Q1 Static multicast route

1 Point

Show the routing table on romeo, after you add the multicast route.

▾ E1_new_table.png Download

```
ty2069@romeo:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.0.1      0.0.0.0         UG    0      0        0 eth0
10.0.0.0        10.10.1.1       255.0.0.0       UG    0      0        0 eth1
10.10.1.0       0.0.0.0         255.255.255.0   U     0      0        0 eth1
172.16.0.0      0.0.0.0         255.240.0.0     U     0      0        0 eth0
224.0.0.0       0.0.0.0         240.0.0.0       U     0      0        0 eth1
```

We are planning to send traffic to the 224.0.0.0 multicast subnet over the experiment interface. Will the routing rule we added determine where traffic for this subnet is sent? What is the range of addresses that this rule will apply to?

Yes, the range is going to be 224.0.0.1 to 239.255.255.254.

Q2 Multicast group membership (Section 7.4, Exercise 2)

1 Point

How many IPv4 multicast groups did each host belong to on the experiment interface, `eth1`?

Annotate a screenshot of the `netstat -g -n` output from each host: circle each IPv4 multicast address that a host belongs to on `eth1`.

▾ E2_netstat1.png Download

```
ty2069@romeo:~$ netstat -g -n
IPv6/IPv4 Group Memberships
Interface       RefCnt Group
--------------- ------ ---------------------
lo              1      224.0.0.1
eth0            1      224.0.0.1
eth1            1      224.0.0.1
lo              1      ff02::1
lo              1      ff01::1
eth0            1      ff02::1:ff90:96cb
eth0            1      ff02::1
eth0            1      ff01::1
eth1            1      ff02::1:ff11:4876
eth1            1      ff02::1
eth1            1      ff01::1
```

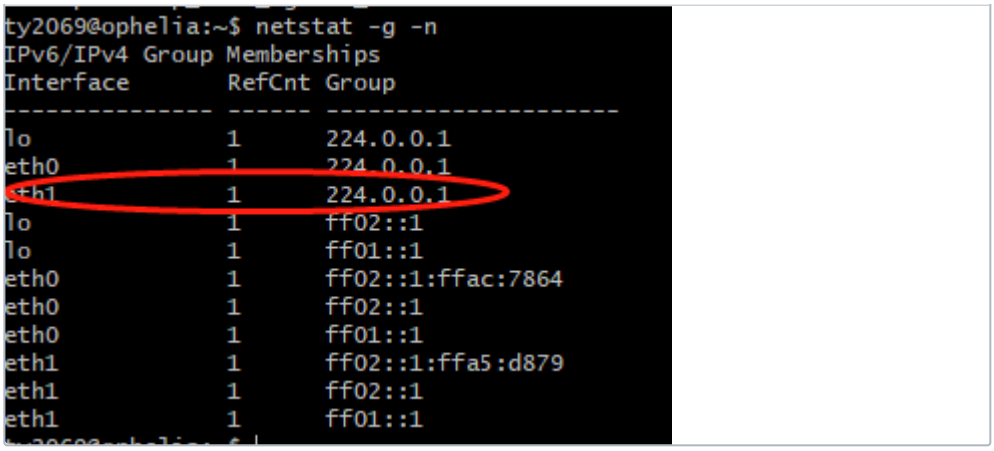
▾ E2_netstat2.png Download

```
ty2069@hamlet:~$ netstat -g -n
IPv6/IPv4 Group Memberships
Interface       RefCnt Group
--------------- ------ ---------------------
lo              1      224.0.0.1
eth0            1      224.0.0.1
eth1            1      224.0.0.1
lo              1      ff02::1
lo              1      ff01::1
eth0            1      ff02::1:ff66:2d54
eth0            1      ff02::1
eth0            1      ff01::1
eth1            1      ff02::1:ffc7:9c25
eth1            1      ff02::1
eth1            1      ff01::1
```

▾ E2_netstat3.png Download

```
ty2069@juliet:~$ netstat -g -n
IPv6/IPv4 Group Memberships
Interface       RefCnt Group
--------------- ------ ---------------------
lo              1      224.0.0.1
eth0            1      224.0.0.1
eth1            1      224.0.0.1
lo              1      ff02::1
lo              1      ff01::1
eth0            1      ff02::1:ff37:1931
eth0            1      ff02::1
eth0            1      ff01::1
eth1            1      ff02::1:ff06:d7a7
eth1            1      ff02::1
eth1            1      ff01::1
```

▾ E2_netstat4.png Download



Refer to the list of multicast group IDs registered with IANA. Is the multicast group that these hosts belong to a special "well-known" group? What is it used for? Explain.

