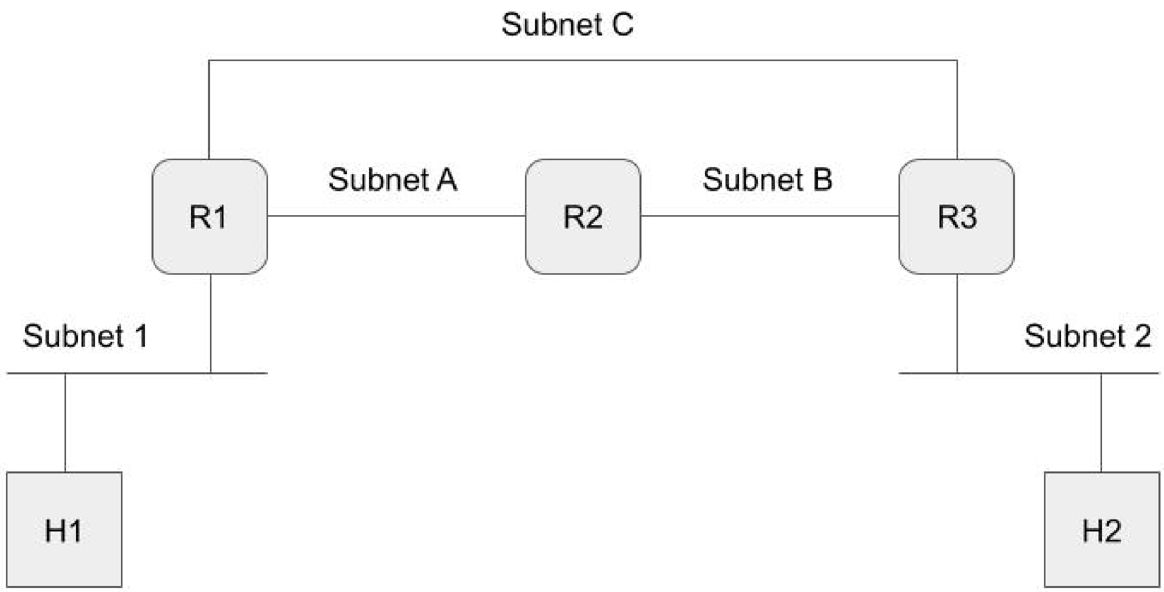
CS3543 Lab Assignment 1

Member 1: Sai Krishna (CS18BTECH11025)
Member 2: Sai Mahesh (CS18BTECH11001)
Member 3: Deepak Reddy (CS18BTECH11016)
Member 4: Sandeep Kumar (CS18BTECH11044)
Member 5: Yaswanth Purli (CS18BTECH11038)

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.

