# CS3543 Lab Assignment 1

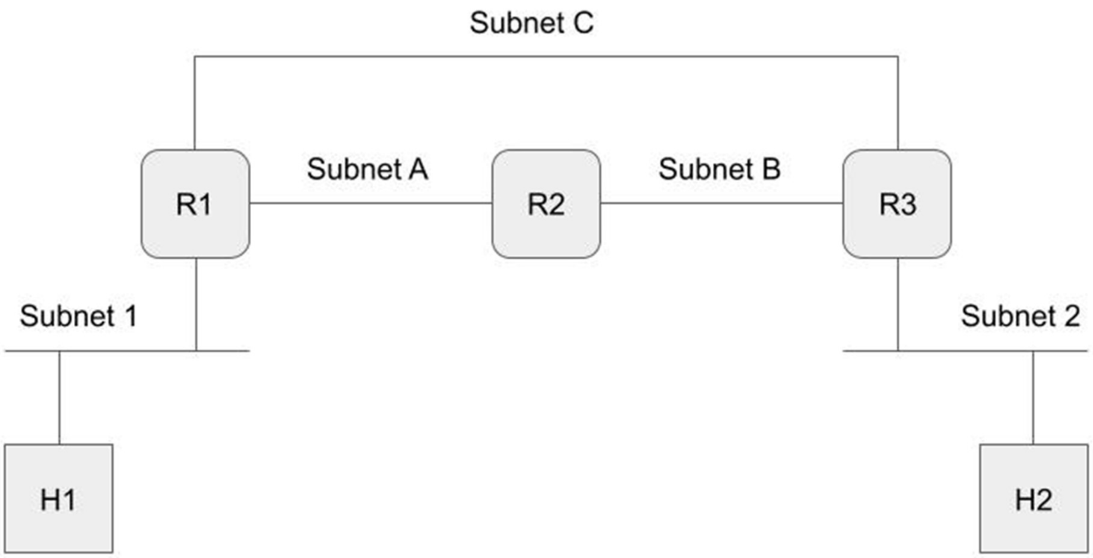
| | | |
|---|---|---|
| Member 1: | RAJ DEEPAKNATH PATIL | (CS18BTECH11039) |
| Member 2: | DEEPANSHU CHOUHAN | (CS18BTECH11009) |
| Member 3: | HIMANSHU BISHNOI | (CS16BTECH11018) |
| Member 4: | VIJAY PONNEKANTI | (CS16BTECH11028) |

# General Instructions

1. This assignment must be conducted and submitted by a group of students (at least 2 members, up to 4 members). The same mark will be offered to the students in the same group 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 group should create a locally copy of this question file and the supplemental presentation file, give the answer to the local copy, and submit in a form of PDF file.
4. Only up to one submission must be made per group.
5. Name and Student ID of all the group members must be mentioned. Any student, whose name and student ID are not properly mentioned, may not receive marks no matter how much his/her contribution could be.
6. Do not send any private comment to separately mention the name and student ID of group members.
7. If you want to send a pcap file from a VyOS VM to your host Ubunt, you can give an IP address to the linux bridge which the VyOS VM connects to. You may enable sshd on the VM and use scp on the host Ubuntu.

# Warming Up

In this lab assignment, each team is requested to form the network using Linux Bridge and VMs running Ubuntu servers and VyOS routers. R1, R2, R3 are routers, H1 and H2 are hosts. The IP address for each subnet has not been fixed. You need to fix the prefix information and properly note down to configure the hosts and routers based on it.



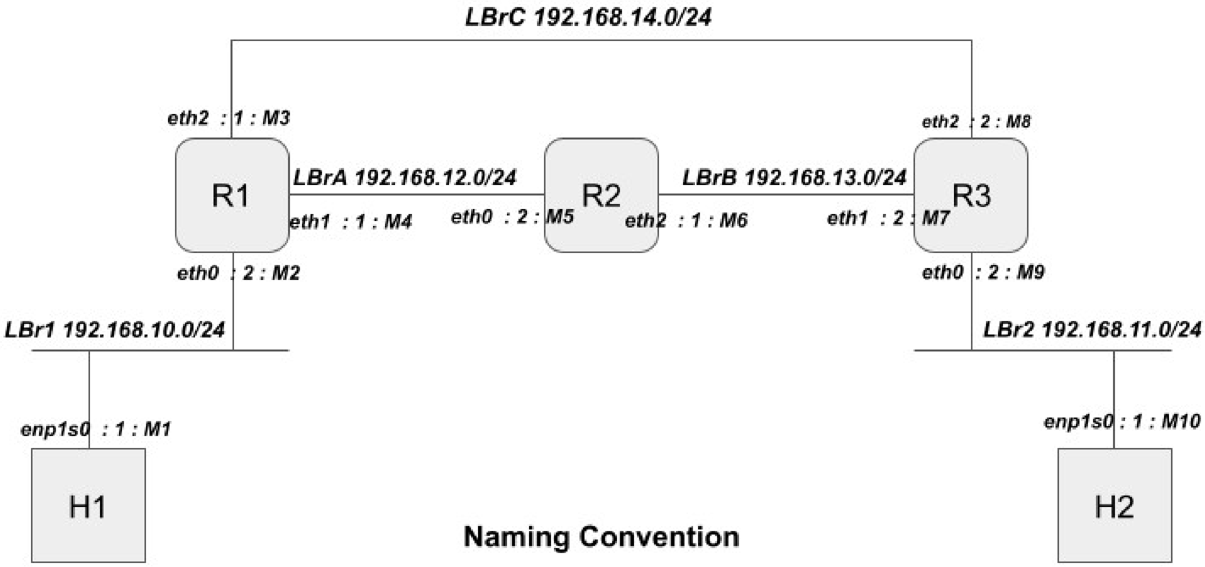
Question 1. (5 marks all together)

Fill the blanks in Table 1 to clarify NIC and IPv4 to belong to Subnets 1 to 4. If there is no corresponding NIC belonging to a subnet, mention "N/A". All the prefixes must be planned by yourself.

| | **Linux Bridge** | **H1** | **H2** | **R1** | **R2** | **R3** |
|---|---|---|---|---|---|---|
| **Subnet 1** | **LBr1** | *eth0*<br>*192.168.10.1/24* | N/A | *eth0*<br>*192.168.10.2/24* | N/A | N/A |
| **Subnet 2** | **LBr2** | N/A | *eth0*<br>*192.168.11.1/24* | N/A | N/A | *eth0*<br>*192.168.11.2/24* |
| **Subnet A** | **LBrA** | N/A | N/A | *eth1*<br>*192.168.12.1/24* | *eth0*<br>*192.168.12.2/24* | N/A |
| **Subnet B** | **LBrB** | N/A | N/A | N/A | *eth1*<br>*192.168.13.1/24* | *eth1*<br>*192.168.13.2/24* |
| **Subnet C** | **LBrC** | N/A | N/A | *eth2*<br>*192.168.14.1/24* | N/A | *eth2*<br>*192.168.14.2/24* |

Question 2. (5 marks)

Illustrate the network diagram that appropriately contains the information given in Table 1. The full mark will be given if the network diagram fully covers the information given in the table. (Linux Bridge and other information must also be mentioned for the sake of explainability of answers to the following questions). The original presentation file can be locally copied to your Google Drive and used to work on this assignment.



