1. Take routing tables screenshot before/after on [r1-r4] (10%)

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        255.255.0.0     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        255.255.0.0     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        255.255.0.0     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. (10%)

R1

Zebra

```
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, 00:01:42
```

Bgpd

```
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      78      81        0    0    0 00:03:47        1

Total number of neighbors 1

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

R2

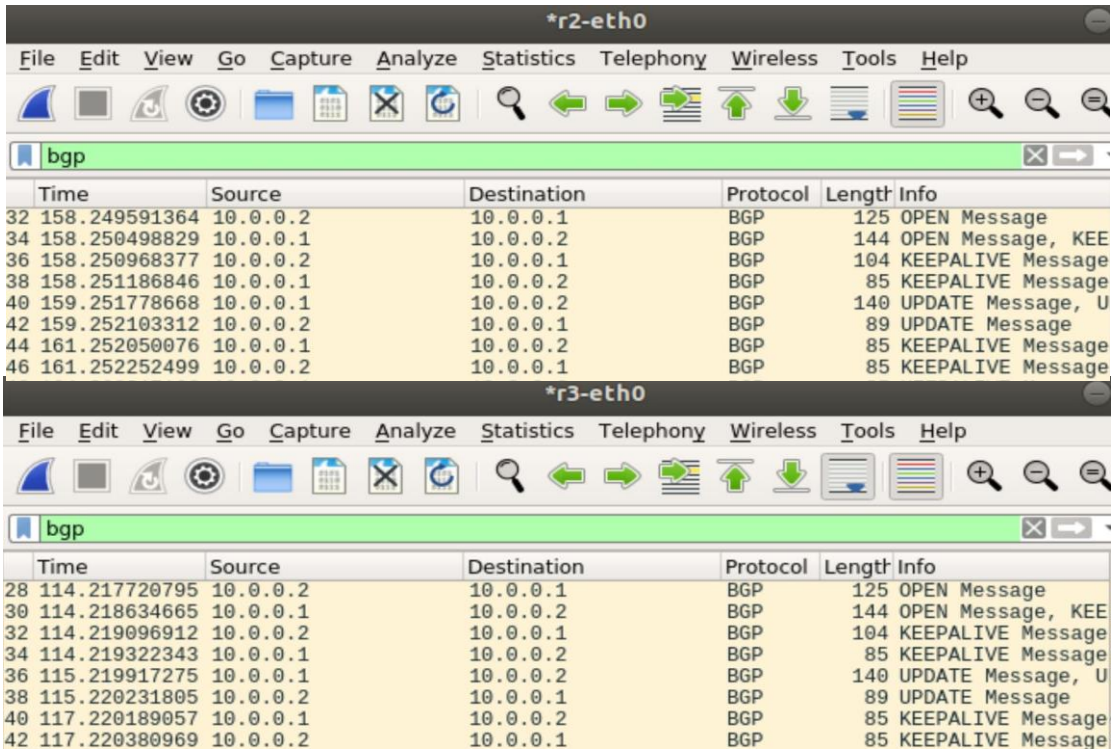
Zebra

```
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, 00:27:28
B>* 140.114.0.0/24 [20/0] via 10.0.0.2, r2-eth0, 00:27:23
```

	Bgpd	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 586 589 0 0 0 00:29:11 1 10.0.1.2 4 65000 587 588 0 0 0 00:29:11 1 Total number of neighbors 2 Total num. Established sessions 2 Total num. of routes received 2
R3	Zebra	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, 00:00:17 B>* 140.114.0.0/24 [20/0] via 10.0.2.3, r3-eth1, 00:00:22
	Bgpd	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 22 23 0 0 0 00:00:55 1 10.0.2.3 4 65003 21 24 0 0 0 00:00:55 1 Total number of neighbors 2 Total num. Established sessions 2 Total num. of routes received 2
R4	Zebra	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, 00:01:55
	Bgpd	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 66 67 0 0 0 00:03:09 1 Total number of neighbors 1 Total num. Established sessions 1 Total num. of routes received 1

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



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

screenshot

