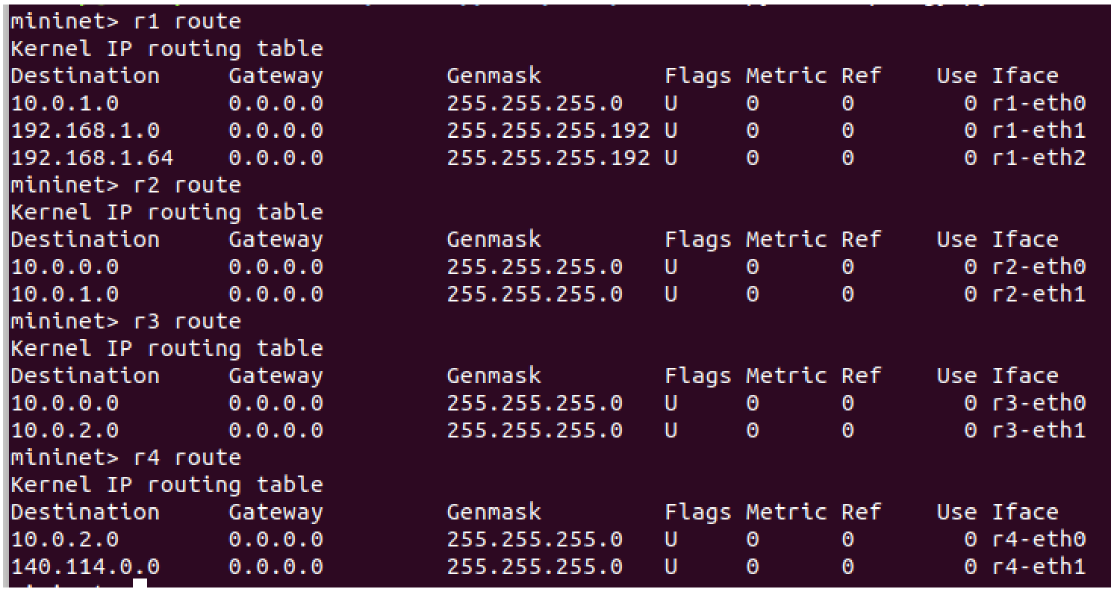
Part1:

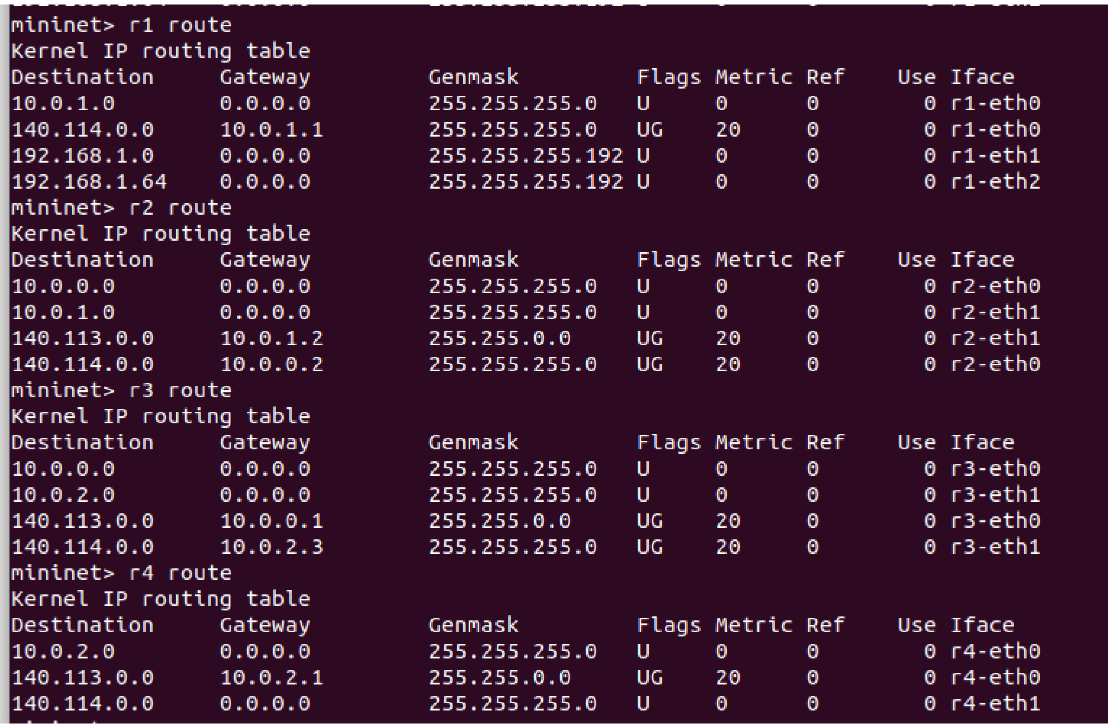
1. Take routing tables screenshot before/after on [r1-r4]