**Naming Convention**

| | |
|---|---|
| For NIC | <NIC name> : <Ipv4 host part(last 8 bits)> : <MacAddr Index(check appended Table)> |
| For Linux Bridge | <Bridge Name> : <Subnet addr/Mask len> |

MAC ADDR INDEX (CHECK FIGURE ABOVE)

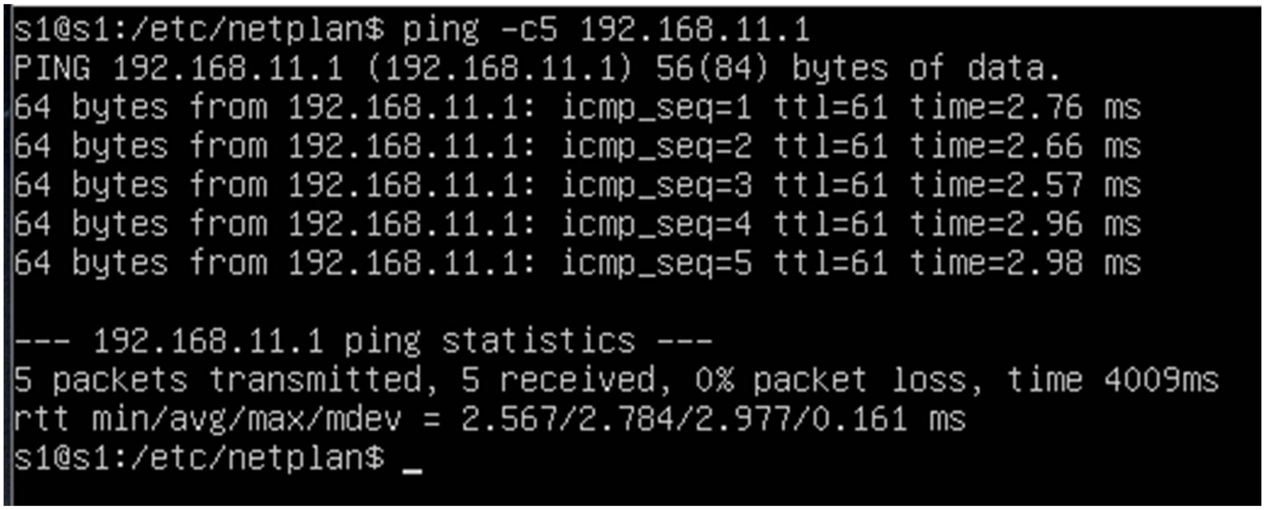
| Mac addr index | Mac addr | Mac addr index | Mac addr |
|---|---|---|---|
| **M1** | *52:54:00:3d:0d:cb* | **M6** | *52:54:00:21:97:c8* |
| **M2** | *52:54:00:65:f9:ba* | **M7** | *52:54:00:67:a6:2b* |
| **M3** | *52:54:00:32:f7:34* | **M8** | *52:54:00:32:ae:79* |
| **M4** | *52:54:00:eb:f9:16* | **M9** | *52:54:00:c5:bd:38* |
| **M5** | *52:54:00:6e:94:af* | **M10** | *52:54:00:63:d4:41* |

# NIC configuration and Static Routing Instruction

1. Configure all NICs of the hosts and routers (H1, H2, R1, R2 and R3) as planned in Table 1 and the network diagram.
2. Manually configure the routing table of all the routers so that 1) the path from H1 to H2 is always {H1 -> R1 -> R3 -> H2} and 2) the path from H2 to H1 is always {H2 -> R3 -> R2 -> R1 -> H1}.
3. Make sure that H1 and H2 can ping with each other.

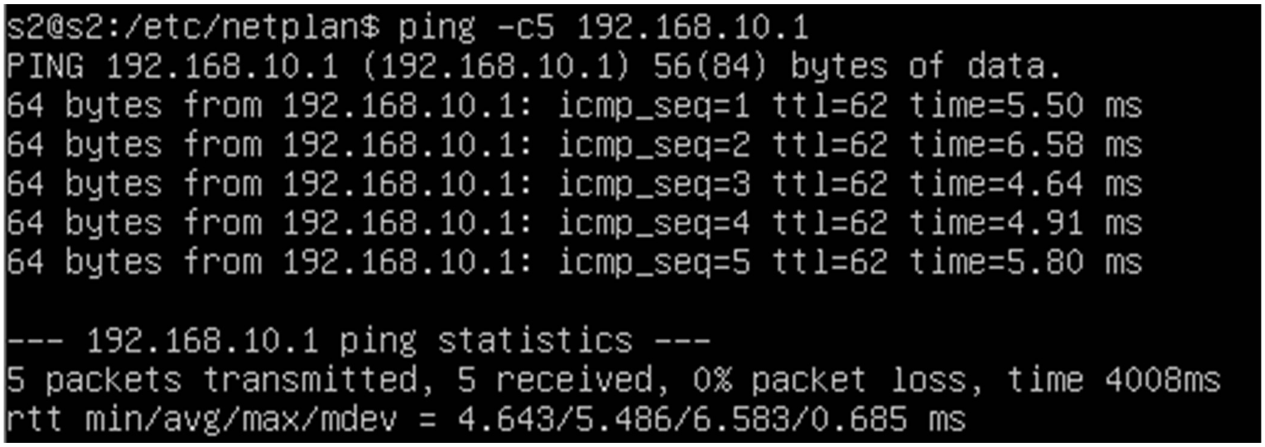
Question 3.1 (5 marks)
Paste the screen capture of the ping command from H1 and H2 to show that the static routing configuration is working to allow H1 and H2 to communicate with each other.

H1 to H2



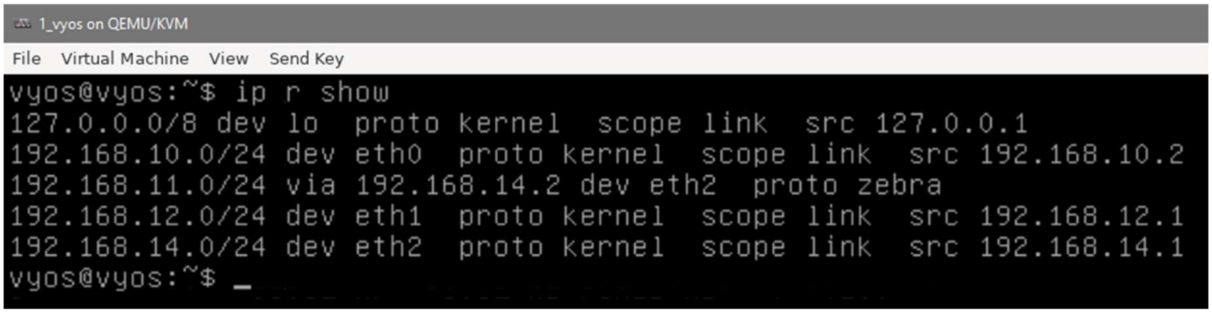
```
s1@s1:/etc/netplan$ ping -c5 192.168.11.1
PING 192.168.11.1 (192.168.11.1) 56(84) bytes of data.
64 bytes from 192.168.11.1: icmp_seq=1 ttl=61 time=2.76 ms
64 bytes from 192.168.11.1: icmp_seq=2 ttl=61 time=2.66 ms
64 bytes from 192.168.11.1: icmp_seq=3 ttl=61 time=2.57 ms
64 bytes from 192.168.11.1: icmp_seq=4 ttl=61 time=2.96 ms
64 bytes from 192.168.11.1: icmp_seq=5 ttl=61 time=2.98 ms

--- 192.168.11.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 2.567/2.784/2.977/0.161 ms
s1@s1:/etc/netplan$ _
```