Just one IPv4 multicast group belongs to the experiment interface. It belongs to all-systems.mcast.net meaning All Systems on this Su bnet

Q3 Multicast vs Broadcast (Section 7.4, Exercise 3)

2 Points

Q3.1 Multicast ping

1 Point

Show the output of the multicast `ping` command on romeo (either as a screenshot, or copy and paste terminal output).

screenshot

ping_multi.png Download

```
ty2069@romeo:~$ ping -c 3 -I eth1 224.0.0.1
PING 224.0.0.1 (224.0.0.1) from 10.10.1.100 eth1: 56(84) bytes of data.
64 bytes from 10.10.1.100: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 10.10.1.102: icmp_seq=1 ttl=64 time=1.43 ms (DUP!)
64 bytes from 10.10.1.101: icmp_seq=1 ttl=64 time=1.78 ms (DUP!)
64 bytes from 10.10.1.100: icmp_seq=2 ttl=64 time=0.061 ms
64 bytes from 10.10.1.102: icmp_seq=2 ttl=64 time=0.866 ms (DUP!)
64 bytes from 10.10.1.101: icmp_seq=2 ttl=64 time=1.08 ms (DUP!)
64 bytes from 10.10.1.100: icmp_seq=3 ttl=64 time=0.052 ms

--- 224.0.0.1 ping statistics ---
3 packets transmitted, 3 received, +4 duplicates, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.049/0.762/1.789/0.668 ms
```

Which of these hosts sent an ICMP echo response when romeo sends an ICMP echo request to the *multicast* address?

- [x] juliet (10.10.1.101)
- [x] hamlet (10.10.1.102)
- [ ] router (10.10.1.1)
- [ ] ophelia (10.10.2.103)

Explain these results. Did the hosts that did *not* respond to the multicast ICMP echo request receive the request, or did they receive the request and choose not to reply? If the former - why did they not receive the request? If the latter - why did they choose not to reply? Answer separately for *each* host that did not reply. Use evidence from `tcpdump` to support your answer.

The host "Ophelia" did not receive the request since no packet ca pture from the tcpdump comment. "Juliet" and "hamlet" reply beca use they are in the same multicast group with "Romeo".

▾ tcp.png Download

```
ty2069@ophelia:~$ sudo tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

▾ tcp_multicast1.png Download

```
ty2069@hamlet:~$ sudo tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
16:02:49.967124 IP 10.10.1.100 > 224.0.0.1: ICMP echo request, id 5656, seq 1, l
ength 64
16:02:49.967199 ARP, Request who-has 10.10.1.100 tell 10.10.1.102, length 28
16:02:49.967715 ARP, Reply 10.10.1.100 is-at 02:fd:50:11:48:76, length 28
16:02:49.967730 IP 10.10.1.102 > 10.10.1.100: ICMP echo reply, id 5656, seq 1, l
ength 64
16:02:49.967769 ARP, Request who-has 10.10.1.100 tell 10.10.1.101, length 46
16:02:49.967962 ARP, Reply 10.10.1.100 is-at 02:fd:50:11:48:76, length 28
16:02:49.968616 IP 10.10.1.101 > 10.10.1.100: ICMP echo reply, id 5656, seq 1, l
ength 64
16:02:50.968503 IP 10.10.1.100 > 224.0.0.1: ICMP echo request, id 5656, seq 2, l
ength 64
16:02:50.968558 IP 10.10.1.102 > 10.10.1.100: ICMP echo reply, id 5656, seq 2, l
ength 64
16:02:50.969309 IP 10.10.1.101 > 10.10.1.100: ICMP echo reply, id 5656, seq 2, l
ength 64
16:02:51.970399 IP 10.10.1.100 > 224.0.0.1: ICMP echo request, id 5656, seq 3, l
ength 64
16:02:51.970437 IP 10.10.1.102 > 10.10.1.100: ICMP echo reply, id 5656, seq 3, l
ength 64
16:02:51.971075 IP 10.10.1.101 > 10.10.1.100: ICMP echo reply, id 5656, seq 3, l
ength 64
```

▾ tcp_multicast2.png Download

```
ty2069@juliet:~$ sudo tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
16:02:49.963842 IP 10.10.1.100 > 224.0.0.1: ICMP echo request, id 5656, seq 1, l
ength 64
16:02:49.963914 ARP, Request who-has 10.10.1.100 tell 10.10.1.101, length 28
16:02:49.964157 ARP, Request who-has 10.10.1.100 tell 10.10.1.102, length 46
16:02:49.964678 ARP, Reply 10.10.1.100 is-at 02:fd:50:11:48:76, length 46
16:02:49.964689 IP 10.10.1.101 > 10.10.1.100: ICMP echo reply, id 5656, seq 1, l
ength 64
16:02:50.965373 IP 10.10.1.100 > 224.0.0.1: ICMP echo request, id 5656, seq 2, l
ength 64
16:02:50.965430 IP 10.10.1.101 > 10.10.1.100: ICMP echo reply, id 5656, seq 2, l
ength 64
16:02:51.967130 IP 10.10.1.100 > 224.0.0.1: ICMP echo request, id 5656, seq 3, l
ength 64
16:02:51.967178 IP 10.10.1.101 > 10.10.1.100: ICMP echo reply, id 5656, seq 3, l
ength 64
```

Q3.2 Broadcast ping

1 Point

Show the output of the broadcast `ping` command on romeo (either as a screenshot, or copy and paste terminal output).

screenshot

▾ ping_broad.png Download