Before:



```
mininet> r1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r1-eth0
192.168.1.0     0.0.0.0         255.255.255.192 U     0      0        0 r1-eth1
192.168.1.64    0.0.0.0         255.255.255.192 U     0      0        0 r1-eth2
mininet> r2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 r2-eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r2-eth1
mininet> r3 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth1
mininet> r4 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 r4-eth0
140.114.0.0     0.0.0.0         255.255.255.0   U     0      0        0 r4-eth1
```

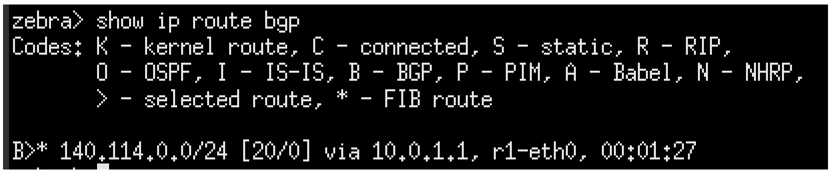
After:



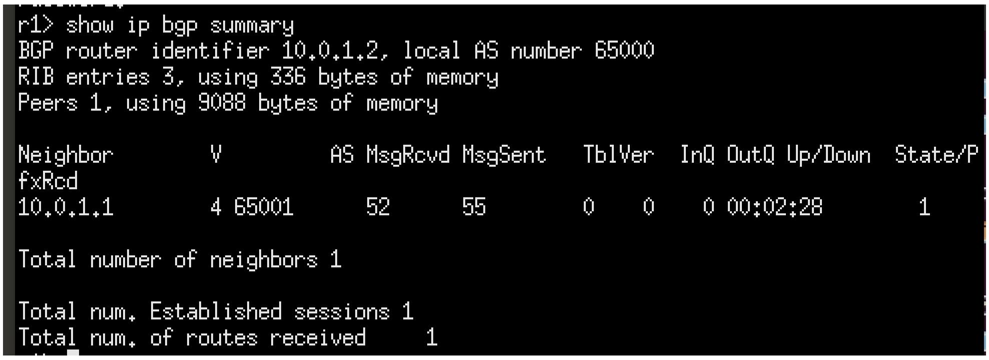
```
mininet> r1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r1-eth0
140.114.0.0     10.0.1.1        255.255.255.0   UG    20     0        0 r1-eth0
192.168.1.0     0.0.0.0         255.255.255.192 U     0      0        0 r1-eth1
192.168.1.64    0.0.0.0         255.255.255.192 U     0      0        0 r1-eth2
mininet> r2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 r2-eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r2-eth1
140.113.0.0     10.0.1.2        255.255.0.0     UG    20     0        0 r2-eth1
140.114.0.0     10.0.0.2        255.255.255.0   UG    20     0        0 r2-eth0
mininet> r3 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth1
140.113.0.0     10.0.0.1        255.255.0.0     UG    20     0        0 r3-eth0
140.114.0.0     10.0.2.3        255.255.255.0   UG    20     0        0 r3-eth1
mininet> r4 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 r4-eth0
140.113.0.0     10.0.2.1        255.255.0.0     UG    20     0        0 r4-eth0
140.114.0.0     0.0.0.0         255.255.255.0   U     0      0        0 r4-eth1
```

2. Telnet zebra and bgpd daemons of [r1-r4] and take screenshots of routes in zebra and bgpd daemons.

r1:

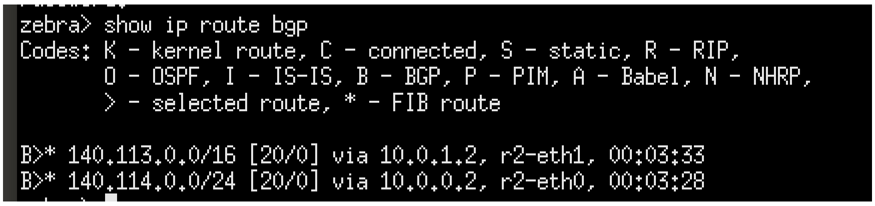


```
zebra> show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

B>* 140.114.0.0/24 [20/0] via 10.0.1.1, r1-eth0, 00:01:27
```