H2 to H1



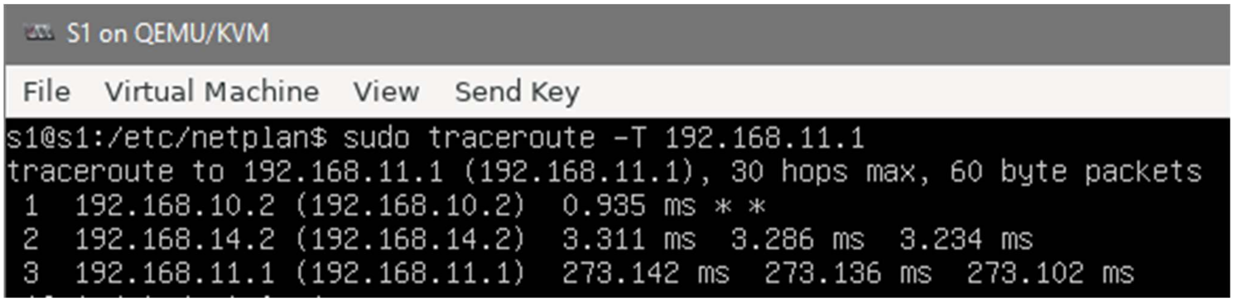
```
s2@s2:/etc/netplan$ ping -c5 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=62 time=5.50 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=62 time=6.58 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=62 time=4.64 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=62 time=4.91 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=62 time=5.80 ms

--- 192.168.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 4.643/5.486/6.583/0.685 ms
```

Question 3.b (5 marks)
Paste the screen capture of the routing table of R1.



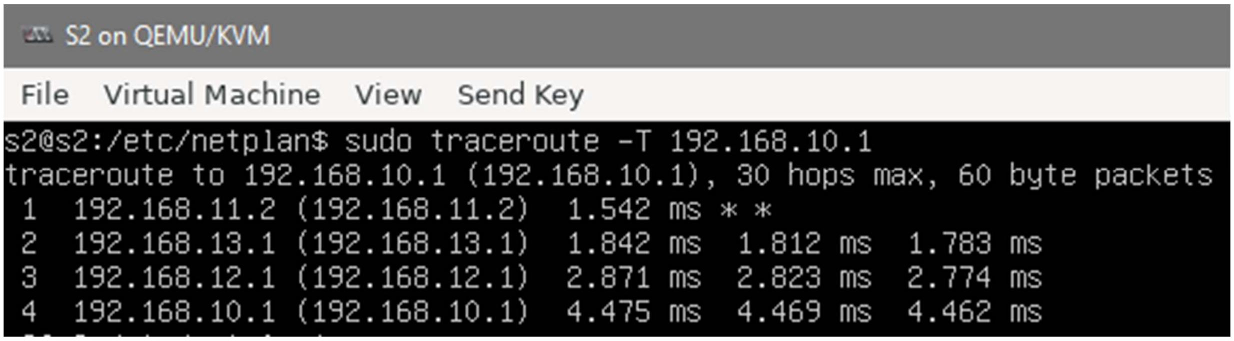
```
vyos@vyos:~$ ip r show
127.0.0.0/8 dev lo  proto kernel  scope link  src 127.0.0.1
192.168.10.0/24 dev eth0  proto kernel  scope link  src 192.168.10.2
192.168.11.0/24 via 192.168.14.2 dev eth2  proto zebra
192.168.12.0/24 dev eth1  proto kernel  scope link  src 192.168.12.1
192.168.14.0/24 dev eth2  proto kernel  scope link  src 192.168.14.1
vyos@vyos:~$ _
```

Question 3.c (5 marks)
Perform traceroute from H1 to H2 so that the path is following the instruction. Paste the screen capture of the traceroute result of H1.



```
s1@s1:/etc/netplan$ sudo traceroute -T 192.168.11.1
traceroute to 192.168.11.1 (192.168.11.1), 30 hops max, 60 byte packets
 1  192.168.10.2 (192.168.10.2)  0.935 ms * *
 2  192.168.14.2 (192.168.14.2)  3.311 ms  3.286 ms  3.234 ms
 3  192.168.11.1 (192.168.11.1)  273.142 ms  273.136 ms  273.102 ms
```

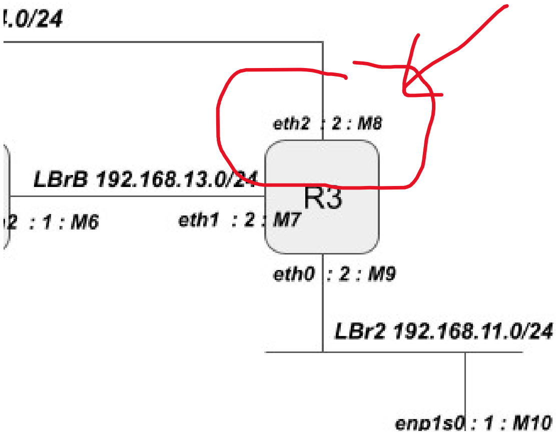
Question 3.d (5 marks)
Perform traceroute from H2 to H1 so that the path is following the instruction. Paste the screen capture of the traceroute result of H2.



```
s2@s2:/etc/netplan$ sudo traceroute -T 192.168.10.1
traceroute to 192.168.10.1 (192.168.10.1), 30 hops max, 60 byte packets
 1  192.168.11.2 (192.168.11.2)  1.542 ms * *
 2  192.168.13.1 (192.168.13.1)  1.842 ms  1.812 ms  1.783 ms
 3  192.168.12.1 (192.168.12.1)  2.871 ms  2.823 ms  2.774 ms
 4  192.168.10.1 (192.168.10.1)  4.475 ms  4.469 ms  4.462 ms
```

Question 3.e (5 marks for the perfect answer.)
When a packet from H1 to H2 is transmitted by R1, whose MAC address is set as the destination address in the Ethernet header? Answer the names of node and NIC respectively.

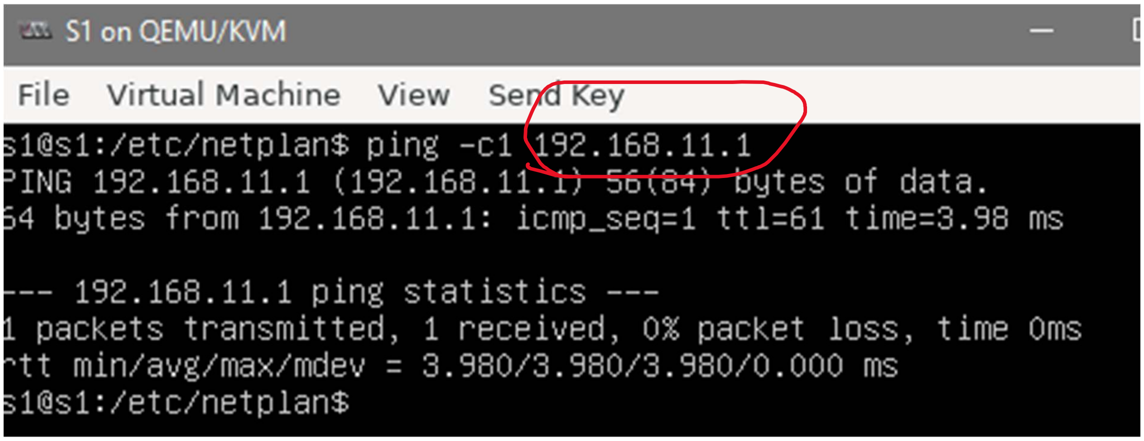
Short Answer:
NIC name and Node: ***eth2 of Router 3(router3) with MacAddress as M8 i.e. :*** *52:54:00:32:ae:79*



