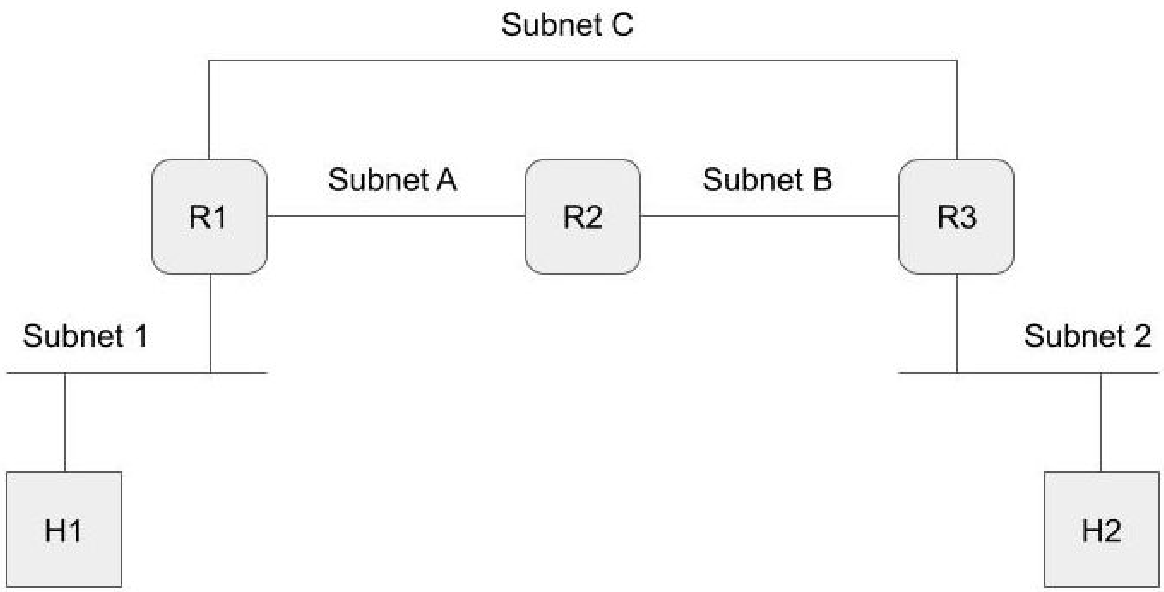
CS3543 Lab Assignment 1

Member 1: Akash Tadwai (ES18BTECH11019)
Member 2: Vinta Reethu (ES18BTECH11028)
Member 3: M Sai Anuraag(CS18BTECH11027)
Member 4: khalid shareef (CS18BTECH11029)

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.

