671 RealtekU_a4:7b:2f -> RealtekU_40:74:95 ARP Who has 192.168.10.2? Tel
l 192.168.10.1
165.510708 RealtekU_40:74:95 -> RealtekU_a4:7b:2f ARP 192.168.10.2 is at 52:54:0
0:40:74:95

While checking tcpdump i noticed **Hello** packets being sent. These packets are multicasted periodically to 224.0.0.5 addresses on all interfaces(ospfv3 equivalent is ff02::5:). This helps the ospf know all its connected neighbours and their respective relationships. Since, it is periodic dynamic neighbors can also be identified and corresponding changes can then be made.
They should contain router ID which we is taken while initializing ospf protocol. It will have interval which tells us the frequency of hello packets.
Area ID (so that same area neighbours can exchange Hello packets only).
Since, neighbor IDs are dynamic, Hello packets don't have fixed size.
In our experiment I noticed 44 bytes of packet in R4 and 48 bytes packet in R2.

OSPF DB Descr - I think these are exchanged when new neighbour is introduced(that being the case here). As name signifies, it exchanges the networks's topology db in multiple packets.
This also includes router ID, area ID, etc

OSPF LS Request - Since, i reconnected my node to subnet A, this packet is responsible for sending any updated content after checking the outdated ones.
This also includes router ID, area ID, etc

I also noticed, **OSPF LS Update and Acknowledge packets**.

Question 7.1. (For OSPF)
Execute ping (only IPv4 is OK) from H1 to H2, and disconnect Subnet (A) by unplugging the LAN cable between the laptops. Keep pinging even after you disconnect the LAN cable (up to 1 minute should be enough), and observe what happens to the ping result.

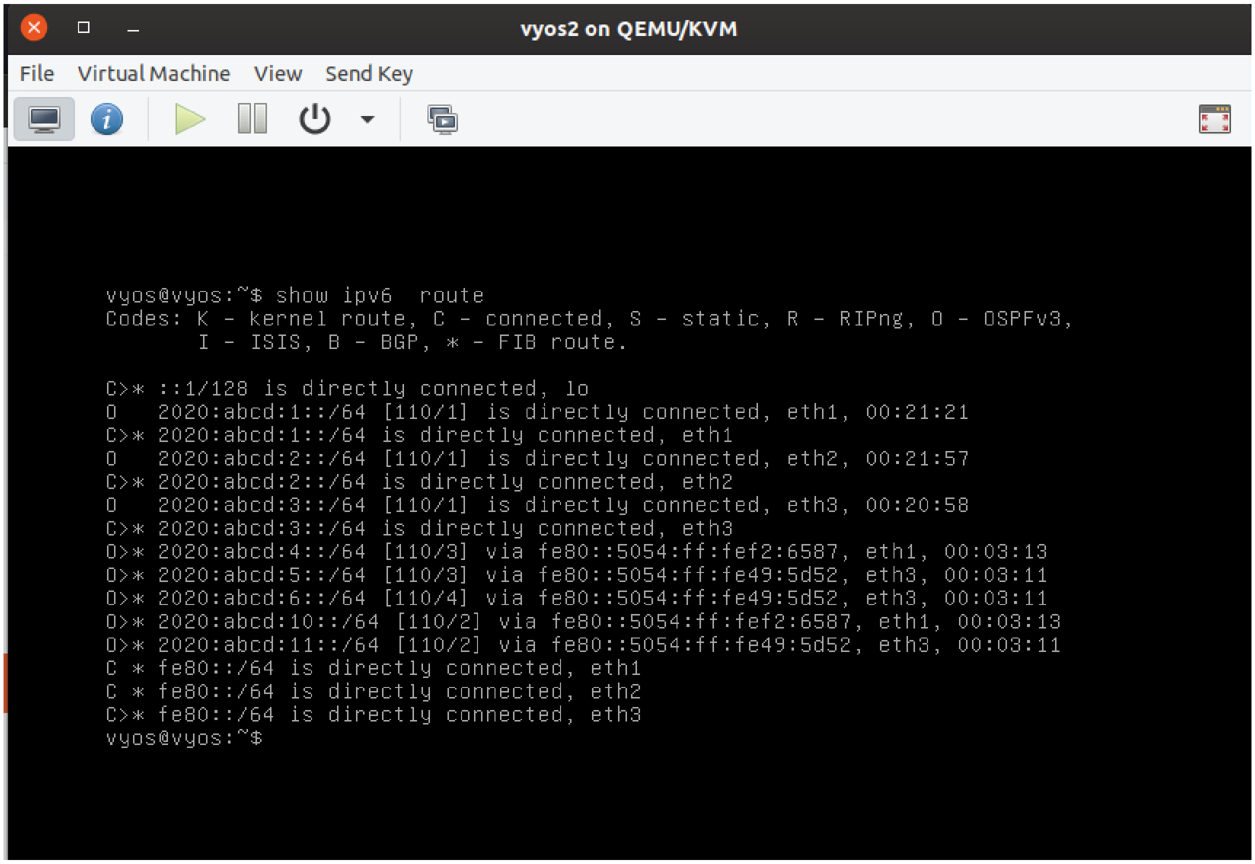
```
64 bytes from 192.168.6.2: icmp_req=8 ttl=60 time=4.30 ms
64 bytes from 192.168.6.2: icmp_req=9 ttl=60 time=4.74 ms
64 bytes from 192.168.6.2: icmp_req=10 ttl=60 time=4.80 ms
64 bytes from 192.168.6.2: icmp_req=12 ttl=60 time=4.96 ms
64 bytes from 192.168.6.2: icmp_req=16 ttl=60 time=4.83 ms
64 bytes from 192.168.6.2: icmp_req=20 ttl=60 time=4.58 ms
64 bytes from 192.168.6.2: icmp_req=24 ttl=60 time=4.72 ms
64 bytes from 192.168.6.2: icmp_req=28 ttl=60 time=4.57 ms
64 bytes from 192.168.6.2: icmp_req=32 ttl=60 time=3.96 ms
64 bytes from 192.168.6.2: icmp_req=36 ttl=60 time=4.48 ms
From 192.168.1.2 icmp_seq=37 Destination Host Unreachable
From 192.168.1.2 icmp_seq=39 Destination Host Unreachable
64 bytes from 192.168.6.2: icmp_req=40 ttl=60 time=4.56 ms
From 192.168.1.2 icmp_seq=41 Destination Host Unreachable
From 192.168.1.2 icmp_seq=43 Destination Host Unreachable
64 bytes from 192.168.6.2: icmp_req=44 ttl=60 time=4.17 ms
From 192.168.1.2 icmp_seq=45 Destination Host Unreachable
From 192.168.1.2 icmp_seq=47 Destination Host Unreachable
64 bytes from 192.168.6.2: icmp_req=48 ttl=60 time=4.53 ms
64 bytes from 192.168.6.2: icmp_req=49 ttl=60 time=4.27 ms
64 bytes from 192.168.6.2: icmp_req=50 ttl=60 time=4.55 ms
64 bytes from 192.168.6.2: icmp_req=51 ttl=60 time=4.45 ms
64 bytes from 192.168.6.2: icmp_req=52 ttl=60 time=4.16 ms
64 bytes from 192.168.6.2: icmp_req=53 ttl=60 time=3.87 ms
_
```

