NSCap Lab Report

Lab 2

Part 1

1. Finish Step 1

   a. h2 can ping h3, because they are under the same subnet and are connect by a switch. The packets deliveries don't need the help of r1.

```
mininet> h2 ping h3 -c 5
PING 192.168.1.66 (192.168.1.66) 56(84) bytes of data.
64 bytes from 192.168.1.66: icmp_seq=1 ttl=64 time=0.292 ms
64 bytes from 192.168.1.66: icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from 192.168.1.66: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 192.168.1.66: icmp_seq=4 ttl=64 time=0.037 ms
64 bytes from 192.168.1.66: icmp_seq=5 ttl=64 time=0.044 ms

--- 192.168.1.66 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4092ms
rtt min/avg/max/mdev = 0.037/0.096/0.292/0.098 ms
```

   b. h2 can not ping h4, because they are in different subnet and the packets deliveries needs the help from correctly-configured routers.

```
mininet> h2 ping h4 -c 5
ping: connect: Network is unreachable
```

2. Show that the configuration is correct.

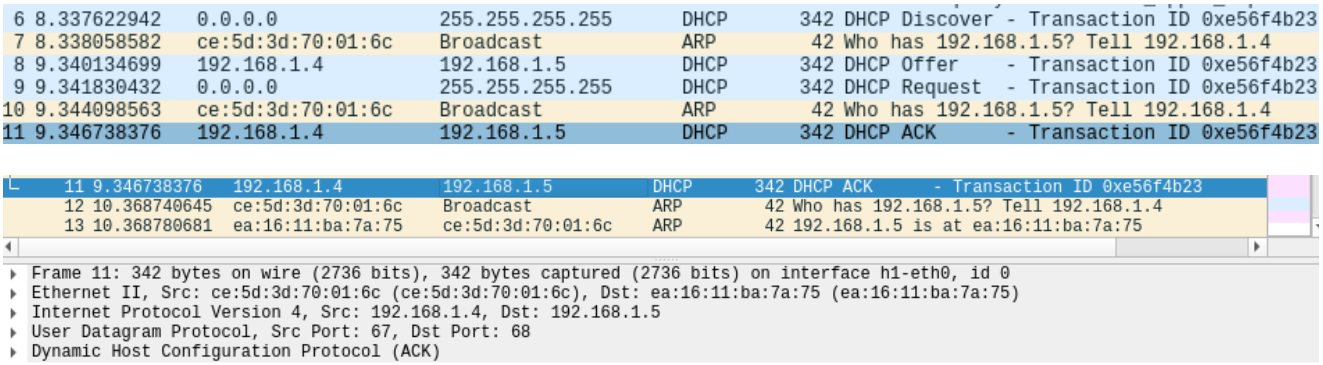
```
mininet> exit
h1 doesn't have connectivity to 192.168.1.65
h1 doesn't have connectivity to 192.168.1.66
h1 doesn't have connectivity to 192.168.3.1
h1 doesn't have connectivity to 192.168.3.2
WRONG ANSWER
[-] Killing DHCP server

# tommytyc @ ubuntu in ~/NSCap/lab2 on git:main x [16:29:53]
$
```

- By the figure above, we can see that all hosts can ping each other except h1.

Part 2

3. Show DHCP messages on h1-eth0



h1(MAC: `ea:16:11:ba:7a:75`) got `192.168.1.5` from DHCPServer(MAC: `ce:5d:3d:70:01:6c`, IP: `192.168.1.4`)

4. Can hosts other than h1 acquire IP addresses from DHCP server?

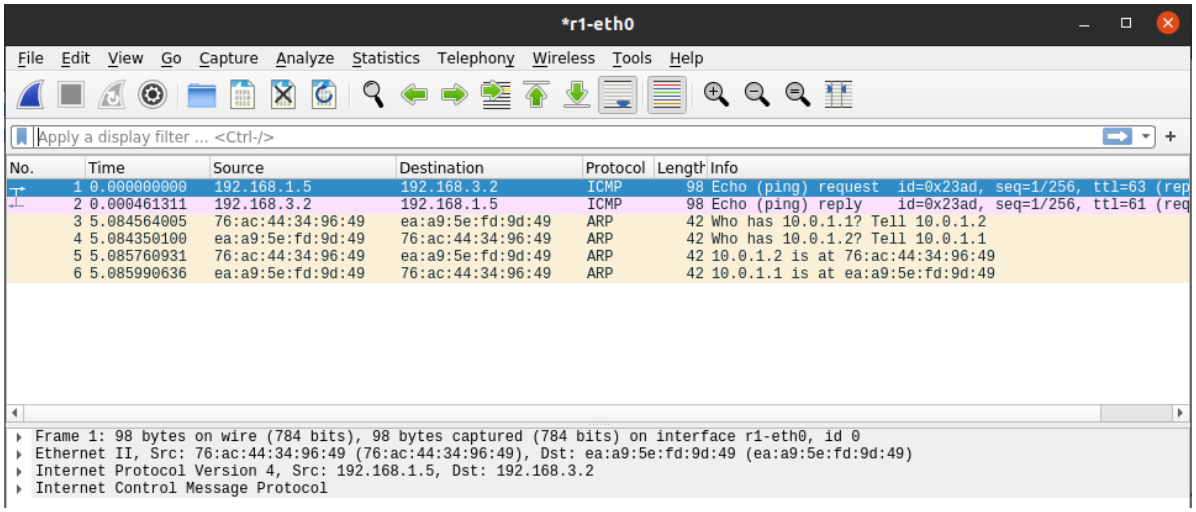
   - No. Because we have set up the ip address for the network interface of each host.

Part 3

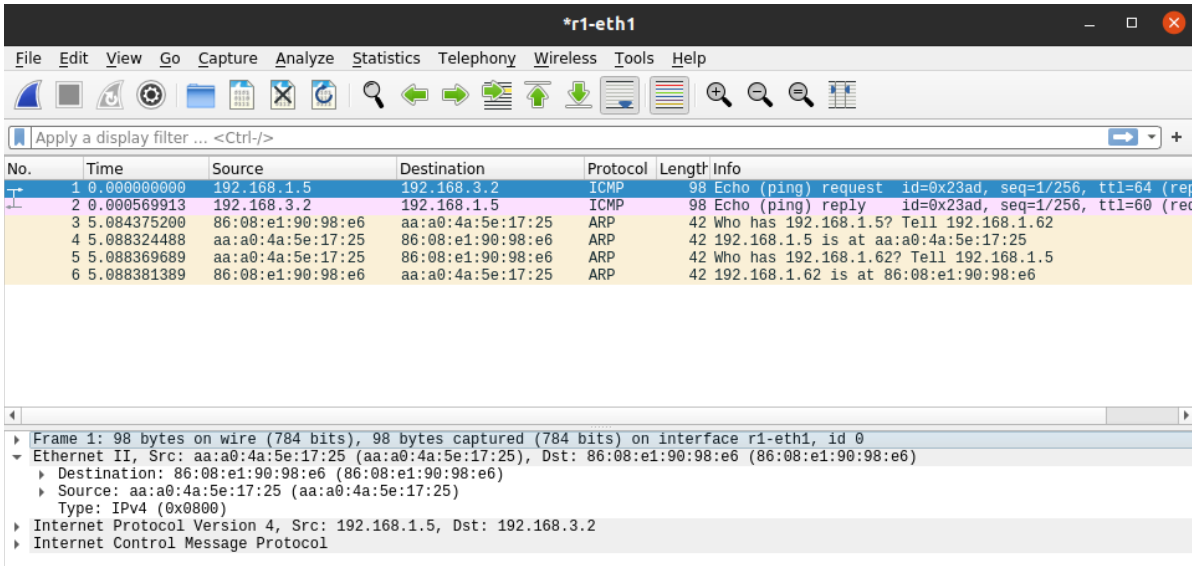
5. What does r1 do on the packets from h1 to h5, and h5 to h1, respectively?

   - h1 to h5

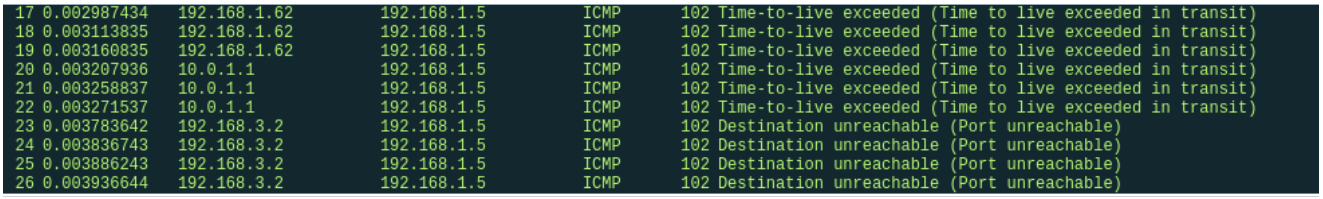
     - r1 helps to send the ICMP packets from h1 to r2(and the packets will go to h5 through r2, r3, r4). r1 will firstly send arp packet to discover r2's MAC address and then sends the ICMP packet to r2.



- h5 to h1
  - r1 helps to send the ICMP packets from h5 to h1. r1 will firstly send arp packet to discover h1's MAC address and then sends the ICMP packets to h1.



6. Capture all ICMP messages received by h1 and explain why h1 can only derive only 1st, 2nd, and 5th hops details.



   - Because the **TTL exceeds ICMP messages** are sent by routers. If we don't manually configure the static routing rule in each router, the router will not be able to decide where to route th coming packets, and finally discards these packets.

7. The types and senders of the ICMP packets come from the first and second hops
   - first
     - type: 11(Time-to-live exceeded)
     - sender: 192.168.1.62(r1)
   - second
     - type: 11(Time-to-live exceeded)
     - sender: 10.0.1.1(r2)

8. The types and senders of the ICMP packets come from the fifth hop
   - type: 3(Destination Unreachable)

- sender: 192.168.3.2(h5)

9. (Bonus)

```
mininet> h1 traceroute h5
traceroute to 192.168.3.2 (192.168.3.2), 30 hops max, 60 byte packets
 1  192.168.1.62 (192.168.1.62)  4.594 ms  4.601 ms  4.594 ms
 2  10.0.1.1 (10.0.1.1)  4.590 ms  4.584 ms  4.577 ms
 3  10.0.0.2 (10.0.0.2)  4.574 ms  4.568 ms  4.563 ms
 4  10.0.2.3 (10.0.2.3)  4.560 ms  4.553 ms  4.547 ms
 5  192.168.3.2 (192.168.3.2)  4.550 ms  4.530 ms  4.522 ms
```

- I add some extra static routing rules into each router. As we can see in the following figure, these rule tell the router how to handle the TTL exceeds ICMP packets sent from other router.

```
# Rule for TTL exceeds ICMP messages from each router
routers['r1'].cmd('route add -net 10.0.0.0/24 gw 10.0.1.1')
routers['r1'].cmd('route add -net 10.0.2.0/24 gw 10.0.1.1')
routers['r2'].cmd('route add -net 10.0.0.0/24 gw 10.0.0.2')
routers['r2'].cmd('route add -net 10.0.2.0/24 gw 10.0.0.2')
routers['r3'].cmd('route add -net 10.0.2.0/24 gw 10.0.2.3')
routers['r3'].cmd('route add -net 10.0.1.0/24 gw 10.0.0.1')
routers['r4'].cmd('route add -net 10.0.2.0/24 gw 10.0.2.1')
routers['r4'].cmd('route add -net 10.0.0.0/24 gw 10.0.2.1')
routers['r4'].cmd('route add -net 10.0.1.0/24 gw 10.0.2.1')
```