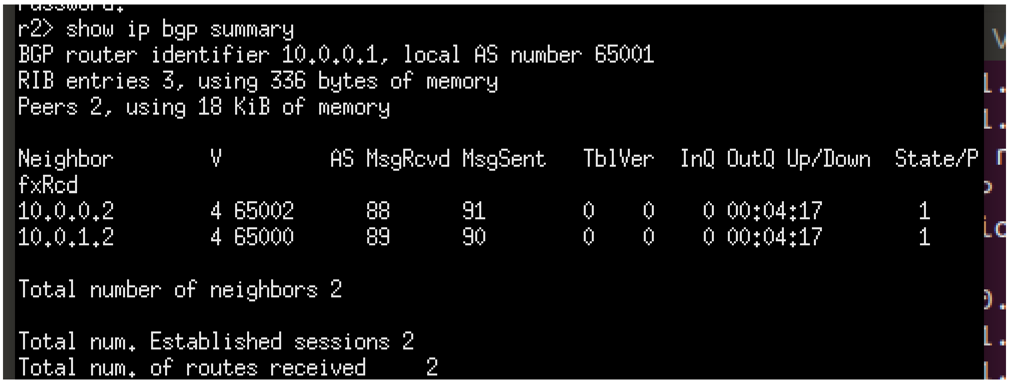


```
r1> show ip bgp summary
BGP router identifier 10.0.1.2, local AS number 65000
RIB entries 3, using 336 bytes of memory
Peers 1, using 9088 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/P
fxRcd
10.0.1.1        4 65001      52      55        0    0    0 00:02:28        1

Total number of neighbors 1

Total num. Established sessions 1
Total num. of routes received     1
```

r2:

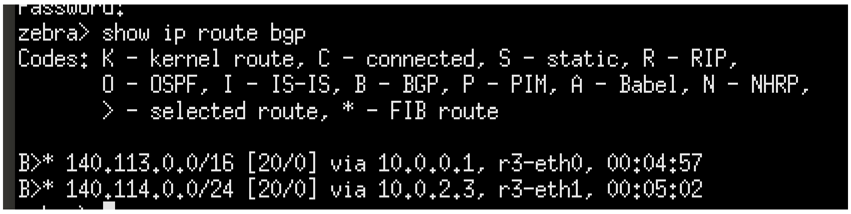


```
zebra> show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

B>* 140.113.0.0/16 [20/0] via 10.0.1.2, r2-eth1, 00:03:33
B>* 140.114.0.0/24 [20/0] via 10.0.0.2, r2-eth0, 00:03:28
```

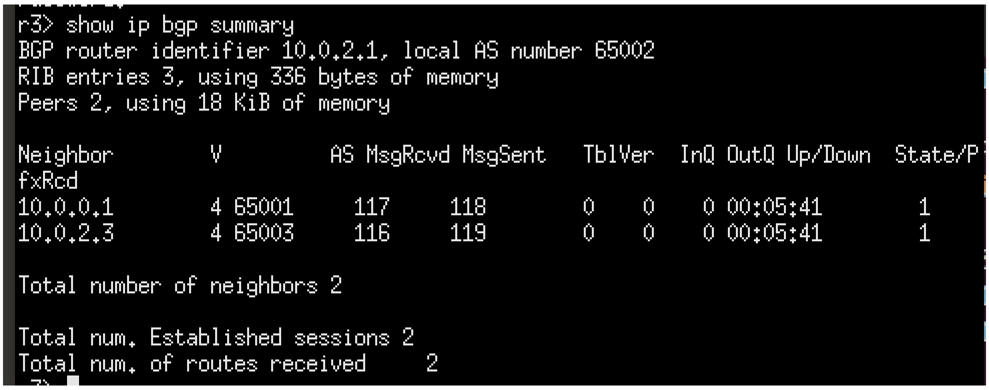


```
r2> show ip bgp summary
BGP router identifier 10.0.0.1, local AS number 65001
RIB entries 3, using 336 bytes of memory
Peers 2, using 18 KiB of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/P
fxRcd
10.0.0.2        4 65002      88      91        0    0    0 00:04:17        1
10.0.1.2        4 65000      89      90        0    0    0 00:04:17        1

Total number of neighbors 2

Total num. Established sessions 2
Total num. of routes received     2
```

r3:

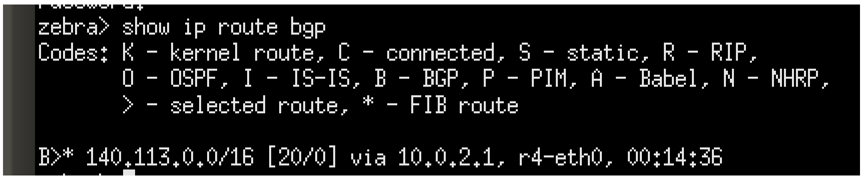


```
zebra> show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

B>* 140.113.0.0/16 [20/0] via 10.0.0.1, r3-eth0, 00:04:57
B>* 140.114.0.0/24 [20/0] via 10.0.2.3, r3-eth1, 00:05:02
```

