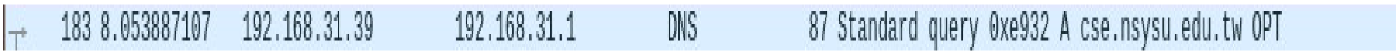
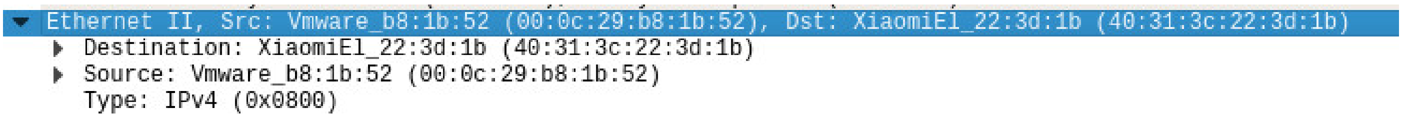
Part.I

1.



```
183 8.053887107   192.168.31.39   192.168.31.1   DNS   87 Standard query 0xe932 A cse.nsysu.edu.tw OPT
```

(1) Examine the Ethernet



```
▾ Ethernet II, Src: Vmware_b8:1b:52 (00:0c:29:b8:1b:52), Dst: XiaomiEl_22:3d:1b (40:31:3c:22:3d:1b)
   ▸ Destination: XiaomiEl_22:3d:1b (40:31:3c:22:3d:1b)
   ▸ Source: Vmware_b8:1b:52 (00:0c:29:b8:1b:52)
     Type: IPv4 (0x0800)
```

a. What is the Ethernet address of the source and destination?

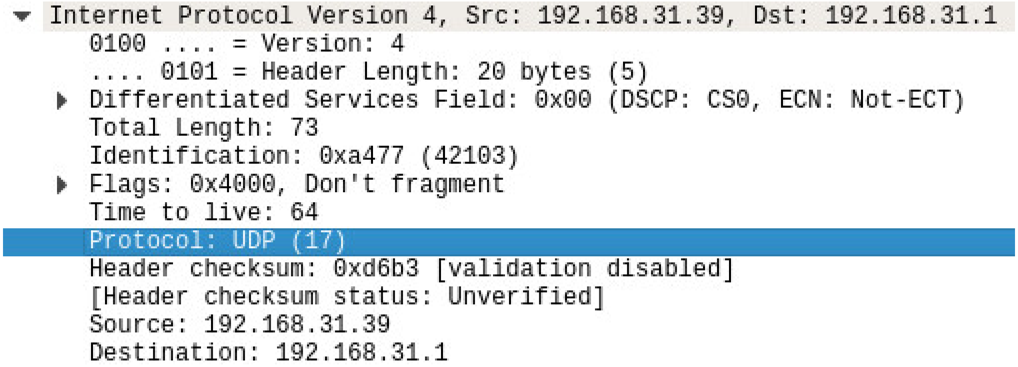
Source: 00:0c:29:b8:1b:52

Destination: 40:31:3c:22:3d:1b

b. What is the content of the type field in the Ethernet frame?

0x0800 (IPv4)

(2) Examine the Internet Protocol



```
▾ Internet Protocol Version 4, Src: 192.168.31.39, Dst: 192.168.31.1
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 73
     Identification: 0xa477 (42103)
   ▸ Flags: 0x4000, Don't fragment
     Time to live: 64
     Protocol: UDP (17)
     Header checksum: 0xd6b3 [validation disabled]
     [Header checksum status: Unverified]
     Source: 192.168.31.39
     Destination: 192.168.31.1
```

a. What is the IP address of the source and destination?

Source: 192.168.31.39

Destination: 192.168.31.1

b. What is the header length? What is the total packet length?

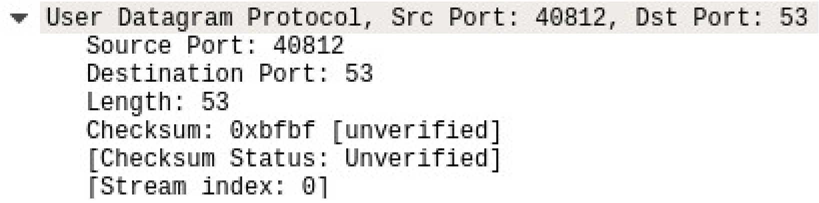
Header length = 20 Bytes

Package length = 73 Bytes

c. Identify the protocol type field. What is the number and type of the protocol in the payload?

17 UDP

(3) Examine the User Datagram Protocol



```
▾ User Datagram Protocol, Src Port: 40812, Dst Port: 53
     Source Port: 40812
     Destination Port: 53
     Length: 53
     Checksum: 0xbfbf [unverified]
     [Checksum Status: Unverified]
     [Stream index: 0]
```

a. Identify the client ephemeral port number and the server well-known port number.

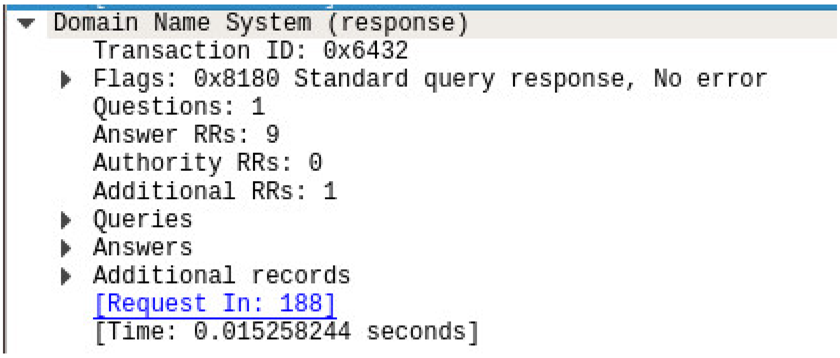
Client port: 40812

Server port: 53

b. What type of application layer protocol is in the payload?