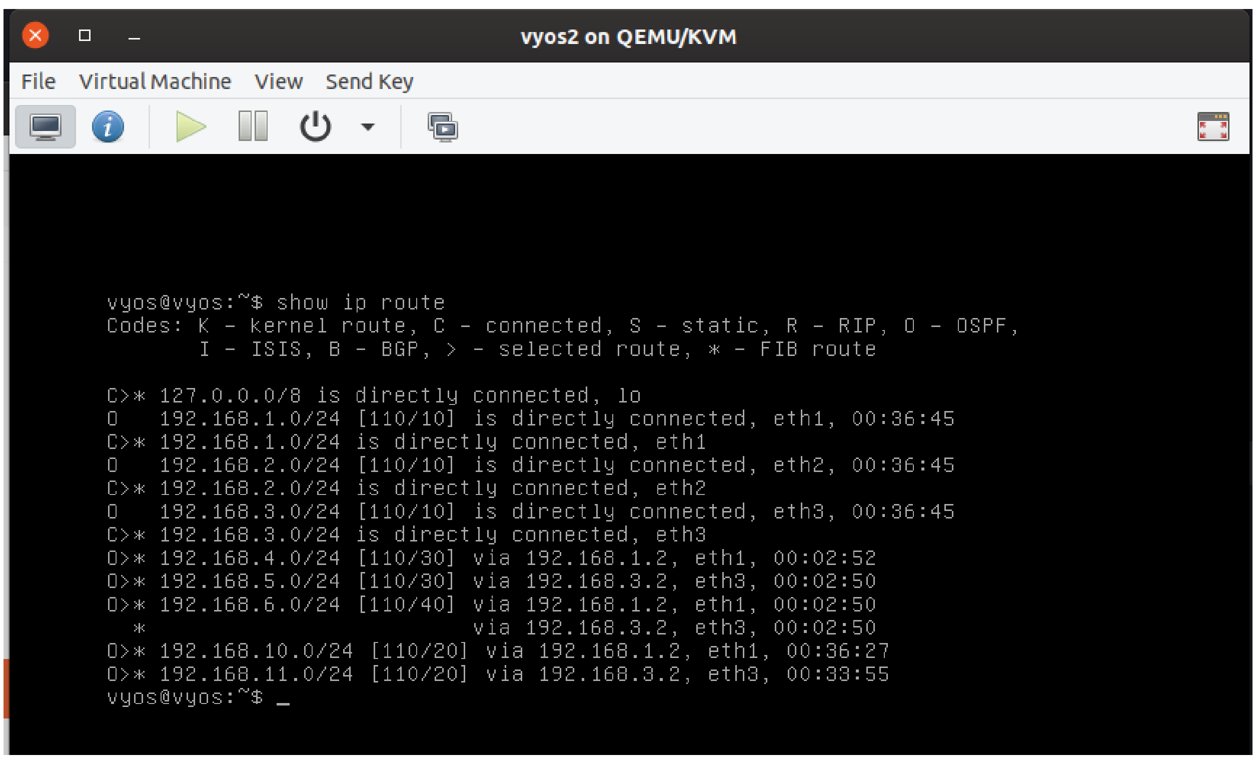
Previously, destination Host unreachable was the only output(static). But this time, ospf realises that a next-hop router is unavailable so it searches if any other path can be constructed such that it becomes fully connected . After finding such a path it imposes it ono all the corresponding ip tables. Hence after a delay (When network is down) the network comes back up with new path.

Question 7.2.

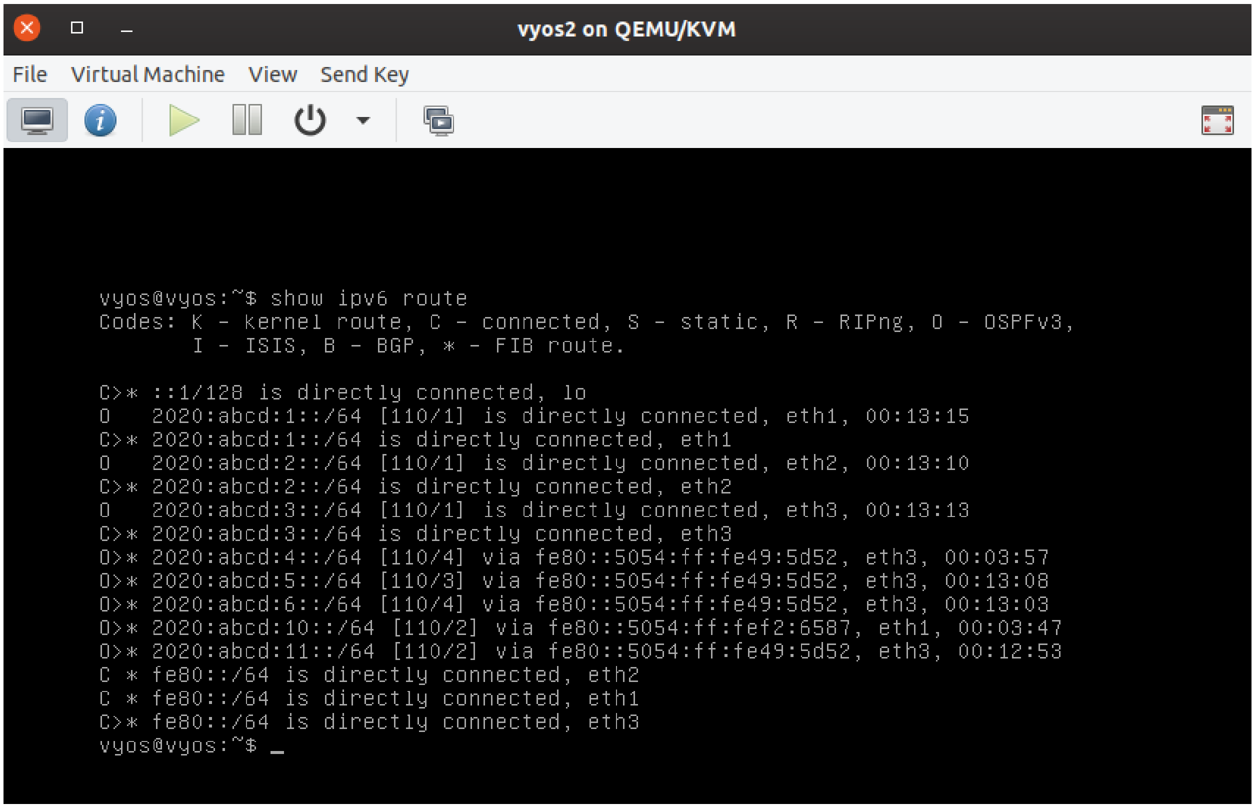
Insert the screen captures of the routing table on R2 to compare those before and after unplugging the LAN cable. And explain which change in the routing table is the cause of observation that you gave in the previous answer in 7.1.

