WORKING WITH ICMP

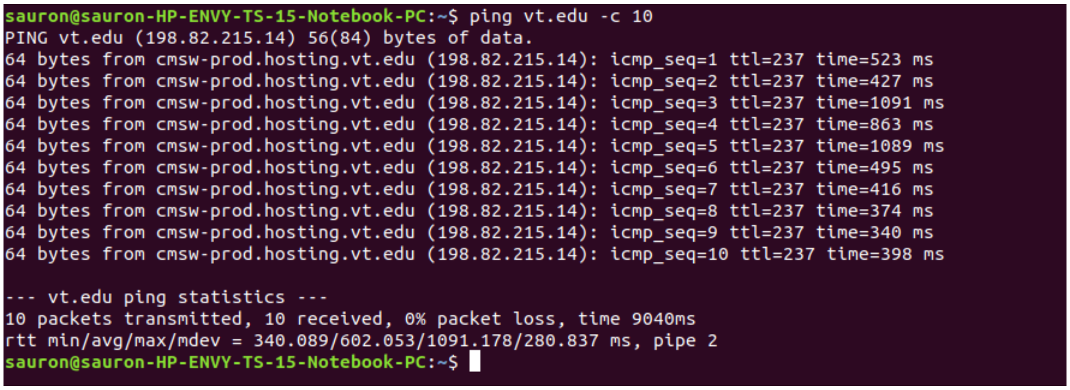
S Srinivas Saurab
CS16B039
Dept. of CSE, IIT-Madras

April 17, 2019

The ICM protocol, simply called ICMP, is one of the three main components of the Network layer. This is used by hosts and routers to specify the network-layer information to each other. The most notable feature about the ICMP is that these messages are carried as payloads inside IP datagrams. However, this does not mean that ICMP is an upper layer protocol. Interestingly, the traceroute programs extensively uses ICMP messages to get the IP addresses and RTT corresponding to the routers in a given path.

Using PING:

Ping is a program that allows us to verify if a host is live or not. Ping makes use of ICMP packets. A **particular type of ICMP packet** is sent to the interested destination. The host at the destination responds back with another typical ICMP packet. Using this packet, the ping program finds out RTT and status of the destination server.



```
sauron@sauron-HP-ENVY-TS-15-Notebook-PC:~$ ping vt.edu -c 10
PING vt.edu (198.82.215.14) 56(84) bytes of data.
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=1 ttl=237 time=523 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=2 ttl=237 time=427 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=3 ttl=237 time=1091 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=4 ttl=237 time=863 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=5 ttl=237 time=1089 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=6 ttl=237 time=495 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=7 ttl=237 time=416 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=8 ttl=237 time=374 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=9 ttl=237 time=340 ms
64 bytes from cmsw-prod.hosting.vt.edu (198.82.215.14): icmp_seq=10 ttl=237 time=398 ms

--- vt.edu ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9040ms
rtt min/avg/max/mdev = 340.089/602.053/1091.178/280.837 ms, pipe 2
sauron@sauron-HP-ENVY-TS-15-Notebook-PC:~$
```

Figure showing a ping command (count = 10) directed to Virginia Tech University

Question 1:

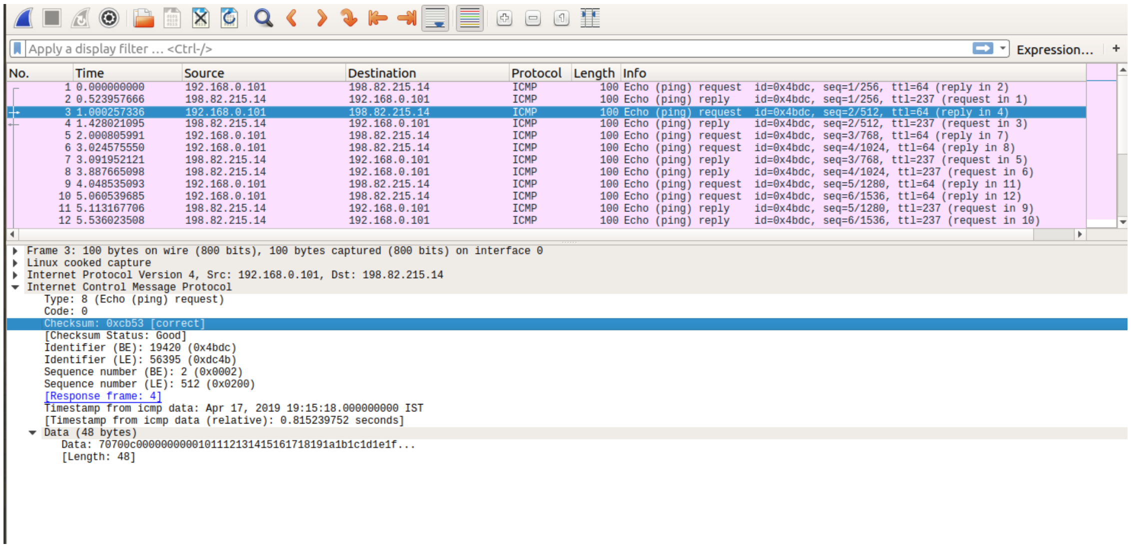
command ran : **ping vt.edu -c 10 (Virginia Tech University)**
Source IP : 192.168.0.101
Destination IP : 192.82.215.14