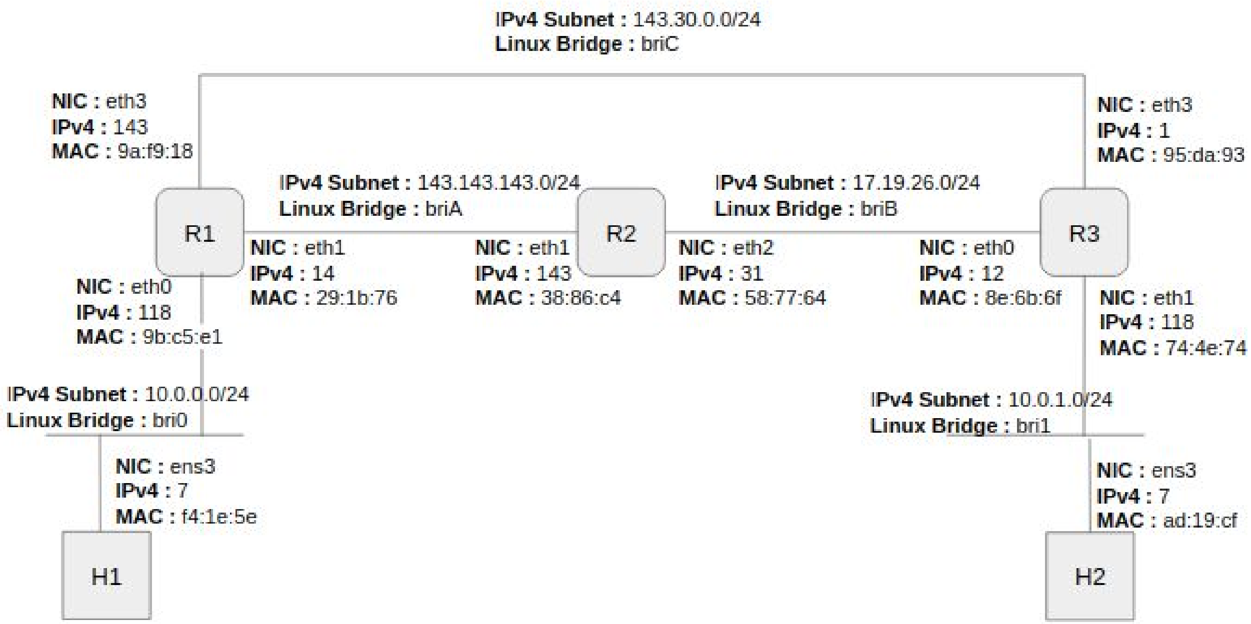
	Linux Bridge	H1	H2	R1	R2	R3
Subnet 1	bri0	**ens3 :** 10.0.0.7 **MAC :** 52:54:00:f4:1e:5e	N/A	**eth0 :** 10.0.0.118/24 **MAC :** 52:54:00:9b:c5:e1	N/A	N/A
Subnet 2	bri1	N/A	**ens3 :** 10.0.1.7/24 **MAC :** 52:54:00:c5:bd:01	N/A	N/A	**eth1 :** 10.0.1.118/24 **MAC:** 52:54:00:74:4e:74
Subnet A	briA	N/A	N/A	**eth1:** 143.143.143.14/24 **MAC:** 52:54:00:29:1b:76	**eth1 :** 143.143.143.143/24 **MAC:** 52:54:00:38:86:c4	N/A

Subnet B	briB	N/A	N/A	N/A	**eth 2 :** 17.19.26.31/24 **MAC :** 52:54:00:58:77:64	**eth 0 :** 17.19.26.12/24 **MAC :** 52:54:00:8e:6b:6f
Subnet C	briC	N/A	N/A	**eth 3 :** 143.30.0.143/24 **MAC :** 52:54:00:9a:f9:18	N/A	**eth3 :** 143.30.0.1/24 **MAC :** 52:54:00:95:da:93

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.

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.



NOTE: We cloned H1 to get H2, hence the user name in the both of these will be host1@akash.

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.
Ping H1 to H2 :

```
host1@akash:~$ ping 10.0.1.7 -c 5
PING 10.0.1.7 (10.0.1.7) 56(84) bytes of data.
64 bytes from 10.0.1.7: icmp_seq=1 ttl=61 time=2.19 ms
64 bytes from 10.0.1.7: icmp_seq=2 ttl=61 time=2.48 ms
64 bytes from 10.0.1.7: icmp_seq=3 ttl=61 time=2.17 ms
64 bytes from 10.0.1.7: icmp_seq=4 ttl=61 time=2.18 ms
64 bytes from 10.0.1.7: icmp_seq=5 ttl=61 time=2.47 ms

--- 10.0.1.7 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.177/2.304/2.489/0.152 ms
host1@akash:~$
```

Ping H2 to H1:

```
host1@akash:~$ ping 10.0.0.7 -c 5
PING 10.0.0.7 (10.0.0.7) 56(84) bytes of data.
64 bytes from 10.0.0.7: icmp_seq=1 ttl=62 time=21.4 ms
64 bytes from 10.0.0.7: icmp_seq=2 ttl=62 time=2.74 ms
64 bytes from 10.0.0.7: icmp_seq=3 ttl=62 time=2.60 ms
64 bytes from 10.0.0.7: icmp_seq=4 ttl=62 time=38.4 ms
64 bytes from 10.0.0.7: icmp_seq=5 ttl=62 time=2.56 ms

--- 10.0.0.7 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 2.564/13.550/38.432/14.409 ms
host1@akash:~$
```

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

```
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>* 10.0.0.0/24 is directly connected, eth0
S>* 10.0.1.0/24 [1/0] via 143.30.0.1, eth3
C>* 127.0.0.0/8 is directly connected, lo
C>* 143.30.0.0/24 is directly connected, eth3
C>* 143.143.143.0/24 is directly connected, eth1
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.