Proof:

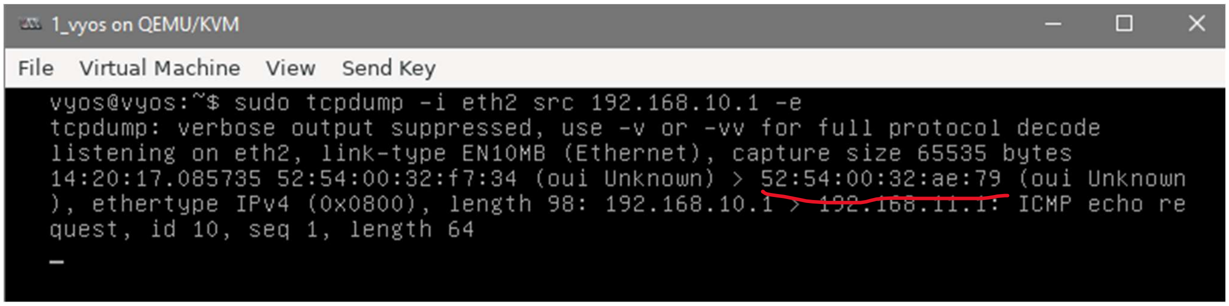
Running tcpdump on R1 and R3 filtering packets as follows and use `-e` to observe the ethernet headers:

Pinging H1 to H2:



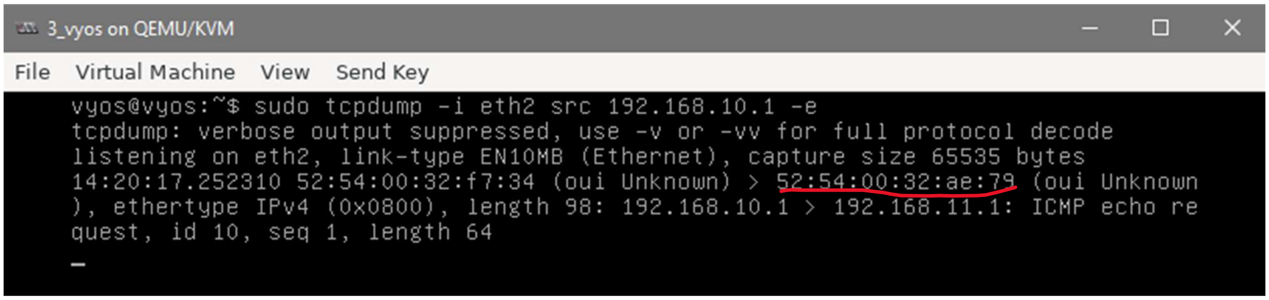
```
s1@s1:/etc/netplan$ ping -c1 192.168.11.1
PING 192.168.11.1 (192.168.11.1) 56(84) bytes of data.
64 bytes from 192.168.11.1: icmp_seq=1 ttl=61 time=3.98 ms

--- 192.168.11.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.980/3.980/3.980/0.000 ms
s1@s1:/etc/netplan$
```

Tcpdump output on R1:



```
vyos@vyos:~$ sudo tcpdump -i eth2 src 192.168.10.1 -e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
14:20:17.085735 52:54:00:32:f7:34 (oui Unknown) > 52:54:00:32:ae:79 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 192.168.10.1 > 192.168.11.1: ICMP echo re
quest, id 10, seq 1, length 64
_
```

Tcpdump output on R3:



```
vyos@vyos:~$ sudo tcpdump -i eth2 src 192.168.10.1 -e
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
14:20:17.252310 52:54:00:32:f7:34 (oui Unknown) > 52:54:00:32:ae:79 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 192.168.10.1 > 192.168.11.1: ICMP echo re
quest, id 10, seq 1, length 64
_
```

#
# Dynamic Routing Instruction using OSPF
#

1. Flush the static routing configuration from all the routers.
2. Enable tcpdump on both of R1's NICs, on which OSPF is enabled,and save (write) the captured packets. The packet capture files will be used to answer a question.
3. Enable OSPF. You can configure all the NICs of routers to belong to Area 0.
4. Make sure that H1 and H2 can ping with each other.

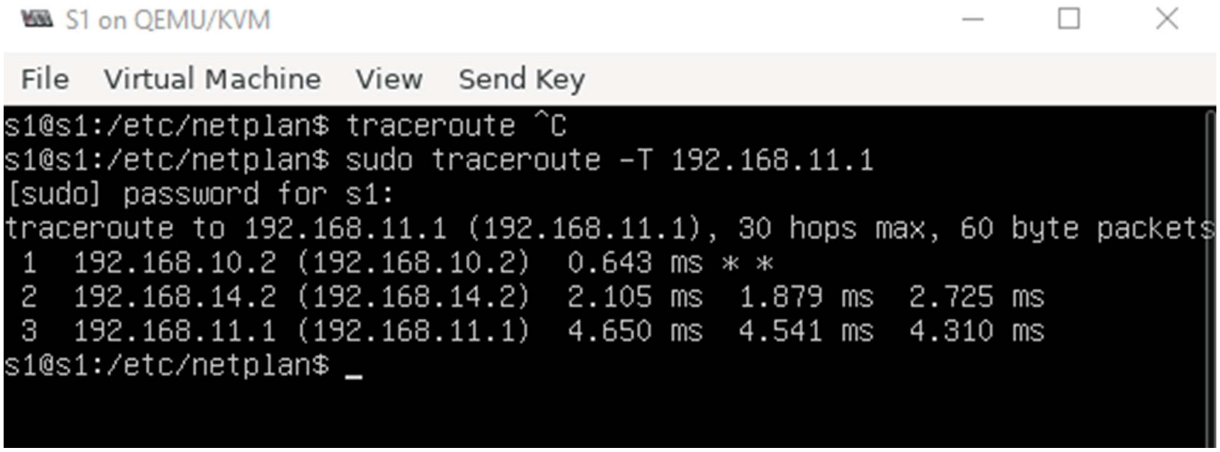
Meta Information:
Router-ids in the configuration have been set as shown below and the all of them have been set to area 0 as requested in the question.

| Router | Router id in ospf configuration |
|---|---|
| R1 | 0.0.0.1 |
| R2 | 0.0.0.2 |
| R3 | 0.0.0.3 |

Question 4.a. (5 marks all together)
Perform traceroute from H1 to H2 as well as from H2 to H1. 1) Explain the path of both directions, 2) paste the screen captures of traceroute for both directions.