```
ty2069@romeo:~$ ping -c 3 -b 10.10.1.255
WARNING: pinging broadcast address
PING 10.10.1.255 (10.10.1.255) 56(84) bytes of data.
64 bytes from 10.10.1.100: icmp_seq=1 ttl=64 time=0.069 ms
64 bytes from 10.10.1.102: icmp_seq=1 ttl=64 time=0.701 ms (DUP!)
64 bytes from 10.10.1.100: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 10.10.1.102: icmp_seq=2 ttl=64 time=0.706 ms (DUP!)
64 bytes from 10.10.1.100: icmp_seq=3 ttl=64 time=0.056 ms

--- 10.10.1.255 ping statistics ---
3 packets transmitted, 3 received, +2 duplicates, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.049/0.316/0.706/0.316 ms
```

Which of these hosts sent an ICMP echo response when romeo sends an ICMP echo request to the *broadcast* address?

- [ ] juliet (10.10.1.101)
- [x] hamlet (10.10.1.102)
- [ ] router (10.10.1.1)
- [ ] ophelia (10.10.2.103)

Explain these results. Did the hosts that did *not* respond to the broadcast ICMP echo request receive the request, or did they receive the request and choose not to reply? If the former - why did they not receive the request? If the latter - why did they choose not to reply? Answer separately for *each* host that did not reply. Use evidence from `tcpdump` to support your answer.

The host "Ophelia" did not receive the request since no packet ca pture from the tcpdump comment and "Juliet" receive the request and choose not to reply. In this case, we change the subnet mask for Juliet so that Romeo's IP address is no longer in the range of J uliet. But juliet's address is in Romeo's range, So it can receive the request but cannot reply

▾ tcp.png Download

```
ty2069@ophelia:~$ sudo tcpdump -i eth1 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

▾ tcp_broadcast.png Download

```
16:04:03.800881 IP 10.10.1.100 > 10.10.1.255: ICMP echo request, id 5658, seq 1,
 length 64
16:04:03.800948 IP 10.10.1.102 > 10.10.1.100: ICMP echo reply, id 5658, seq 1, l
ength 64
16:04:04.801223 IP 10.10.1.100 > 10.10.1.255: ICMP echo request, id 5658, seq 2,
 length 64
16:04:04.801283 IP 10.10.1.102 > 10.10.1.100: ICMP echo reply, id 5658, seq 2, l
ength 64
16:04:05.802538 IP 10.10.1.100 > 10.10.1.255: ICMP echo request, id 5658, seq 3,
 length 64
16:04:05.802597 IP 10.10.1.102 > 10.10.1.100: ICMP echo reply, id 5658, seq 3, l
ength 64
16:04:08.811910 ARP, Request who-has 10.10.1.100 tell 10.10.1.102, length 28
16:04:08.812575 ARP, Reply 10.10.1.100 is-at 02:fd:50:11:48:76, length 28
```

▾ tcp_broadcast2.png Download

```
16:04:03.797722 IP 10.10.1.100 > 10.10.1.255: ICMP echo request, id 5658, seq 1,
 length 64
16:04:04.798069 IP 10.10.1.100 > 10.10.1.255: ICMP echo request, id 5658, seq 2,
 length 64
16:04:05.799347 IP 10.10.1.100 > 10.10.1.255: ICMP echo request, id 5658, seq 3,
 length 64
```

Q4 Multicast MAC Addresses (Section 7.4, Exercise 4)