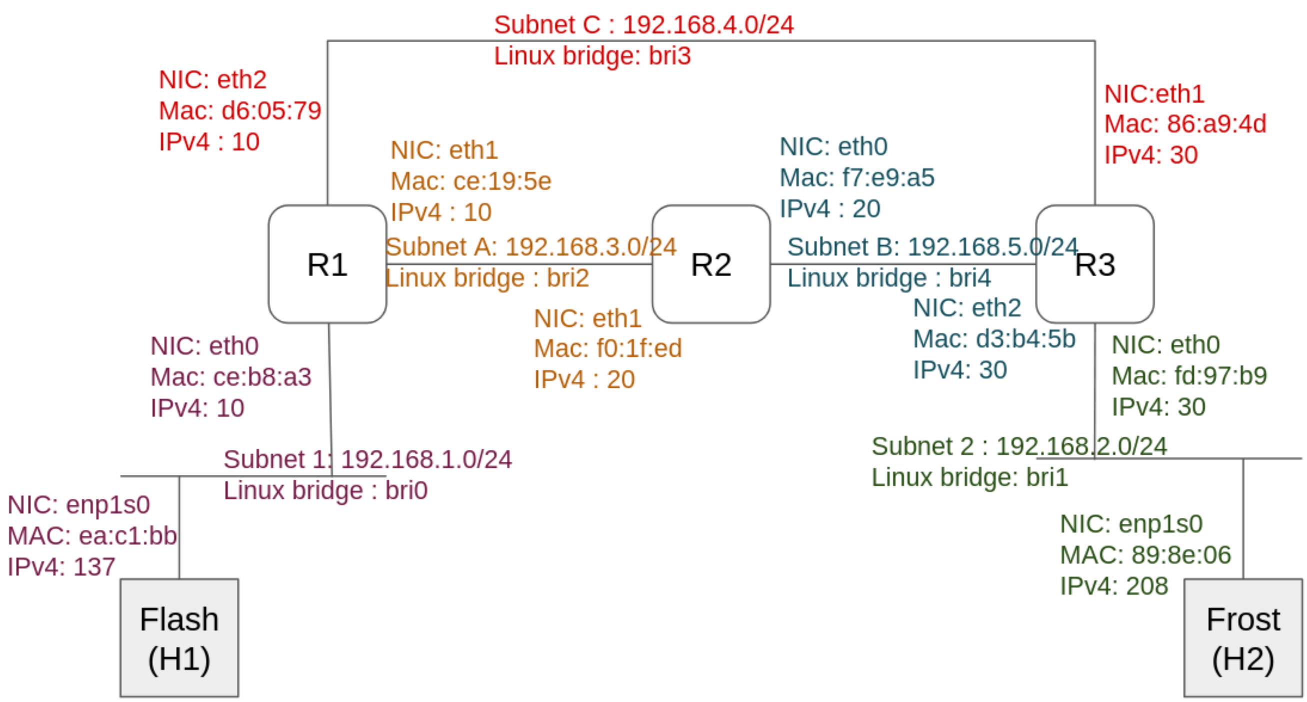


Question 1. (5 marks all together)
Fill the blanks in Table 1 to clarify NIC and IPv4 to belong to Subnets 1 to 4. If there is no corresponding NIC belonging to a subnet, mention “N/A”. All the prefixes must be planned by yourself.

	Linux Bridge	H1	H2	R1	R2	R3
# Example Subnet 1	bri0	eth0: 192.168.0.1/24	N/A	eth0: 192.168.0.2/24	N/A	N/A
Subnet 1	bri0	enp1s0: 192.168.1.137	N/A	eth0: 192.168.1.10	N/A	N/A
Subnet 2	bri1	N/A	enp1s0: 192.168.2.208	N/A	N/A	eth0: 192.168.2.30
Subnet A	bri2	N/A	N/A	eth1: 192.168.3.10	eth1: 192.168.3.20	N/A
Subnet B	bri4	N/A	N/A	N/A	eth0: 192.168.5.20	eth2: 192.168.5.30
Subnet C	bri3	N/A	N/A	eth2: 192.168.4.10	N/A	eth1: 192.168.4.30

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.

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.

```
krish@flash:~$ ping 192.168.2.208 -c 10
PING 192.168.2.208 (192.168.2.208) 56(84) bytes of data.
64 bytes from 192.168.2.208: icmp_seq=1 ttl=61 time=4.73 ms
64 bytes from 192.168.2.208: icmp_seq=2 ttl=61 time=2.69 ms
64 bytes from 192.168.2.208: icmp_seq=3 ttl=61 time=4.14 ms
64 bytes from 192.168.2.208: icmp_seq=4 ttl=61 time=4.39 ms
64 bytes from 192.168.2.208: icmp_seq=5 ttl=61 time=4.61 ms
64 bytes from 192.168.2.208: icmp_seq=6 ttl=61 time=4.41 ms
64 bytes from 192.168.2.208: icmp_seq=7 ttl=61 time=4.61 ms
64 bytes from 192.168.2.208: icmp_seq=8 ttl=61 time=4.49 ms
64 bytes from 192.168.2.208: icmp_seq=9 ttl=61 time=4.88 ms
64 bytes from 192.168.2.208: icmp_seq=10 ttl=61 time=5.24 ms

--- 192.168.2.208 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9090ms
rtt min/avg/max/mdev = 2.692/4.419/5.236/0.642 ms
```

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

```
Vyos_R1 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
C>* 192.168.1.0/24 is directly connected, eth0
S>* 192.168.2.0/24 [1/0] via 192.168.4.30, eth2
C>* 192.168.3.0/24 is directly connected, eth1
C>* 192.168.4.0/24 is directly connected, eth2
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.

```
krish@flash:~$ traceroute 192.168.2.208
traceroute to 192.168.2.208 (192.168.2.208), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.10)  1.402 ms  1.707 ms  1.910 ms
 2  192.168.4.30 (192.168.4.30)  11.697 ms  11.634 ms  11.491 ms
 3  192.168.2.208 (192.168.2.208)  17.281 ms  18.204 ms  19.045 ms
krish@flash:~$
```

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.