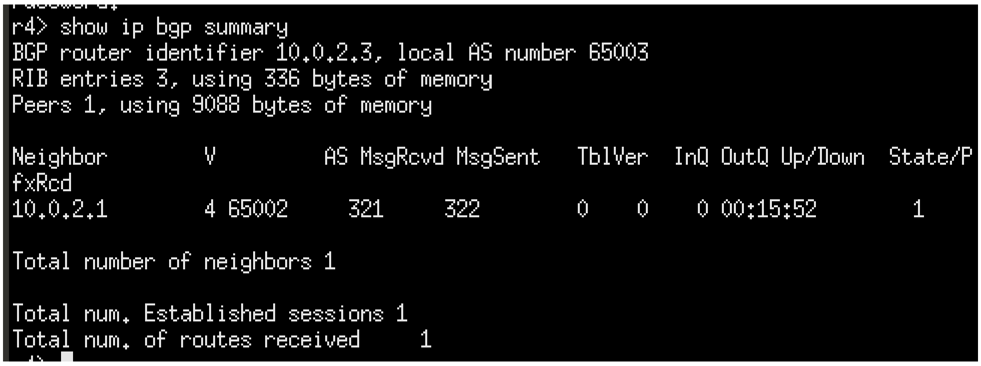


```
r3> show ip bgp summary
BGP router identifier 10.0.2.1, local AS number 65002
RIB entries 3, using 336 bytes of memory
Peers 2, using 18 KiB of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/P
fxRcd
10.0.0.1        4 65001     117     118        0    0    0 00:05:41        1
10.0.2.3        4 65003     116     119        0    0    0 00:05:41        1

Total number of neighbors 2

Total num. Established sessions 2
Total num. of routes received     2
```

r4:



```
zebra> show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel, N - NHRP,
       > - selected route, * - FIB route

B>* 140.113.0.0/16 [20/0] via 10.0.2.1, r4-eth0, 00:14:36
```

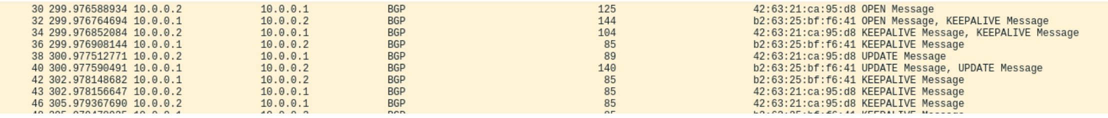


```
r4> show ip bgp summary
BGP router identifier 10.0.2.3, local AS number 65003
RIB entries 3, using 336 bytes of memory
Peers 1, using 9088 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/P
fxRcd
10.0.2.1        4 65002     321     322        0    0    0 00:15:52        1

Total number of neighbors 1

Total num. Established sessions 1
Total num. of routes received     1
```

3. Capture BGP packets from wireshark and take screenshot to verify your answer for the following questions

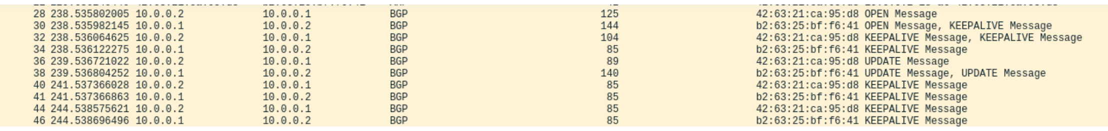
3-1. Show BGP packets (OPEN, UPDATE, KEEP ALIVE) exchanged by r2 and r3

r2-eth0:



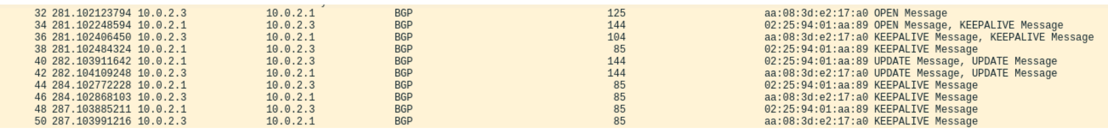
```
30 299.976588934 10.0.0.2     10.0.0.1     BGP     125     42:63:21:ca:95:d8 OPEN Message
32 299.976764694 10.0.0.1     10.0.0.2     BGP     144     b2:63:25:bf:f6:41 OPEN Message, KEEPALIVE Message
34 299.976852084 10.0.0.2     10.0.0.1     BGP     104     42:63:21:ca:95:d8 KEEPALIVE Message, KEEPALIVE Message
36 299.976908144 10.0.0.1     10.0.0.2     BGP      85     b2:63:25:bf:f6:41 KEEPALIVE Message
38 300.977512771 10.0.0.2     10.0.0.1     BGP      89     42:63:21:ca:95:d8 UPDATE Message
40 300.977590491 10.0.0.1     10.0.0.2     BGP     140     b2:63:25:bf:f6:41 UPDATE Message, UPDATE Message
42 302.978148682 10.0.0.1     10.0.0.2     BGP      85     b2:63:25:bf:f6:41 KEEPALIVE Message
43 302.978156647 10.0.0.2     10.0.0.1     BGP      85     42:63:21:ca:95:d8 KEEPALIVE Message
46 305.979367690 10.0.0.2     10.0.0.1     BGP      85     42:63:21:ca:95:d8 KEEPALIVE Message
```

r3-eth0:



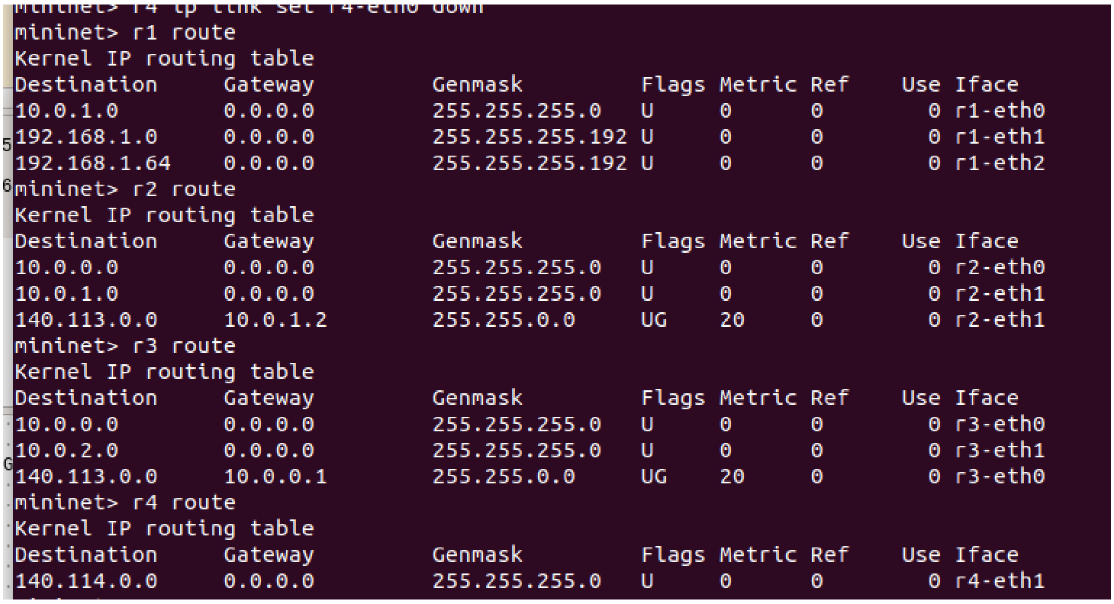
```
28 238.535802005 10.0.0.2     10.0.0.1     BGP     125     42:63:21:ca:95:d8 OPEN Message
30 238.535982145 10.0.0.1     10.0.0.2     BGP     144     b2:63:25:bf:f6:41 OPEN Message, KEEPALIVE Message
32 238.536064625 10.0.0.2     10.0.0.1     BGP     104     42:63:21:ca:95:d8 KEEPALIVE Message, KEEPALIVE Message
34 238.536122275 10.0.0.1     10.0.0.2     BGP      85     b2:63:25:bf:f6:41 KEEPALIVE Message
36 239.536721022 10.0.0.2     10.0.0.1     BGP      89     42:63:21:ca:95:d8 UPDATE Message
38 239.536804252 10.0.0.1     10.0.0.2     BGP     140     b2:63:25:bf:f6:41 UPDATE Message, UPDATE Message
40 241.537366028 10.0.0.2     10.0.0.1     BGP      85     42:63:21:ca:95:d8 KEEPALIVE Message
41 241.537366863 10.0.0.1     10.0.0.2     BGP      85     b2:63:25:bf:f6:41 KEEPALIVE Message
44 244.538575621 10.0.0.2     10.0.0.1     BGP      85     42:63:21:ca:95:d8 KEEPALIVE Message
46 244.538696496 10.0.0.1     10.0.0.2     BGP      85     b2:63:25:bf:f6:41 KEEPALIVE Message
```