Question 2:

In the ping pakcets, we can see that the ICMP payload has not mention to any kind of address. **ICMP is carried as a payload within an IP datagram** and does not have an header associated with it. Within this payload, there is no mention about any address. This is not similar to TCP/UDP (have port addresses) which too are data carried inside IP datagrams.

This is because, when the protocol in IP header is ICMP the host/router has **well defined** way of handling this ICMP data message. There is no ambiguity as to which single application needs to collect this information. Therefore there is no need of address inside ICMP message. The IP address of the destination in the IP header is enough for the packet to be tramsitted appropriately, unlike TCP/UDP where port numbers are needed.

Question 3:



```
No.  Time         Source          Destination     Protocol Length Info
   1 0.000000000  192.168.0.101   198.82.215.14   ICMP     100 Echo (ping) request  id=0x4bdc, seq=1/256, ttl=64 (reply in 2)
   2 0.523957666  198.82.215.14   192.168.0.101   ICMP     100 Echo (ping) reply    id=0x4bdc, seq=1/256, ttl=237 (request in 1)
   3 1.000257336  192.168.0.101   198.82.215.14   ICMP     100 Echo (ping) request  id=0x4bdc, seq=2/512, ttl=64 (reply in 4)
   4 1.428021095  198.82.215.14   192.168.0.101   ICMP     100 Echo (ping) reply    id=0x4bdc, seq=2/512, ttl=237 (request in 3)
   5 2.000805991  192.168.0.101   198.82.215.14   ICMP     100 Echo (ping) request  id=0x4bdc, seq=3/768, ttl=64 (reply in 7)
   6 3.024575550  192.168.0.101   198.82.215.14   ICMP     100 Echo (ping) request  id=0x4bdc, seq=4/1024, ttl=64 (reply in 8)
   7 3.091952121  198.82.215.14   192.168.0.101   ICMP     100 Echo (ping) reply    id=0x4bdc, seq=3/768, ttl=237 (request in 5)
   8 3.887665098  198.82.215.14   192.168.0.101   ICMP     100 Echo (ping) reply    id=0x4bdc, seq=4/1024, ttl=237 (request in 6)
   9 4.048535093  192.168.0.101   198.82.215.14   ICMP     100 Echo (ping) request  id=0x4bdc, seq=5/1280, ttl=64 (reply in 11)
  10 5.060539685  192.168.0.101   198.82.215.14   ICMP     100 Echo (ping) request  id=0x4bdc, seq=6/1536, ttl=64 (reply in 12)
  11 5.113167706  198.82.215.14   192.168.0.101   ICMP     100 Echo (ping) reply    id=0x4bdc, seq=5/1280, ttl=237 (request in 9)
  12 5.536023508  198.82.215.14   192.168.0.101   ICMP     100 Echo (ping) reply    id=0x4bdc, seq=6/1536, ttl=237 (request in 10)
```