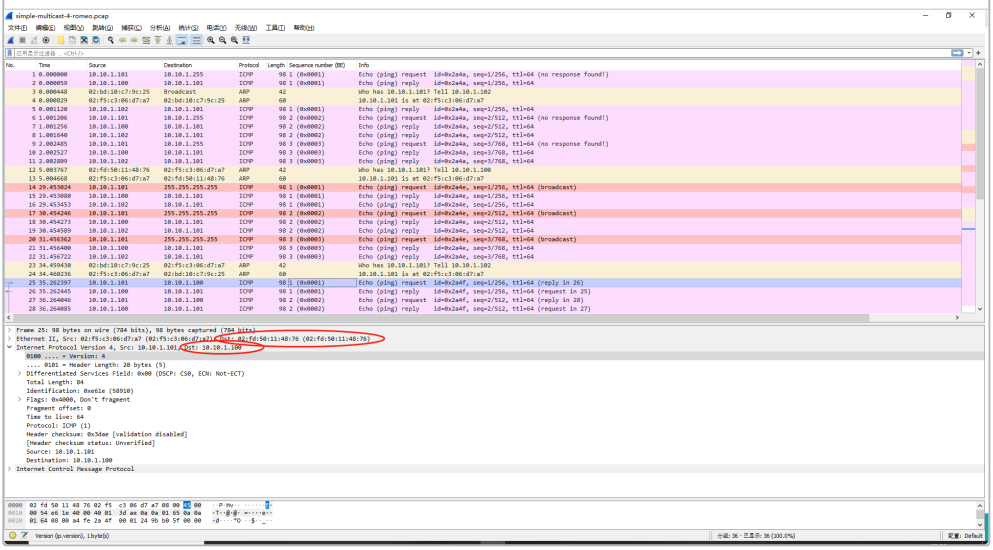
3 Points

Q4.1 Multicast MAC Addresses

2 Points

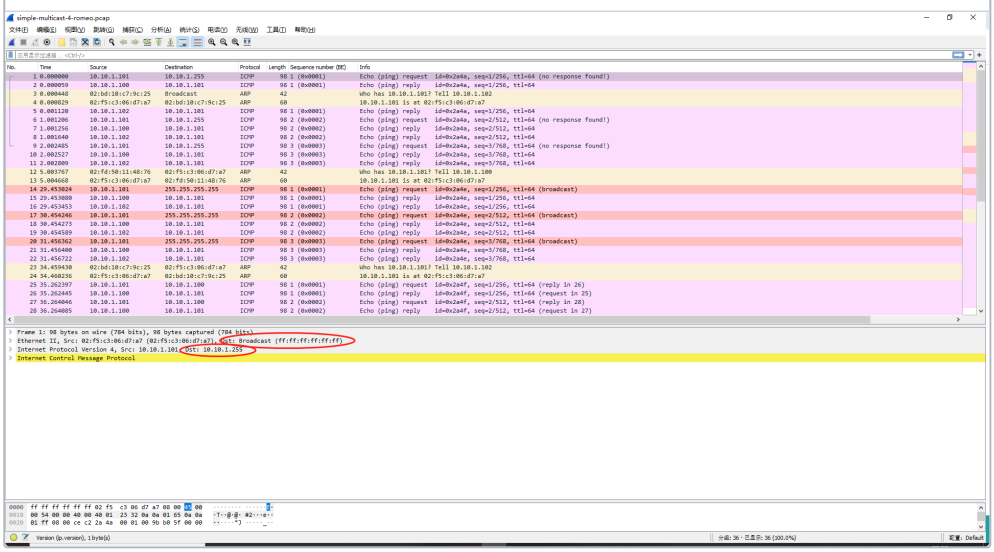
Show screenshots of your Wireshark or `tcpdump` output from this experiment - annotate these screenshots to circle the destination IP address AND the destination MAC address for one ICMP echo request to each of the five destination addresses.

