CMPT471 Networking II

Assignment 4 Low-level Socket Programming

Student Name: **Rui Zheng**

Student Number: **301200960**

Student Email: **rza31@sfu.ca**

Professor: **Lou Hafer**

TA: **Ehsan Tavakoli**

USAGE

I assume that the code will be executed on August. The code will send an Echo request to Year by default.

However, you can run the code on any host in CS-VNL. The execution can be asked to pretend to be arbitrary host and send Echo Request to any legal IPv4 address. But if you fail to assign the argument properly, you are unable to receive the Echo Reply message.

Follow the terminal command below to compile&execute the code(obtain the root privilege first please):

```
g++ -o echo a4skel.cpp
./echo
```

Usage: echo [src_IPv4_addr dst_IPv4_addr [src_Ether_addr dst_Ether_addr]]

```
echo
```

- send Ethernet Frame containing echo request from 172.17.1.8@august to 172.19.1.18@year(hard coding)

```
echo src_IPv4_address dst_IPv4_address
```

- send IP packet from src_IPv4_addr to dst_IPv4_addr. Only IP packet is crafted since no enough link-layer info

```
echo src_IPv4_addr dst_IPv4_addr src_Ether_addr dst_Ether_addr
```

- send Ethernet Frame from src_Ether_addr to dst_Ether_addr and the frame contains a crafted packet with src_IPv4_addr and dst_IPv4_addr

Warning: In the 3rd usage, the value of last argument should be the MAC address of next-hop. In other words, if the frame intends to pass through some routers, the value should be the MAC address of the interface directly connected to the host rather than that of the interface attached under terminal receiver.

CODE DESIGN

The skeleton code given deals with most of the annoying issues such as how to start a socket, how to configure the socket addresses and how to send out prepared packet at the end of the code. So we just need to focus on constructing the packet only. Many thanks to Lou and Ehsan.

I decide to craft the data, ICMP header, IP header and Ethernet header in turn and separately. Then I copy those part into the correct position into the buffer(using pointer arithmetic operation).

ICMP Header

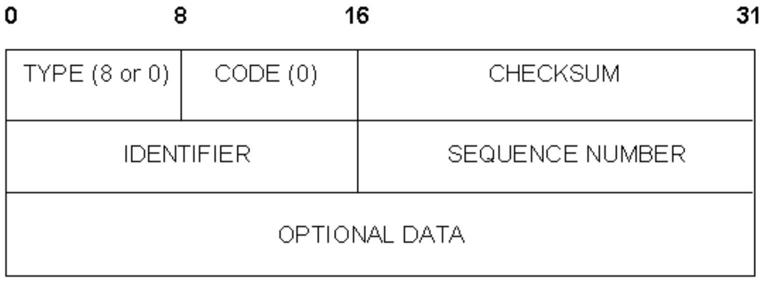


Fig. 5.2: ICMP echo request/reply message format.

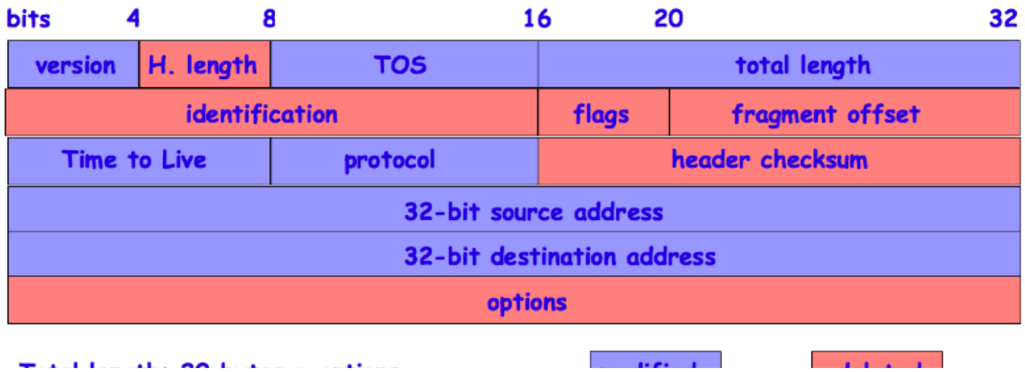
Fig. 5.2 is a diagram of ICMP echo message[1]. The fields mentioned in the diagram is necessary to ICMP echo message. These fields are where I have to manually assign some value. Here is an ideal crafted ICMP header at this stage:

8	0	0(Calculated later)
Auto-configured		Auto-configured
nil		

[1] zap2sandhu, "Ping", http://zap2sandhu.tripod.com/ping/ping.htm#5

IP Header

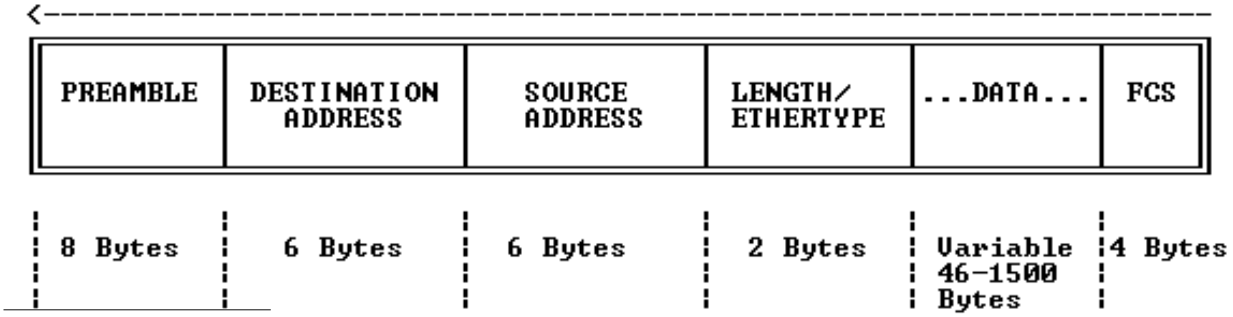
The IP header is a little bit more complicated. We have quite a lot of fields need to be configured . Here is an ideal filling up IP header[2]:



4(IPv4)	5(20bytes)	0	20+8+50(IP header+ICMP header +DataLen)	
htons(0x777)(arbitrary #)			0x4000	0(do not frag)
255(any #>4)		1(ICMP)	0(calculate later)	
172.17.1.8 or user given address				
172.19.1.18 or given address				

Ethernet Header

For Ethernet Header, we only need to specify the source and destination address as well as the type of Ethernet.[3]



[2] Jen Linkova, "IPv4 Header Format vs. IPv6 (IPv6: What, Why, How - Slide", http://www.openwall.com/presentations/IPv6/img18.html

[3] Tampa Bay Interactive, Inc. "Ethernet Frame Types", http://telecom.tbi.net/frmlan.html

AutoConfig	00:50:56:a4:05:33 or given MAC address	00:50:56:a4:0b:bb or given MAC address	htons(ETHERTYPE_IP)	…	AutoCalculated

Filling up Buffer

Next, I fill up the buffer in the order shown below. In the meanwhile, I keep tracks of the offset of each component.

Etherner Header	IP Header	ICMP Header	Data	FCS	Unused part

Ethernet Header Offset=frame;

IP Header Offset=Ethernet Header Offset+ sizeof(ether_header)

ICMP Header Offset=IP Header Offset+ sizeof(ip_header)

Data Offset=ICMP Header Offset+sizeof(icmphdr)

Calculate Checksum

We can use the offset we record above to located the address of IP header checksum and ICMP header checksum.

IP header checksum is calculated according to IP header only. In contrast, the ICMP header checksum need to count the data part in as well.

As a result,

```
        ipHeader.ip_sum=calcsum((unsigned short*)&ipHeader,sizeof(struct ip));
        icmpPtr->checksum=calcsum((unsigned short*)icmpOffset,sizeof(icmphdr)+DATALEN);
```

Total Length of Frame

We want to find out the total length of frame so that we can only send necessary part of buffer. Also, we can check if the frame is too large to the buffer reversely. (usually MTU=1500, len(frame)=1000).

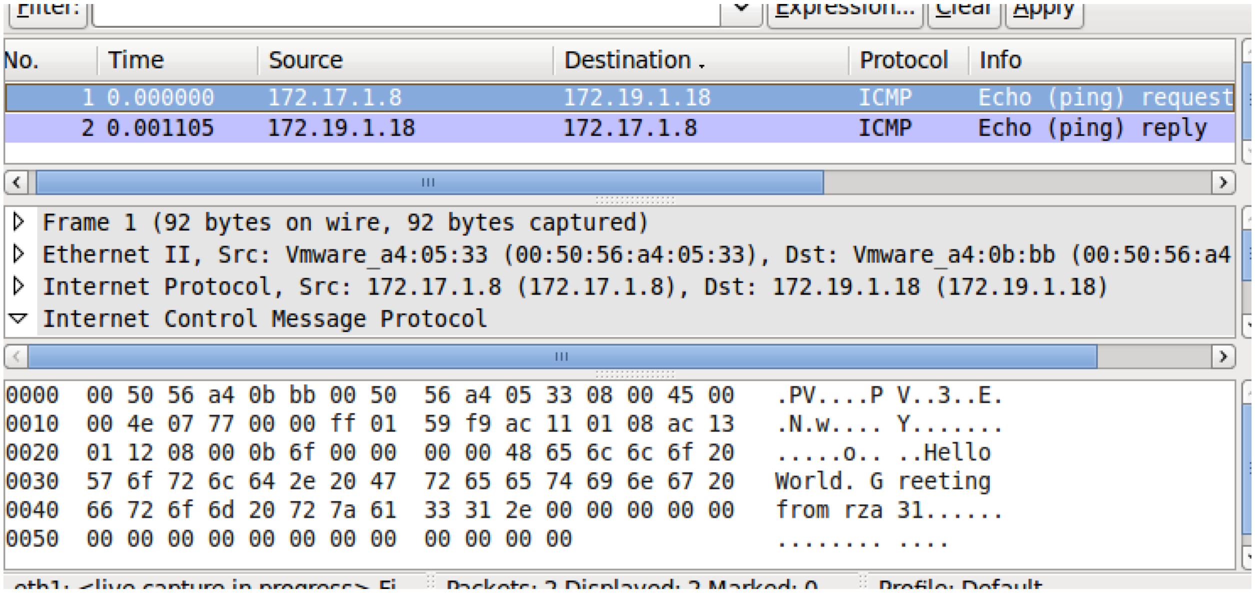
frameLen=sizeof(ether_header)+sizeof(ip)+sizeof(icmphdr)+DATALEN

CAPTURE

Case 1 Sending echo request from August to Year @ August (By default)

From 172.17.1.8
To 172.19.1.18
srcMAC:00:50:56:a4:05:33
dstMAC:00:50:56:a4:0b:bb

Result: message is delivered to January and receives Echo Reply successfully.

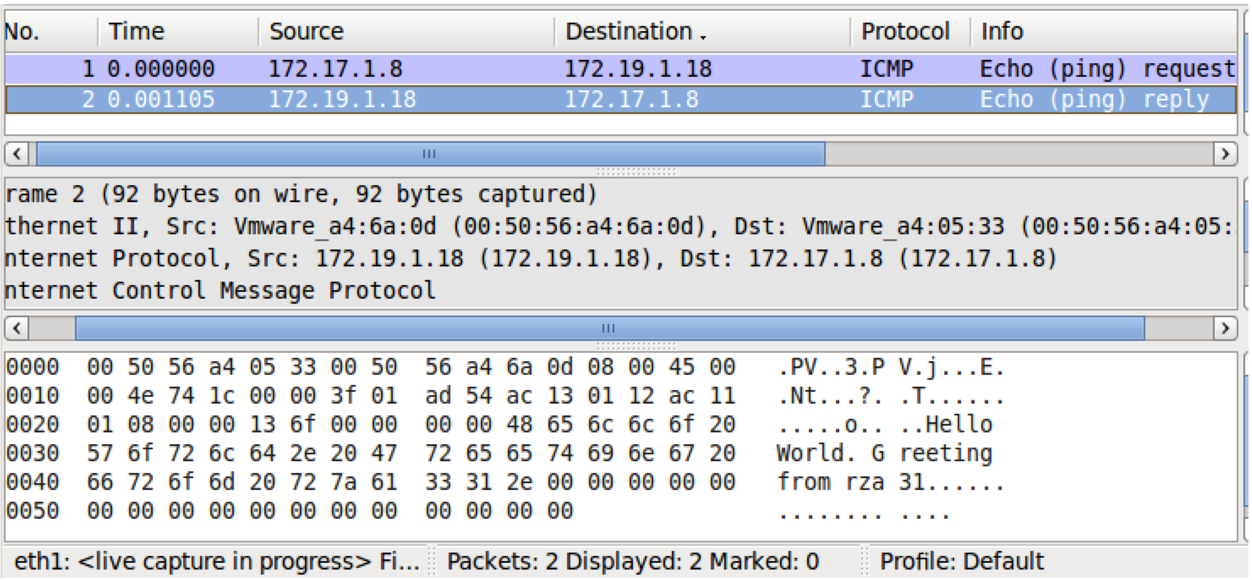


```
No.   Time       Source        Destination   Protocol  Info
    1 0.000000   172.17.1.8    172.19.1.18   ICMP      Echo (ping) request
    2 0.001105   172.19.1.18   172.17.1.8    ICMP      Echo (ping) reply
```

```
▷ Frame 1 (92 bytes on wire, 92 bytes captured)
▷ Ethernet II, Src: Vmware_a4:05:33 (00:50:56:a4:05:33), Dst: Vmware_a4:0b:bb (00:50:56:a4
▷ Internet Protocol, Src: 172.17.1.8 (172.17.1.8), Dst: 172.19.1.18 (172.19.1.18)
▽ Internet Control Message Protocol
```

```
0000  00 50 56 a4 0b bb 00 50  56 a4 05 33 08 00 45 00   .PV....P V..3..E.
0010  00 4e 07 77 00 00 ff 01  59 f9 ac 11 01 08 ac 13   .N.w.... Y.......
0020  01 12 08 00 0b 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi... Packets: 2 Displayed: 2 Marked: 0 Profile: Default



```
No.   Time       Source        Destination   Protocol  Info
    1 0.000000   172.17.1.8    172.19.1.18   ICMP      Echo (ping) request
    2 0.001105   172.19.1.18   172.17.1.8    ICMP      Echo (ping) reply
```

```
rame 2 (92 bytes on wire, 92 bytes captured)
thernet II, Src: Vmware_a4:6a:0d (00:50:56:a4:6a:0d), Dst: Vmware_a4:05:33 (00:50:56:a4:05:
nternet Protocol, Src: 172.19.1.18 (172.19.1.18), Dst: 172.17.1.8 (172.17.1.8)
nternet Control Message Protocol
```

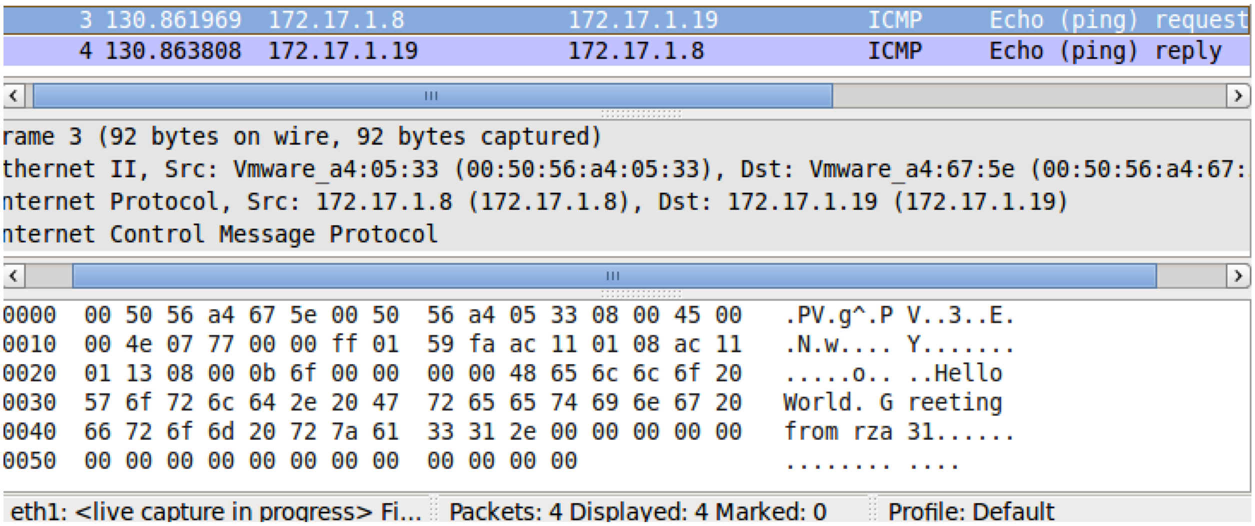
```
0000  00 50 56 a4 05 33 00 50  56 a4 6a 0d 08 00 45 00   .PV..3.P V.j...E.
0010  00 4e 74 1c 00 00 3f 01  ad 54 ac 13 01 12 ac 11   .Nt...?. .T......
0020  01 08 00 00 13 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi... Packets: 2 Displayed: 2 Marked: 0 Profile: Default

Case2 send Echo Request to directly connected host and Ethernet frame is configured perfectly.

From 172.17.1.8
To 172.17.1.19
srcMAC:00:50:56:a4:05:33
dstMAC:00:50:56:a4:67:5e

Result: Echo Reply received.