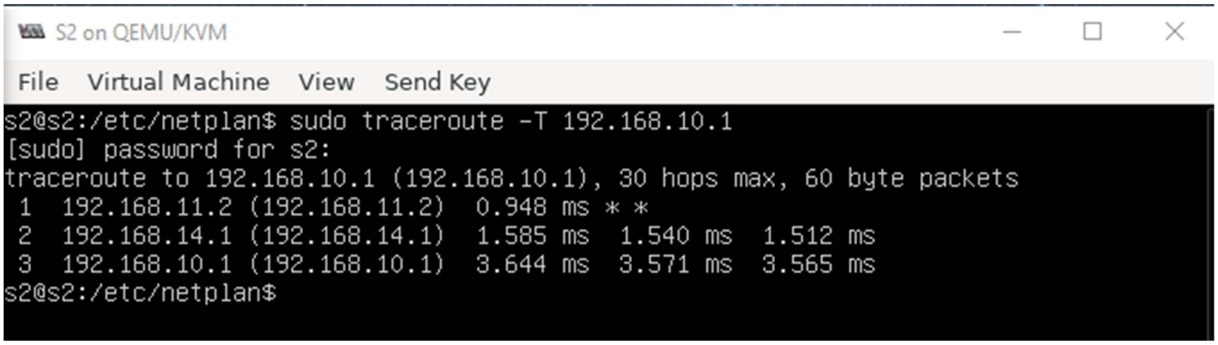
1) In both cases, this time, The shorter path is chosen – the one that skips R2 completely i.e. the one that uses the linux Bridge LbrC.

- Output for H1 to H2



```
s1@s1:/etc/netplan$ traceroute ^C
s1@s1:/etc/netplan$ sudo traceroute -T 192.168.11.1
[sudo] password for s1:
traceroute to 192.168.11.1 (192.168.11.1), 30 hops max, 60 byte packets
 1  192.168.10.2 (192.168.10.2)  0.643 ms * *
 2  192.168.14.2 (192.168.14.2)  2.105 ms  1.879 ms  2.725 ms
 3  192.168.11.1 (192.168.11.1)  4.650 ms  4.541 ms  4.310 ms
s1@s1:/etc/netplan$ _
```

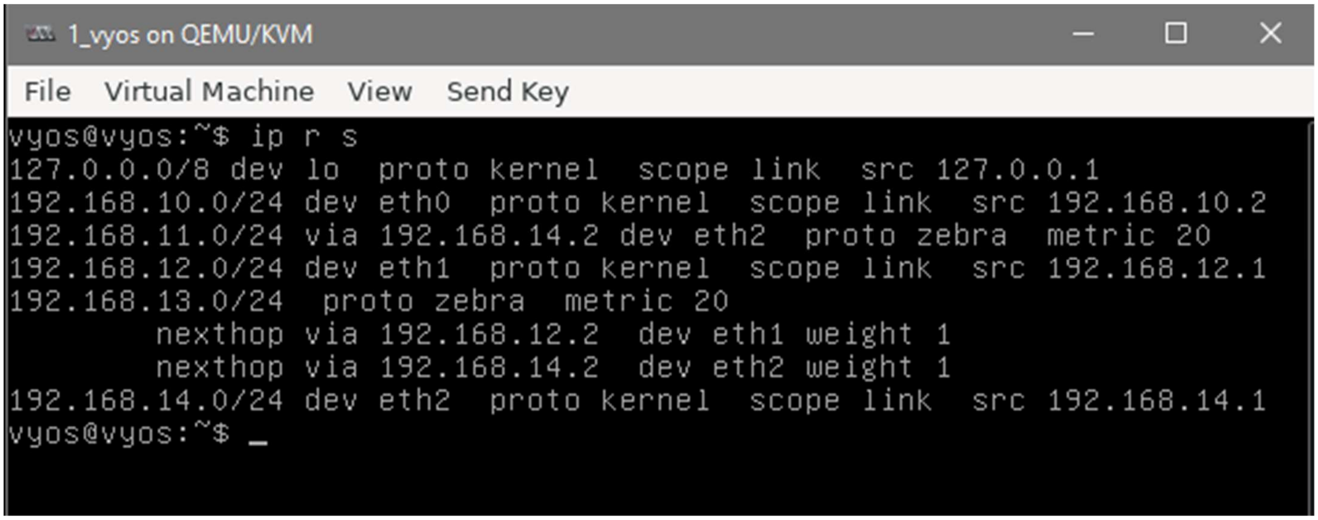
- Output for H2 to H1



```
s2@s2:/etc/netplan$ sudo traceroute -T 192.168.10.1
[sudo] password for s2:
traceroute to 192.168.10.1 (192.168.10.1), 30 hops max, 60 byte packets
 1  192.168.11.2 (192.168.11.2)  0.948 ms * *
 2  192.168.14.1 (192.168.14.1)  1.585 ms  1.540 ms  1.512 ms
 3  192.168.10.1 (192.168.10.1)  3.644 ms  3.571 ms  3.565 ms
s2@s2:/etc/netplan$
```

Question 4.b (5 marks all together)
Paste screen captures of 1) the routing table of R1, and 2) the list of OSPF neighbors.

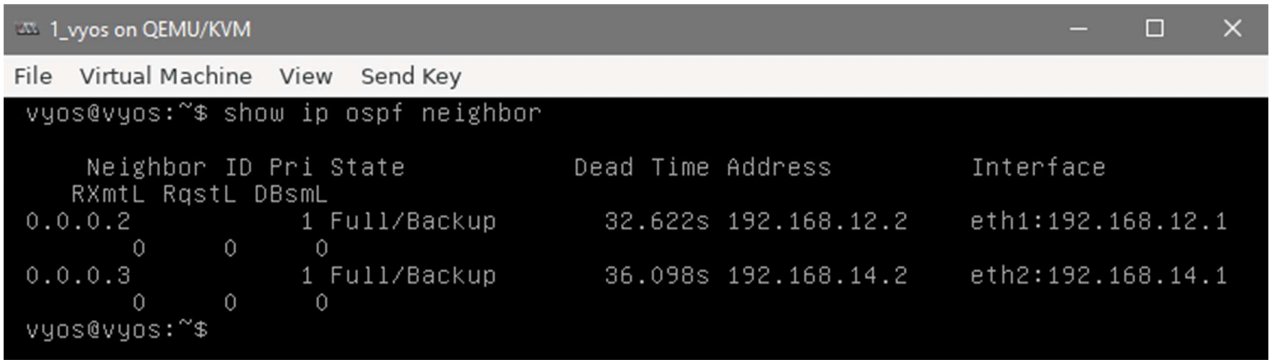
1) Routing table of R1



```
vyos@vyos:~$ ip r s
127.0.0.0/8 dev lo  proto kernel  scope link  src 127.0.0.1
192.168.10.0/24 dev eth0  proto kernel  scope link  src 192.168.10.2
192.168.11.0/24 via 192.168.14.2 dev eth2  proto zebra  metric 20
192.168.12.0/24 dev eth1  proto kernel  scope link  src 192.168.12.1
192.168.13.0/24  proto zebra  metric 20
        nexthop via 192.168.12.2  dev eth1 weight 1
        nexthop via 192.168.14.2  dev eth2 weight 1
192.168.14.0/24 dev eth2  proto kernel  scope link  src 192.168.14.1
vyos@vyos:~$ _
```