DNS

(4) Examine the Domain Name System (query)



```
▼ Domain Name System (response)
    Transaction ID: 0x6432
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 9
    Authority RRs: 0
    Additional RRs: 1
  ▶ Queries
  ▶ Answers
  ▶ Additional records
    [Request In: 188]
    [Time: 0.015258244 seconds]
```

a. What field indicates whether the message is a query or a response?

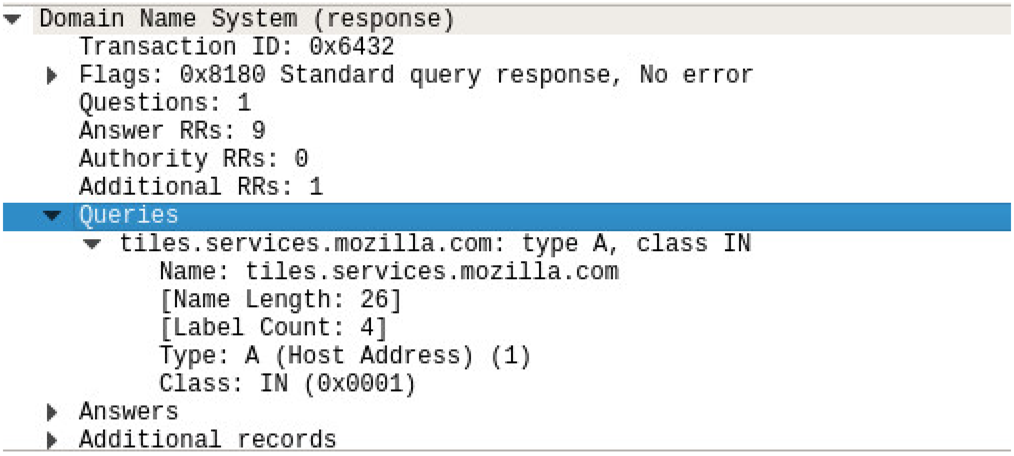
Flags

b. What is the query transaction ID?

0x6432

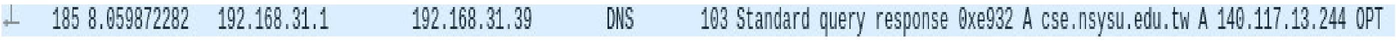
c. Identify the fields that carry the type and class of the query.

Queries



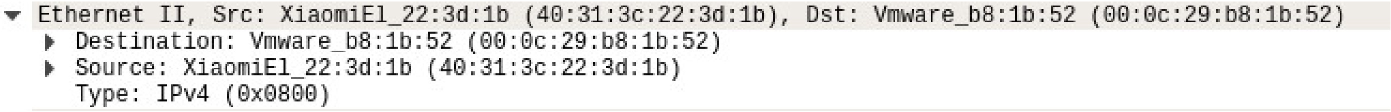
```
▼ Domain Name System (response)
    Transaction ID: 0x6432
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 9
    Authority RRs: 0
    Additional RRs: 1
  ▼ Queries
    ▼ tiles.services.mozilla.com: type A, class IN
        Name: tiles.services.mozilla.com
        [Name Length: 26]
        [Label Count: 4]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ▶ Answers
  ▶ Additional records
```

2.



```
185 8.059872282   192.168.31.1   192.168.31.39   DNS   103 Standard query response 0xe932 A cse.nsysu.edu.tw A 140.117.13.244 OPT
```

(1) Examine the Ethernet



```
▼ Ethernet II, Src: XiaomiEl_22:3d:1b (40:31:3c:22:3d:1b), Dst: Vmware_b8:1b:52 (00:0c:29:b8:1b:52)
  ▶ Destination: Vmware_b8:1b:52 (00:0c:29:b8:1b:52)
  ▶ Source: XiaomiEl_22:3d:1b (40:31:3c:22:3d:1b)
    Type: IPv4 (0x0800)
```

a. What is the Ethernet address of the source and destination?

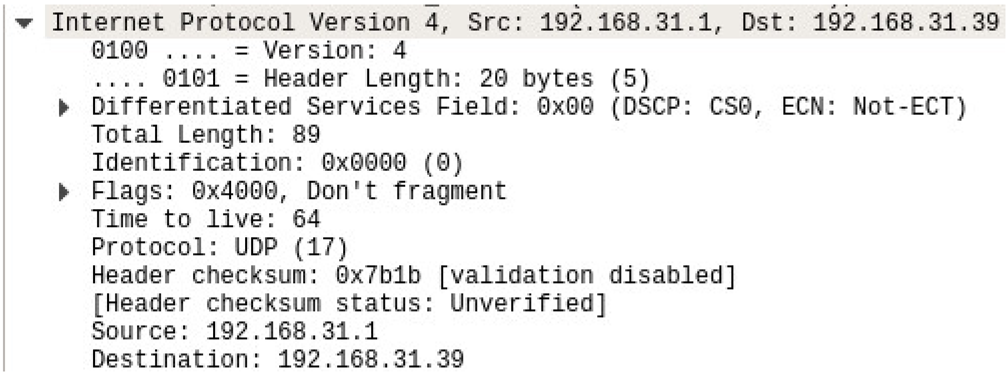
Source: 40:31:3c:22:3d:1b

Destination: 00:0c:29:b8:1b:52

b. What is the content of the type field in the Ethernet frame?

IPv4

(2) Examine the Internet Protocol & Domain Name System (response)



```
▼ Internet Protocol Version 4, Src: 192.168.31.1, Dst: 192.168.31.39
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 89
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0x7b1b [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.31.1
    Destination: 192.168.31.39
```

a. What is the IP address of the source and destination?

Source: 192.168.31.1

Destination: 192.168.31.39

b. What is the header length? What is the total packet length? Is it longer than the query?