```
sai@frost:~$ traceroute 192.168.1.137
traceroute to 192.168.1.137 (192.168.1.137), 30 hops max, 60 byte packets
 1  _gateway (192.168.2.30)  1.526 ms  1.669 ms  1.927 ms
 2  192.168.5.20 (192.168.5.20)  6.042 ms  6.260 ms  6.648 ms
 3  192.168.3.10 (192.168.3.10)  7.151 ms  7.510 ms  7.892 ms
 4  192.168.1.137 (192.168.1.137)  14.557 ms  15.507 ms  15.365 ms
sai@frost:~$ traceroute ironman.com
traceroute to ironman.com (192.168.1.137), 30 hops max, 60 byte packets
 1  _gateway (192.168.2.30)  2.985 ms  3.400 ms  3.344 ms
 2  192.168.5.20 (192.168.5.20)  26.924 ms  27.320 ms  27.604 ms
 3  192.168.3.10 (192.168.3.10)  31.269 ms  31.223 ms  31.175 ms
 4  192.168.1.137 (192.168.1.137)  31.123 ms  31.075 ms  31.025 ms
sai@frost:~$
```

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.

A. When a packet is transmitted from H1 to H2 by R1, it is transmitted using R1 -> R3 subnet (as done wrt instructions given). So the destination MAC address will be the MAC address looked up using the next-hop IP address(i.e., ethernet interface of R3 connected to the R1 = eth1)
   Destination Address in Ethernet Header is
   NIC:eth1
   MAC: 86:a9:4d
   IPv4: 192.168.4.30/24

   Further, at R3, the destination address is regenerated by looking up into the next-hop IP address(i.e., enp1s0 of H2)
   NIC: enp1s0
   MAC: 89:8e:06
   IPv4: 192.168.2.208/24

#
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.

1) H1 to H2:
   The possible paths for a packet to go from H1 -> H2 are:
   <u>Path1:</u> H1 ---> R1 ---> R3 ---> H2
   <u>Path2:</u> H1 ---> R1 ---> R2 ---> R3 ---> H2
   As the ospf cost for path2 is more than the path1(as it is including R2), the ospf selects path1 rather than path2

   H2 to H1:
   <u>Path1:</u> H2 ---> R3 ---> R1 ---> H1
   <u>Path2:</u> H3 ---> R3 ---> R2 ---> R1 ---> H1
   As the ospf cost for path2 is more than the path1(as it is including R2), the ospf selects path1 rather than path2

2) Traceroute from H1 to H2:

```
krish@flash:~$ traceroute 192.168.2.208
traceroute to 192.168.2.208 (192.168.2.208), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.10)  1.140 ms  1.659 ms  2.196 ms
 2  192.168.4.30 (192.168.4.30)  10.106 ms  10.314 ms  10.705 ms
 3  192.168.2.208 (192.168.2.208)  13.421 ms  14.377 ms  14.343 ms
krish@flash:~$
```

   Traceroute from H2 to H1:

```
sai@frost:~$ traceroute ironman.com
traceroute to ironman.com (192.168.1.137), 30 hops max, 60 byte packets
 1  _gateway (192.168.2.30)  4.070 ms  4.210 ms  4.663 ms
 2  192.168.4.10 (192.168.4.10)  10.311 ms  10.390 ms  10.424 ms
 3  192.168.1.137 (192.168.1.137)  15.127 ms  14.822 ms  17.049 ms
sai@frost:~$ traceroute 192.168.1.137
traceroute to 192.168.1.137 (192.168.1.137), 30 hops max, 60 byte packets
 1  _gateway (192.168.2.30)  1.132 ms  1.440 ms  2.247 ms
 2  192.168.4.10 (192.168.4.10)  7.201 ms  7.357 ms  7.564 ms
 3  192.168.1.137 (192.168.1.137)  9.779 ms  9.841 ms  10.135 ms
sai@frost:~$
```

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:~$ show ip ospf route
============ OSPF network routing table ============
N    192.168.1.0/24        [10] area: 0.0.0.0
                           directly attached to eth0
N    192.168.2.0/24        [20] area: 0.0.0.0
                           via 192.168.4.30, eth2
N    192.168.3.0/24        [10] area: 0.0.0.0
                           directly attached to eth1
N    192.168.4.0/24        [10] area: 0.0.0.0
                           directly attached to eth2
N    192.168.5.0/24        [20] area: 0.0.0.0
                           via 192.168.3.20, eth1
                           via 192.168.4.30, eth2

============ OSPF router routing table =============

============ OSPF external routing table ===========

vyos@vyos:~$ _
```

2) List of OSPF neighbours of R1:

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

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
192.168.3.20      1 Full/DR           33.326s 192.168.3.20    eth1:192.168.3.10
       0     0     0
192.168.4.30      1 Full/Backup       32.259s 192.168.4.30    eth2:192.168.4.10
       0     0     0
```

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) Originally the path {H1 <---> R1 <---> R3 <---> H2} would have been taken by the ospf as the cost of path1 {R1 <---> R2 <---> R3} is greater than the cost of path2 {R1 <---> R3}. So if we want to choose path1 instead of path2, we revise the cost of path2 to its maximum so that the ospf will take the least path(i.e., path1). using the following command in both R1 & R3 with their respective ethernet interfaces.