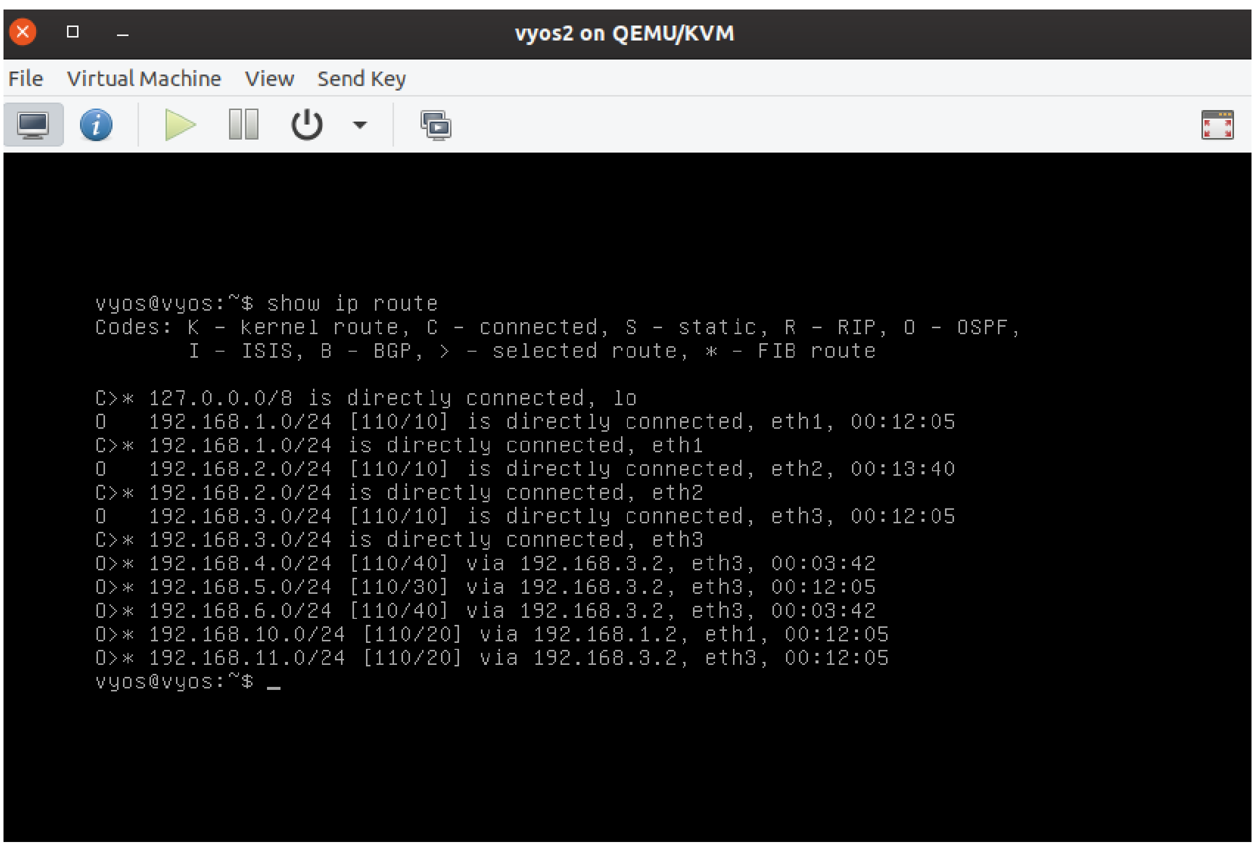
Before unplugging





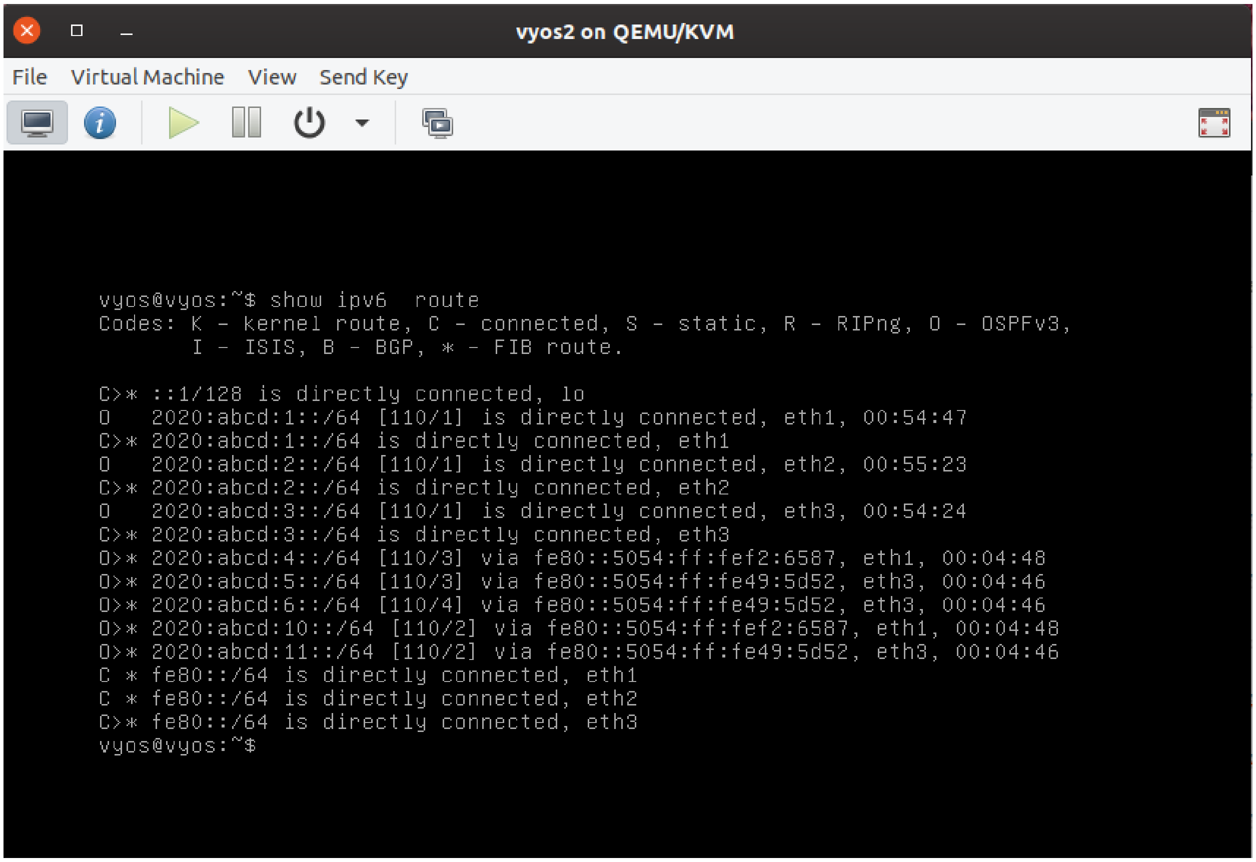
After unplugging


vyos2 on QEMU/KVM
File Virtual Machine View Send Key
vyos@vyos:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
I - ISIS, B - BGP, * - FIB route.
C>* ::1/128 is directly connected, lo
O 2020:abcd:1::/64 [110/1] is directly connected, eth1, 00:13:15
C>* 2020:abcd:1::/64 is directly connected, eth1
O 2020:abcd:2::/64 [110/1] is directly connected, eth2, 00:13:10
C>* 2020:abcd:2::/64 is directly connected, eth2
O 2020:abcd:3::/64 [110/1] is directly connected, eth3, 00:13:13
C>* 2020:abcd:3::/64 is directly connected, eth3
O>* 2020:abcd:4::/64 [110/4] via fe80::5054:ff:fe49:5d52, eth3, 00:03:57
O>* 2020:abcd:5::/64 [110/3] via fe80::5054:ff:fe49:5d52, eth3, 00:13:08
O>* 2020:abcd:6::/64 [110/4] via fe80::5054:ff:fe49:5d52, eth3, 00:13:03
O>* 2020:abcd:10::/64 [110/2] via fe80::5054:ff:fef2:6587, eth1, 00:03:47
O>* 2020:abcd:11::/64 [110/2] via fe80::5054:ff:fe49:5d52, eth3, 00:12:53
C * fe80::/64 is directly connected, eth2
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth3
vyos@vyos:~$ _

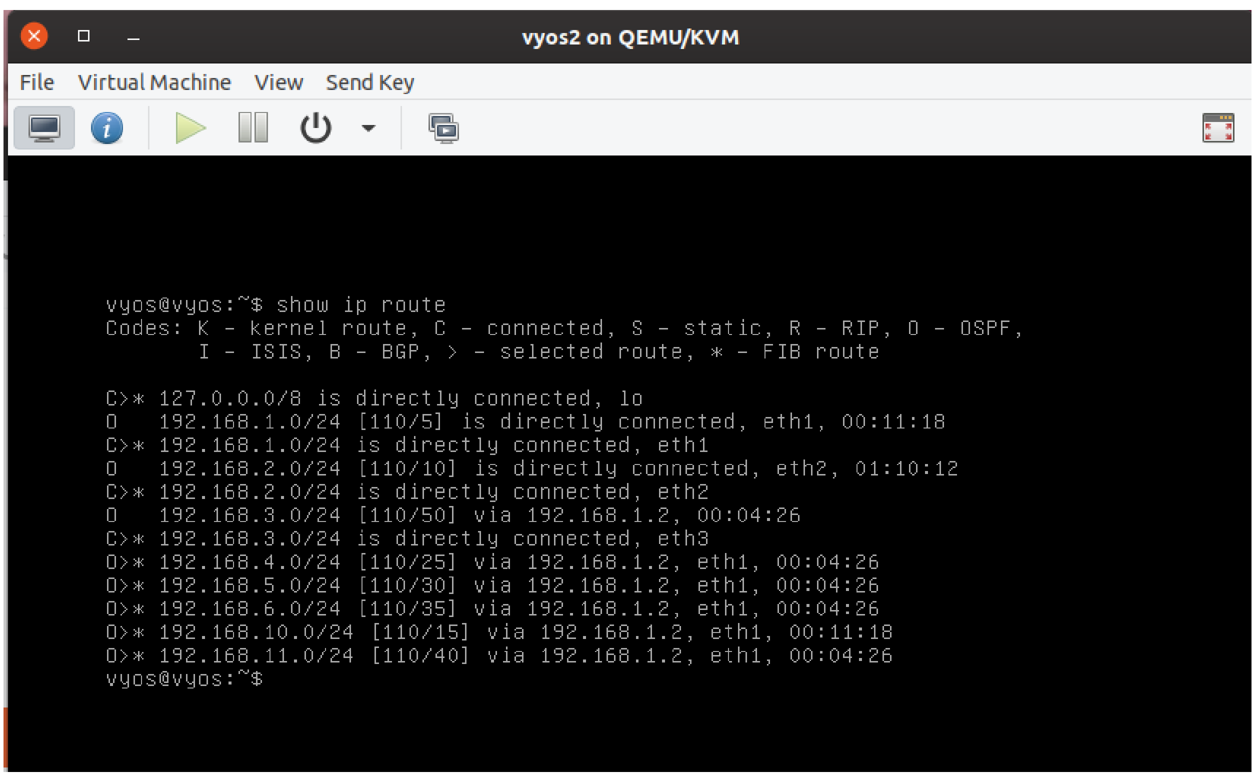

vyos2 on QEMU/KVM
