Venkata Naga Dheeraj Pavuluri
N16788493

A. Why must we ensure that our subnets do not overlap? Discuss one example of something that could go wrong. (10 points)

Subnets shouldn't intersect with each other. As a result, each subnet should have a range of addresses that is distinct from all the others. When a router wants to transmit a packet to an IP address inside that range of overlapping addresses in actual networks, if two subnets overlap, the router may send the message to the incorrect subnet. In network design and setup, it is essential to avoid subnet overlap since it can cause a number of issues and operational difficulties. IP address disputes can occur as a result of overlapping subnets. If two or more devices in distinct subnets share an IP address, it will complicate network communication. For instance, routing issues and unusual activity may arise if two hosts with the IP address 192.168.1.10 are present, one in subnet A and the other in subnet B.

<u>Example:-</u>
Consider a scenario where you have two distant branch offices, each of which utilizes the same IP subnet as an example, 192.168.1.0/24 for its internal network. Imagine that a worker from Branch Office A brings their laptop to Branch Office B and connects it to the local network. The laptop can end up with an IP address that clashes with a device already connected to Branch Office B's network since both offices use the same subnet. In addition to making it more challenging to debug and locate the conflict's origin, this may result in issues with network connectivity.

B. Suppose there is another Router (R5) directly connected to the HUB between R3 and R4. Explain whether or not we would need to reconfigure the IP subnets on R3 and R4 in order to communicate with R5. (10 points)

In order to ensure good communication with R5, it may need to modify the IP subnets on R3 and R4 if another router (R5) is directly linked to the same hub as R3 and R4. Network's routing setup and architecture will determine the specific step that must be taken. Introducing a new router would increase the number of hosts causing us to reconfigure our IP subnet with a mask of /29

Let's assume a scenario:-

- ➔ You should set R5 with an IP address in the same subnet as R3 and R4 if you want R3, R4, and R5 to be in the same IP subnet. In this scenario, you would have to modify R3 and R4 to make enough space to accommodate an additional device (R5) inside the same subnet.
- ➔ This implies that the hub that connects R3, R4, and R5 is a layer-2 hub (a standard

Ethernet switch) and doesn't carry out any routing or subnet segregation. The IP addresses of all connected devices should be in the same subnet range.

In conclusion, your network design objectives will determine whether or not you need to change the IP subnets on R3 and R4. You would need to adjust if you wanted all of your devices to be on the same subnet. Without altering the IP subnets of R3 and R4, routing configurations can be added if different subnets are desired while maintaining routing. Your specific network's requirements and architecture will have an effect on your decision.

C. Run Wireshark on R2 (eth1). Now *ping* R3(eth1) from R2. Identify what type of packet is used in *ping.* Why is R2 unable to reach R3 (eth1)? (10 points)

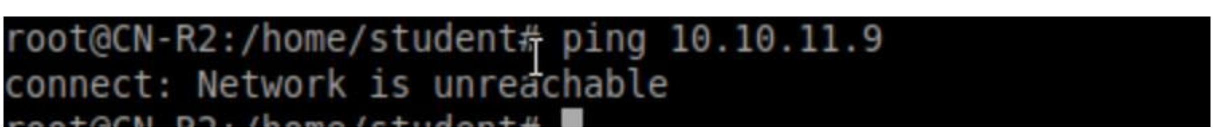
A typical troubleshooting procedure to identify network connectivity issues involves running Wireshark on R2 (eth1) and pinging R3 (eth1) from R2.

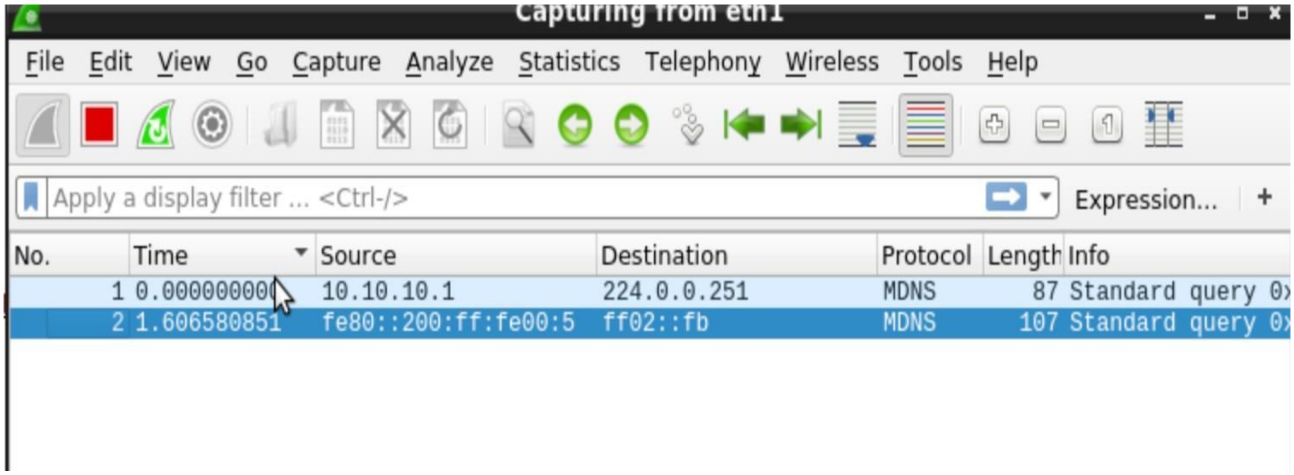
The Internet Control Message Protocol (ICMP) is used while pinging R3 (eth1) from R2. The ping action specifically uses the ICMP Echo Request and Echo Reply messages.

- ➔ R3 receives an ICMP Echo Request packet from R2 (sender).
- ➔ The ICMP Echo Request is processed by R3 (the receiver), who then sends back an ICMP Echo Reply packet.

R2 may not be able to connect to R3 (eth1) for a variety of reasons. Here are a few typical causes which are Network Configuration Issues, Routing Issues, IP Address Conflicts, Firewall or Security Rules.

- ➔ To make sure they are in the same subnet, check the IP setup on R2 (eth1) and R3 (eth1). You won't have effective communication if they aren't.
- ➔ Make sure that R2's routing table contains the appropriate route to reach R3's (eth1) subnet. R2 will be unable to send packets to R3 if there is no route.
- ➔ Make sure that R2 (eth1) and R3 (eth1) have separate IP addresses that don't clash with anything else on the network.
- ➔ Verify whether R2 or R3 have any firewall rules or security policies that could be preventing ICMP traffic (ping) from passing through. It is possible to set up firewalls to deny incoming ICMP queries.





Since R2 and R3 are on different subnets there is no hub to connect them. So, no packets were captured on wireshark, and R3 is not reachable from R2.

D. Briefly describe how Wireshark results compare when you *ping* R3 (eth0) from R2 (eth1). (5 points)

The results of pinging R3 (eth0) from R2 (eth1) and using Wireshark to record the network traffic usually reveal the exchange of ICMP packets (Echo Request and Echo Reply) between the two devices. Up on running Wireshark on R2 (eth1) and pinging R3 (eth0), we notice some, ping echo messages use ICMP protocol to exchange requests and replys. These include some ARP messages.

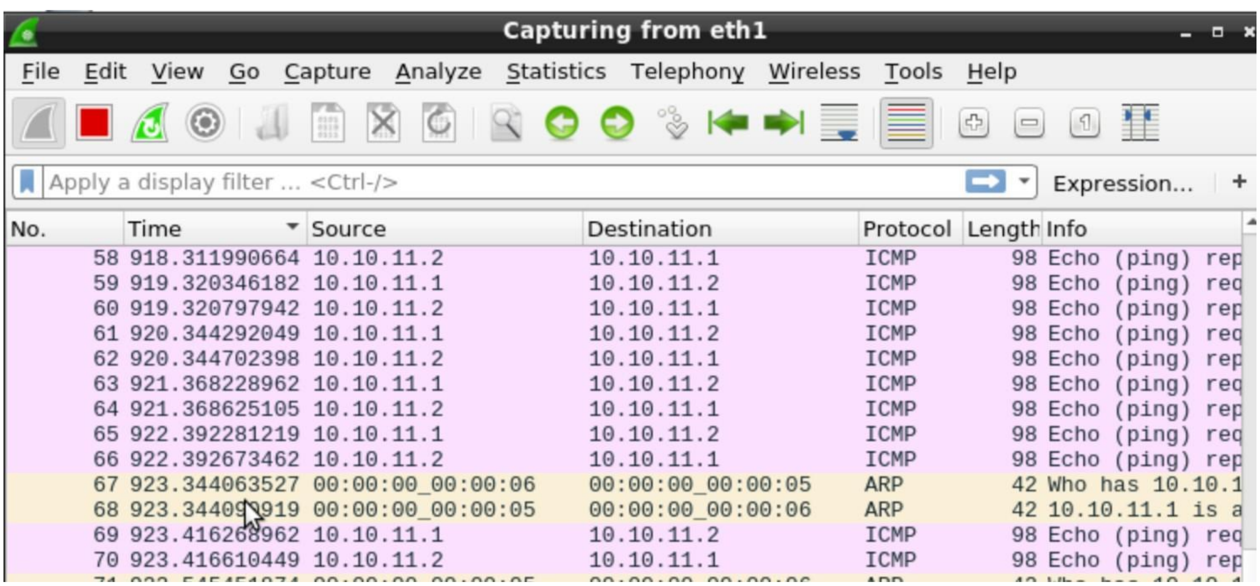
- ➔ Essentially, in this case, we are checking the connection between two separate interfaces (eth1 on R2 and eth0 on R3) of the same routers, R2 and R3.

- ➔ As R2 transmits Echo Request packets to R3's eth0 and R3 replies with Echo Reply packets, we would anticipate seeing ICMP activity.

- ➔ The Wireshark data may be used to verify whether or not R3's eth0 interface is receiving the ping queries and if R3 is responding as intended.

- ➔ By looking at the collected traffic in Wireshark, we may see any network difficulties like firewall rules or routing issues.

In general, keeping track of the Wireshark findings in this situation will assist in confirming the connection and communication between certain interfaces (eth1 on R2 and eth0 on R3) of the network's routers.



No.	Time	Source	Destination	Protocol	Length	Info
58	918.311990664	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) rep
59	919.320346182	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) req
60	919.320797942	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) rep
61	920.344292049	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) req
62	920.344702398	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) rep
63	921.368228962	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) req
64	921.368625105	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) rep
65	922.392281219	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) req
66	922.392673462	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) rep
67	923.344063527	00:00:00_00:00:06	00:00:00_00:00:05	ARP	42	Who has 10.10.1
68	923.34409[illegible]919	00:00:00_00:00:05	00:00:00_00:00:06	ARP	42	10.10.11.1 is a
69	923.4162[illegible]962	10.10.11.1	10.10.11.2	ICMP	98	Echo (ping) req
70	923.416610449	10.10.11.2	10.10.11.1	ICMP	98	Echo (ping) rep

SUBMISSIONS:-

Screenshot of the .conf file under /etc/frr/frr.conf from R2, R3, and R4

```
root@CN-R2:/home/student# cat /etc/frr/frr.conf
frr version 7.1
frr defaults traditional
hostname CN-R2
log syslog informational
service integrated-vtysh-config
!
interface eth1
 ip address 10.10.11.1/30
!
interface eth2
 ip address 10.10.11.5/30
!
line vty
!
```

```
root@CN-R3:/home/student# cat /etc/frr/frr.conf
frr version 7.1
frr defaults traditional
hostname CN-R3
log syslog informational
service integrated-vtysh-config
!
interface eth0
 ip address 10.10.11.2/30
!
interface eth1
 ip address 10.10.11.9/30
!
line vty
!
root@CN-R3:/home/student#
```

```
CN-R4# exit
root@CN-R4:/home/student# cat /etc/frr/frr.conf
frr version 7.1
frr defaults traditional
hostname CN-R4
log syslog informational
service integrated-vtysh-config
!
interface eth0
 ip address 10.10.11.10/30
!
interface eth1
 ip address 10.10.11.6/30
!
interface eth2
 ip address 10.10.11.17/28
!
line vty
!
root@CN-R4:/home/student#
```

IP subnet table

VM (interface)	IP Address	Network Address	Broadcast Address	Range (usable addresses)
R2 (eth1)	10.10.11.1/30	10.10.11.0/30	10.10.11.3/30	10.10.11.1-2/30
R3 (eth0)	10.10.11.2/30			
R2 (eth2)	10.10.11.5/30	10.10.11.4/30	10.10.11.7/30	10.10.11.5-6/30
R4 (eth1)	10.10.11.6/30			
R3 (eth1)	10.10.11.9/30	10.10.11.8/30	10.10.11.11/30	10.10.11.9-10/30
R4 (eth0)	10.10.11.10/30			
R4 (eth2)	10.10.11.17/28	10.10.11.16/28	10.10.11.31/28	10.10.11.17-30/28

Screenshot of the ARP tables on R2, R3, and R4

```
root@CN-R2:/home/student# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.10.11.2               ether   00:00:00:00:00:06   C                     eth1
10.10.11.6               ether   00:00:00:00:00:0a   C                     eth2
```

```
root@CN-R3:/home/student# arp -n
Address                  HWtype  HWaddress           Flags Mask            Ifa
10.10.11.1               ether   00:00:00:00:00:05   C                     eth
```

```
student@CN-R4:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.10.11.9               ether   00:00:00:00:00:07   C                     eth0
10.10.11.5               ether   00:00:00:00:00:09   C                     eth1
```

Screenshot showing that pinging works between R2, R3, and R4

```
root@CN-R2:/home/student# ping 10.10.11.2
PING 10.10.11.2 (10.10.11.2) 56(84) bytes of data.
64 bytes from 10.10.11.2: icmp_seq=1 ttl=64 time=0.353 ms
64 bytes from 10.10.11.2: icmp_seq=2 ttl=64 time=0.489 ms
64 bytes from 10.10.11.2: icmp_seq=3 ttl=64 time=0.460 ms
64 bytes from 10.10.11.2: icmp_seq=4 ttl=64 time=0.433 ms
64 bytes from 10.10.11.2: icmp_seq=5 ttl=64 time=0.428 ms
64 bytes from 10.10.11.2: icmp_seq=6 ttl=64 time=0.385 ms
64 bytes from 10.10.11.2: icmp_seq=7 ttl=64 time=0.407 ms
64 bytes from 10.10.11.2: icmp_seq=8 ttl=64 time=0.447 ms
64 bytes from 10.10.11.2: icmp_seq=9 ttl=64 time=0.350 ms
64 bytes from 10.10.11.2: icmp_seq=10 ttl=64 time=0.443 ms
64 bytes from 10.10.11.2: icmp_seq=11 ttl=64 time=0.404 ms
64 bytes from 10.10.11.2: icmp_seq=12 ttl=64 time=0.424 ms
64 bytes from 10.10.11.2: icmp_seq=13 ttl=64 time=0.392 ms
64 bytes from 10.10.11.2: icmp_seq=14 ttl=64 time=0.325 ms
64 bytes from 10.10.11.2: icmp_seq=15 ttl=64 time=0.496 ms
```

```
PING 10.10.11.1 (10.10.11.1) 56(84) bytes of data.
64 bytes from 10.10.11.1: icmp_seq=1 ttl=64 time=0.746 ms
64 bytes from 10.10.11.1: icmp_seq=2 ttl=64 time=0.361 ms
64 bytes from 10.10.11.1: icmp_seq=3 ttl=64 time=0.388 ms
64 bytes from 10.10.11.1: icmp_seq=4 ttl=64 time=0.408 ms
64 bytes from 10.10.11.1: icmp_seq=5 ttl=64 time=0.451 ms
64 bytes from 10.10.11.1: icmp_seq=6 ttl=64 time=0.460 ms
64 bytes from 10.10.11.1: icmp_seq=7 ttl=64 time=0.336 ms
```

```
student@CN-R3:~$ ping 10.10.11.10
PING 10.10.11.10 (10.10.11.10) 56(84) bytes of data.
64 bytes from 10.10.11.10: icmp_seq=1 ttl=64 time=0.720 ms
64 bytes from 10.10.11.10: icmp_seq=2 ttl=64 time=0.291 ms
64 bytes from 10.10.11.10: icmp_seq=3 ttl=64 time=0.435 ms
64 bytes from 10.10.11.10: icmp_seq=4 ttl=64 time=0.336 ms
64 bytes from 10.10.11.10: icmp_seq=5 ttl=64 time=0.444 ms
64 bytes from 10.10.11.10: icmp_seq=6 ttl=64 time=0.399 ms
^C
--- 10.10.11.10 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 111ms
rtt min/avg/max/mdev = 0.291/0.437/0.720/0.138 ms
```

```
root@CN-R2:/home/student# ping 10.10.11.6
PING 10.10.11.6 (10.10.11.6) 56(84) bytes of data.
64 bytes from 10.10.11.6: icmp_seq=1 ttl=64 time=0.710 ms
64 bytes from 10.10.11.6: icmp_seq=2 ttl=64 time=0.392 ms
64 bytes from 10.10.11.6: icmp_seq=3 ttl=64 time=1.13 ms
64 bytes from 10.10.11.6: icmp_seq=4 ttl=64 time=0.425 ms
64 bytes from 10.10.11.6: icmp_seq=5 ttl=64 time=0.382 ms
64 bytes from 10.10.11.6: icmp_seq=6 ttl=64 time=0.354 ms
^C
--- 10.10.11.6 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 86ms
rtt min/avg/max/mdev = 0.354/0.565/1.127/0.278 ms
```

```
root@CN-R4:/home/student# ping 10.10.11.5
PING 10.10.11.5 (10.10.11.5) 56(84) bytes of data.
64 bytes from 10.10.11.5: icmp_seq=1 ttl=64 time=0.907 ms
64 bytes from 10.10.11.5: icmp_seq=2 ttl=64 time=0.382 ms
64 bytes from 10.10.11.5: icmp_seq=3 ttl=64 time=0.401 ms
64 bytes from 10.10.11.5: icmp_seq=4 ttl=64 time=0.464 ms
64 bytes from 10.10.11.5: icmp_seq=5 ttl=64 time=0.329 ms
64 bytes from 10.10.11.5: icmp_seq=6 ttl=64 time=0.411 ms
64 bytes from 10.10.11.5: icmp_seq=7 ttl=64 time=0.536 ms
64 bytes from 10.10.11.5: icmp_seq=8 ttl=64 time=0.398 ms
64 bytes from 10.10.11.5: icmp_seq=9 ttl=64 time=0.433 ms
^X64 bytes from 10.10.11.5: icmp_seq=10 ttl=64 time=0.414 ms
64 bytes from 10.10.11.5: icmp_seq=11 ttl=64 time=0.510 ms
^C
--- 10.10.11.5 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 211ms
rtt min/avg/max/mdev = 0.329/0.471/0.907/0.149 ms
root@CN-R4:/home/student#
```