

Part 1

```
eric070021@ubuntu:~/Desktop$ sudo python topology.py
h1 doesn't have connectivity to 192.168.1.129
h1 doesn't have connectivity to 192.168.1.130
h1 doesn't have connectivity to 192.168.3.2
h1 doesn't have connectivity to 192.168.3.3
h2 doesn't have connectivity to 192.168.1.129
h2 doesn't have connectivity to 192.168.1.130
h2 Software Updater connectivity to 192.168.3.2
h2 doesn't have connectivity to 192.168.3.3
h3 doesn't have connectivity to 192.168.3.2
h3 doesn't have connectivity to 192.168.3.3
h4 doesn't have connectivity to 192.168.3.2
h4 doesn't have connectivity to 192.168.3.3
h5 doesn't have connectivity to 192.168.1.129
h5 doesn't have connectivity to 192.168.1.130
h6 doesn't have connectivity to 192.168.1.129
h6 doesn't have connectivity to 192.168.1.130
WRONG ANSWER
```

(the connect status after complete step 1-1)

1(a).

Yes, it can. Since h3 and h4 are in the same subnet.

1(b).

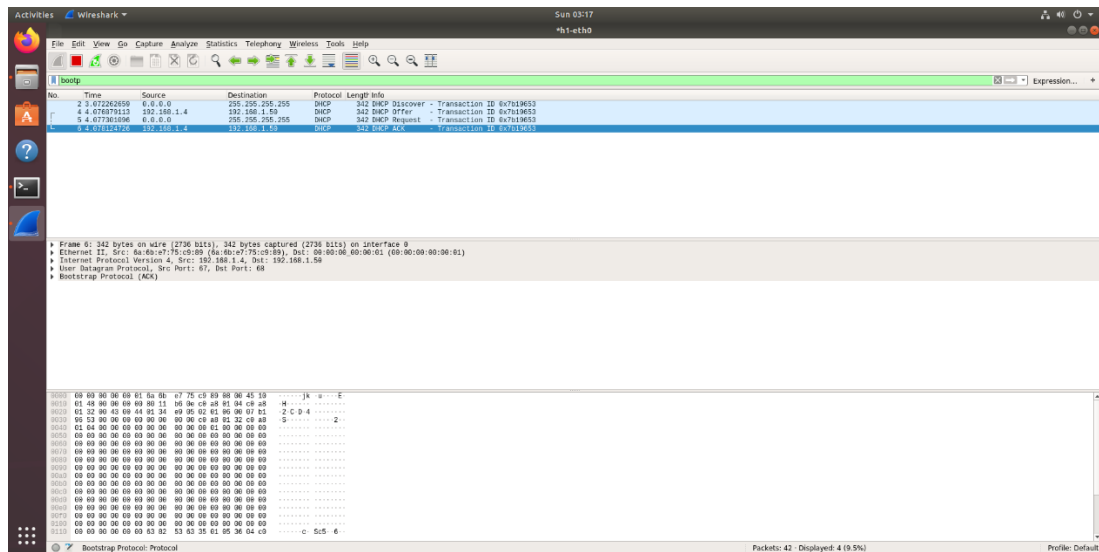
No, it can't. Since h3 and h5 are not in the same subnet and we haven't configured static routing rules yet.

2.

```
eric070021@ubuntu:~/Desktop$ sudo python topology.py
h1 doesn't have connectivity to 192.168.1.129
h1 doesn't have connectivity to 192.168.1.130
h1 doesn't have connectivity to 192.168.3.2
h1 doesn't have connectivity to 192.168.3.3
h2 doesn't have connectivity to 192.168.1.129
h2 doesn't have connectivity to 192.168.1.130
h2 doesn't have connectivity to 192.168.3.2
h2 doesn't have connectivity to 192.168.3.3
WRONG ANSWER
```

Part 2

3.



4.

No, since the one who need DHCP service has no ip yet, it can only broadcast the DHCP discover message in its subnet. So, it can only acquire IP address from DHCP server in its subnet.