```
host1@akash:~$ traceroute 10.0.1.7
traceroute to 10.0.1.7 (10.0.1.7), 30 hops max, 60 byte packets
 1  10.0.0.118 (10.0.0.118)  0.769 ms  0.660 ms  0.557 ms
 2  143.30.0.1 (143.30.0.1)  67.041 ms  117.397 ms  117.381 ms
 3  10.0.1.7 (10.0.1.7)  117.355 ms  117.316 ms  117.287 ms
host1@akash:~$
```

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.

```
host1@akash:~$ sudo traceroute 10.0.0.7 -w 10 -I
traceroute to 10.0.0.7 (10.0.0.7), 30 hops max, 60 byte packets
 1  10.0.1.118 (10.0.1.118)  0.904 ms  0.910 ms *
 2  17.19.26.31 (17.19.26.31)  1.599 ms  1.621 ms  1.627 ms
 3  143.143.143.14 (143.143.143.14)  2.261 ms  2.275 ms  2.272 ms
 4  10.0.0.7 (10.0.0.7)  2.824 ms  2.839 ms  2.828 ms
host1@akash:~$
```

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.

Explanation :

From the tcpdump on R1 by using the “-e” flag we can check the ethernet header. We saw that The destination address was of R3’s eth3. This is because the routers follow NAT protocol and the translation table is updated accordingly.

NIC Name : eth3
Destination address : 52:54:00:95:da:93
Node : R3

```
vyos@vyos:~$ sudo tcpdump -i eth3 -e -p icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth3, link-type EN10MB (Ethernet), capture size 65535 bytes
15:56:20.342034 52:54:00:9a:f9:18 (oui Unknown) > 52:54:00:95:da:93 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 10.0.0.7 > 10.0.1.7: ICMP echo request, i
d 1976, seq 87, length 64
15:56:20.343445 52:54:00:95:da:93 (oui Unknown) > 52:54:00:9a:f9:18 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 10.0.1.7 > 10.0.0.7: ICMP echo reply, id
1976, seq 87, length 64
15:56:21.343861 52:54:00:9a:f9:18 (oui Unknown) > 52:54:00:95:da:93 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 10.0.0.7 > 10.0.1.7: ICMP echo request, i
d 1976, seq 88, length 64
15:56:21.344584 52:54:00:95:da:93 (oui Unknown) > 52:54:00:9a:f9:18 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 10.0.1.7 > 10.0.0.7: ICMP echo reply, id
1976, seq 88, length 64
15:56:22.345645 52:54:00:9a:f9:18 (oui Unknown) > 52:54:00:95:da:93 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 10.0.0.7 > 10.0.1.7: ICMP echo request, i
d 1976, seq 89, length 64
15:56:22.346964 52:54:00:95:da:93 (oui Unknown) > 52:54:00:9a:f9:18 (oui Unknown
), ethertype IPv4 (0x0800), length 98: 10.0.1.7 > 10.0.0.7: ICMP echo reply, id
```

#
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.

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.

Explanation : From the traceroute to H2 from H1, we can see that the path taken by the packet is H1->R1->R3->H2 as it is the shortest path.

Traceroute from H1 to H2 :

```
host1@akash:~$ traceroute 10.0.1.7
traceroute to 10.0.1.7 (10.0.1.7), 30 hops max, 60 byte packets
 1  10.0.0.118 (10.0.0.118)  0.758 ms  0.727 ms  0.642 ms
 2  143.30.0.1 (143.30.0.1)  1.502 ms  1.449 ms  1.384 ms
 3  10.0.1.7 (10.0.1.7)  3.195 ms  3.069 ms  2.892 ms
host1@akash:~$ _
```

Explanation : From the traceroute of H1 from H2 we can see that the path taken by H2 is H2->R3->R1->H1 as it is the shortest path.

Traceroute from H2 to H1 :