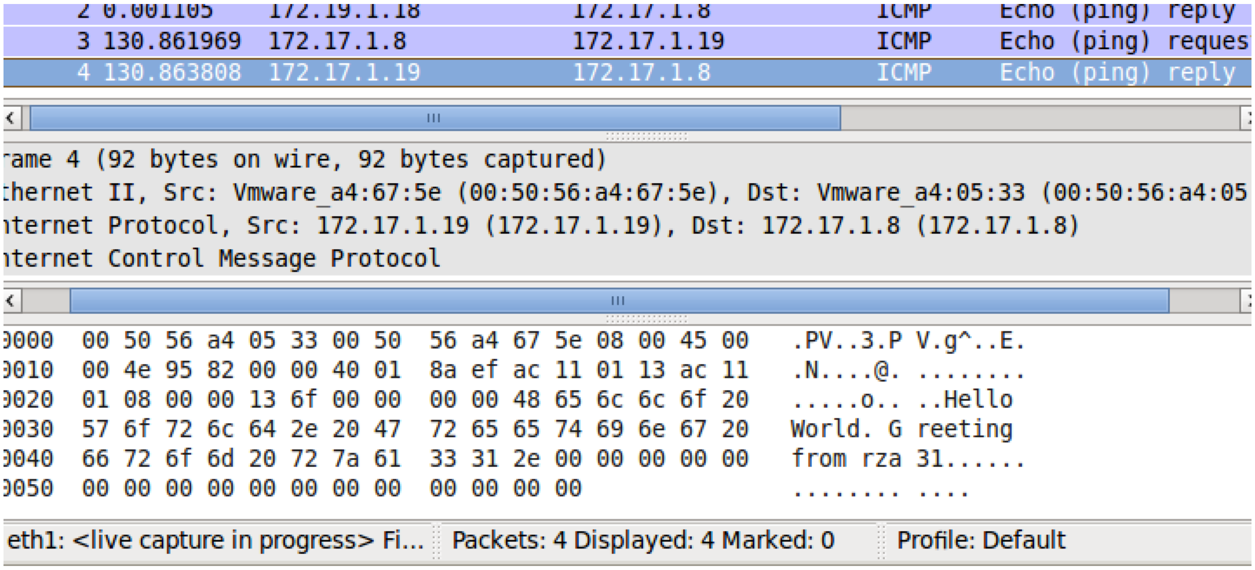


```
  3 130.861969  172.17.1.8           172.17.1.19          ICMP     Echo (ping) request
  4 130.863808  172.17.1.19          172.17.1.8           ICMP     Echo (ping) reply
```

```
rame 3 (92 bytes on wire, 92 bytes captured)
thernet II, Src: Vmware_a4:05:33 (00:50:56:a4:05:33), Dst: Vmware_a4:67:5e (00:50:56:a4:67:
nternet Protocol, Src: 172.17.1.8 (172.17.1.8), Dst: 172.17.1.19 (172.17.1.19)
nternet Control Message Protocol
```

```
0000  00 50 56 a4 67 5e 00 50  56 a4 05 33 08 00 45 00   .PV.g^.P V..3..E.
0010  00 4e 07 77 00 00 ff 01  59 fa ac 11 01 08 ac 11   .N.w.... Y.......
0020  01 13 08 00 0b 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi… Packets: 4 Displayed: 4 Marked: 0 Profile: Default



```
  2 0.001105    172.19.1.18          172.17.1.8           ICMP     Echo (ping) reply
  3 130.861969  172.17.1.8           172.17.1.19          ICMP     Echo (ping) reques
  4 130.863808  172.17.1.19          172.17.1.8           ICMP     Echo (ping) reply
```

```
rame 4 (92 bytes on wire, 92 bytes captured)
thernet II, Src: Vmware_a4:67:5e (00:50:56:a4:67:5e), Dst: Vmware_a4:05:33 (00:50:56:a4:05
nternet Protocol, Src: 172.17.1.19 (172.17.1.19), Dst: 172.17.1.8 (172.17.1.8)
nternet Control Message Protocol
```

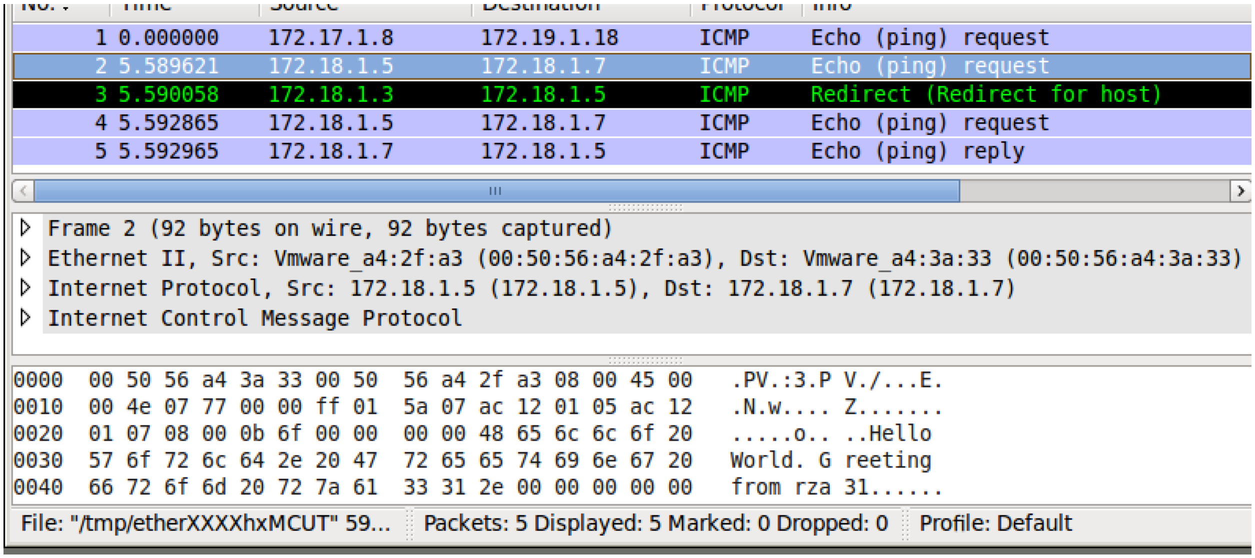
```
0000  00 50 56 a4 05 33 00 50  56 a4 67 5e 08 00 45 00   .PV..3.P V.g^..E.
0010  00 4e 95 82 00 00 40 01  8a ef ac 11 01 13 ac 11   .N....@. ........
0020  01 08 00 00 13 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi… Packets: 4 Displayed: 4 Marked: 0 Profile: Default

Case 3 Sending Echo Request to local host but the Ethernet frame is sent to some Router

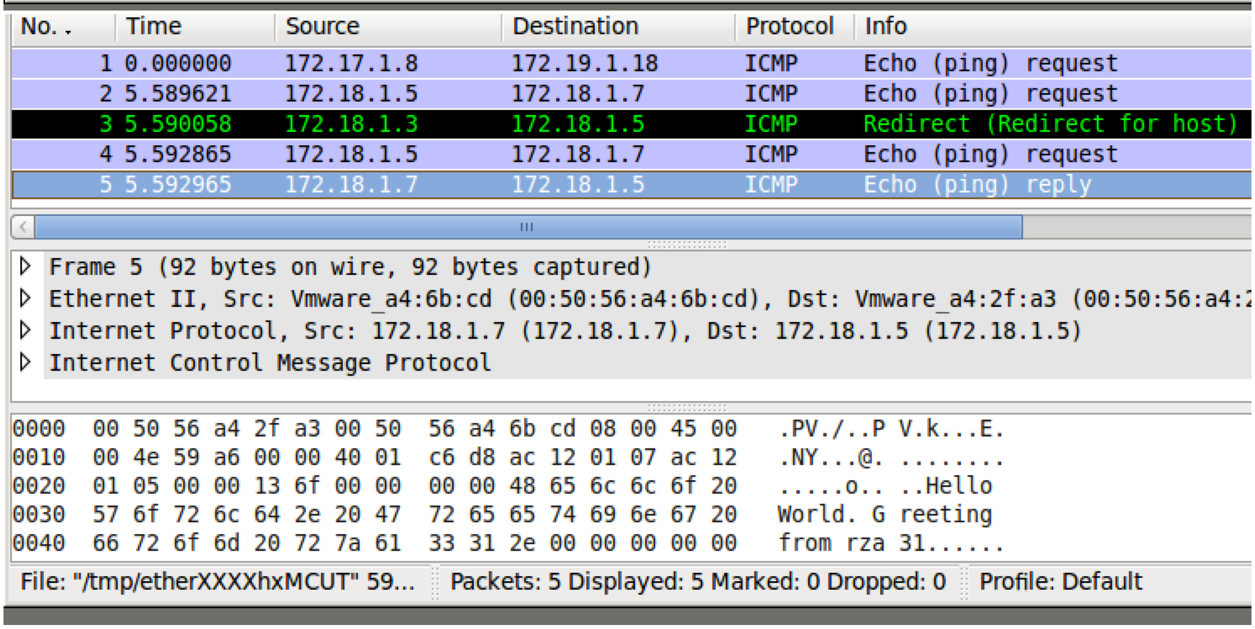
From 172.18.1.5
To 172.18.1.7
srcMAC:00:50:56:a4:05:33
dstMAC:00:50:56:a4:67:5e (172.18.1.3)

Result: May gets an ICMP Redirect message and re-wrap the IP packet with a new frame to July



```
    1 0.000000   172.17.1.8    172.19.1.18    ICMP     Echo (ping) request
    2 5.589621   172.18.1.5    172.18.1.7     ICMP     Echo (ping) request
    3 5.590058   172.18.1.3    172.18.1.5     ICMP     Redirect (Redirect for host)
    4 5.592865   172.18.1.5    172.18.1.7     ICMP     Echo (ping) request
    5 5.592965   172.18.1.7    172.18.1.5     ICMP     Echo (ping) reply

▷ Frame 2 (92 bytes on wire, 92 bytes captured)
▷ Ethernet II, Src: Vmware_a4:2f:a3 (00:50:56:a4:2f:a3), Dst: Vmware_a4:3a:33 (00:50:56:a4:3a:33)
▷ Internet Protocol, Src: 172.18.1.5 (172.18.1.5), Dst: 172.18.1.7 (172.18.1.7)
▷ Internet Control Message Protocol

0000  00 50 56 a4 3a 33 00 50  56 a4 2f a3 08 00 45 00   .PV.:3.P V./...E.
0010  00 4e 07 77 00 00 ff 01  5a 07 ac 12 01 05 ac 12   .N.w.... Z.......
0020  01 07 08 00 0b 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
```

File: "/tmp/etherXXXXhxMCUT" 59... Packets: 5 Displayed: 5 Marked: 0 Dropped: 0 Profile: Default



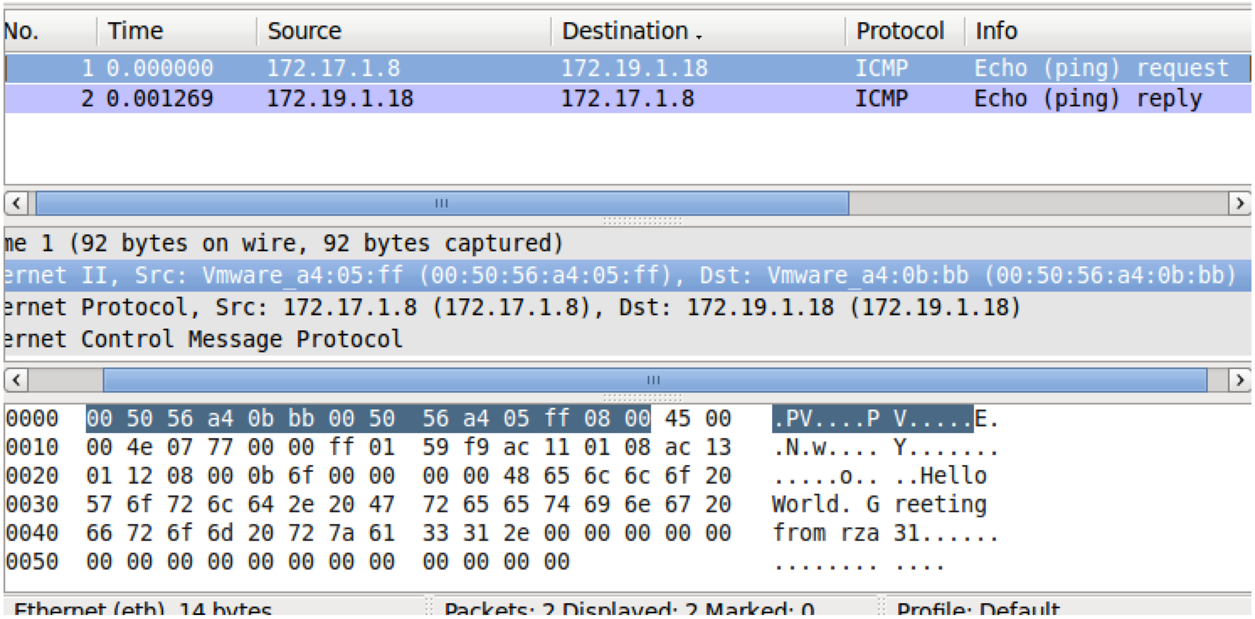
```
No. .  Time        Source        Destination    Protocol  Info
    1 0.000000   172.17.1.8    172.19.1.18    ICMP      Echo (ping) request
    2 5.589621   172.18.1.5    172.18.1.7     ICMP      Echo (ping) request
    3 5.590058   172.18.1.3    172.18.1.5     ICMP      Redirect (Redirect for host)
    4 5.592865   172.18.1.5    172.18.1.7     ICMP      Echo (ping) request
    5 5.592965   172.18.1.7    172.18.1.5     ICMP      Echo (ping) reply

▷ Frame 5 (92 bytes on wire, 92 bytes captured)
▷ Ethernet II, Src: Vmware_a4:6b:cd (00:50:56:a4:6b:cd), Dst: Vmware_a4:2f:a3 (00:50:56:a4:2
▷ Internet Protocol, Src: 172.18.1.7 (172.18.1.7), Dst: 172.18.1.5 (172.18.1.5)
▷ Internet Control Message Protocol

0000  00 50 56 a4 2f a3 00 50  56 a4 6b cd 08 00 45 00   .PV./..P V.k...E.
0010  00 4e 59 a6 00 00 40 01  c6 d8 ac 12 01 07 ac 12   .NY...@. ........
0020  01 05 00 00 13 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
```

File: "/tmp/etherXXXXhxMCUT" 59... Packets: 5 Displayed: 5 Marked: 0 Dropped: 0 Profile: Default

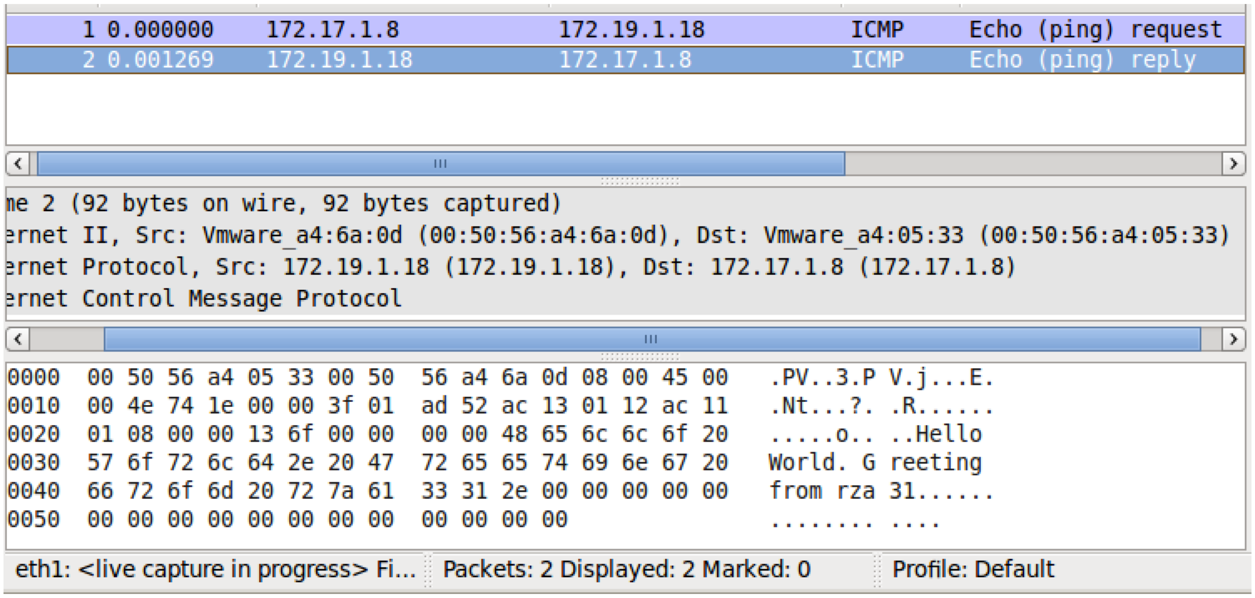
Case 4 Sending Echo Request with wrong source MAC address

From 172.17.1.8
To 172.19.1.18
srcMAC:00:50:56:a4:05:*ff* (should be 00:50:56:a4:05:33)
dstMAC:00:50:56:a4:0b:bb

Result: Source address dose not matter. Echo Reply received.



```
No.    Time        Source          Destination     Protocol  Info
     1 0.000000    172.17.1.8      172.19.1.18     ICMP      Echo (ping) request
     2 0.001269    172.19.1.18     172.17.1.8      ICMP      Echo (ping) reply

me 1 (92 bytes on wire, 92 bytes captured)
ernet II, Src: Vmware_a4:05:ff (00:50:56:a4:05:ff), Dst: Vmware_a4:0b:bb (00:50:56:a4:0b:bb)
ernet Protocol, Src: 172.17.1.8 (172.17.1.8), Dst: 172.19.1.18 (172.19.1.18)
ernet Control Message Protocol

0000  00 50 56 a4 0b bb 00 50  56 a4 05 ff 08 00 45 00   .PV....P V.....E.
0010  00 4e 07 77 00 00 ff 01  59 f9 ac 11 01 08 ac 13   .N.w.... Y.......
0020  01 12 08 00 0b 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....

Ethernet (eth), 14 bytes    Packets: 2 Displayed: 2 Marked: 0    Profile: Default
```

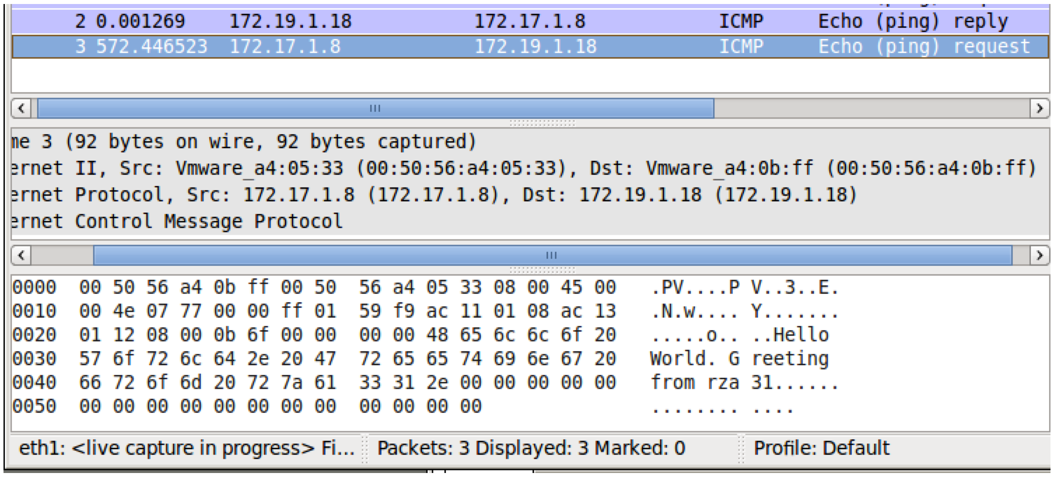


```
     1 0.000000    172.17.1.8      172.19.1.18     ICMP      Echo (ping) request
     2 0.001269    172.19.1.18     172.17.1.8      ICMP      Echo (ping) reply

me 2 (92 bytes on wire, 92 bytes captured)
ernet II, Src: Vmware_a4:6a:0d (00:50:56:a4:6a:0d), Dst: Vmware_a4:05:33 (00:50:56:a4:05:33)
ernet Protocol, Src: 172.19.1.18 (172.19.1.18), Dst: 172.17.1.8 (172.17.1.8)
ernet Control Message Protocol

0000  00 50 56 a4 05 33 00 50  56 a4 6a 0d 08 00 45 00   .PV..3.P V.j...E.
0010  00 4e 74 1e 00 00 3f 01  ad 52 ac 13 01 12 ac 11   .Nt...?. .R......
0020  01 08 00 00 13 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....

eth1: <live capture in progress> Fi...    Packets: 2 Displayed: 2 Marked: 0    Profile: Default
```

Case 5 Sending Echo Request with wrong destination MAC address

From 172.17.1.8
To 172.19.1.18
srcMAC:00:50:56:a4:05:33
dstMAC:00:50:56:a4:0b:*ff (should be 00:50:56:a4:0b:bb)*

Result: No Echo Reply comes back.



```
      2 0.001269    172.19.1.18           172.17.1.8            ICMP     Echo (ping) reply
      3 572.446523  172.17.1.8            172.19.1.18           ICMP     Echo (ping) request
```

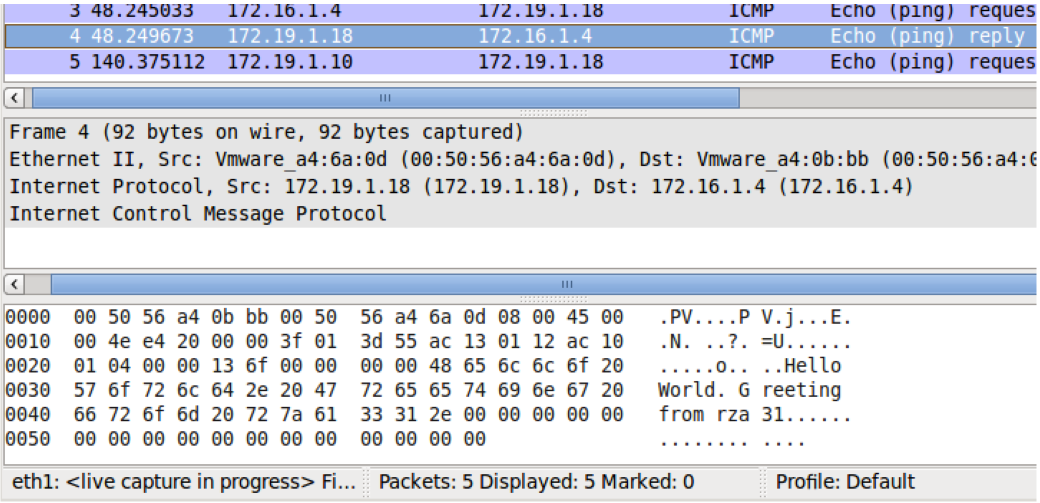
```
me 3 (92 bytes on wire, 92 bytes captured)
ernet II, Src: Vmware_a4:05:33 (00:50:56:a4:05:33), Dst: Vmware_a4:0b:ff (00:50:56:a4:0b:ff)
ernet Protocol, Src: 172.17.1.8 (172.17.1.8), Dst: 172.19.1.18 (172.19.1.18)
ernet Control Message Protocol
```

```
0000  00 50 56 a4 0b ff 00 50  56 a4 05 33 08 00 45 00   .PV....P V..3..E.
0010  00 4e 07 77 00 00 ff 01  59 f9 ac 11 01 08 ac 13   .N.w.... Y.......
0020  01 12 08 00 0b 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi...
Packets: 3 Displayed: 3 Marked: 0
Profile: Default

Case 6 Sending Echo Request with source IP different from hosting workstation

@ 172.17.1.8
From 172.16.1.4
To 172.19.1.18
srcMAC:00:50:56:a4:05:33
dstMAC: *00:50:56:a4:0b:bb*



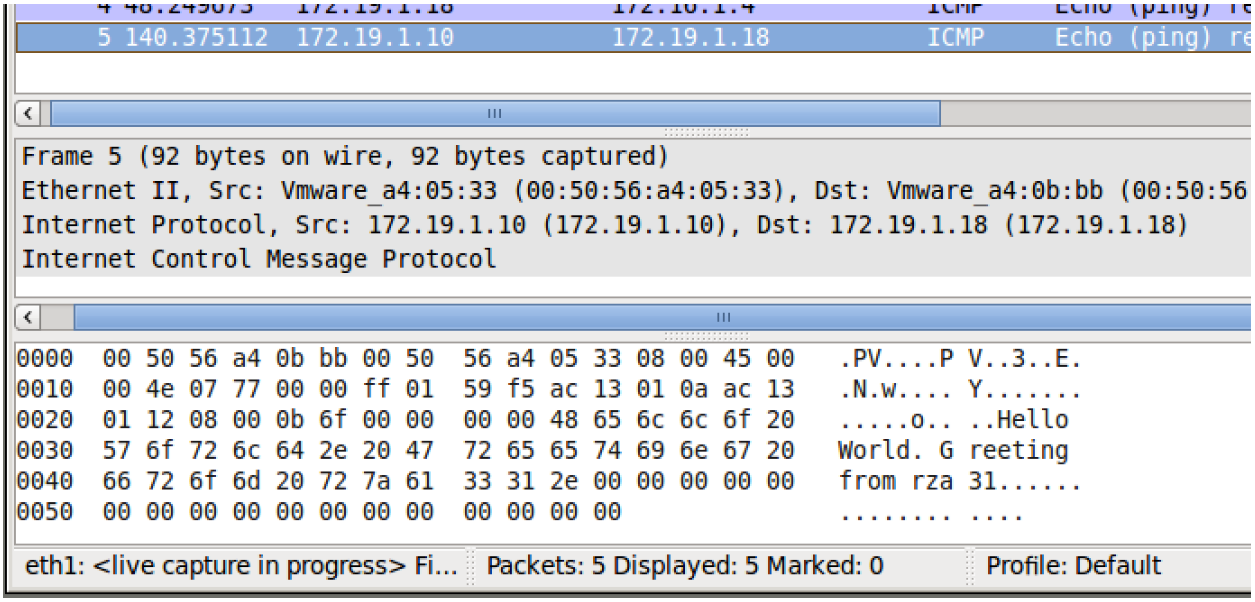
```
      3 48.245033   172.16.1.4            172.19.1.18           ICMP     Echo (ping) reques
      4 48.249673   172.19.1.18           172.16.1.4            ICMP     Echo (ping) reply
      5 140.375112  172.19.1.10           172.19.1.18           ICMP     Echo (ping) reques
```

```
Frame 4 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: Vmware_a4:6a:0d (00:50:56:a4:6a:0d), Dst: Vmware_a4:0b:bb (00:50:56:a4:0
Internet Protocol, Src: 172.19.1.18 (172.19.1.18), Dst: 172.16.1.4 (172.16.1.4)
Internet Control Message Protocol
```

```
0000  00 50 56 a4 0b bb 00 50  56 a4 6a 0d 08 00 45 00   .PV....P V.j...E.
0010  00 4e e4 20 00 00 3f 01  3d 55 ac 13 01 12 ac 10   .N. ..?. =U......
0020  01 04 00 00 13 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi...
Packets: 5 Displayed: 5 Marked: 0
Profile: Default

@ 172.17.1.8
From 172.19.1.10
To 172.19.1.18
srcMAC:00:50:56:a4:05:33
dstMAC: *00:50:56:a4:0b:bb11*



```
    4 48.249073   172.19.1.18          172.16.1.4           ICMP      Echo (ping) re
    5 140.375112  172.19.1.10          172.19.1.18          ICMP      Echo (ping) re
```

```
Frame 5 (92 bytes on wire, 92 bytes captured)
Ethernet II, Src: Vmware_a4:05:33 (00:50:56:a4:05:33), Dst: Vmware_a4:0b:bb (00:50:56
Internet Protocol, Src: 172.19.1.10 (172.19.1.10), Dst: 172.19.1.18 (172.19.1.18)
Internet Control Message Protocol
```

```
0000  00 50 56 a4 0b bb 00 50  56 a4 05 33 08 00 45 00   .PV....P V..3..E.
0010  00 4e 07 77 00 00 ff 01  59 f5 ac 13 01 0a ac 13   .N.w.... Y.......
0020  01 12 08 00 0b 6f 00 00  00 00 48 65 6c 6c 6f 20   .....o.. ..Hello
0030  57 6f 72 6c 64 2e 20 47  72 65 65 74 69 6e 67 20   World. G reeting
0040  66 72 6f 6d 20 72 7a 61  33 31 2e 00 00 00 00 00   from rza 31......
0050  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

eth1: <live capture in progress> Fi... Packets: 5 Displayed: 5 Marked: 0 Profile: Default

Result: the Echo Reply is sent back to the host of given IP address. If that host is on the route from sender host to receive host. sender host can still capture the Echo Reply message. If that host is not on the path, sender host will never see the Reply message.