```
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
15:46:18.430     Main Warn QStandardPaths: XDG_RUNTIME_DIR not set, defaulting
to '/tmp/runtime-root'
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
15:49:23.917     Main Warn QStandardPaths: XDG_RUNTIME_DIR not set, defaulting
to '/tmp/runtime-root'
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
```

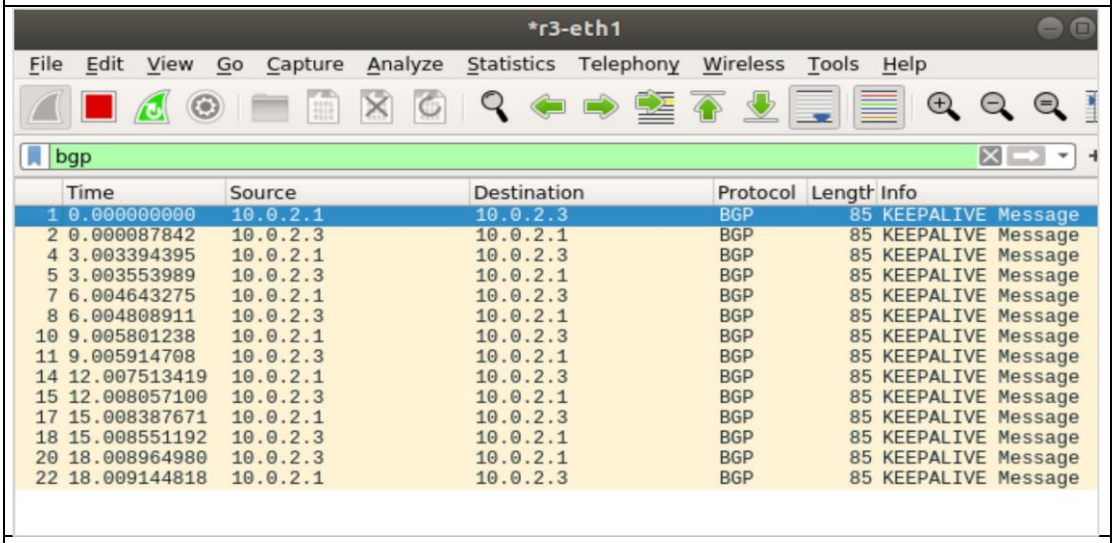
Explain

Routers will remove the prefixes previously learn over the BGP session and reroute over alternate paths.

Thus, routes to network under r4 (140.114.0.0/16) disappeared on r1~r3, and the routes to outside-network were also removed from the table of r4.

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

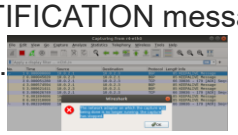
screenshot



	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message
2	0.000087842	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
4	3.003394395	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message
5	3.003553989	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
7	6.004643275	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message
8	6.004808911	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
10	9.005801238	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message
11	9.005914708	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
14	12.007513419	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message
15	12.008057100	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
17	15.008387671	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message
18	15.008551192	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