File Virtual Machine View Send Key
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
I - ISIS, B - BGP, > - selected route, * - FIB route
C>* 127.0.0.0/8 is directly connected, lo
O 192.168.1.0/24 [110/10] is directly connected, eth1, 00:12:05
C>* 192.168.1.0/24 is directly connected, eth1
O 192.168.2.0/24 [110/10] is directly connected, eth2, 00:13:40
C>* 192.168.2.0/24 is directly connected, eth2
O 192.168.3.0/24 [110/10] is directly connected, eth3, 00:12:05
C>* 192.168.3.0/24 is directly connected, eth3
O>* 192.168.4.0/24 [110/40] via 192.168.3.2, eth3, 00:03:42
O>* 192.168.5.0/24 [110/30] via 192.168.3.2, eth3, 00:12:05
O>* 192.168.6.0/24 [110/40] via 192.168.3.2, eth3, 00:03:42
O>* 192.168.10.0/24 [110/20] via 192.168.1.2, eth1, 00:12:05
O>* 192.168.11.0/24 [110/20] via 192.168.3.2, eth3, 00:12:05
vyos@vyos:~$ _

Here, after comparing the screen captures we notice that the entries corresponding to subnet 4,5,6 are updated after removed subnet A from topology (switching off the corresponding vlan i/f). As for Host2's ip, previously it has two paths to reach it.(first screen shot). It then got updated to only ip entry as shown in the latter screenshot.