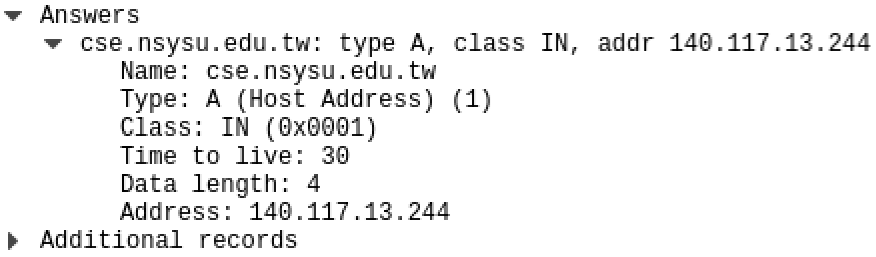
Header length: 20 Bytes

Total length: 89 Bytes

It’s longer than query packet (73 Bytes).

c. How many answers are provided in the response message?

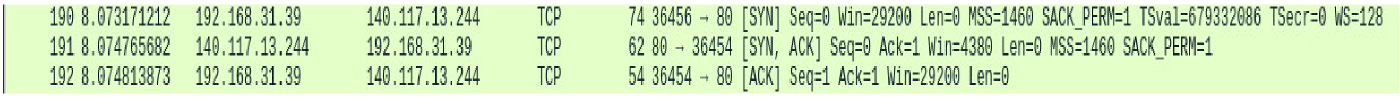
Compare the answers and their time-to-live values.



```
▼ Answers
  ▼ cse.nsysu.edu.tw: type A, class IN, addr 140.117.13.244
      Name: cse.nsysu.edu.tw
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 30
      Data length: 4
      Address: 140.117.13.244
▶ Additional records
```

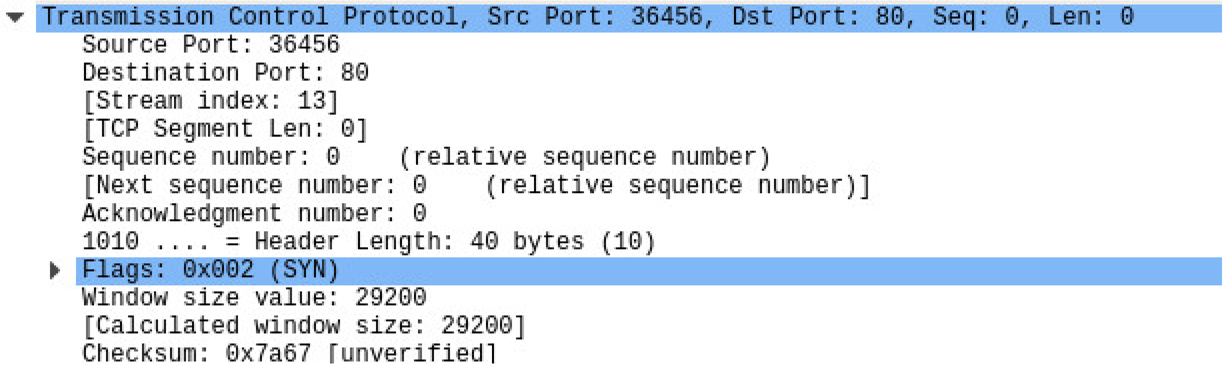
1 answer, time-to-live: 30

3.



```
190 8.073171212   192.168.31.39    140.117.13.244   TCP   74 36456 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=679332086 TSecr=0 WS=128
191 8.074765682   140.117.13.244   192.168.31.39    TCP   62 80 → 36454 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 MSS=1460 SACK_PERM=1
192 8.074813873   192.168.31.39    140.117.13.244   TCP   54 36454 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
```

(1) Examine the Transmission Control Protocol



```
▼ Transmission Control Protocol, Src Port: 36456, Dst Port: 80, Seq: 0, Len: 0
      Source Port: 36456
      Destination Port: 80
      [Stream index: 13]
      [TCP Segment Len: 0]
      Sequence number: 0    (relative sequence number)
      [Next sequence number: 0    (relative sequence number)]
      Acknowledgment number: 0
      1010 .... = Header Length: 40 bytes (10)
    ▶ Flags: 0x002 (SYN)
      Window size value: 29200
      [Calculated window size: 29200]
      Checksum: 0x7a67 [unverified]
```

a. What are the ephemeral port number used by the client and the well-known port number used by the server?

Client port: 36456

Server port: 80

b. What is the length of the TCP segment?

0

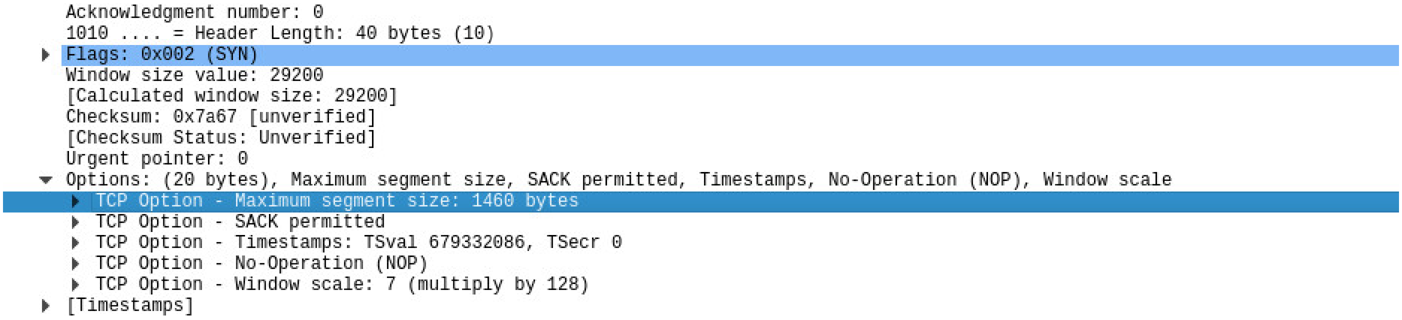
c. What is the initial sequence number for the segments from the client to the server?