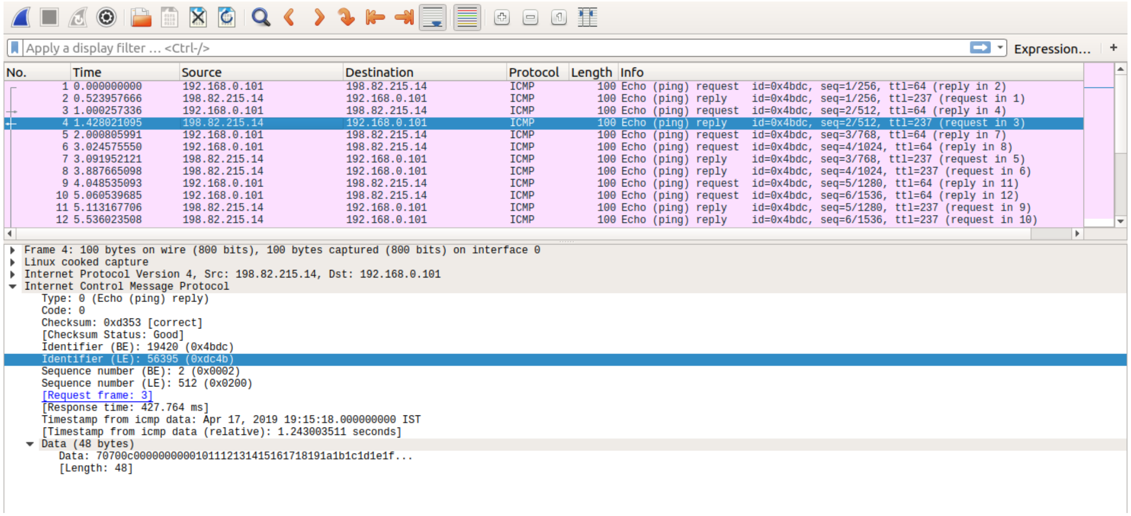
```
▸ Frame 3: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 198.82.215.14
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xcb53 [correct]
    [Checksum Status: Good]
    Identifier (BE): 19420 (0x4bdc)
    Identifier (LE): 56395 (0xdc4b)
    Sequence number (BE): 2 (0x0002)
    Sequence number (LE): 512 (0x0200)
    [Response frame: 4]
    Timestamp from icmp data: Apr 17, 2019 19:15:18.000000000 IST
    [Timestamp from icmp data (relative): 0.815239752 seconds]
  ▾ Data (48 bytes)
      Data: 70700c0000000000101112131415161718191a1b1c1d1e1f...
      [Length: 48]
```

In the above figure, we see a capture of expanded version of one of the ping requests.

The fields in ICMP are :
1) Type = 8 : corresponds to ping request
2) Code = 0

3) Checksum : 2 bytes
4) Identifier : 2 bytes
5) Sequence number : 2 bytes
6) Timestamp
7) Data

Question 4:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.0.101	198.82.215.14	ICMP	100	Echo (ping) request id=0x4bdc, seq=1/256, ttl=64 (reply in 2)
2	0.523957666	198.82.215.14	192.168.0.101	ICMP	100	Echo (ping) reply id=0x4bdc, seq=1/256, ttl=237 (request in 1)
3	1.000257336	192.168.0.101	198.82.215.14	ICMP	100	Echo (ping) request id=0x4bdc, seq=2/512, ttl=64 (reply in 4)
4	1.428021095	198.82.215.14	192.168.0.101	ICMP	100	Echo (ping) reply id=0x4bdc, seq=2/512, ttl=237 (request in 3)
5	2.000805991	192.168.0.101	198.82.215.14	ICMP	100	Echo (ping) request id=0x4bdc, seq=3/768, ttl=64 (reply in 7)
6	3.024575550	192.168.0.101	198.82.215.14	ICMP	100	Echo (ping) request id=0x4bdc, seq=4/1024, ttl=64 (reply in 8)
7	3.091952121	198.82.215.14	192.168.0.101	ICMP	100	Echo (ping) reply id=0x4bdc, seq=3/768, ttl=237 (request in 5)
8	3.887665098	198.82.215.14	192.168.0.101	ICMP	100	Echo (ping) reply id=0x4bdc, seq=4/1024, ttl=237 (request in 6)
9	4.048535093	192.168.0.101	198.82.215.14	ICMP	100	Echo (ping) request id=0x4bdc, seq=5/1280, ttl=64 (reply in 11)
10	5.060539685	192.168.0.101	198.82.215.14	ICMP	100	Echo (ping) request id=0x4bdc, seq=6/1536, ttl=64 (reply in 12)
11	5.113167706	198.82.215.14	192.168.0.101	ICMP	100	Echo (ping) reply id=0x4bdc, seq=5/1280, ttl=237 (request in 9)
12	5.536023508	198.82.215.14	192.168.0.101	ICMP	100	Echo (ping) reply id=0x4bdc, seq=6/1536, ttl=237 (request in 10)

```
▸ Frame 4: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 198.82.215.14, Dst: 192.168.0.101
▾ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xd353 [correct]
    [Checksum Status: Good]
    Identifier (BE): 19420 (0x4bdc)
    Identifier (LE): 56395 (0xdc4b)
    Sequence number (BE): 2 (0x0002)
    Sequence number (LE): 512 (0x0200)
    [Request frame: 3]
    [Response time: 427.764 ms]
    Timestamp from icmp data: Apr 17, 2019 19:15:18.000000000 IST
    [Timestamp from icmp data (relative): 1.243003511 seconds]
  ▾ Data (48 bytes)
      Data: 70700c0000000000101112131415161718191a1b1c1d1e1f...
      [Length: 48]
```