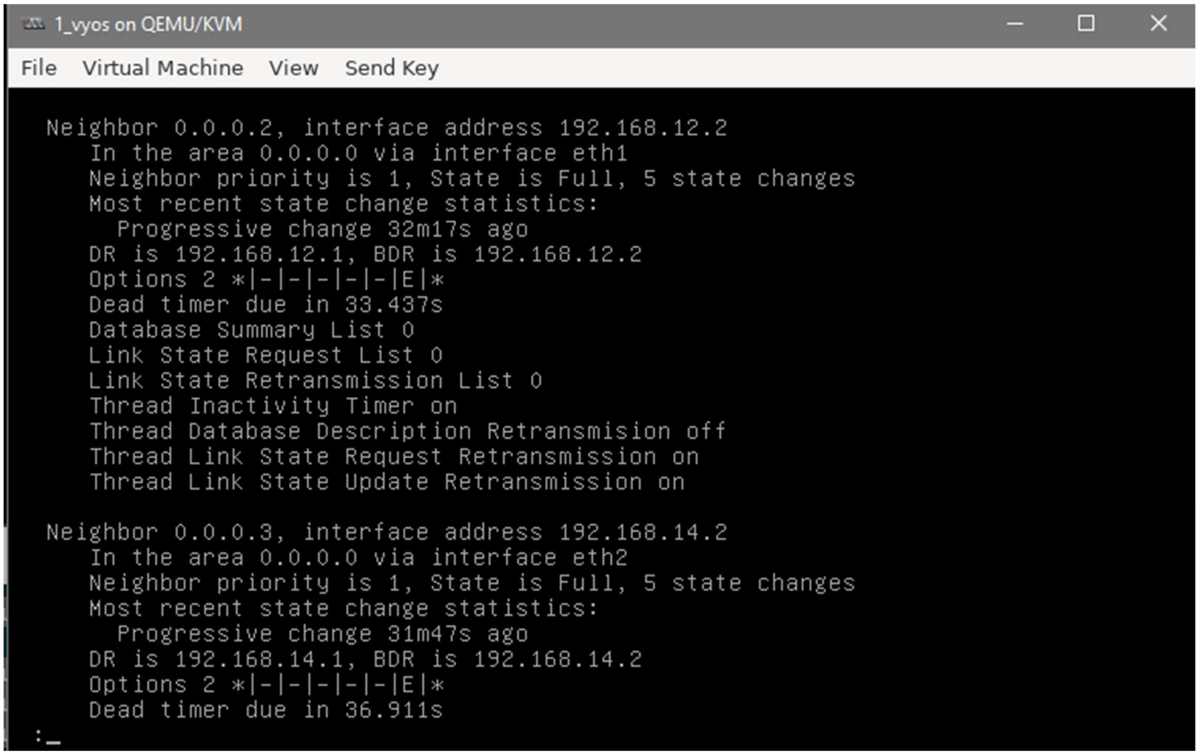
2) OSPF Neighbors

- Summary



```
vyos@vyos:~$ show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
0.0.0.2           1 Full/Backup       32.622s 192.168.12.2    eth1:192.168.12.1
      0     0     0
0.0.0.3           1 Full/Backup       36.098s 192.168.14.2    eth2:192.168.14.1
      0     0     0
vyos@vyos:~$
```

- Complete info



```
   Neighbor 0.0.0.2, interface address 192.168.12.2
      In the area 0.0.0.0 via interface eth1
      Neighbor priority is 1, State is Full, 5 state changes
      Most recent state change statistics:
        Progressive change 32m17s ago
      DR is 192.168.12.1, BDR is 192.168.12.2
      Options 2 *|-|-|-|-|-|E|*
      Dead timer due in 33.437s
      Database Summary List 0
      Link State Request List 0
      Link State Retransmission List 0
      Thread Inactivity Timer on
      Thread Database Description Retransmision off
      Thread Link State Request Retransmission on
      Thread Link State Update Retransmission on

   Neighbor 0.0.0.3, interface address 192.168.14.2
      In the area 0.0.0.0 via interface eth2
      Neighbor priority is 1, State is Full, 5 state changes
      Most recent state change statistics:
        Progressive change 31m47s ago
      DR is 192.168.14.1, BDR is 192.168.14.2
      Options 2 *|-|-|-|-|-|E|*
      Dead timer due in 36.911s
 :_
```

Question 4.c (5 marks all together)

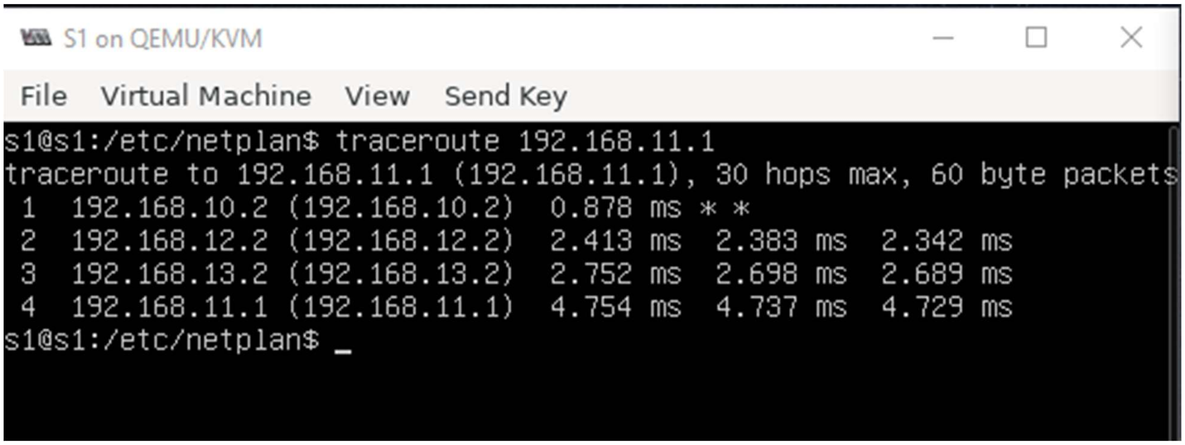
Revise your OSPF configuration of each router so that the traffic between H1 and H2 always goes through the path {H1 <---> R1 <---> R2 <---> R3 <---> H2}.  1) Explain what kind of revision you made on which router.  Also, 2) paste the screen capture of traceroute between H1 and H2 to show that the above mentioned path is successfully implemented.

1) To change the link-weights of the path between R1←→ R2 ←→ R3, I explicitly lowered the costs on those interfaces using the following commands for the each of the routers.
   Note that the default cost set by vyos was found to be 10 so I used the minimum possible cost (1) as with the following commands:

   1_vyos(R1):
      *set interfaces ethernet eth1 ip ospf cost 1*
   2_vyos(R2):
      *set interfaces ethernet eth1 ip ospf cost 1*
      *set interfaces ethernet eth0 ip ospf cost 1*
   3_vyos(R2):
      *set interfaces ethernet eth1 ip ospf cost 1*

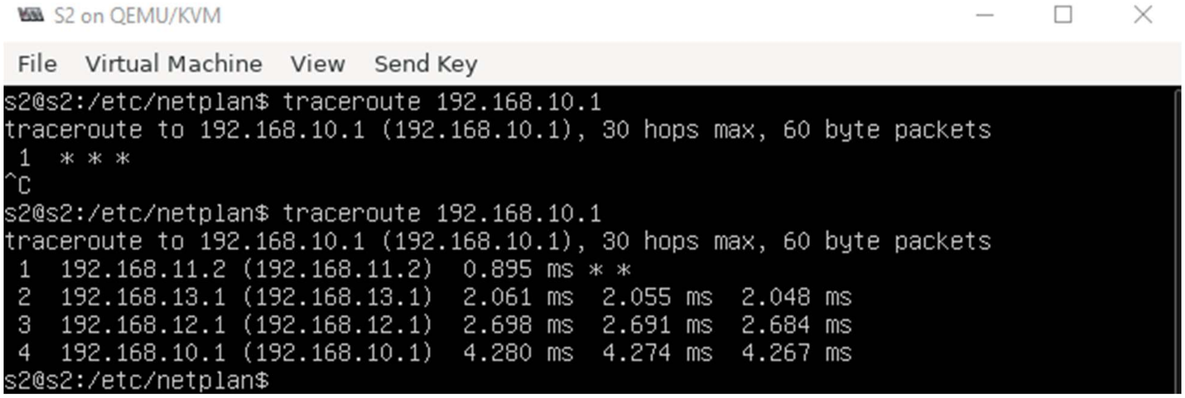
   Now, The routers use the length 2 path via R2 rather than the other length 10 path.