Question 8.
Configure OSPF Link Costs on the routers so that 1) ping traffic from H1 to H2 goes through Subnet (A), and 2) that from H2 to H1 goes through Subnet (B) using both IPv4 and IPv6. Answer by inserting the screen captures of the routing table on R2 and R6, and the tcpdump result on R1 and R5 on those you should observe the traffic is one way.
R2 routing info

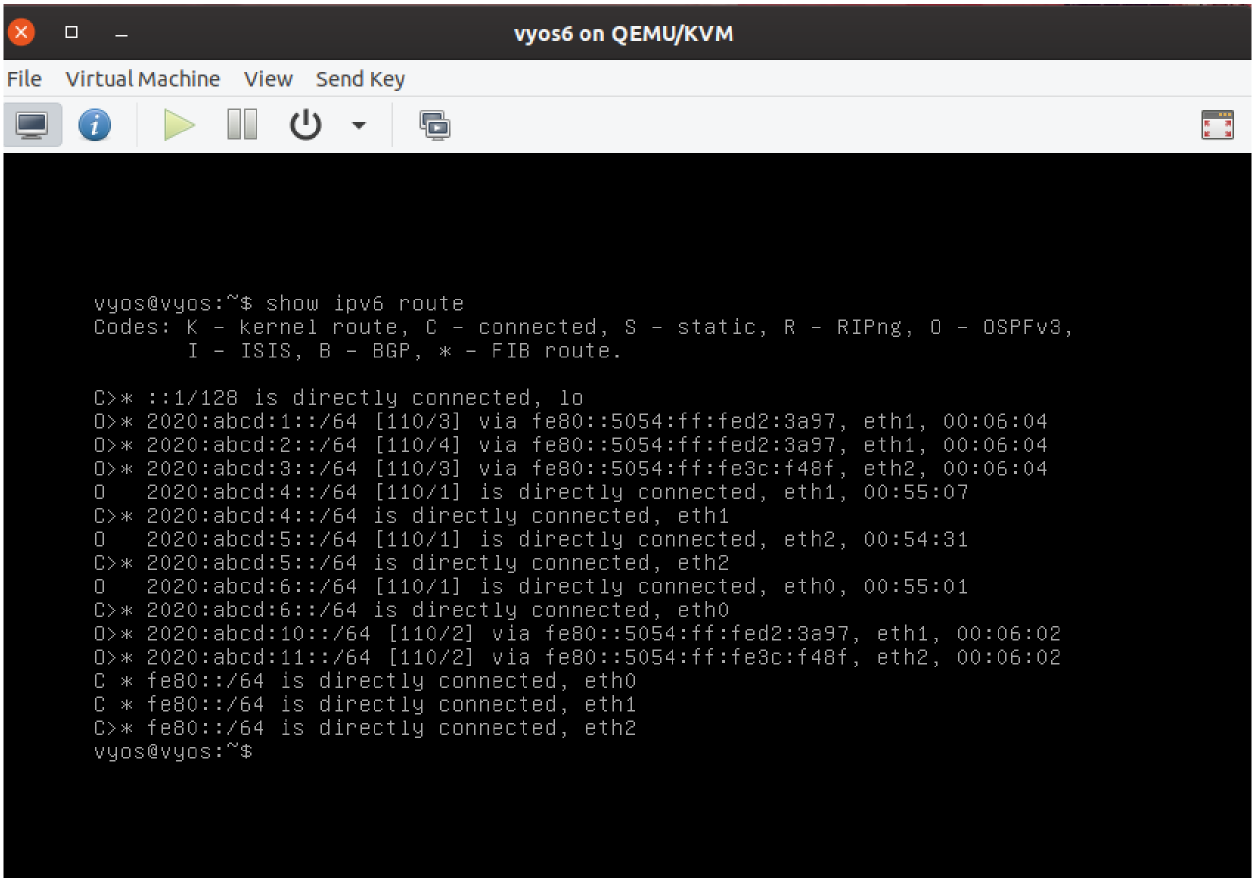


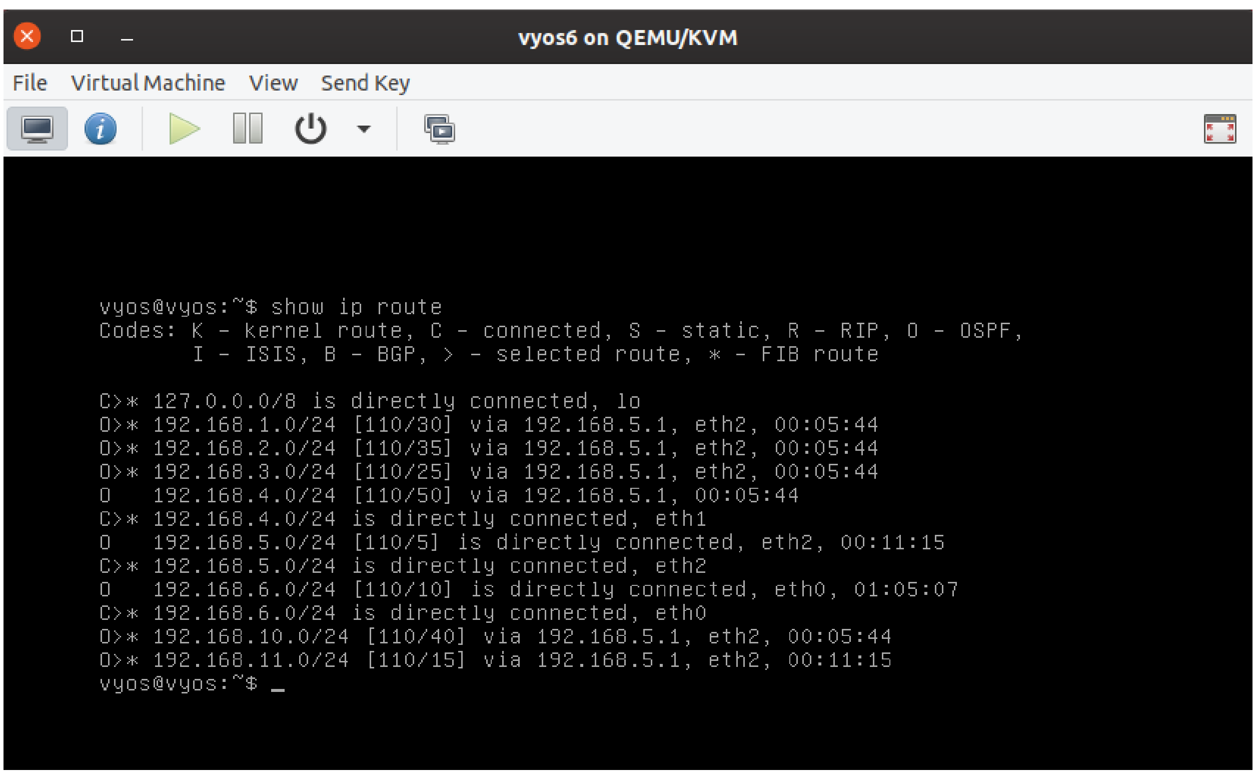

```
vyos@vyos:~$ show ipv6  route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
O   2020:abcd:1::/64 [110/1] is directly connected, eth1, 00:54:47
C>* 2020:abcd:1::/64 is directly connected, eth1
O   2020:abcd:2::/64 [110/1] is directly connected, eth2, 00:55:23
C>* 2020:abcd:2::/64 is directly connected, eth2
O   2020:abcd:3::/64 [110/1] is directly connected, eth3, 00:54:24
C>* 2020:abcd:3::/64 is directly connected, eth3
O>* 2020:abcd:4::/64 [110/3] via fe80::5054:ff:fef2:6587, eth1, 00:04:48
O>* 2020:abcd:5::/64 [110/3] via fe80::5054:ff:fe49:5d52, eth3, 00:04:46
O>* 2020:abcd:6::/64 [110/4] via fe80::5054:ff:fe49:5d52, eth3, 00:04:46
O>* 2020:abcd:10::/64 [110/2] via fe80::5054:ff:fef2:6587, eth1, 00:04:48
O>* 2020:abcd:11::/64 [110/2] via fe80::5054:ff:fe49:5d52, eth3, 00:04:46
C * fe80::/64 is directly connected, eth1
C * fe80::/64 is directly connected, eth2
C>* fe80::/64 is directly connected, eth3
vyos@vyos:~$
```