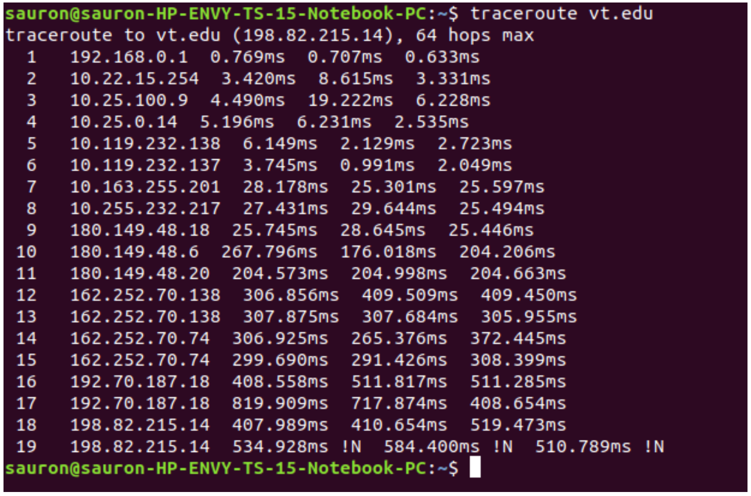
In the above figure, we see a capture of expanded version of one of the ping replies.

The fields in ICMP are :
1) Type = 0 : corresponds to ping response
2) Code = 0
3) Checksum : 2 bytes
4) Identifier : 2 bytes
5) Sequence number : 2 bytes
6) Timestamp
7) Data

Using TRACEROUTE:

Traceroute is a program that makes use of UDP packets to get the IPs of all the routers along a given path and the time delays between them. This is achieved by sending a **sequence of UDP packets** with TTL ranging from 1 to 64 (MAX). Everytime a router identifies that a packet has expired, it sends a corresponding **ICMP error message** to the source host. The traceroute then identifies the details about the router from this ICMP message.

command used : traceroute vt.edu (Vriginia Tech University)



```
sauron@sauron-HP-ENVY-TS-15-Notebook-PC:~$ traceroute vt.edu
traceroute to vt.edu (198.82.215.14), 64 hops max
  1   192.168.0.1  0.769ms  0.707ms  0.633ms
  2   10.22.15.254  3.420ms  8.615ms  3.331ms
  3   10.25.100.9  4.490ms  19.222ms  6.228ms
  4   10.25.0.14  5.196ms  6.231ms  2.535ms
  5   10.119.232.138  6.149ms  2.129ms  2.723ms
  6   10.119.232.137  3.745ms  0.991ms  2.049ms
  7   10.163.255.201  28.178ms  25.301ms  25.597ms
  8   10.255.232.217  27.431ms  29.644ms  25.494ms
  9   180.149.48.18  25.745ms  28.645ms  25.446ms
 10   180.149.48.6  267.796ms  176.018ms  204.206ms
 11   180.149.48.20  204.573ms  204.998ms  204.663ms
 12   162.252.70.138  306.856ms  409.509ms  409.450ms
 13   162.252.70.138  307.875ms  307.684ms  305.955ms
 14   162.252.70.74  306.925ms  265.376ms  372.445ms
 15   162.252.70.74  299.690ms  291.426ms  308.399ms
 16   192.70.187.18  408.558ms  511.817ms  511.285ms
 17   192.70.187.18  819.909ms  717.874ms  408.654ms
 18   198.82.215.14  407.989ms  410.654ms  519.473ms
 19   198.82.215.14  534.928ms !N  584.400ms !N  510.789ms !N
sauron@sauron-HP-ENVY-TS-15-Notebook-PC:~$ 
```