0

d. What is the initial window size?

29200

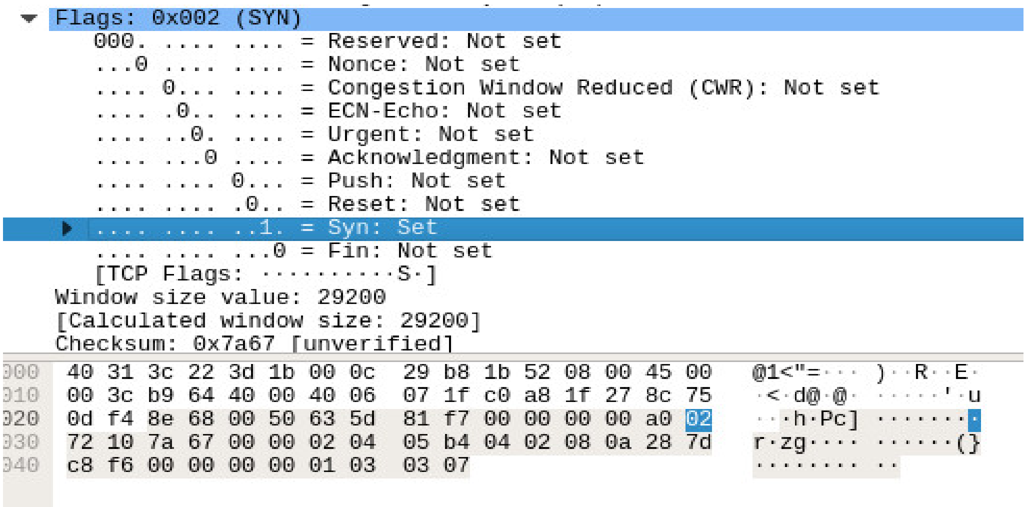
e. What is the maximum segment size?



```
   Acknowledgment number: 0
   1010 .... = Header Length: 40 bytes (10)
 ▶ Flags: 0x002 (SYN)
   Window size value: 29200
   [Calculated window size: 29200]
   Checksum: 0x7a67 [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
 ▼ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
    ▶ TCP Option - Maximum segment size: 1460 bytes
    ▶ TCP Option - SACK permitted
    ▶ TCP Option - Timestamps: TSval 679332086, TSecr 0
    ▶ TCP Option - No-Operation (NOP)
    ▶ TCP Option - Window scale: 7 (multiply by 128)
 ▶ [Timestamps]
```

1460 Bytes

f. Find the hex character that contains the SYN flag bit

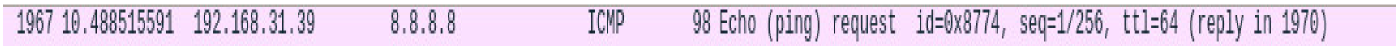


```
▼ Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Nonce: Not set
    .... 0... .... = Congestion Window Reduced (CWR): Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
  ▶ .... .... ..1. = Syn: Set
    .... .... ...0 = Fin: Not set
    [TCP Flags: ··········S·]
  Window size value: 29200
  [Calculated window size: 29200]
  Checksum: 0x7a67 [unverified]
000  40 31 3c 22 3d 1b 00 0c  29 b8 1b 52 08 00 45 00   @1<"=··· )··R··E·
010  00 3c b9 64 40 00 40 06  07 1f c0 a8 1f 27 8c 75   ·<·d@·@· ·····'·u
020  0d f4 8e 68 00 50 63 5d  81 f7 00 00 00 00 a0 02   ···h·Pc] ········
030  72 10 7a 67 00 00 02 04  05 b4 04 02 08 0a 28 7d   r·zg···· ······(}
040  c8 f6 00 00 00 00 01 03  03 07                     ········ ··
```

Part.II

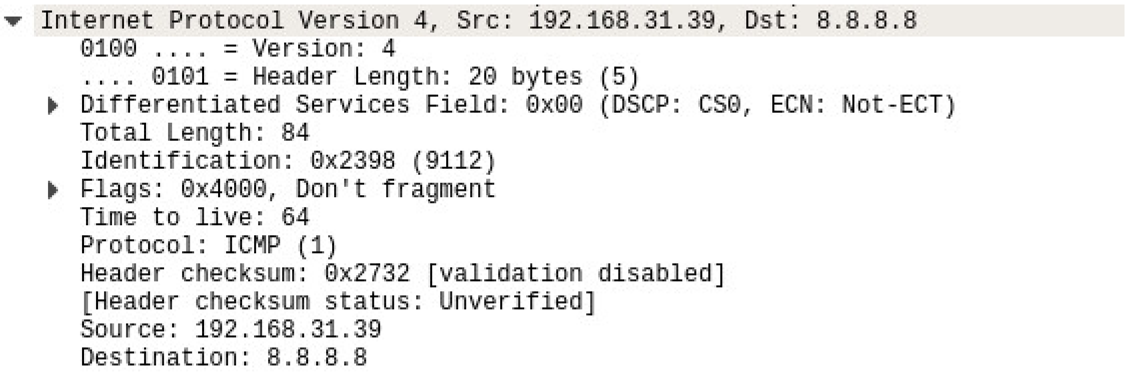
1.