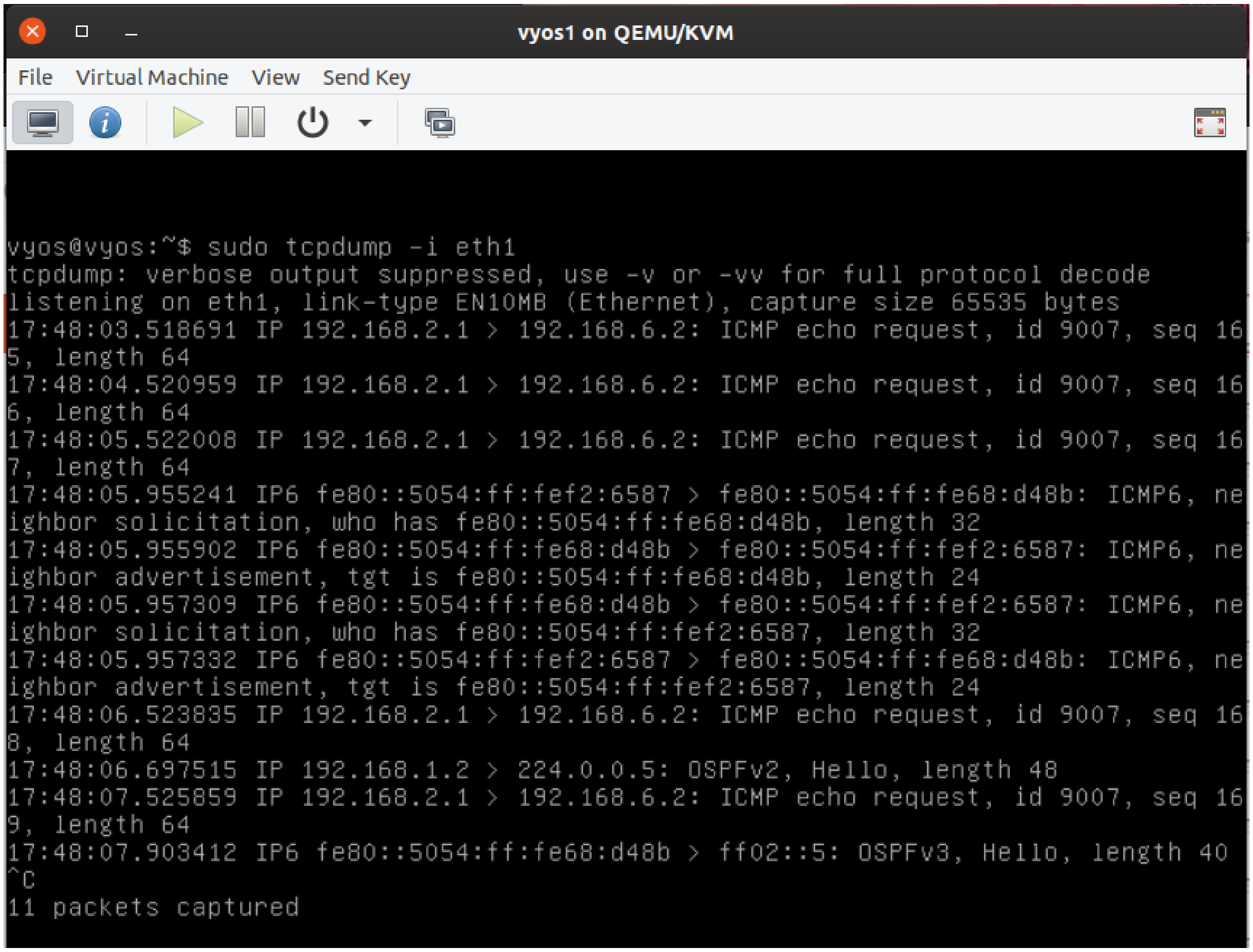

```
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo
O   192.168.1.0/24 [110/5] is directly connected, eth1, 00:11:18
C>* 192.168.1.0/24 is directly connected, eth1
O   192.168.2.0/24 [110/10] is directly connected, eth2, 01:10:12
C>* 192.168.2.0/24 is directly connected, eth2
O   192.168.3.0/24 [110/50] via 192.168.1.2, 00:04:26
C>* 192.168.3.0/24 is directly connected, eth3
O>* 192.168.4.0/24 [110/25] via 192.168.1.2, eth1, 00:04:26
O>* 192.168.5.0/24 [110/30] via 192.168.1.2, eth1, 00:04:26
O>* 192.168.6.0/24 [110/35] via 192.168.1.2, eth1, 00:04:26
O>* 192.168.10.0/24 [110/15] via 192.168.1.2, eth1, 00:11:18
O>* 192.168.11.0/24 [110/40] via 192.168.1.2, eth1, 00:04:26
vyos@vyos:~$
```