20	18.008964980	10.0.2.3	10.0.2.1	BGP	85	KEEPALIVE Message
22	18.009144818	10.0.2.1	10.0.2.3	BGP	85	KEEPALIVE Message

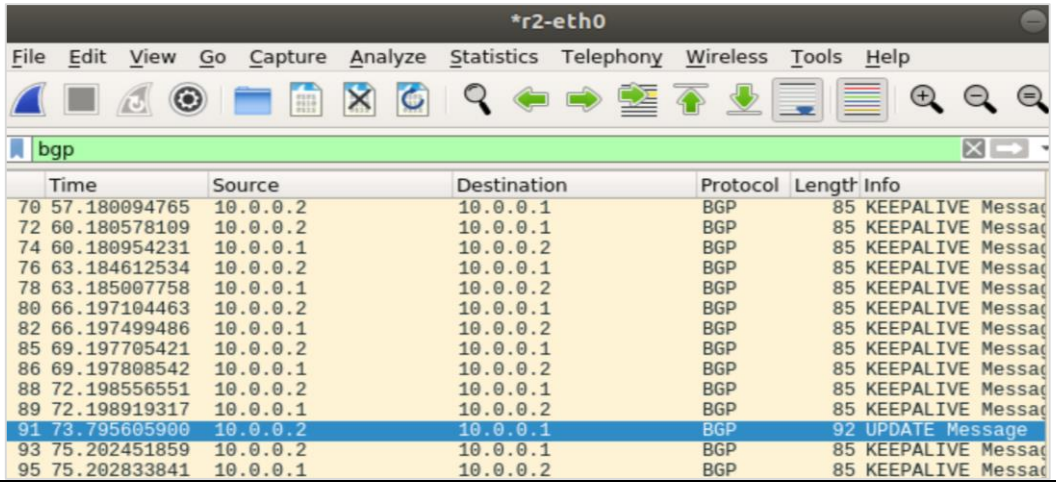
Explain

If r3 doesn't receive any KEEPALIVE from r4 till the hold time expired (set by r3.config), it will decide r4 to be dead, sending a NOTIFICATION message with cease code (I didn't catch it but did got an error msg).



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

screenshot



	Time	Source	Destination	Protocol	Length	Info
70	57.180094765	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
72	60.180578109	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
74	60.180954231	10.0.0.1	10.0.0.2	BGP	85	KEEPALIVE Messa
76	63.184612534	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
78	63.185007758	10.0.0.1	10.0.0.2	BGP	85	KEEPALIVE Messa
80	66.197104463	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
82	66.197499486	10.0.0.1	10.0.0.2	BGP	85	KEEPALIVE Messa
85	69.197705421	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
86	69.197808542	10.0.0.1	10.0.0.2	BGP	85	KEEPALIVE Messa
88	72.198556551	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
89	72.198919317	10.0.0.1	10.0.0.2	BGP	85	KEEPALIVE Messa
91	73.795605900	10.0.0.2	10.0.0.1	BGP	92	UPDATE Message
93	75.202451859	10.0.0.2	10.0.0.1	BGP	85	KEEPALIVE Messa
95	75.202833841	10.0.0.1	10.0.0.2	BGP	85	KEEPALIVE Messa

Explain

As r3's table is changed, r2 will receive an UPDATE Message from r3, and therefore withdraw the previously advertised routes (which contained network under r4) and learned the new one.

1. Take screenshot of curl result (10%)

```
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=".git/">.git/</a>
<li><a href=".gitattributes">.gitattributes</a>
<li><a href=".github/">.github/</a>
<li><a href=".gitignore">.gitignore</a>
<li><a href=".pylint">.pylint</a>
<li><a href="bin/">bin/</a>
<li><a href="configs/">configs/</a>
<li><a href="CONTRIBUTORS">CONTRIBUTORS</a>
<li><a href="custom/">custom/</a>
<li><a href="debian/">debian/</a>
<li><a href="dhcpd.conf">dhcpd.conf</a>
<li><a href="doc/">doc/</a>
<li><a href="example/">example/</a>
<li><a href="example.py">example.py</a>
<li><a href="examples/">examples/</a>
<li><a href="INSTALL">INSTALL</a>
<li><a href="lab1/">lab1/</a>
<li><a href="lab2/">lab2/</a>
<li><a href="LICENSE">LICENSE</a>
<li><a href="Makefile">Makefile</a>
<li><a href="mininet/">mininet/</a>
<li><a href="mn.1">mn.1</a>
<li><a href="mnexec">mnexec</a>
<li><a href="mnexec.1">mnexec.1</a>
<li><a href="mnexec.c">mnexec.c</a>
<li><a href="README.md">README.md</a>
<li><a href="setup.py">setup.py</a>
<li><a href="topo.py">topo.py</a>
<li><a href="topology.py">topology.py</a>
<li><a href="util/">util/</a>
</ul>
<hr>
</body>
</html>
```

2. Check reachability and take screenshot (10%)

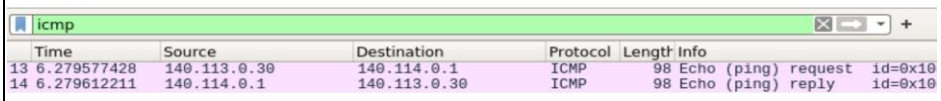
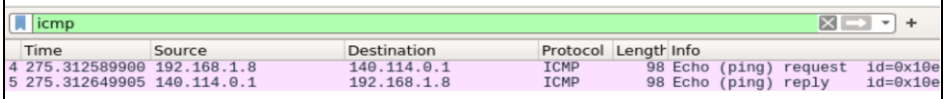
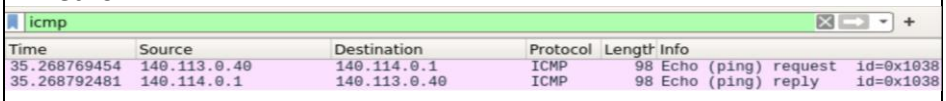
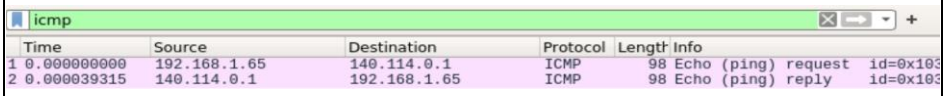
h1 ping h4	`mininet> h1 dhclient h1-eth0`
	`mininet> h1 ping h4 -c 1`
	`PING 140.114.0.1 (140.114.0.1) 56(84) bytes of data.`
	`64 bytes from 140.114.0.1: icmp_seq=1 ttl=60 time=0.396 ms`
	
	`--- 140.114.0.1 ping statistics ---`
	`1 packets transmitted, 1 received, 0% packet loss, time 0ms`
	`rtt min/avg/max/mdev = 0.396/0.396/0.396/0.000 ms`

h2 ping h4	`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=0.292 ms`
	
	`--- 140.114.0.1 ping statistics ---`
	`1 packets transmitted, 1 received, 0% packet loss, time 0ms`
	`rtt min/avg/max/mdev = 0.292/0.292/0.292/0.000 ms`

h3 ping h4	mininet> h3 ping h4 -c 1 Serving HTTP on 0.0.0.0 port 8000 ... 140.114.0.1 - - [28/Mar/2021 20:30:16] "GET / HTTP/1.1" 200 - 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=0.359 ms --- 140.114.0.1 ping statistics --- 1 packets transmitted, 1 received, 0% packet loss, time 0ms rtt min/avg/max/mdev = 0.359/0.359/0.359/0.000 ms

3. Run wireshark on r1 to take screenshot of input/output packet (10%)

■ Explain the difference of packet headers

h1 ping h4	r1-eth0  icmp Time Source Destination Protocol Length Info 13 6.279577428 140.113.0.30 140.114.0.1 ICMP 98 Echo (ping) request id=0x10 14 6.279612211 140.114.0.1 140.113.0.30 ICMP 98 Echo (ping) reply id=0x10
	r1-eth1  icmp Time Source Destination Protocol Length Info 4 275.312589900 192.168.1.8 140.114.0.1 ICMP 98 Echo (ping) request id=0x10e 5 275.312649905 140.114.0.1 192.168.1.8 ICMP 98 Echo (ping) reply id=0x10e
	Explain : The packet did SNAT to replace the source ip (gained from dhcp) from internal ip to public ip when it leaved eth2. Eth0 then passed out the packet using that public IP.
h2 ping h4	r1-eth0  icmp Time Source Destination Protocol Length Info 35.268769454 140.113.0.40 140.114.0.1 ICMP 98 Echo (ping) request id=0x1038 35.268792481 140.114.0.1 140.113.0.40 ICMP 98 Echo (ping) reply id=0x1038
	r1-eth2  icmp Time Source Destination Protocol Length Info 1 0.000000000 192.168.1.65 140.114.0.1 ICMP 98 Echo (ping) request id=0x103 2 0.000039315 140.114.0.1 192.168.1.65 ICMP 98 Echo (ping) reply id=0x103
	Explain : The packet did SNAT on POSTROUTING, so the source ip was modified as the packet leaved eth1. Eth1 then passed out the packet using that modified ip.