5.

```
mininet> h2 dhclient h2-eth0
mininet> h2 ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.34 netmask 255.255.255.128 broadcast 192.168.1.127
    inet6 fe80::200:ff:fe00:2 prefixlen 64 scopeid 0x20<link>
    ether 00:00:00:00:00:02 txqueuelen 1000 (Ethernet)
    RX packets 67 bytes 7239 (7.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15 bytes 1750 (1.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

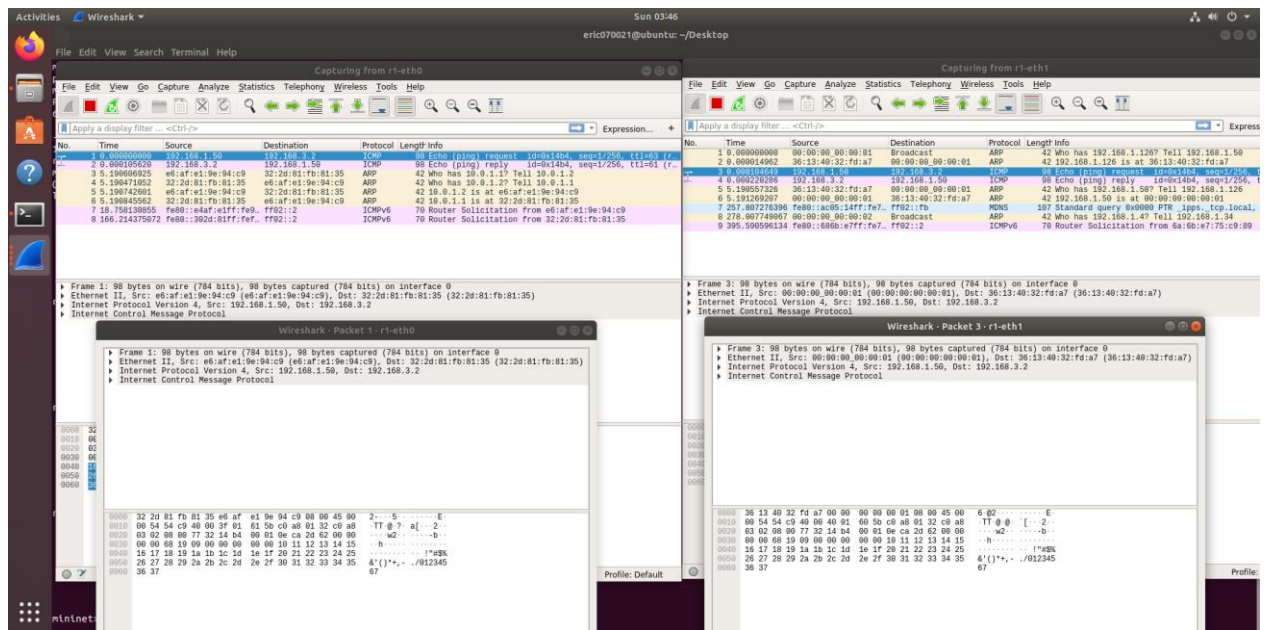
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

6.

I use “host” statement in dhcpd.conf to assign fixed address to h2. It can recognize host by mac address, since h2’s mac address is fixed (00:00:00:00:00:02)

Part 3

7.



H1 to h5: h1 send packet to r1, r1 substitute the source mac address r1’s mac address, relay the packet to r2.

H5 to h1: r2 send packet to r1, substitute the source mac address r1’s mac address, r1 send the packet to h1.

8(a)(b).

R1:



Wireshark interface showing a packet capture on interface *r3-eth0. The filter is set to icmp. The packet list shows 36 ICMP packets. The packet details pane shows the selected packet (Frame 2) with the following information:

- Frame 2: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0
- Ethernet II, Src: f2:1c:59:c8:53:26 (f2:1c:59:c8:53:26), Dst: 1e:f1:a0:9b:10:fb (1e:f1:a0:9b:10:fb)
- Internet Protocol Version 4, Src: 10.0.0.2, Dst: 192.168.1.50
- Internet Control Message Protocol

Hex dump of the selected packet (Frame 2) showing the ICMP Echo (ping) request. The hex data is as follows:

```
0000 1e f1 a0 9b 10 fb f2 1c 59 c8 53 26 08 00 45 c0 .....Y54..E
0010 00 5e a7 3b 08 00 40 01 00 c0 0a 00 00 02 c0 a8 X:;@.....
0020 01 32 0b 00 24 ee 00 00 00 00 45 00 00 3c a1 24 2:;...E<.<$
0030 00 00 01 11 93 08 c0 a8 01 32 c0 a8 03 02 02 7f .....2.....
0040 82 a0 00 20 85 be 40 41 42 43 44 45 46 47 48 49 ...(-@A BCDEFGHI
0050 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 JKLMNOPQ RSTUVWXY
0060 5a 5b 5c 5d 5e 5f Z[\]^_
```

Internet Control Message Protocol: Protocol Packets: 40 - Displayed: 18 (45.0%) Profile: Default

R4:

Wireshark interface showing a packet capture on interface *r4-eth0. The filter is set to icmp. The packet list shows 24 ICMP packets. The packet details pane shows the selected packet (Frame 2) with the following information:

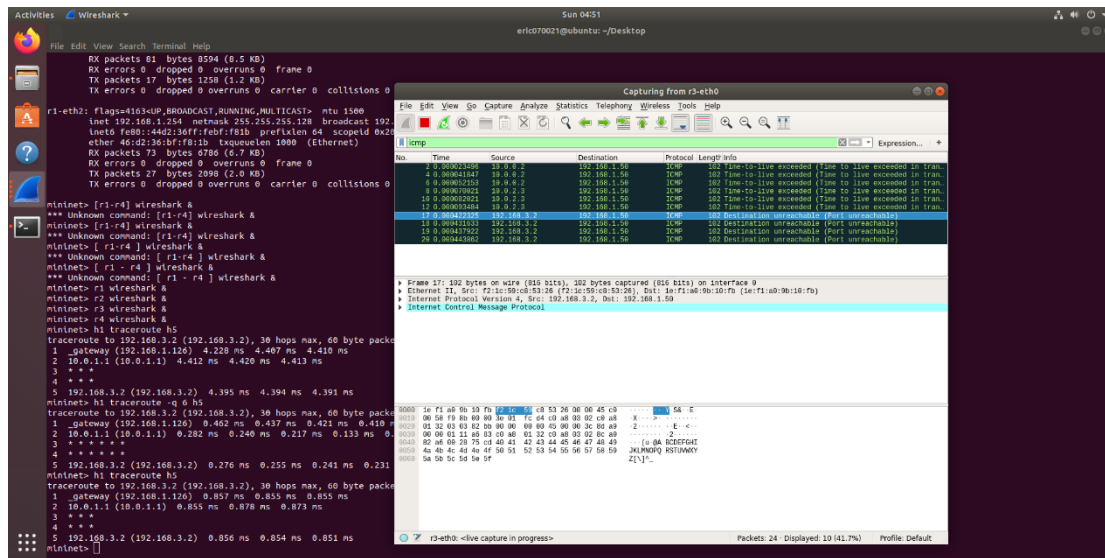
- Frame 2: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface 0
- Ethernet II, Src: 6a:fb:59:c7:51:72 (6a:fb:59:c7:51:72), Dst: 66:10:39:b7:c4:39 (66:10:39:b7:c4:39)
- Internet Protocol Version 4, Src: 10.0.2.3, Dst: 192.168.1.50
- Internet Control Message Protocol

Hex dump of the selected packet (Frame 2) showing the ICMP Echo (ping) request. The hex data is as follows:

```
0000 66 10 39 b7 c4 39 6a fb 59 c7 51 72 08 00 45 c0 f:9:0j YQr..E
0010 00 5e a7 3b 08 00 40 01 00 c0 0a 00 00 02 c0 a8 XI:;@.....
0020 01 32 0b 00 24 ee 00 00 00 00 45 00 00 3c a1 2a 2:;...E<.<.*
0030 00 00 01 11 93 02 c0 a8 01 32 c0 a8 03 02 94 ce .....2.....
0040 82 a0 00 20 85 be 40 41 42 43 44 45 46 47 48 49 ...(-@A BCDEFGHI
0050 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58 59 JKLMNOPQ RSTUVWXY
0060 5a 5b 5c 5d 5e 5f Z[\]^_
```

Internet Control Message Protocol: Protocol Packets: 29 - Displayed: 12 (41.4%) Profile: Default

9.



Traceroute requires a response from the target server and each of the intermediate hops to create its output. If a router doesn't generate a Time-to-leave exceeded response, traceroute will not know anything about that hop. So, the *** means that router didn't response. In the picture, we can see we only have 6 Time-to-leave exceeded response, meaning there are definitely some routers didn't response.