```
host1@akash:~$ traceroute 10.0.0.7
traceroute to 10.0.0.7 (10.0.0.7), 30 hops max, 60 byte packets
 1  10.0.1.118 (10.0.1.118)  0.401 ms * *
 2  143.30.0.143 (143.30.0.143)  20.385 ms  20.381 ms  20.355 ms
 3  10.0.0.7 (10.0.0.7)  20.571 ms  20.573 ms  20.533 ms
host1@akash:~$ _
```

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

Routing table of R1 :

```
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>* 10.0.0.0/24 is directly connected, eth0
O>* 10.0.1.0/24 [110/20] via 143.30.0.1, eth3, 00:05:37
O>* 17.19.26.0/24 [110/20] via 143.30.0.1, eth3, 00:05:38
  *                       via 143.143.143.143, eth1, 00:05:38
C>* 127.0.0.0/8 is directly connected, lo
O   143.30.0.0/24 [110/10] is directly connected, eth3, 00:10:53
C>* 143.30.0.0/24 is directly connected, eth3
O   143.143.143.0/24 [110/10] is directly connected, eth1, 00:10:53
C>* 143.143.143.0/24 is directly connected, eth1
vyos@vyos:~$
```

List of OSPF neighbours of R1 :

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

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
127.0.0.2         1 Full/DR           34.956s 143.143.143.143 eth1:143.143.143.1
4      0     0     0
127.0.0.3         1 Full/DR           31.972s 143.30.0.1      eth3:143.30.0.143
       0     0     0
vyos@vyos:~$
```

List of OSPF neighbours of R2 :

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

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
127.0.0.1         1 Full/Backup       37.973s 143.143.143.14  eth1:143.143.143.1
43     0     0     0
127.0.0.3         1 Full/DR           36.460s 17.19.26.12     eth2:17.19.26.31
       0     0     0
vyos@vyos:~$ _
```

List of OSPF neighbours of R3 :

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

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
127.0.0.2         1 Full/Backup       34.625s 17.19.26.31     eth0:17.19.26.12
       0     0     0
127.0.0.1         1 Full/Backup       33.163s 143.30.0.143    eth3:143.30.0.1
       0     0     0
vyos@vyos:~$
```

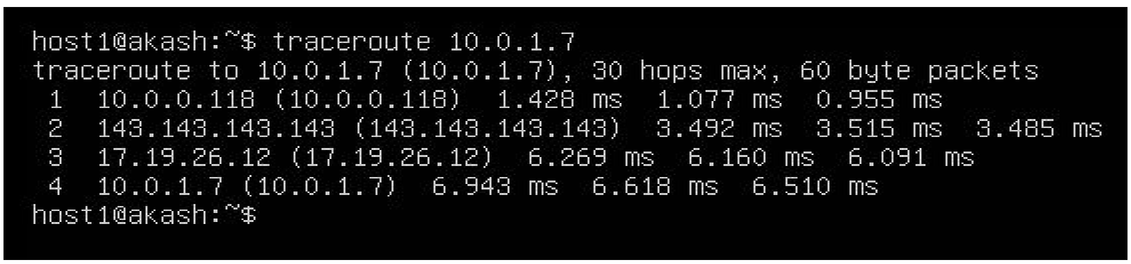
Question 4.c (5 marks all together)
Revise your OSPF configuration of each router so that the traffic between H1 and H2 always goes through the path {H1 <---> R1 <---> R2 <---> R3 <---> H2}. 1) Explain what kind of revision you made on which router. Also, 2) paste the screen capture of traceroute between H1 and H2 to show that the above mentioned path is successfully implemented.

Explanation :

As shown in the screenshots, the path is now as required(not through Subnet C). This is achieved by increasing the costs of respective interfaces at both the routers. They are set to the maximum possible cost(65,535).

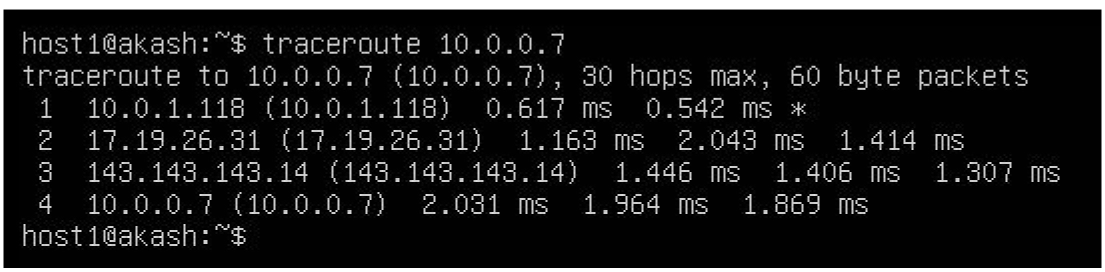
Interfaces whose costs are changed are eth3 of R1 and eth3 of R3. Now, when OSPF is run to find the least cost path, the R1<--->R3 path is always ignored as its cost will always be greater than other interfaces'(since other interfaces' default cost is just 10).