r3-eth1:



```
32 281.102123794 10.0.2.3     10.0.2.1     BGP     125     aa:08:3d:e2:17:a0 OPEN Message
34 281.102248594 10.0.2.1     10.0.2.3     BGP     144     02:25:94:01:aa:89 OPEN Message, KEEPALIVE Message
36 281.102406450 10.0.2.3     10.0.2.1     BGP     104     aa:08:3d:e2:17:a0 KEEPALIVE Message, KEEPALIVE Message
38 281.102484324 10.0.2.1     10.0.2.3     BGP      85     02:25:94:01:aa:89 KEEPALIVE Message
40 282.103911642 10.0.2.1     10.0.2.3     BGP     144     02:25:94:01:aa:89 UPDATE Message, UPDATE Message
42 282.104109248 10.0.2.3     10.0.2.1     BGP     144     aa:08:3d:e2:17:a0 UPDATE Message, UPDATE Message
44 284.102772228 10.0.2.1     10.0.2.3     BGP      85     02:25:94:01:aa:89 KEEPALIVE Message
46 284.102868103 10.0.2.3     10.0.2.1     BGP      85     aa:08:3d:e2:17:a0 KEEPALIVE Message
48 287.103885211 10.0.2.1     10.0.2.3     BGP      85     02:25:94:01:aa:89 KEEPALIVE Message
50 287.103991216 10.0.2.3     10.0.2.1     BGP      85     aa:08:3d:e2:17:a0 KEEPALIVE Message
```

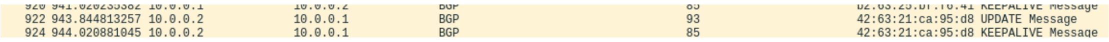
3-2. What will happen to the routing table if you set r4-eth0 down?



```
mininet> r4 ip link set r4-eth0 down
mininet> r1 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r1-eth0
192.168.1.0     0.0.0.0         255.255.255.192 U     0      0        0 r1-eth1
192.168.1.64    0.0.0.0         255.255.255.192 U     0      0        0 r1-eth2
mininet> r2 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 r2-eth0
10.0.1.0        0.0.0.0         255.255.255.0   U     0      0        0 r2-eth1
140.113.0.0     10.0.1.2        255.255.0.0     UG    20     0        0 r2-eth1
mininet> r3 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth0
10.0.2.0        0.0.0.0         255.255.255.0   U     0      0        0 r3-eth1
140.113.0.0     10.0.0.1        255.255.0.0     UG    20     0        0 r3-eth0
mininet> r4 route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
140.114.0.0     0.0.0.0         255.255.255.0   U     0      0        0 r4-eth1
```

Routing table 會自動移除 r4 底下的全部 subnet

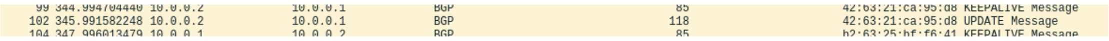
3-3. How does r3 know r4 is unreachable? Explain how



```
920 941.020235382 10.0.0.1     10.0.0.2     BGP     85     b2:63:25:bf:f6:41 KEEPALIVE Message
922 943.844813257 10.0.0.2     10.0.0.1     BGP     93     42:63:21:ca:95:d8 UPDATE Message
924 944.020881045 10.0.0.2     10.0.0.1     BGP     85     42:63:21:ca:95:d8 KEEPALIVE Message
```

r3-eth1 會與 r4 終止交換訊息，r3-eth0 則會與 r2 交換 update message

3-4. How does r2 know r4 is unreachable? Explain how

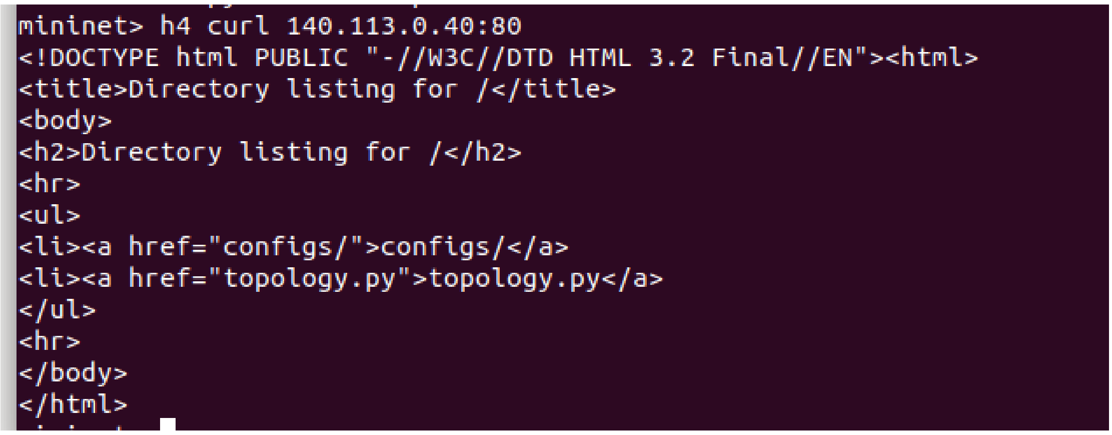


```
 99 344.994704440 10.0.0.2     10.0.0.1     BGP     85     42:63:21:ca:95:d8 KEEPALIVE Message
102 345.991582248 10.0.0.2     10.0.0.1     BGP     118    42:63:21:ca:95:d8 UPDATE Message
104 347.996013479 10.0.0.1     10.0.0.2     BGP     85     b2:63:25:bf:f6:41 KEEPALIVE Message
```