(1) Find the first ICMP Echo Request packet.



```
1967 10.488515591  192.168.31.39     8.8.8.8          ICMP     98 Echo (ping) request  id=0x8774, seq=1/256, ttl=64 (reply in 1970)
```

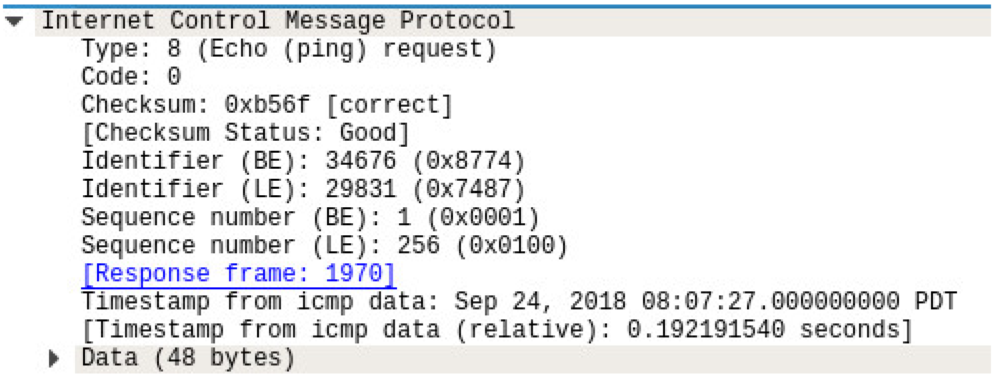
a. First, examine the Internet Protocol. What is the Time-to-Live?



```
▼ Internet Protocol Version 4, Src: 192.168.31.39, Dst: 8.8.8.8
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 84
     Identification: 0x2398 (9112)
   ▶ Flags: 0x4000, Don't fragment
     Time to live: 64
     Protocol: ICMP (1)
     Header checksum: 0x2732 [validation disabled]
     [Header checksum status: Unverified]
     Source: 192.168.31.39
     Destination: 8.8.8.8
```

Time-to-live: 64

b. Next examine the Internet Control Message Protocol. What is the ICMP message type?



```
▼ Internet Control Message Protocol
     Type: 8 (Echo (ping) request)
     Code: 0
     Checksum: 0xb56f [correct]
     [Checksum Status: Good]
     Identifier (BE): 34676 (0x8774)
     Identifier (LE): 29831 (0x7487)
     Sequence number (BE): 1 (0x0001)
     Sequence number (LE): 256 (0x0100)
     [Response frame: 1970]
     Timestamp from icmp data: Sep 24, 2018 08:07:27.000000000 PDT
     [Timestamp from icmp data (relative): 0.192191540 seconds]
   ▶ Data (48 bytes)
```

Echo (ping) request

c. What is the message identifier and sequence number?

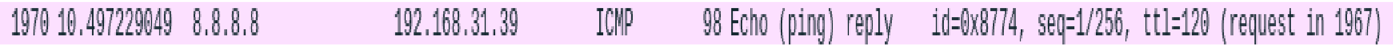
Identifier (BE): 34676 (0x8774)

Identifier (LE): 29831 (0x7487)

Sequence number (BE): 1 (0x0001)

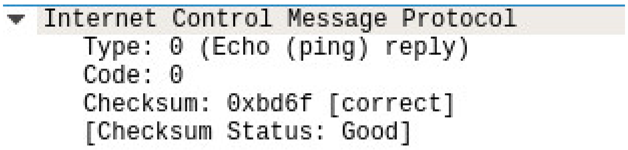
Sequence number (LE): 256 (0x0100)

(2) Find the first ICMP Echo Reply packet.



```
1970 10.497229049  8.8.8.8          192.168.31.39     ICMP     98 Echo (ping) reply    id=0x8774, seq=1/256, ttl=120 (request in 1967)
```

a. Now examine the Internet Control Message Protocol. What is the ICMP message type?

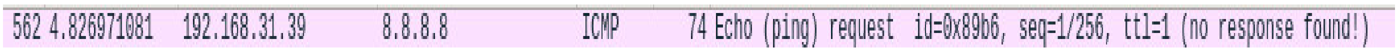


```
▼ Internet Control Message Protocol
     Type: 0 (Echo (ping) reply)
     Code: 0
     Checksum: 0xbd6f [correct]
     [Checksum Status: Good]
```

Echo (ping) reply

2.

(1) Find the first ICMP Echo Request packet.



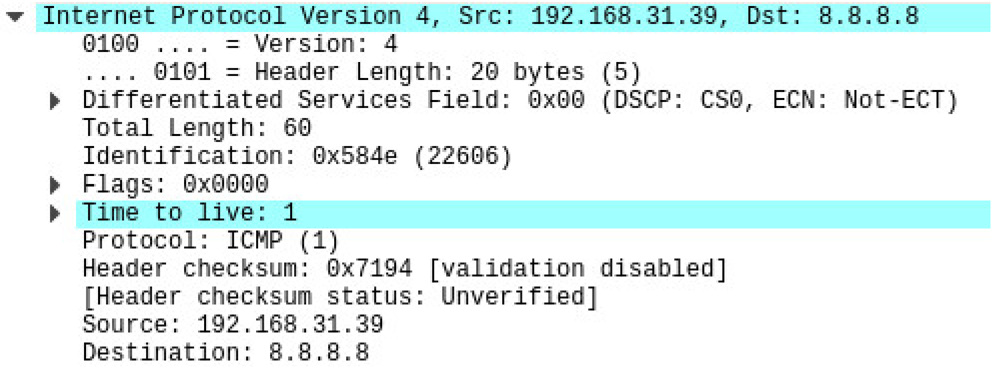
```
562 4.826971081   192.168.31.39        8.8.8.8             ICMP      74 Echo (ping) request  id=0x89b6, seq=1/256, ttl=1 (no response found!)
```

a. Examine the Internet Protocol. What are the source and destination addresses?

Source: 192.168.31.39

Destination: 8.8.8.8

b. What are the protocol type and the Time-to-Live in the IP packet?