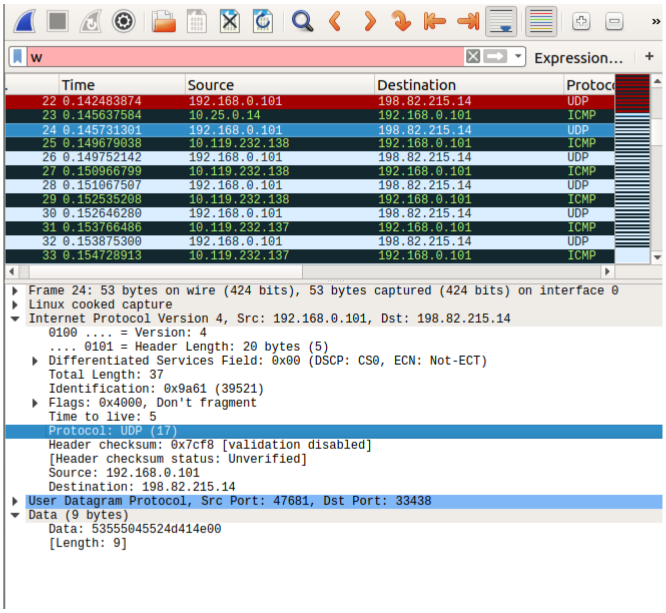
Question 5:

IP address of our host : 192.168.0.101
IP address of Virginia Tech host : 198.82.215.14

The destination IP address can be seen from the last echo. Alternatively this can also be seen from any echo because, every response (TTL expire warning) also contains the original packet the host tried to send.

Question 6:

Different upper layer protocols are assigned different values for the protocol field in the IP header. For ICMP it is 01, where as for UDP it **will be 17**. This is a must because, if the IP datagrams encapsulate UDP packets, then having 17 as the protocol number is important in processing the packet when it arrives at the host. If it were falsely written as 1 instead, the host will try to break the payload as an ICMP packet which **has different fields** and demarcations. The same can be seen from the UDP probes sent from our Linux system.



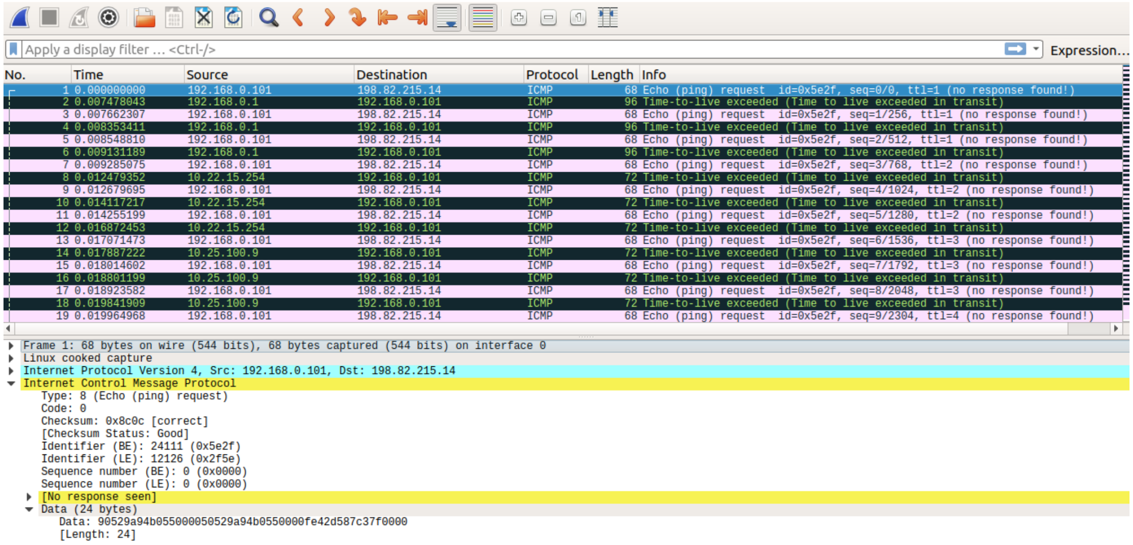
```
.    Time         Source          Destination     Protoc
  22 0.142483874  192.168.0.101   198.82.215.14   UDP
  23 0.145637584  10.25.0.14      192.168.0.101   ICMP
  24 0.145731301  192.168.0.101   198.82.215.14   UDP
  25 0.149679038  10.119.232.138  192.168.0.101   ICMP
  26 0.149752142  192.168.0.101   198.82.215.14   UDP
  27 0.150966799  10.119.232.138  192.168.0.101   ICMP
  28 0.151067507  192.168.0.101   198.82.215.14   UDP
  29 0.152535208  10.119.232.138  192.168.0.101   ICMP
  30 0.152646280  192.168.0.101   198.82.215.14   UDP
  31 0.153766486  10.119.232.137  192.168.0.101   ICMP
  32 0.153875300  192.168.0.101   198.82.215.14   UDP
  33 0.154728913  10.119.232.137  192.168.0.101   ICMP
```

```
▸ Frame 24: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface 0
▸ Linux cooked capture
▾ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 198.82.215.14
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 37
    Identification: 0x9a61 (39521)
  ▸ Flags: 0x4000, Don't fragment
    Time to live: 5
    Protocol: UDP (17)
    Header checksum: 0x7cf8 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.0.101
    Destination: 198.82.215.14
▸ User Datagram Protocol, Src Port: 47681, Dst Port: 33438
▾ Data (9 bytes)
    Data: 53555045524d414e00
    [Length: 9]
```