r2 與 r3 交換 update message 之後，便可得知 r4 的狀態

Part2:

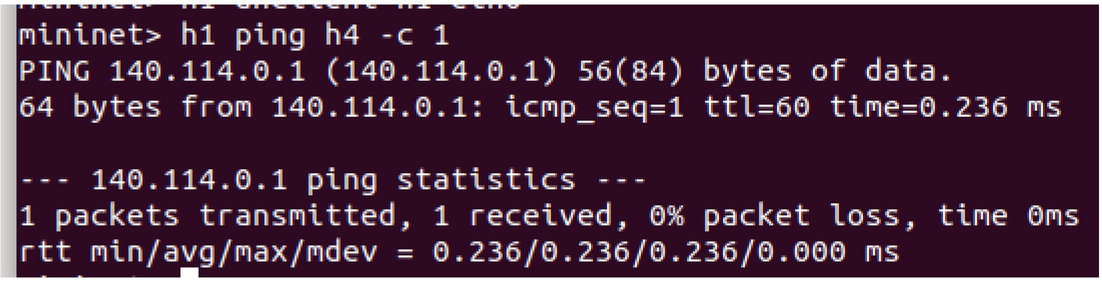
1. Take screenshot of curl result



```
mininet> h4 curl 140.113.0.40:80
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href="configs/">configs/</a>
<li><a href="topology.py">topology.py</a>
</ul>
<hr>
</body>
</html>
mininet> 
```

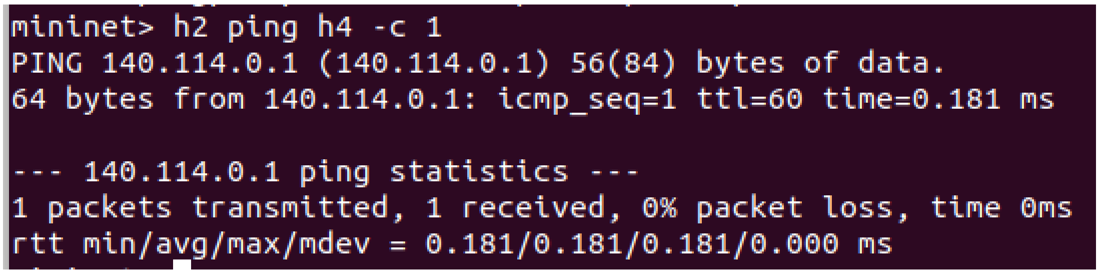
2. Check reachability and take screenshot

h1:



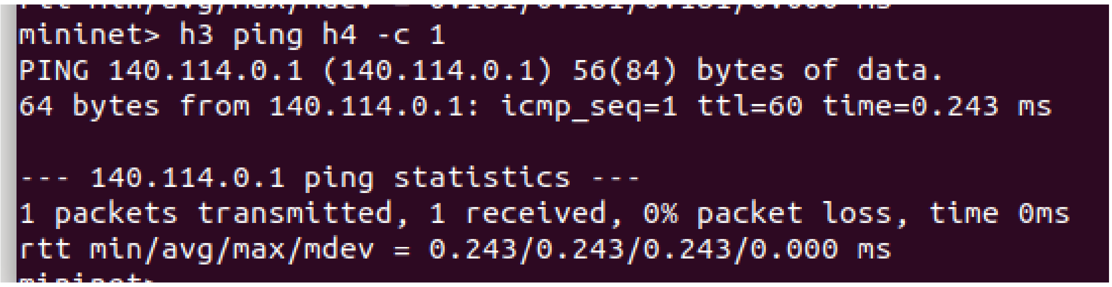
```
mininet> h1 ping h4 -c 1
PING 140.114.0.1 (140.114.0.1) 56(84) bytes of data.
64 bytes from 140.114.0.1: icmp_seq=1 ttl=60 time=0.236 ms

--- 140.114.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.236/0.236/0.236/0.000 ms
```

h2:



```
mininet> h2 ping h4 -c 1
PING 140.114.0.1 (140.114.0.1) 56(84) bytes of data.
64 bytes from 140.114.0.1: icmp_seq=1 ttl=60 time=0.181 ms

--- 140.114.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.181/0.181/0.181/0.000 ms
```

h3:



```
mininet> h3 ping h4 -c 1
PING 140.114.0.1 (140.114.0.1) 56(84) bytes of data.
64 bytes from 140.114.0.1: icmp_seq=1 ttl=60 time=0.243 ms

--- 140.114.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.243/0.243/0.243/0.000 ms
```

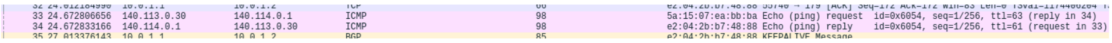
由上面三張圖可觀察到，h1,h2,h3 皆可 ping 到 h4，同時表示 NAT 正常運行

3. Run wireshark on r1 to take screenshot of input/output packet

- Explain the difference of packet headers

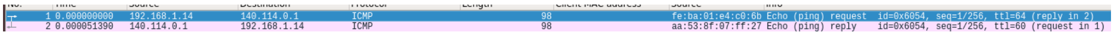
h1 ping h4:

r1-eth0:



```
32 24.012184990 10.0.1.1       10.0.1.2       TCP   66  e2:04:2b:b7:48:88 55740 → 179 [ACK] Seq=172 Ack=172 Win=83 Len=0 TSval=1174406204 T
33 24.672806656 140.113.0.30   140.114.0.1    ICMP  98  5a:15:07:ea:bb:ba Echo (ping) request  id=0x6054, seq=1/256, ttl=63 (reply in 34)
34 24.672833166 140.114.0.1    140.113.0.30   ICMP  98  e2:04:2b:b7:48:88 Echo (ping) reply    id=0x6054, seq=1/256, ttl=61 (request in 33)
35 27.013376143 10.0.1.1       10.0.1.2       BGP   85  e2:04:2b:b7:48:88 KEEPALIVE Message
```

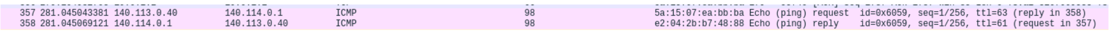
r1-eth1:



```
No. Time        Source         Destination    Protocol Length Client MAC address Source           Info
1 0.000000000   192.168.1.14   140.114.0.1    ICMP     98     fe:ba:01:e4:c0:6b Echo (ping) request  id=0x6054, seq=1/256, ttl=64 (reply in 2)
2 0.000051390   140.114.0.1    192.168.1.14   ICMP     98     aa:53:8f:07:ff:27 Echo (ping) reply    id=0x6054, seq=1/256, ttl=60 (request in 1)
```

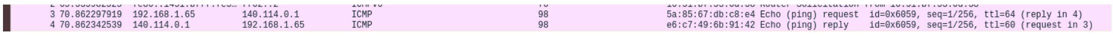
h2 ping h4:

r1-eth0:



```
357 281.045043381 140.113.0.40   140.114.0.1    ICMP  98  5a:15:07:ea:bb:ba Echo (ping) request  id=0x6059, seq=1/256, ttl=63 (reply in 358)
358 281.045069121 140.114.0.1    140.113.0.40   ICMP  98  e2:04:2b:b7:48:88 Echo (ping) reply    id=0x6059, seq=1/256, ttl=61 (request in 357)
```

r1-eth2:



```
3 70.862297919  192.168.1.65   140.114.0.1    ICMP  98  5a:85:67:db:c8:e4 Echo (ping) request  id=0x6059, seq=1/256, ttl=64 (reply in 4)
4 70.862342539  140.114.0.1    192.168.1.65   ICMP  98  e6:c7:49:6b:91:42 Echo (ping) reply    id=0x6059, seq=1/256, ttl=60 (request in 3)
```

由 h1 送出的封包，source IP 是 192.168.1.14，而經過 NAT 之後，r1 送出到外部網路的封包，source IP 會改為 140.113.0.30。反之，由外部要送到 h1

的封包，destination IP 會是 140.113.0.30，經由 r1 的 NAT 處理後，會再改為 h1 在內網的 IP。同樣地，由 h2 送出或是要送往 h2 的封包，IP 則會以 140.113.0.40 顯示。