2) Traceroute from H1 to H2:



```
s1@s1:/etc/netplan$ traceroute 192.168.11.1
traceroute to 192.168.11.1 (192.168.11.1), 30 hops max, 60 byte packets
 1  192.168.10.2 (192.168.10.2)  0.878 ms * *
 2  192.168.12.2 (192.168.12.2)  2.413 ms  2.383 ms  2.342 ms
 3  192.168.13.2 (192.168.13.2)  2.752 ms  2.698 ms  2.689 ms
 4  192.168.11.1 (192.168.11.1)  4.754 ms  4.737 ms  4.729 ms
s1@s1:/etc/netplan$ _
```

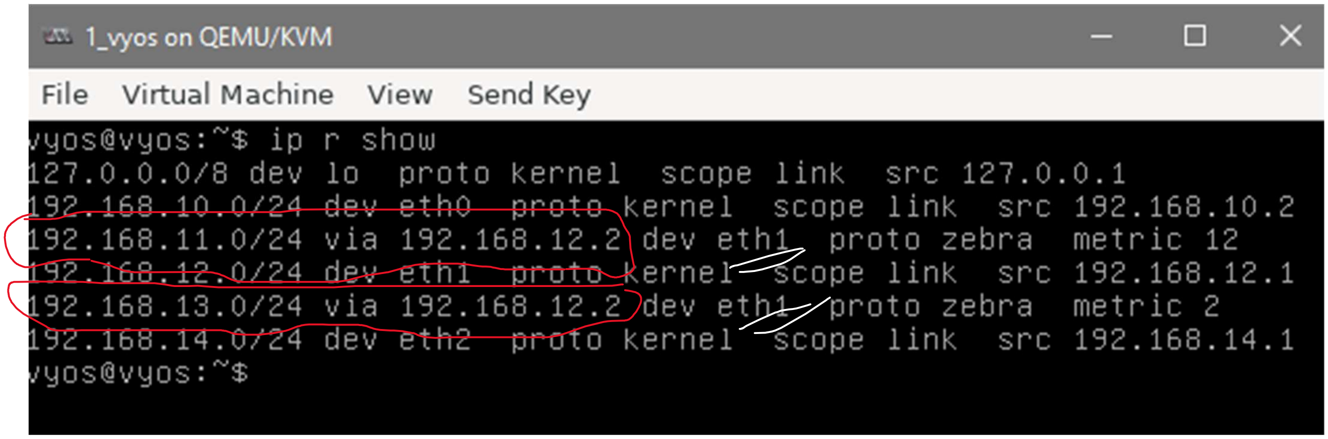
3) Traceroute from H2 to H1:



```
s2@s2:/etc/netplan$ traceroute 192.168.10.1
traceroute to 192.168.10.1 (192.168.10.1), 30 hops max, 60 byte packets
 1  * * *
^C
s2@s2:/etc/netplan$ traceroute 192.168.10.1
traceroute to 192.168.10.1 (192.168.10.1), 30 hops max, 60 byte packets
 1  192.168.11.2 (192.168.11.2)  0.895 ms * *
 2  192.168.13.1 (192.168.13.1)  2.061 ms  2.055 ms  2.048 ms
 3  192.168.12.1 (192.168.12.1)  2.698 ms  2.691 ms  2.684 ms
 4  192.168.10.1 (192.168.10.1)  4.280 ms  4.274 ms  4.267 ms
s2@s2:/etc/netplan$
```

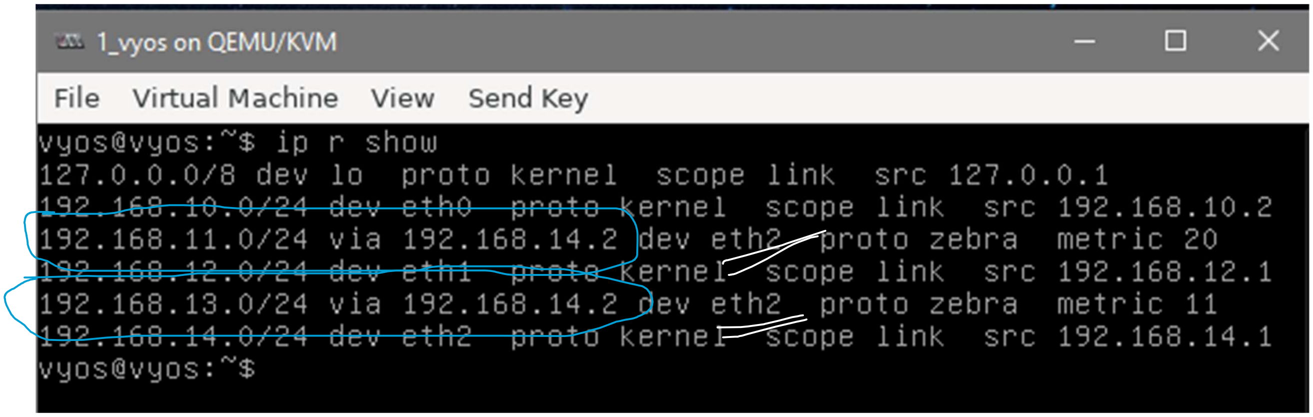
Question 4.d (5 marks)
Shutdown R2, and explain what happens to the routing table of R1 after R2 becomes down.

Routing table of R1 :
**BEFORE** SHUTDOWN of R2



```
vyos@vyos:~$ ip r show
127.0.0.0/8 dev lo  proto kernel  scope link  src 127.0.0.1
192.168.10.0/24 dev eth0  proto kernel  scope link  src 192.168.10.2
192.168.11.0/24 via 192.168.12.2 dev eth1  proto zebra  metric 12
192.168.12.0/24 dev eth1  proto kernel  scope link  src 192.168.12.1
192.168.13.0/24 via 192.168.12.2 dev eth1  proto zebra  metric 2
192.168.14.0/24 dev eth2  proto kernel  scope link  src 192.168.14.1
vyos@vyos:~$
```

**AFTER** SHUTDOWN of R2

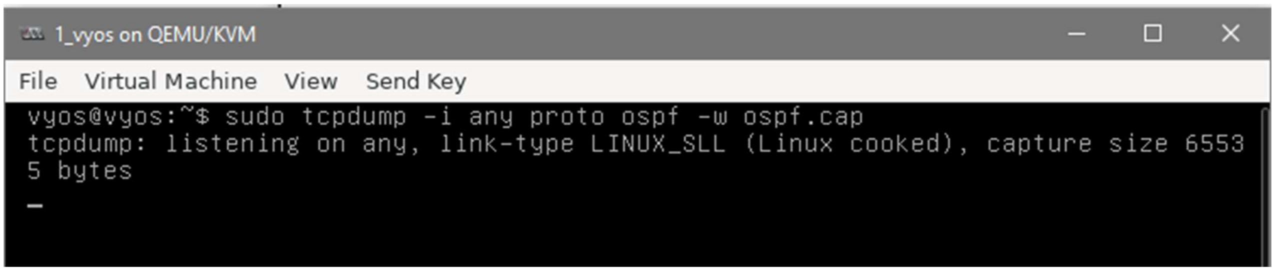


```
vyos@vyos:~$ ip r show
127.0.0.0/8 dev lo  proto kernel  scope link  src 127.0.0.1
192.168.10.0/24 dev eth0  proto kernel  scope link  src 192.168.10.2
192.168.11.0/24 via 192.168.14.2 dev eth2  proto zebra  metric 20
192.168.12.0/24 dev eth1  proto kernel  scope link  src 192.168.12.1
192.168.13.0/24 via 192.168.14.2 dev eth2  proto zebra  metric 11
192.168.14.0/24 dev eth2  proto kernel  scope link  src 192.168.14.1
vyos@vyos:~$
```