Traceroute from H1 to H2 :



```
host1@akash:~$ traceroute 10.0.1.7
traceroute to 10.0.1.7 (10.0.1.7), 30 hops max, 60 byte packets
 1  10.0.0.118 (10.0.0.118)  1.428 ms  1.077 ms  0.955 ms
 2  143.143.143.143 (143.143.143.143)  3.492 ms  3.515 ms  3.485 ms
 3  17.19.26.12 (17.19.26.12)  6.269 ms  6.160 ms  6.091 ms
 4  10.0.1.7 (10.0.1.7)  6.943 ms  6.618 ms  6.510 ms
host1@akash:~$
```

Traceroute from H2 to H1 :



```
host1@akash:~$ traceroute 10.0.0.7
traceroute to 10.0.0.7 (10.0.0.7), 30 hops max, 60 byte packets
 1  10.0.1.118 (10.0.1.118)  0.617 ms  0.542 ms *
 2  17.19.26.31 (17.19.26.31)  1.163 ms  2.043 ms  1.414 ms
 3  143.143.143.14 (143.143.143.14)  1.446 ms  1.406 ms  1.307 ms
 4  10.0.0.7 (10.0.0.7)  2.031 ms  1.964 ms  1.869 ms
host1@akash:~$
```

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

Explanation : Since R2 is down there will be no acknowledgment messages from R2, so R1 will think that there is no path there. Hence it will transmit all the packets it recieve to R3.

Below is a screenshot showing the new ospf routes in R1(i.e after R2 is down).

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

O>* 10.0.1.0/24 [110/20] via 143.30.0.1, eth3, 00:11:34
O>* 17.19.26.0/24 [110/20] via 143.30.0.1, eth3, 00:00:55
O   143.30.0.0/24 [110/10] is directly connected, eth3, 00:16:50
O   143.143.143.0/24 [110/10] is directly connected, eth1, 00:16:50
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>* 10.0.0.0/24 is directly connected, eth0
O>* 10.0.1.0/24 [110/20] via 143.30.0.1, eth3, 00:11:41
O>* 17.19.26.0/24 [110/20] via 143.30.0.1, eth3, 00:01:02
C>* 127.0.0.0/8 is directly connected, lo
O   143.30.0.0/24 [110/10] is directly connected, eth3, 00:16:57
C>* 143.30.0.0/24 is directly connected, eth3
O   143.143.143.0/24 [110/10] is directly connected, eth1, 00:16:57
C>* 143.143.143.0/24 is directly connected, eth1
vyos@vyos:~$ _
```

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.

Explanation :

In OSPF, a ‘Hello’ packet is sent to the multicast address 224.0.0.5 every 10 seconds. After receiving a ‘Hello’ packet an OSPF router sends a ‘Hello’ packet back. If a router did not receive any ‘Hello’ packet from a link for 40 secs, that link is considered to be dead.

In our case, R1 sends ‘Hello’ packets on ‘Subnet A’ link, but does not receive any ‘Hello’ packets back. This is because R2 is down. So, R1 considers that link(Subnet A) to be dead. When R1 checks if there is a connection between R1 and R3, we can see that ‘Hello’ packets sent from both the routers are captured. This is because R3 is present at that link and is replying back. This can be seen from the tcpdumps shown below.

Tcpdump of R1 at interface connected to R3 (After R2 is down) :

```
vyos@vyos:~$ sudo tcpdump -i eth3 -c 10
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth3, link-type EN10MB (Ethernet), capture size 65535 bytes
15:35:04.206664 IP 143.30.0.1 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:12.689513 IP 143.30.0.143 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:14.207058 IP 143.30.0.1 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:22.689765 IP 143.30.0.143 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:24.207423 IP 143.30.0.1 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:32.690010 IP 143.30.0.143 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:34.208014 IP 143.30.0.1 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:42.690302 IP 143.30.0.143 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:44.208680 IP 143.30.0.1 > 224.0.0.5: OSPFv2, Hello, length 48
15:35:52.690630 IP 143.30.0.143 > 224.0.0.5: OSPFv2, Hello, length 48
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

Tcpdump of R1 at interface connected to R2 (After R2 is down) :

```
vyos@vyos:~$ sudo tcpdump -i eth1 -c 10
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 65535 bytes
15:37:12.818327 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:37:22.818626 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:37:32.818921 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:37:42.819220 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:37:52.819323 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:38:02.819678 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:38:12.820111 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:38:22.820416 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:38:32.820470 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
15:38:42.820623 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
10 packets captured
10 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _
```

Below screenshot is the tcpdump of R1 when R2 was up and running. We can see that both R1and R2 are sending 'Hello' packets to 224.0.0.5 and the link is considered to be alive. When R2 is shutting down, it sends an IGMP 'Leave group' message indicating that multicast transmissions are no longer required at that address. This screenshot shows messages when R2 was up(first 4 packets), when it was shutting down(IGMP messages) and after its down(packets after IGMP messages).

Tcpdump in R1 during shut down of R2 :

```
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
17:24:14.776244 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 48
17:24:16.241685 IP 143.143.143.143 > 224.0.0.5: OSPFv2, Hello, length 48
17:24:24.776693 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 48
17:24:26.242281 IP 143.143.143.143 > 224.0.0.5: OSPFv2, Hello, length 48
17:24:34.523363 IP 143.143.143.143 > 224.0.0.22: igmp v3 report, 2 group record(
s)
17:24:34.713537 IP 143.143.143.143 > 224.0.0.22: igmp v3 report, 2 group record(
s)
17:24:34.777173 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 48
17:24:44.777477 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 48
17:24:54.778136 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 48
17:25:04.778734 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 48
17:25:14.778953 IP 143.143.143.14 > 224.0.0.5: OSPFv2, Hello, length 44
^C
11 packets captured
11 packets received by filter
0 packets dropped by kernel
vyos@vyos:~$ _
```

Below is the screenshot of tcpdump in R1 on the interface connected to R3. After R2 is down we observe 2 link state updates and two link state acknowledgment updates sent through multicast address.

Link state updates after shutdown of R2:

```
18:32:24.507279 IP 10.0.1.7 > 10.0.0.7: ICMP echo reply, id 1993, seq 1476, leng
th 64
18:32:24.551545 IP 143.30.0.143 > 224.0.0.5: OSPFv2, LS-Update, length 108
18:32:24.582271 IP 143.30.0.1 > 224.0.0.5: OSPFv2, LS-Update, length 108
18:32:24.643735 IP 143.30.0.143 > 224.0.0.5: OSPFv2, Hello, length 48
18:32:24.963676 IP 143.30.0.143 > 224.0.0.5: OSPFv2, LS-Ack, length 64
18:32:25.209979 IP 143.30.0.1 > 224.0.0.5: OSPFv2, LS-Ack, length 64
18:32:25.506934 IP 10.0.0.7 > 10.0.1.7: ICMP echo request, id 1993, seq 1477, le
ngth 64
18:32:25.508370 IP 10.0.1.7 > 10.0.0.7: ICMP echo reply, id 1993, seq 1477, leng
th 64
18:32:26.508975 IP 10.0.0.7 > 10.0.1.7: ICMP echo request, id 1993, seq 1478, le
ngth 64
18:32:26.510096 IP 10.0.1.7 > 10.0.0.7: ICMP echo reply, id 1993, seq 1478, leng
th 64
18:32:27.510871 IP 10.0.0.7 > 10.0.1.7: ICMP echo request, id 1993, seq 1479, le
ngth 64
18:32:27.512176 IP 10.0.1.7 > 10.0.0.7: ICMP echo reply, id 1993, seq 1479, leng
th 64
^C
112 packets captured
112 packets received by filter
0 packets dropped by kernel
[edit]
vyos@vyos#
```