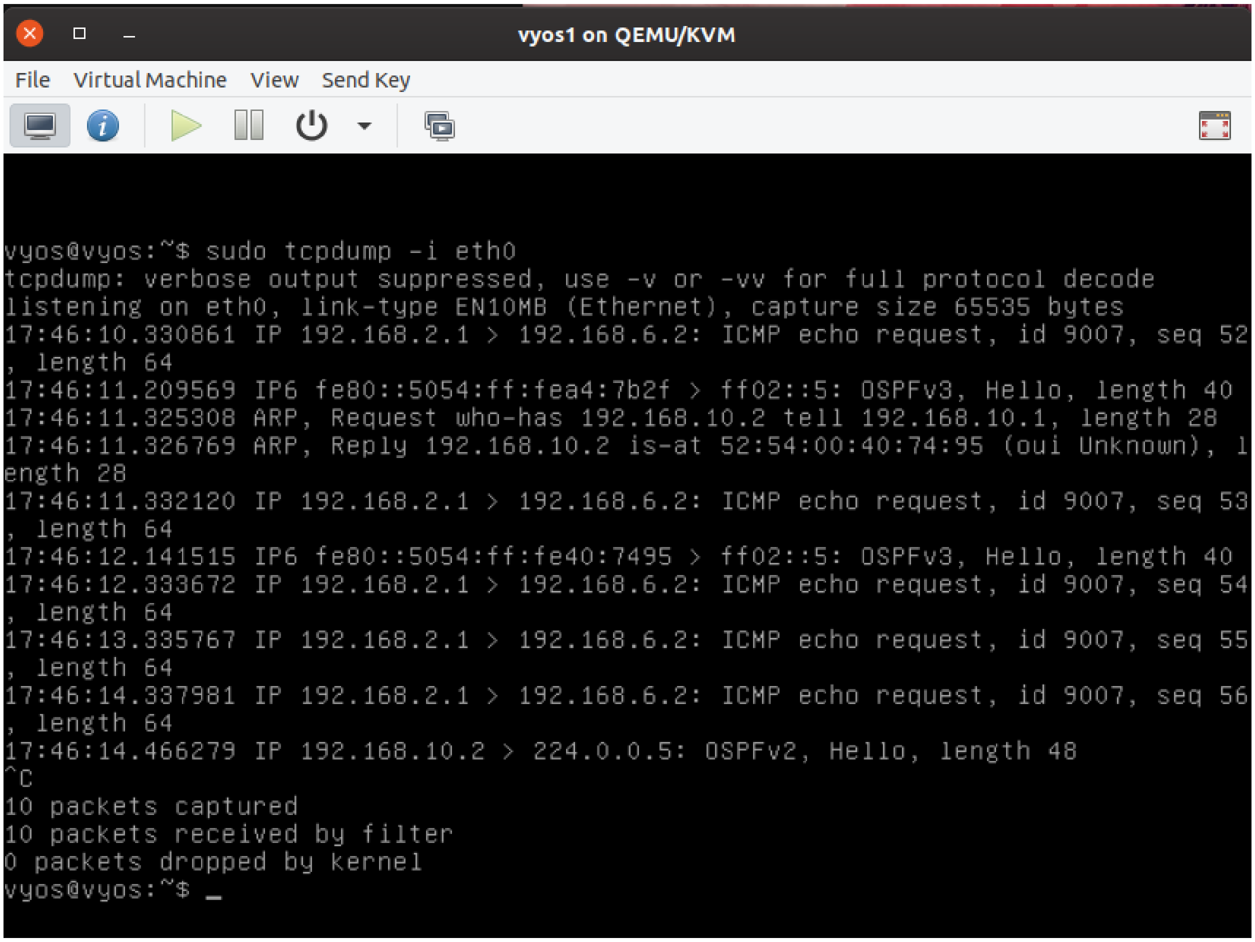
R6 routing info


vyos6 on QEMU/KVM
File
Virtual Machine
View
Send Key
vyos@vyos:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
I - ISIS, B - BGP, * - FIB route.
C>* ::1/128 is directly connected, lo
O>* 2020:abcd:1::/64 [110/3] via fe80::5054:ff:fed2:3a97, eth1, 00:06:04
O>* 2020:abcd:2::/64 [110/4] via fe80::5054:ff:fed2:3a97, eth1, 00:06:04
O>* 2020:abcd:3::/64 [110/3] via fe80::5054:ff:fe3c:f48f, eth2, 00:06:04
O   2020:abcd:4::/64 [110/1] is directly connected, eth1, 00:55:07
C>* 2020:abcd:4::/64 is directly connected, eth1
O   2020:abcd:5::/64 [110/1] is directly connected, eth2, 00:54:31
C>* 2020:abcd:5::/64 is directly connected, eth2
O   2020:abcd:6::/64 [110/1] is directly connected, eth0, 00:55:01
C>* 2020:abcd:6::/64 is directly connected, eth0
O>* 2020:abcd:10::/64 [110/2] via fe80::5054:ff:fed2:3a97, eth1, 00:06:02
O>* 2020:abcd:11::/64 [110/2] via fe80::5054:ff:fe3c:f48f, eth2, 00:06:02
C * fe80::/64 is directly connected, eth0
C * fe80::/64 is directly connected, eth1
C>* fe80::/64 is directly connected, eth2
vyos@vyos:~$