Question 7:

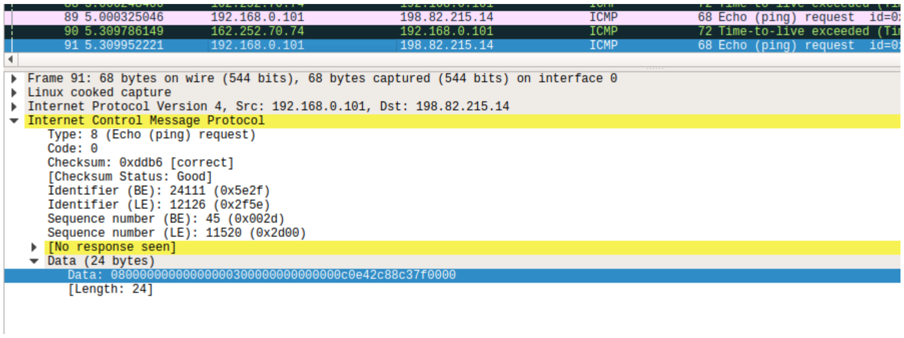
The ICMP packet resembles ping (request and reply) packet upto the IP header. From IP header, the whole payload is a lot different. This is because, the fields in the ping response (identification, sequence number) are absent in the traceroute echo messages. A more important difference is the contents of the data within ICMP payload. This **data has the copy of the original packet as such** that was failed to be forwarded because of TTL expiration. Another difference is the Type field value which is 11 and the code is 0 unlike ping packets (Type : 0,8).

running the command : sudo traceroute -I vt.edu
(uses echo ICMP instead of UDPs : refer to Traceroute2.pcapng)



```
Apply a display filter ... <Ctrl-/>                                              Expression...
No.  Time          Source          Destination     Protocol  Length Info
   1 0.000000000   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=0/0, ttl=1 (no response found!)
   2 0.007478043   192.168.0.1     192.168.0.101   ICMP      96 Time-to-live exceeded (Time to live exceeded in transit)
   3 0.007662307   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=1/256, ttl=1 (no response found!)
   4 0.008353411   192.168.0.1     192.168.0.101   ICMP      96 Time-to-live exceeded (Time to live exceeded in transit)
   5 0.008548810   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=2/512, ttl=1 (no response found!)
   6 0.009131189   192.168.0.1     192.168.0.101   ICMP      96 Time-to-live exceeded (Time to live exceeded in transit)
   7 0.009285075   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=3/768, ttl=2 (no response found!)
   8 0.012479352   10.22.15.254    192.168.0.101   ICMP      72 Time-to-live exceeded (Time to live exceeded in transit)
   9 0.012679695   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=4/1024, ttl=2 (no response found!)
  10 0.014117217   10.22.15.254    192.168.0.101   ICMP      72 Time-to-live exceeded (Time to live exceeded in transit)
  11 0.014255199   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=5/1280, ttl=2 (no response found!)
  12 0.016872453   10.22.15.254    192.168.0.101   ICMP      72 Time-to-live exceeded (Time to live exceeded in transit)
  13 0.017071473   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=6/1536, ttl=3 (no response found!)
  14 0.017887222   10.25.100.9     192.168.0.101   ICMP      72 Time-to-live exceeded (Time to live exceeded in transit)
  15 0.018014602   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=7/1792, ttl=3 (no response found!)
  16 0.018801199   10.25.100.9     192.168.0.101   ICMP      72 Time-to-live exceeded (Time to live exceeded in transit)
  17 0.018923582   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=8/2048, ttl=3 (no response found!)
  18 0.019841909   10.25.100.9     192.168.0.101   ICMP      72 Time-to-live exceeded (Time to live exceeded in transit)
  19 0.019964968   192.168.0.101   198.82.215.14   ICMP      68 Echo (ping) request  id=0x5e2f, seq=9/2304, ttl=4 (no response found!)

▸ Frame 1: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 198.82.215.14
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x8c0c [correct]
    [Checksum Status: Good]
    Identifier (BE): 24111 (0x5e2f)
    Identifier (LE): 12126 (0x2f5e)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
  ▸ [No response seen]
  ▾ Data (24 bytes)
      Data: 90529a94b055000050529a94b0550000fe42d587c37f0000
      [Length: 24]
```