▾ 10_10_1_100.png Download



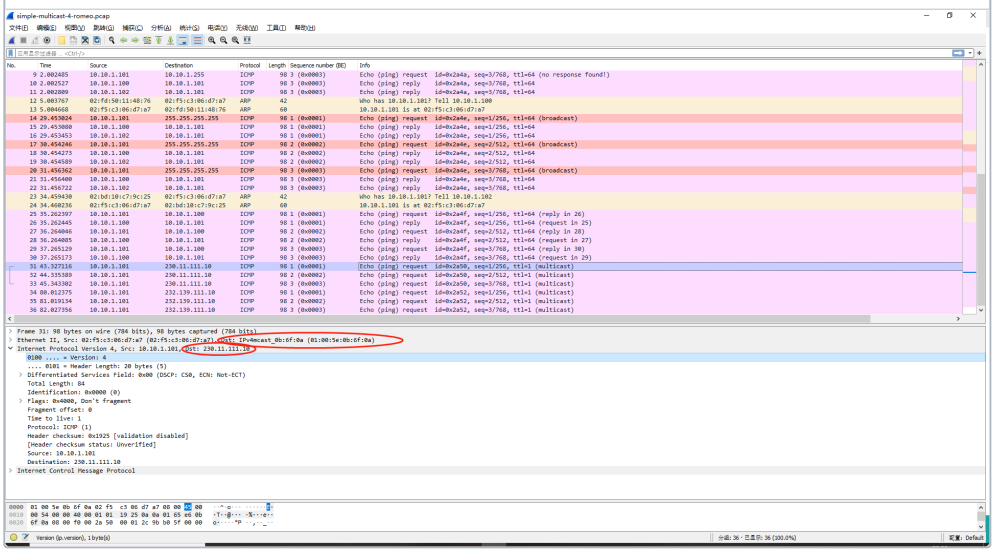
10_10_1_255.png

Download



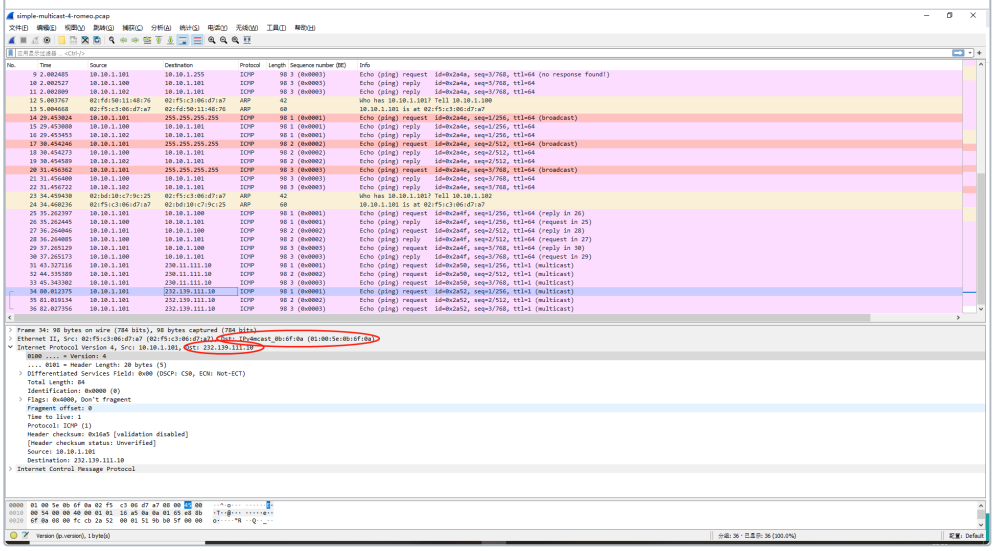
230_11_111_10.png

Download

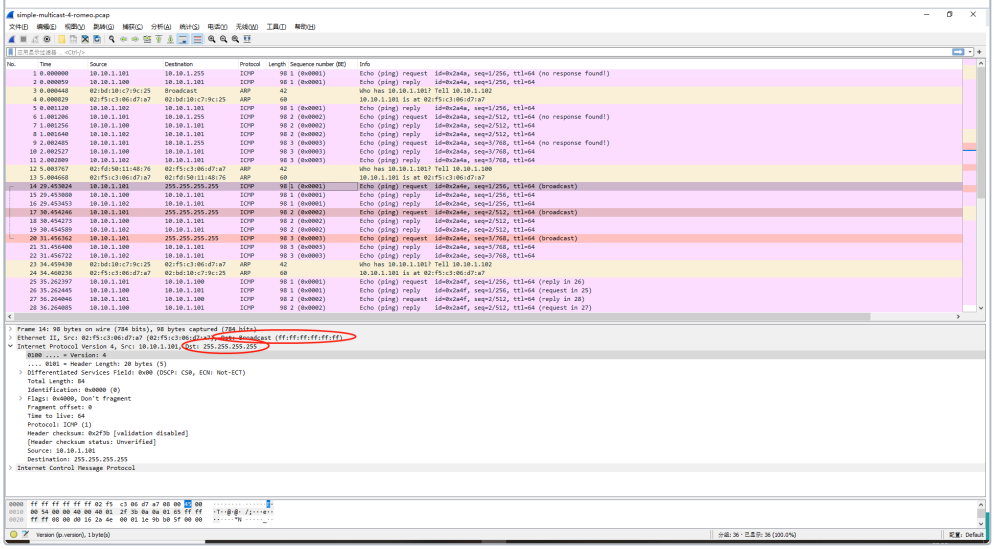


232_139_111_10.png

Download



255_255_255_255.png Download



For each destination IP address -

- 10.10.1.255
- 255.255.255.255
- 10.10.1.100
- 230.11.111.10
- 232.139.111.10

what was the destination MAC address, and how does the host find out the destination MAC address to put in the Ethernet header? Explain each case separately.

10.10.1.255: ff:ff:ff:ff:ff:ff
255.255.255.255: ff:ff:ff:ff:ff:ff
10.10.1.100: 02:fd:50:11:48:76
230.11.111.10: 01:00:5e:0b:6f:0a
232.139.111.10: 01:00:5e:0b:6f:0a

The first two cases are broadcast, so the MAC address is broadcas t MAC address.
The third case is using ARP to learn the destination MAC address and the last two cases are using MAC address mapping.

By looking at a MAC address, can we identify whether it is a broadcast, unicast, or multicast address? Explain.

Yes, if the MAC address is ff:ff:ff:ff:ff:ff then it is a broadcast. If the fi rst octet has a value of 1

in the least-significant bit and the rest are unicast.

Q4.2 MAC Address Mapping
1 Point

Use the frames with a multicast destination address to explain how a multicast group address is mapped to a multicast MAC address.

For example, we look at the IP address 230.11.111.10 which is 11100110.00001011.01101111.00001010 in binary.
Look at the last 23 bits which are 000 1011 0110 1111 0000 1010
And convert to hexadecimal
We have 0 b 6 f 0 a
So we will have a MAC address of 01:00:5E:0B:6F:0A

For the two multicast frames captured, do they have the same destination MAC address? Why?

Because 230.11.111.10 and 232.139.111.10 are identical for the last 23 bits.

Will *all* multicast frames have the same destination MAC address? Explain.

Nope.

Q5 Multicast Copies (Section 7.4, Exercise 5)
1 Point

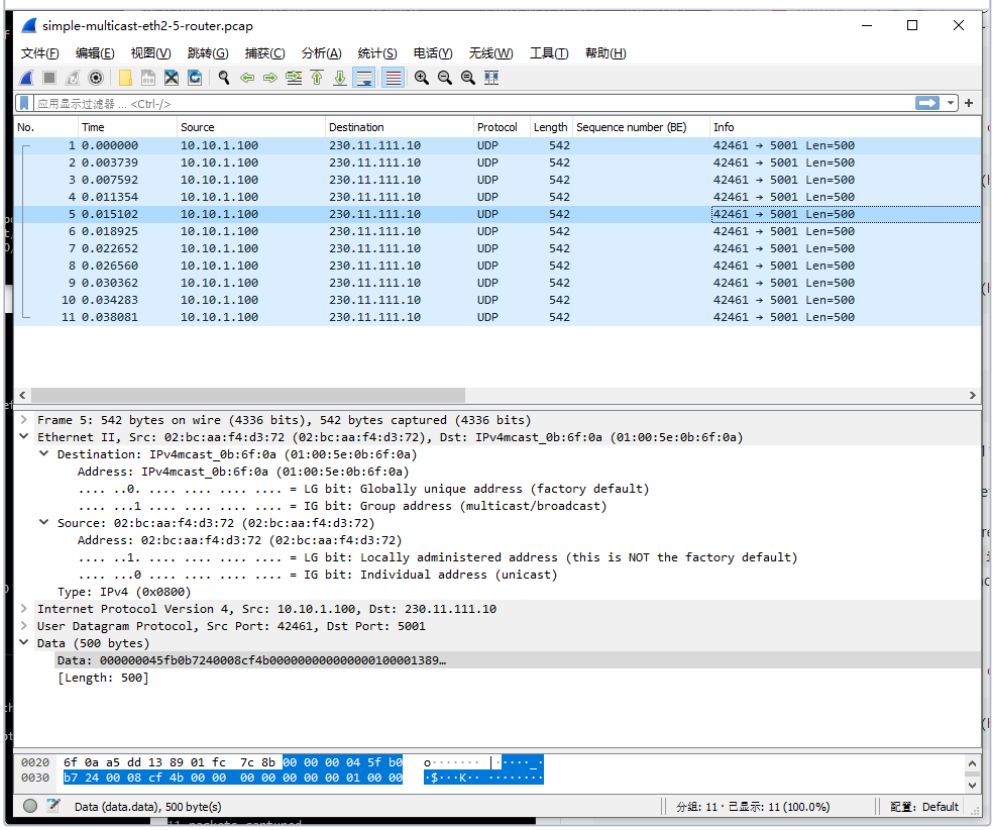
On the 10.10.1.0/24 LAN, how many hosts received the datagrams sent by `iperf`? On the 10.10.2.0/24 LAN, how many hosts received the datagrams sent by `iperf`?

Did the sending host send a *copy* of each of the ten datagrams for each host that received the datagrams, or did it send a single instance of each datagram?

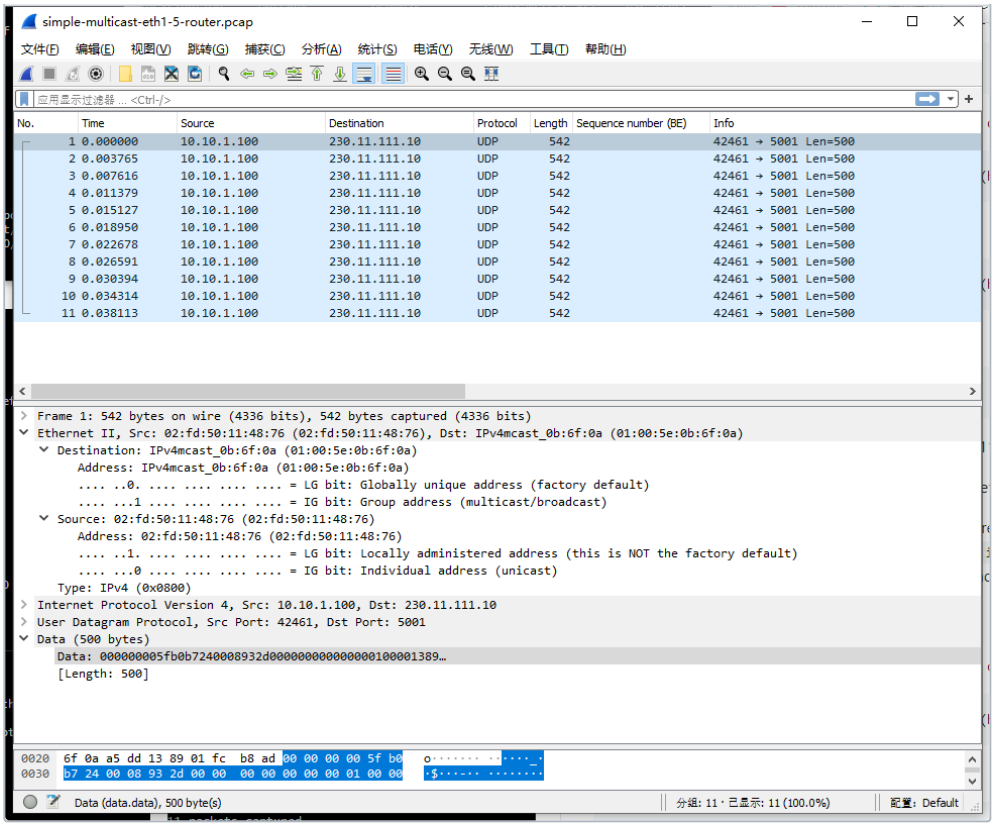
3 hosts received the datagrams sent by iperf. it sends a single inst ance of each datagram.

Show a Wireshark screenshot to support your answer.

▾ tcpdump.png Download



▾ tcpdump2.png Download



Q6 Simple Multicast Exercise (Section 7.4, Exercise 6)

2 Points

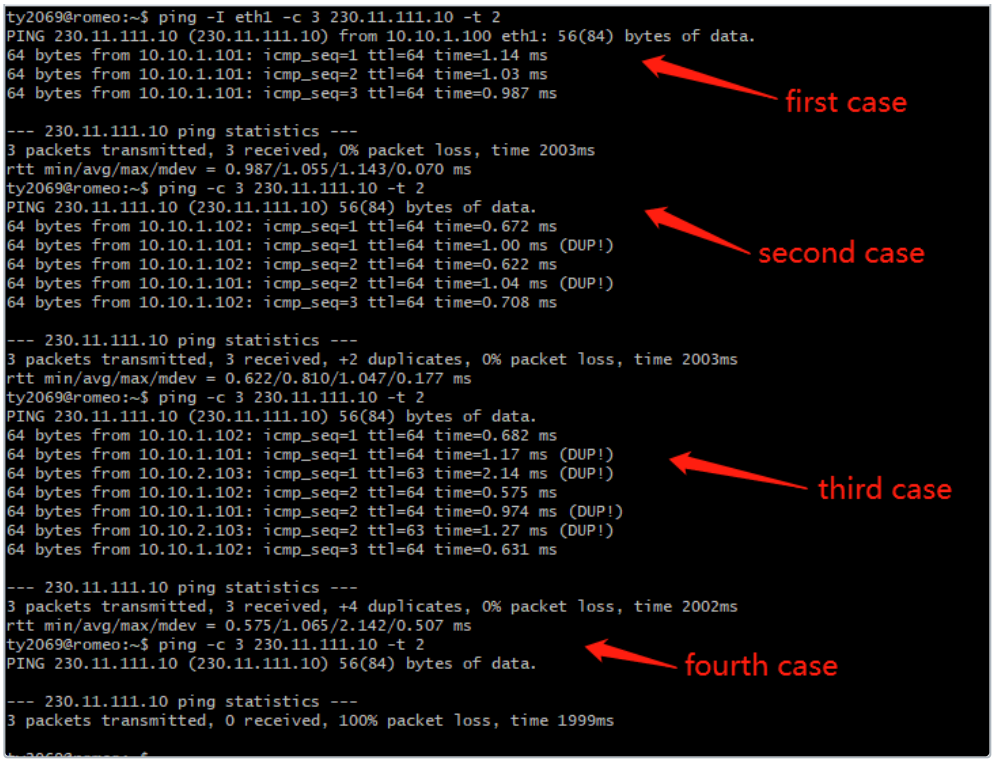
Q6.1 Ping Responses

1 Point

Identify which hosts responded to the ping in each case (four separate cases). Explain *why* the same `ping` command got different responses.

Support your answer with screenshots of the `ping` output on romeo.

▼ ping.png Download



Juliet responded to the ping in the first case.
Juliet and Hamlet responded to the ping in the second case.
Juliet, Hamlet, and Ophelia responded to the ping in the third cas e.
No host response in the last case.

Because the hosts join the multicast group by running iperf instanc e. When it joins the group, it will respond to the ping. When no hos ts in the group, then no hosts will respond to the ping.

Q6.2 Multicast group membership

1 Point

Show the output of

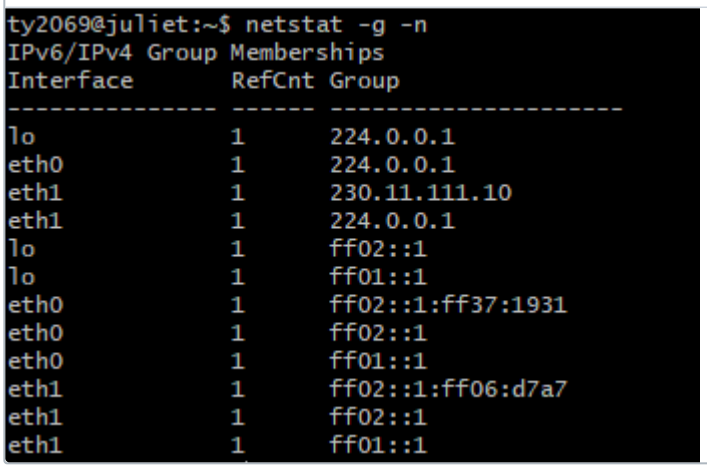
```
netstat -g -n
```

on juliet, before and after you started the `iperf` server.

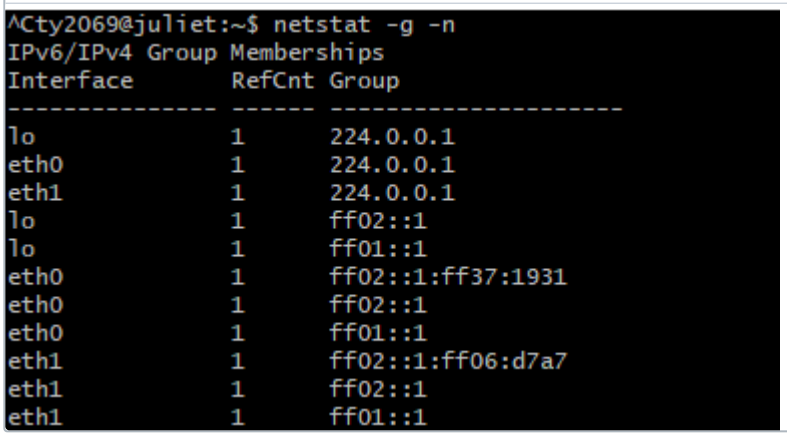
What IPv4 multicast groups is juliet a member of in each case? Explain.

Before we start the iperf server, Juliet is a member of the 224.0.0.1 multicast group. After started the iperf server, Juliet is a member o f the 224.0.0.1 multicast group and 230.11.111.10 multicast group. By running the iperf server, Juliet join the new multicast group.

juliet_after.png Download



juliet_before.png Download



Q7 Delete your resources, please

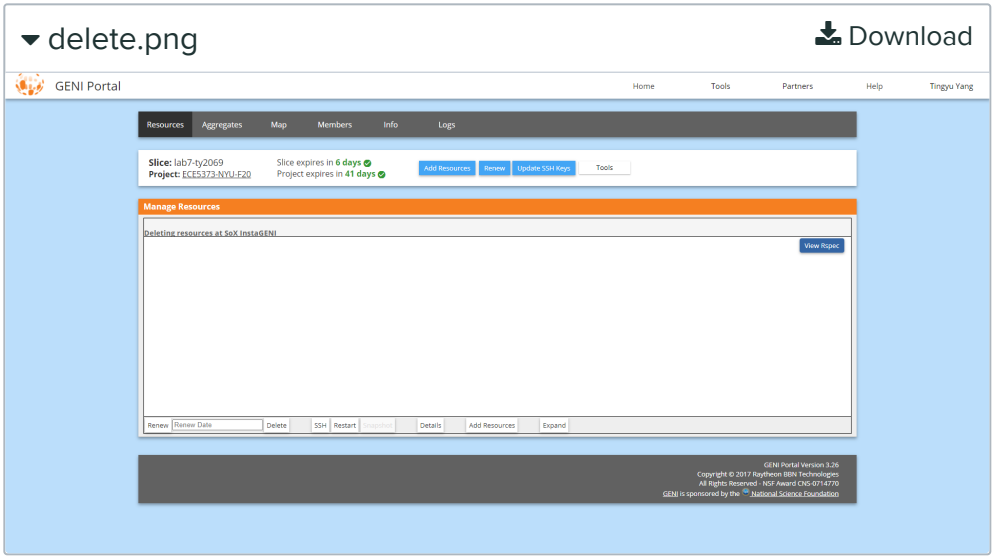
0 Points

Did you delete your resources in the GENI Portal? After you have finished submitting your answers to the questions above, delete your resources so that they will be available to other experimenters.

- [x] Yes, I deleted my resources.

Upload a screenshot of the slice page for each of the slices that you used for lab 5. Your screenshots should show that there are no resources left in your slice.

delete.png Download



Lab 7: Multicast

UNGRADED

STUDENT
Tingyu Yang

TOTAL POINTS
- / 10 pts

QUESTION 1
Static multicast route — 1 pt

QUESTION 2
Multicast group membership (Section 7.4, Exercise 2) — 1 pt

QUESTION 3
Multicast vs Broadcast (Section 7.4, Exercise 3) — 2 pts

3.1 Multicast ping — 1 pt

3.2 Broadcast ping — 1 pt

QUESTION 4

Multicast MAC Addresses (Section 7.4, Exercise 4) 3 pts

4.1 Multicast MAC Addresses 2 pts

4.2 MAC Address Mapping 1 pt

QUESTION 5

Multicast Copies (Section 7.4, Exercise 5) 1 pt

QUESTION 6

Simple Multicast Exercise (Section 7.4, Exercise 6) 2 pts

6.1 Ping Responses 1 pt

6.2 Multicast group membership 1 pt

QUESTION 7

Delete your resources, please 0 pts