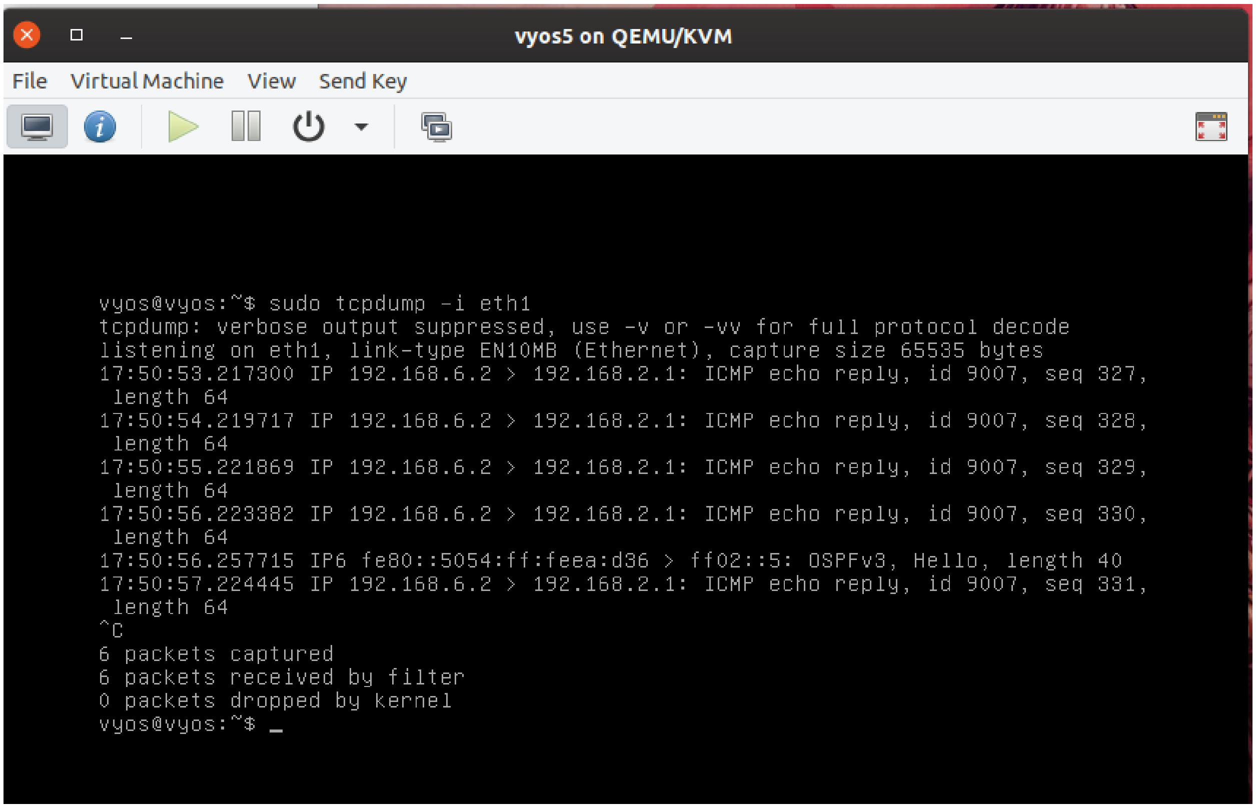

vyos6 on QEMU/KVM
File
Virtual Machine
View
Send Key
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
I - ISIS, B - BGP, > - selected route, * - FIB route
C>* 127.0.0.0/8 is directly connected, lo
O>* 192.168.1.0/24 [110/30] via 192.168.5.1, eth2, 00:05:44
O>* 192.168.2.0/24 [110/35] via 192.168.5.1, eth2, 00:05:44
O>* 192.168.3.0/24 [110/25] via 192.168.5.1, eth2, 00:05:44
O   192.168.4.0/24 [110/50] via 192.168.5.1, 00:05:44
C>* 192.168.4.0/24 is directly connected, eth1
O   192.168.5.0/24 [110/5] is directly connected, eth2, 00:11:15
C>* 192.168.5.0/24 is directly connected, eth2
O   192.168.6.0/24 [110/10] is directly connected, eth0, 01:05:07
C>* 192.168.6.0/24 is directly connected, eth0
O>* 192.168.10.0/24 [110/40] via 192.168.5.1, eth2, 00:05:44
O>* 192.168.11.0/24 [110/15] via 192.168.5.1, eth2, 00:11:15
vyos@vyos:~$ _

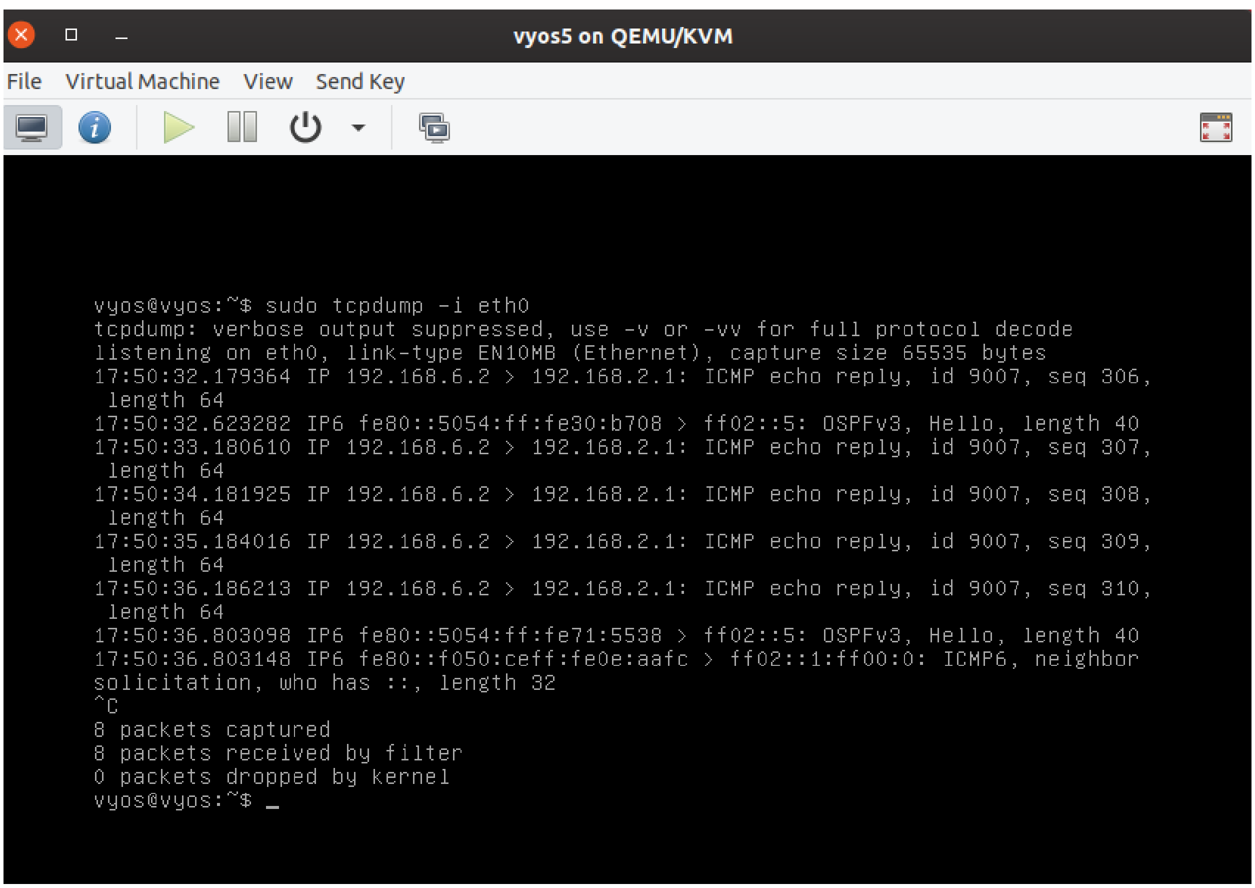
tcpdump result on R1





tcpdump result on R5


vyos5 on QEMU/KVM
File Virtual Machine View Send Key
vyos@vyos:~$ sudo tcpdump -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
17:50:53.217300 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 327, length 64
17:50:54.219717 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 328, length 64
17:50:55.221869 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 329, length 64
17:50:56.223382 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 330, length 64
17:50:56.257715 IP6 fe80::5054:ff:feea:d36 > ff02::5: OSPFv3, Hello, length 40
17:50:57.224445 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 331, length 64
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _


vyos5 on QEMU/KVM
File Virtual Machine View Send Key
vyos@vyos:~$ sudo tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
17:50:32.179364 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 306, length 64
17:50:32.623282 IP6 fe80::5054:ff:fe30:b708 > ff02::5: OSPFv3, Hello, length 40
17:50:33.180610 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 307, length 64
17:50:34.181925 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 308, length 64
17:50:35.184016 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 309, length 64
17:50:36.186213 IP 192.168.6.2 > 192.168.2.1: ICMP echo reply, id 9007, seq 310, length 64
17:50:36.803098 IP6 fe80::5054:ff:fe71:5538 > ff02::5: OSPFv3, Hello, length 40
17:50:36.803148 IP6 fe80::f050:ceff:fe0e:aafc > ff02::1:ff00:0: ICMP6, neighbor solicitation, who has ::, length 32
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _

Done!!

Some notes:

1. OSPF ECMP was omitted because it makes less sense in this scenario.
2. If you're stuck in VLAN configuration, ping TAs or come to the lab session.