Notice the changes highlighted in the BEFORE AND AFTER screenshots. As R2 was shutdown, R1 now has to route directly to R3 which can be seen on the NIC Ip addrs that are paired with the highlighted subnets. These subnets correspond to LBr2 and LBrB which were previously being accessed via R3. To further clarify that argument note the changes in the preferred changes corresponding from eth1(connects to R2) to eth2(connect R3).

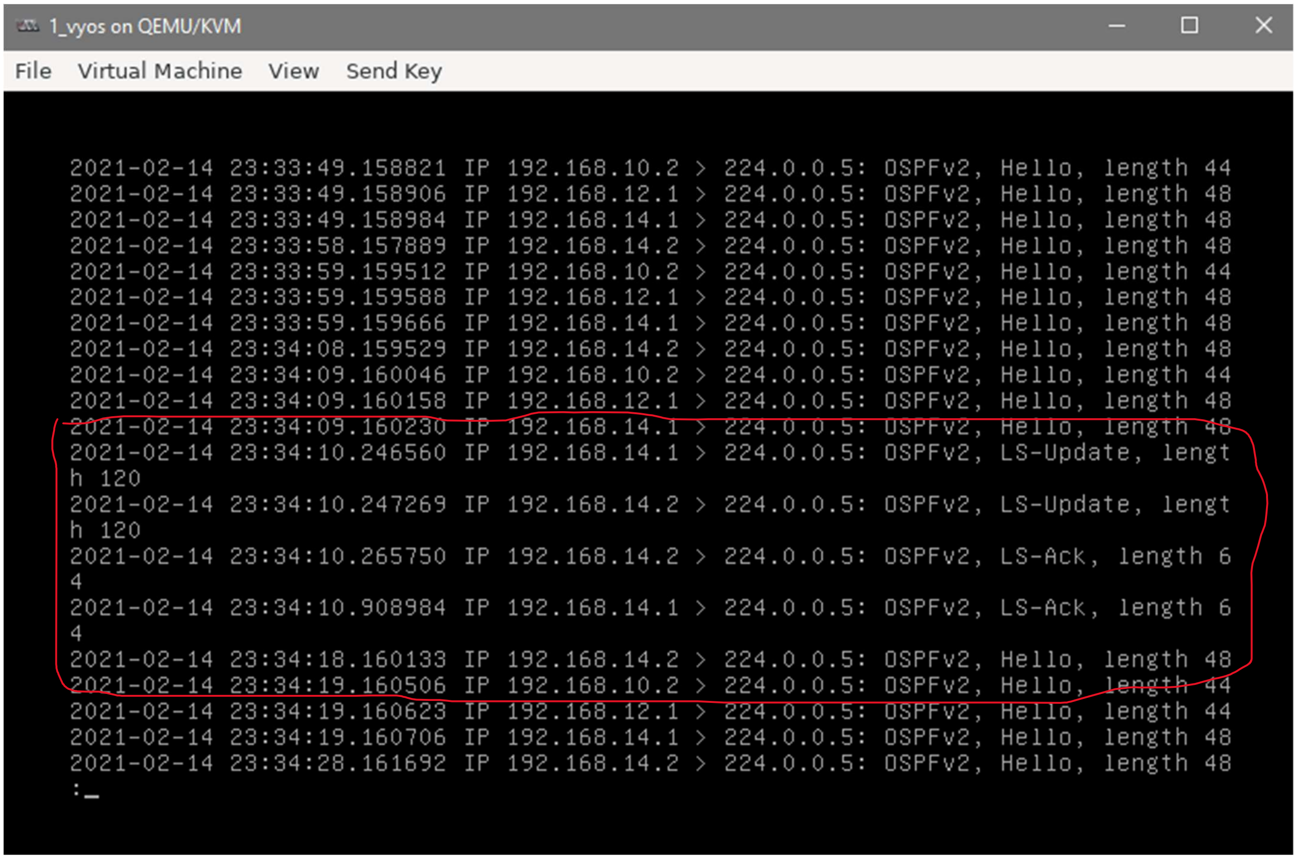
Question 4.e (5 marks)
Observing the packet capture data at R1, explain what kind of OSPF messages flew from/to R1 after R2 becomes down.

Running the following command on R1 before shutting down R2



```
vyos@vyos:~$ sudo tcpdump -i any proto ospf -w ospf.cap
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked), capture size 6553
5 bytes
_
```

And observing the capture logs:
Using sudo tcpdump -ttttnnr ospf.cap



```
2021-02-14 23:33:49.158821 IP 192.168.10.2 > 224.0.0.5: OSPFv2, Hello, length 44
2021-02-14 23:33:49.158906 IP 192.168.12.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:33:49.158984 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:33:58.157889 IP 192.168.14.2 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:33:59.159512 IP 192.168.10.2 > 224.0.0.5: OSPFv2, Hello, length 44
2021-02-14 23:33:59.159588 IP 192.168.12.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:33:59.159666 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:34:08.159529 IP 192.168.14.2 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:34:09.160046 IP 192.168.10.2 > 224.0.0.5: OSPFv2, Hello, length 44
2021-02-14 23:34:09.160158 IP 192.168.12.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:34:09.160230 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:34:10.246560 IP 192.168.14.1 > 224.0.0.5: OSPFv2, LS-Update, lengt
h 120
2021-02-14 23:34:10.247269 IP 192.168.14.2 > 224.0.0.5: OSPFv2, LS-Update, lengt
h 120
2021-02-14 23:34:10.265750 IP 192.168.14.2 > 224.0.0.5: OSPFv2, LS-Ack, length 6
4
2021-02-14 23:34:10.908984 IP 192.168.14.1 > 224.0.0.5: OSPFv2, LS-Ack, length 6
4
2021-02-14 23:34:18.160133 IP 192.168.14.2 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:34:19.160506 IP 192.168.10.2 > 224.0.0.5: OSPFv2, Hello, length 44
2021-02-14 23:34:19.160623 IP 192.168.12.1 > 224.0.0.5: OSPFv2, Hello, length 44
2021-02-14 23:34:19.160706 IP 192.168.14.1 > 224.0.0.5: OSPFv2, Hello, length 48
2021-02-14 23:34:28.161692 IP 192.168.14.2 > 224.0.0.5: OSPFv2, Hello, length 48
:_
```

Ignoring the Hello before and after the shutdown (which are just maintenance messages(they keep of verifying the existence of the links))

Note the multicast messages from .14.1 and .14.2 : they are both corresponding to the NICs on LBrC are LS-Update and LS-Ack messages (a pair for both of them).

These are link-state update and acknowledgement messages which update the shortest path database on all the routers after R2 is down.

Done!!