```
vyos@vyos# set interfaces ethernet eth2 ip ospf cost 65535
[edit]
```

2) Traceroute for H1 to H2:

```
krish@flash:~$ traceroute 192.168.2.208
traceroute to 192.168.2.208 (192.168.2.208), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.10)  0.964 ms  1.249 ms  1.402 ms
 2  192.168.3.20 (192.168.3.20)  3.573 ms  3.675 ms  4.986 ms
 3  192.168.5.30 (192.168.5.30)  7.014 ms  6.978 ms  6.992 ms
 4  192.168.2.208 (192.168.2.208)  7.839 ms  7.849 ms  248.783 ms
krish@flash:~$
```

Traceroute from H2 to H1:

```
sai@frost:~$ traceroute 192.168.1.137
traceroute to 192.168.1.137 (192.168.1.137), 30 hops max, 60 byte packets
 1  _gateway (192.168.2.30)  1.314 ms  1.366 ms  1.488 ms
 2  192.168.5.20 (192.168.5.20)  5.890 ms  5.960 ms  8.383 ms
 3  192.168.3.10 (192.168.3.10)  11.149 ms  11.080 ms  11.016 ms
 4  192.168.1.137 (192.168.1.137)  15.847 ms  15.799 ms  15.736 ms
sai@frost:~$ traceroute ironman.com
traceroute to ironman.com (192.168.1.137), 30 hops max, 60 byte packets
 1  _gateway (192.168.2.30)  1.251 ms  1.377 ms  1.612 ms
 2  192.168.5.20 (192.168.5.20)  5.372 ms  6.452 ms  6.372 ms
 3  192.168.3.10 (192.168.3.10)  11.418 ms  11.314 ms  11.642 ms
 4  192.168.1.137 (192.168.1.137)  14.067 ms  14.527 ms  14.716 ms
sai@frost:~$
```

Question 4.d (5 marks)

Shutdown R2, and explain what happens to the routing table of R1 after R2 becomes down.

```
vyos@vyos# run show ip ospf route
============ OSPF network routing table ============
N    192.168.1.0/24        [10] area: 0.0.0.0
                           directly attached to eth0
N    192.168.2.0/24        [65545] area: 0.0.0.0
                           via 192.168.4.30, eth2
N    192.168.3.0/24        [10] area: 0.0.0.0
                           directly attached to eth1
N    192.168.4.0/24        [65535] area: 0.0.0.0
                           directly attached to eth2
N    192.168.5.0/24        [65545] area: 0.0.0.0
                           via 192.168.4.30, eth2

============ OSPF router routing table =============

============ OSPF external routing table ===========
```

```
vyos@vyos# run show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
192.168.4.30      1 Full/Backup       32.501s 192.168.4.30    eth2:192.168.4.10
       0     0     0
[edit]
```

After R2 becomes shut down, the R2 is no longer a neighbour to R1(can be seen in the above image) and therefore the path {R1 -> R2 -> R3} will no longer be seen in the routing table of R1.

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.

Here we are capturing from eth2 interface of R1

```
14:16:24.735034 IP 192.168.4.30 > 224.0.0.5: OSPFv2, Hello, length 48
14:16:24.922424 IP 192.168.4.10 > 224.0.0.5: OSPFv2, Hello, length 48
14:16:29.340957 IP 192.168.4.10 > 224.0.0.5: OSPFv2, LS-Update, length 60
14:16:30.025667 IP 192.168.4.30 > 224.0.0.5: OSPFv2, LS-Ack, length 44
14:16:34.736314 IP 192.168.4.30 > 224.0.0.5: OSPFv2, Hello, length 48
14:16:34.864498 IP 192.168.4.30 > 224.0.0.5: OSPFv2, LS-Update, length 120
14:16:34.878610 IP 192.168.4.10 > 224.0.0.5: OSPFv2, LS-Update, length 120
14:16:34.893749 IP 192.168.4.10 > 224.0.0.5: OSPFv2, LS-Ack, length 64
14:16:34.922719 IP 192.168.4.10 > 224.0.0.5: OSPFv2, Hello, length 48
14:16:34.927982 IP 192.168.4.10 > 192.168.4.30: OSPFv2, LS-Update, length 60
14:16:34.930137 IP 192.168.4.30 > 192.168.4.10: OSPFv2, LS-Ack, length 44
14:16:35.029972 IP 192.168.4.30 > 224.0.0.5: OSPFv2, LS-Ack, length 64
```

Msgs from R1:

- Hello msg is broadcasted
- LS-Update is broadcasted & sent to R3(192.168.4.30)
- LS-Ack is broadcasted as a response of broadcast msg of R3

Msgs to R1:

- LS-Ack : Receives acknowledgement packet as a response of LS-Update packet sent to R3.

Hello, length 48: Hello msg is used to send the presence of R1 to others in network

LS-Update : Link State update msg is sent whenever there is an update in the Linking state of R1 routing table. Broadcasting the update to peers in the network

LS-Ack : Link state Acknowledgment msg is sent as response to LS-Update msg.

Done!!