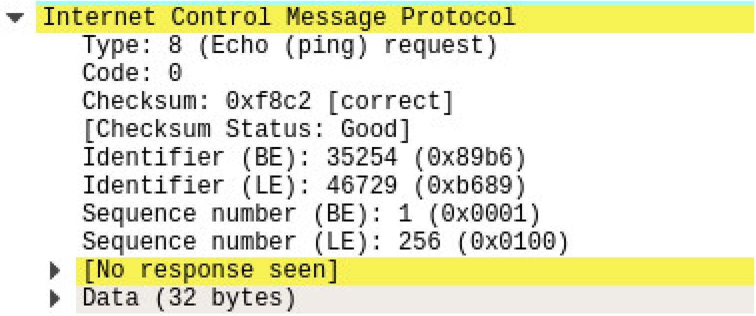
```
▼ Internet Protocol Version 4, Src: 192.168.31.39, Dst: 8.8.8.8
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 60
     Identification: 0x584e (22606)
   ▶ Flags: 0x0000
   ▶ Time to live: 1
     Protocol: ICMP (1)
     Header checksum: 0x7194 [validation disabled]
     [Header checksum status: Unverified]
     Source: 192.168.31.39
     Destination: 8.8.8.8
```

Protocol type: ICMP

Time-to-live: 1

c. Next, examine the Internet Control Message Protocol.

What is the ICMP message type? What are the message identifier and sequence number?



```
▼ Internet Control Message Protocol
     Type: 8 (Echo (ping) request)
     Code: 0
     Checksum: 0xf8c2 [correct]
     [Checksum Status: Good]
     Identifier (BE): 35254 (0x89b6)
     Identifier (LE): 46729 (0xb689)
     Sequence number (BE): 1 (0x0001)
     Sequence number (LE): 256 (0x0100)
   ▶ [No response seen]
   ▶ Data (32 bytes)
```

Type: Echo (ping) request

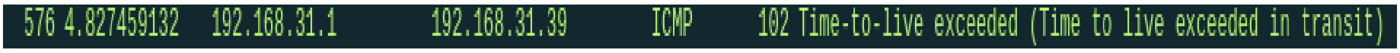
Identifier (BE): 35254 (0x89b6)

Identifier (LE): 46729 (0xb689)

Sequence number (BE): 1 (0x0001)

Sequence number (LE): 256 (0x0100)

(2) Find an ICMP Time-to-live exceeded packet.



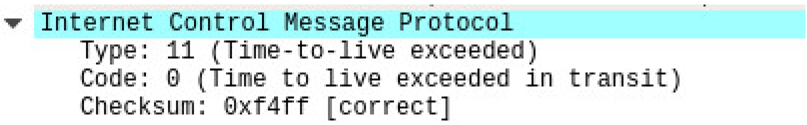
```
576 4.827459132   192.168.31.1         192.168.31.39       ICMP      102 Time-to-live exceeded (Time to live exceeded in transit)
```

a. Examine the Internet Protocol. What are the source and destination addresses?

Source: 192.168.31.1

Destination: 192.168.31.39

b. Next, examine the Internet Control Message Protocol. What is the ICMP message type?

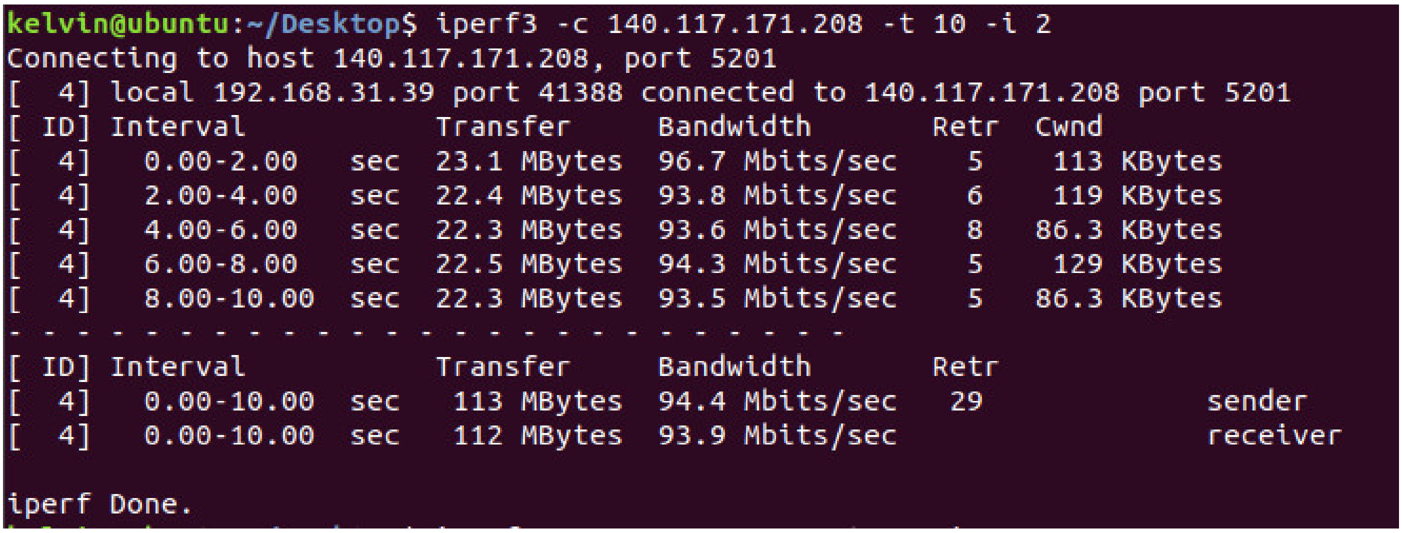


```
▼ Internet Control Message Protocol
     Type: 11 (Time-to-live exceeded)
     Code: 0 (Time to live exceeded in transit)
     Checksum: 0xf4ff [correct]
```

Time-to-live exceeded

Part.III

1. Measure the bandwidth for Transmission Control Protocol Type “iperf3 –c 140.117.171.208 -t 10 -i 2”



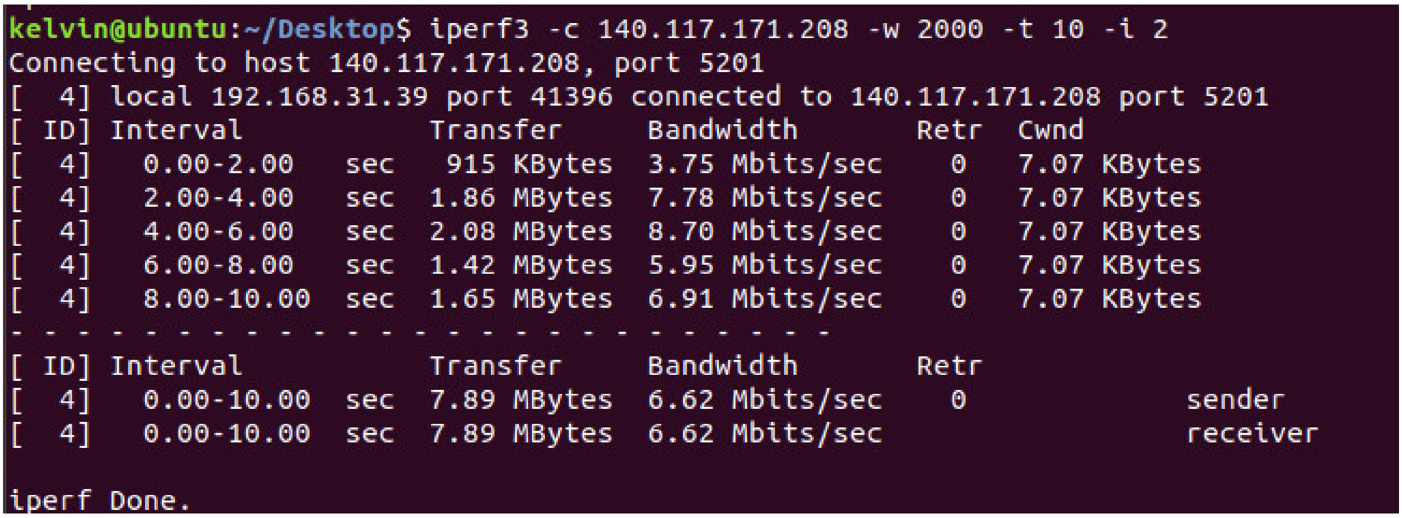
```
kelvin@ubuntu:~/Desktop$ iperf3 -c 140.117.171.208 -t 10 -i 2
Connecting to host 140.117.171.208, port 5201
[  4] local 192.168.31.39 port 41388 connected to 140.117.171.208 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-2.00   sec  23.1 MBytes  96.7 Mbits/sec    5    113 KBytes
[  4]   2.00-4.00   sec  22.4 MBytes  93.8 Mbits/sec    6    119 KBytes
[  4]   4.00-6.00   sec  22.3 MBytes  93.6 Mbits/sec    8   86.3 KBytes
[  4]   6.00-8.00   sec  22.5 MBytes  94.3 Mbits/sec    5    129 KBytes
[  4]   8.00-10.00  sec  22.3 MBytes  93.5 Mbits/sec    5   86.3 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   113 MBytes  94.4 Mbits/sec   29             sender
[  4]   0.00-10.00  sec   112 MBytes  93.9 Mbits/sec                  receiver

iperf Done.
```

Uplink Bandwidth: 94.4Mb

Downlink Bandwidth: 93.9Mb

2. Adjust the window size for Transmission Control Protocol. See what’s different.

Type “iperf3 -c 140.117.171.208 -w 2000 -t 10 -i 2”



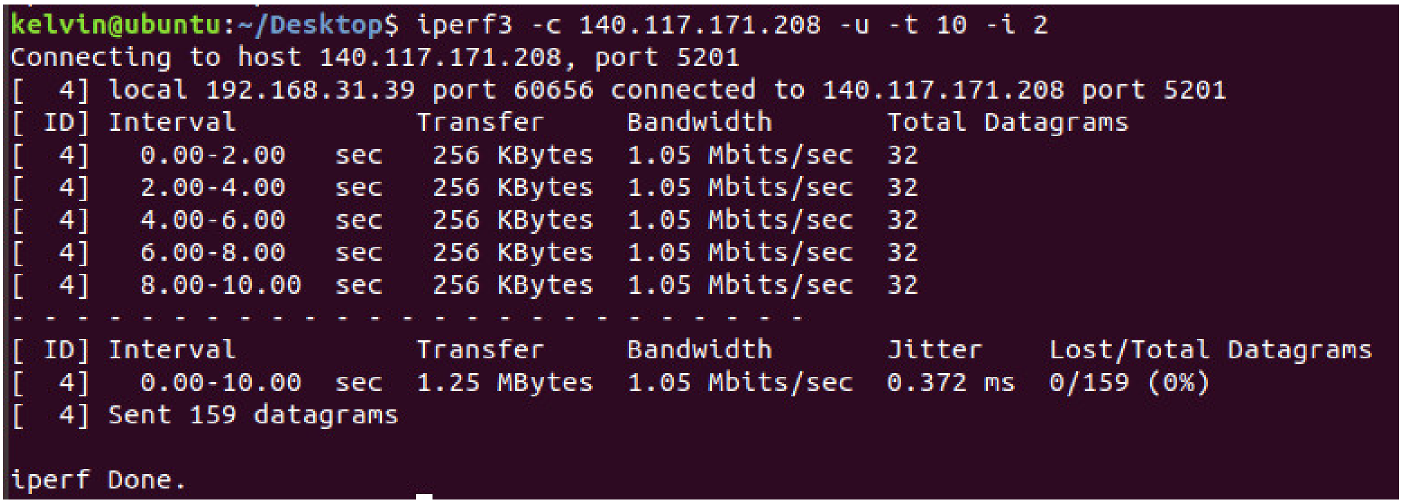
```
kelvin@ubuntu:~/Desktop$ iperf3 -c 140.117.171.208 -w 2000 -t 10 -i 2
Connecting to host 140.117.171.208, port 5201
[  4] local 192.168.31.39 port 41396 connected to 140.117.171.208 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-2.00   sec   915 KBytes  3.75 Mbits/sec    0   7.07 KBytes
[  4]   2.00-4.00   sec  1.86 MBytes  7.78 Mbits/sec    0   7.07 KBytes
[  4]   4.00-6.00   sec  2.08 MBytes  8.70 Mbits/sec    0   7.07 KBytes
[  4]   6.00-8.00   sec  1.42 MBytes  5.95 Mbits/sec    0   7.07 KBytes
[  4]   8.00-10.00  sec  1.65 MBytes  6.91 Mbits/sec    0   7.07 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  7.89 MBytes  6.62 Mbits/sec    0             sender
[  4]   0.00-10.00  sec  7.89 MBytes  6.62 Mbits/sec                  receiver

iperf Done.
```

Uplink Bandwidth: 6.62Mb

Downlink Bandwidth: 6.62Mb

速度遠低於正常狀態。

3. Measure the bandwidth for User Datagram Protocol Type “iperf3 –c 140.117.171.208 -u -t 10 -i 2”

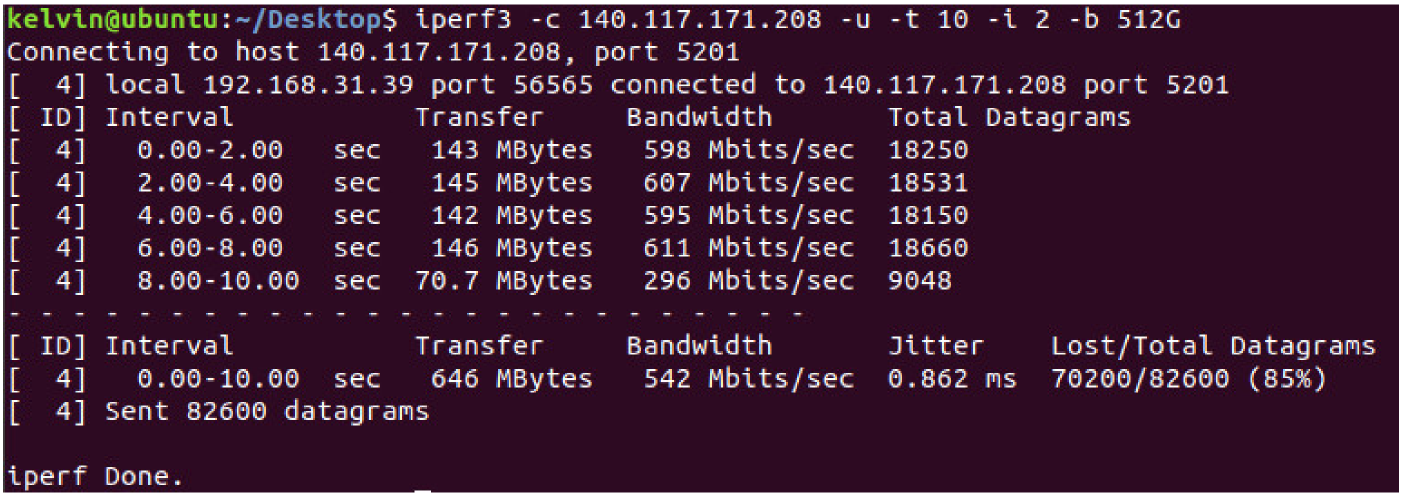


```
kelvin@ubuntu:~/Desktop$ iperf3 -c 140.117.171.208 -u -t 10 -i 2
Connecting to host 140.117.171.208, port 5201
[  4] local 192.168.31.39 port 60656 connected to 140.117.171.208 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-2.00   sec   256 KBytes  1.05 Mbits/sec  32
[  4]   2.00-4.00   sec   256 KBytes  1.05 Mbits/sec  32
[  4]   4.00-6.00   sec   256 KBytes  1.05 Mbits/sec  32
[  4]   6.00-8.00   sec   256 KBytes  1.05 Mbits/sec  32
[  4]   8.00-10.00  sec   256 KBytes  1.05 Mbits/sec  32
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-10.00  sec  1.25 MBytes  1.05 Mbits/sec  0.372 ms  0/159 (0%)
[  4] Sent 159 datagrams

iperf Done.
```

Bandwidth: 1.05Mb

4. Adjust the bandwidth for User Datagram Protocol. Measure the package lost rate or any else happened.

Type “iperf3 -c 140.117.171.208 -u -t 10 -i 2 -b 512G”



```
kelvin@ubuntu:~/Desktop$ iperf3 -c 140.117.171.208 -u -t 10 -i 2 -b 512G
Connecting to host 140.117.171.208, port 5201
[  4] local 192.168.31.39 port 56565 connected to 140.117.171.208 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-2.00   sec   143 MBytes   598 Mbits/sec  18250
[  4]   2.00-4.00   sec   145 MBytes   607 Mbits/sec  18531
[  4]   4.00-6.00   sec   142 MBytes   595 Mbits/sec  18150
[  4]   6.00-8.00   sec   146 MBytes   611 Mbits/sec  18660
[  4]   8.00-10.00  sec  70.7 MBytes   296 Mbits/sec  9048
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-10.00  sec   646 MBytes   542 Mbits/sec  0.862 ms  70200/82600 (85%)
[  4] Sent 82600 datagrams

iperf Done.
```

Bandwidth: 542Mb

Lost/Total Datagrams: 70200/82600 (85%)

封包大量遺失。