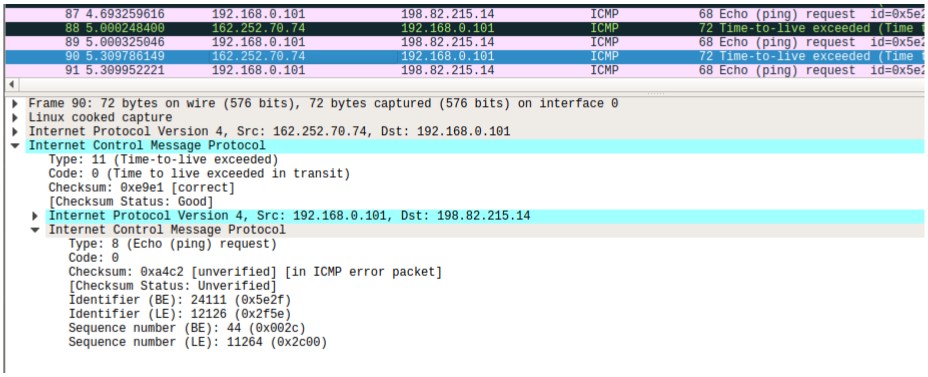


```
  89 5.000325046   192.168.0.101   198.82.215.14   ICMP   68 Echo (ping) request  id=0:
  90 5.309786149   162.252.70.74   192.168.0.101   ICMP   72 Time-to-live exceeded (Ti
  91 5.309952221   192.168.0.101   198.82.215.14   ICMP   68 Echo (ping) request  id=0:

▸ Frame 91: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 198.82.215.14
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xddb6 [correct]
    [Checksum Status: Good]
    Identifier (BE): 24111 (0x5e2f)
    Identifier (LE): 12126 (0x2f5e)
    Sequence number (BE): 45 (0x002d)
    Sequence number (LE): 11520 (0x2d00)
  ▸ [No response seen]
  ▾ Data (24 bytes)
      Data: 08000000000000000300000000000000c0e42c88c37f0000
      [Length: 24]
```

On opening a Echo message and comparing with the ping packet used in ping command, we can find that there is no difference in the fields names i.e the structure of the packet. The only diffrence here is that TTL is set differently (from 1 to 17 in our case). These TTL variations can be seen from the wireshark capture.

Question 8:



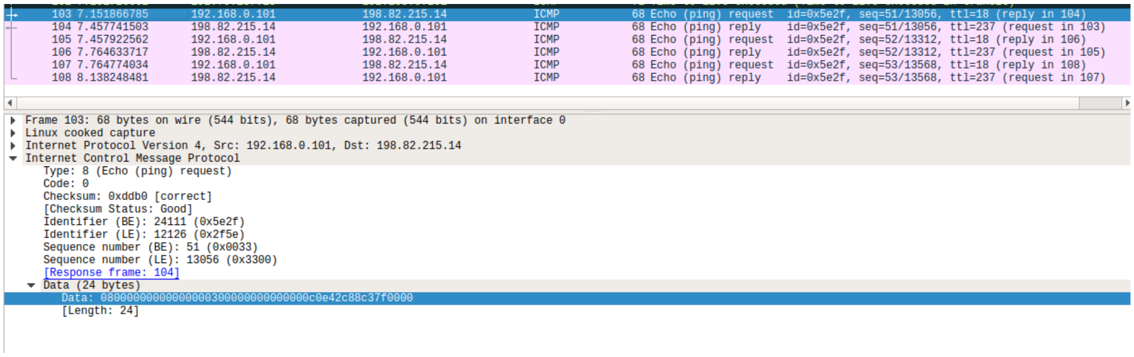
```
   87 4.693259616     192.168.0.101          198.82.215.14          ICMP        68 Echo (ping) request  id=0x5e2
   88 5.000248400     162.252.70.74          192.168.0.101          ICMP        72 Time-to-live exceeded (Time t
   89 5.000325046     192.168.0.101          198.82.215.14          ICMP        68 Echo (ping) request  id=0x5e2
   90 5.309786149     162.252.70.74          192.168.0.101          ICMP        72 Time-to-live exceeded (Time t
   91 5.309952221     192.168.0.101          198.82.215.14          ICMP        68 Echo (ping) request  id=0x5e2

▶ Frame 90: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 162.252.70.74, Dst: 192.168.0.101
▼ Internet Control Message Protocol
    Type: 11 (Time-to-live exceeded)
    Code: 0 (Time to live exceeded in transit)
    Checksum: 0xe9e1 [correct]
    [Checksum Status: Good]
  ▶ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 198.82.215.14
  ▼ Internet Control Message Protocol
      Type: 8 (Echo (ping) request)
      Code: 0
      Checksum: 0xa4c2 [unverified] [in ICMP error packet]
      [Checksum Status: Unverified]
      Identifier (BE): 24111 (0x5e2f)
      Identifier (LE): 12126 (0x2f5e)
      Sequence number (BE): 44 (0x002c)
      Sequence number (LE): 11264 (0x2c00)
```

Upto IP header, the ICMP error as well as the echo messages take the same format. However their payloads, which in turn contain the ICMP data messages, are entirely different. Both ICMP data messages have Type, Code fields as well as checksum. The **differences** between the ICMP error message and the ICMP ping messages are:

1) Echo packet has Identifier and a Sequence number field which is used by the ping program. These fields are not present in the ICMP error message.
2) Echo packet has a small data field (usually 24 bytes). However ICMP error message has the **entire packet that was initially transmitted** by the router (with an insufficient TTL) starting from the network layer header. This is the reason why it appears to have more fields because it encapsulates the original ping packet in its payload.

Question 9:



```
   103 7.151866785   192.168.0.101        198.82.215.14        ICMP        68 Echo (ping) request  id=0x5e2f, seq=51/13056, ttl=18 (reply in 104)
   104 7.457741503   198.82.215.14        192.168.0.101        ICMP        68 Echo (ping) reply    id=0x5e2f, seq=51/13056, ttl=237 (request in 103)
   105 7.457922562   192.168.0.101        198.82.215.14        ICMP        68 Echo (ping) request  id=0x5e2f, seq=52/13312, ttl=18 (reply in 106)
   106 7.764633717   198.82.215.14        192.168.0.101        ICMP        68 Echo (ping) reply    id=0x5e2f, seq=52/13312, ttl=237 (request in 105)
   107 7.764774034   192.168.0.101        198.82.215.14        ICMP        68 Echo (ping) request  id=0x5e2f, seq=53/13568, ttl=18 (reply in 108)
   108 8.138248481   198.82.215.14        192.168.0.101        ICMP        68 Echo (ping) reply    id=0x5e2f, seq=53/13568, ttl=237 (request in 107)

▶ Frame 103: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
▶ Linux cooked capture
▶ Internet Protocol Version 4, Src: 192.168.0.101, Dst: 198.82.215.14
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xddb0 [correct]
    [Checksum Status: Good]
    Identifier (BE): 24111 (0x5e2f)
    Identifier (LE): 12126 (0x2f5e)
    Sequence number (BE): 51 (0x0033)
    Sequence number (LE): 13056 (0x3300)
    [Response frame: 104]
  ▼ Data (24 bytes)
      Data: 08000000000000000300000000000000c0e42c88c37f0000
      [Length: 24]
```

Figure showing the last 6 messages

The last 6 packets are different fromt the rest of the packets because the responses are **not TTL errors but are ping replies**. This is as though a ping has been sent to the destination host. This is because, the TTL is enough to reach the destination server and that server replies with a ping response (Code : 0, Type : 0).

Question 10: