به نام خدا

دانشگاه علم و صنعت ایران

درس شبکه‌های کامپیوتری

تمرین سری چهارم

مدرس درس:

سرکار خانم دکتر موحدی

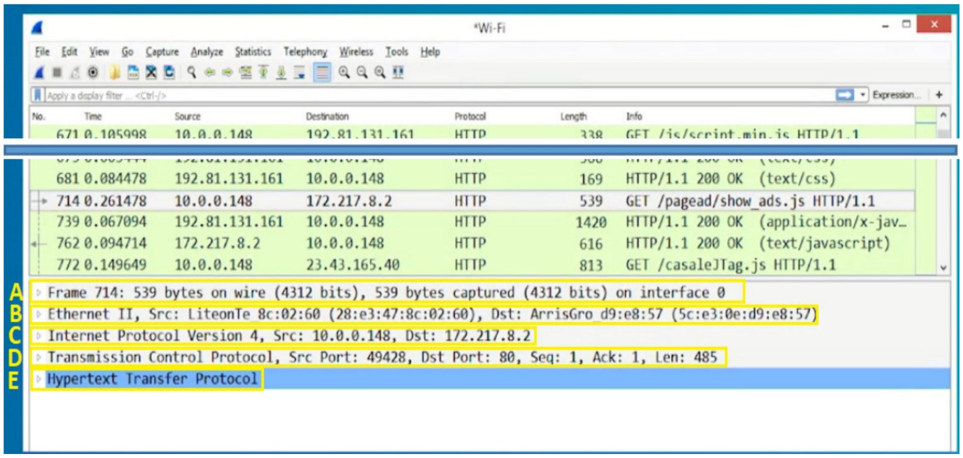
تهیه شده توسط:

الناز رضایی ۹۸۴۱۱۳۸۷

تاریخ ارسال: ۱۴۰۲/۰۲/۲۵

Question 1)

The following picture depicts the HTTP packets:



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

```
No.   Time       Source           Destination      Protocol   Length  Info
  671 0.105998   10.0.0.148       192.81.131.161   HTTP          338  GET /js/script.min.js HTTP/1.1
  681 0.084478   192.81.131.161   10.0.0.148       HTTP          169  HTTP/1.1 200 OK  (text/css)
  714 0.261478   10.0.0.148       172.217.8.2      HTTP          539  GET /pagead/show_ads.js HTTP/1.1
  739 0.067094   192.81.131.161   10.0.0.148       HTTP         1420  HTTP/1.1 200 OK  (application/x-jav…
  762 0.094714   172.217.8.2      10.0.0.148       HTTP          616  HTTP/1.1 200 OK  (text/javascript)
  772 0.149649   10.0.0.148       23.43.165.40     HTTP          813  GET /casaleJTag.js HTTP/1.1
```

```
A  Frame 714: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits) on interface 0
B  Ethernet II, Src: LiteonTe_8c:02:60 (28:e3:47:8c:02:60), Dst: ArrisGro_d9:e8:57 (5c:e3:0e:d9:e8:57)
C  Internet Protocol Version 4, Src: 10.0.0.148, Dst: 172.217.8.2
D  Transmission Control Protocol, Src Port: 49428, Dst Port: 80, Seq: 1, Ack: 1, Len: 485
E  Hypertext Transfer Protocol
```

In frame 714, which section in packet details represents the frame header? Why? (Choose between A, B, C, D and E) (Multiple choice could be correct)

Answer:

Section B represents the frame headers. This occurs because HTTP frames are transmitted over the network using the Ethernet protocol, and the Ethernet II header encompasses crucial details like the source and destination MAC addresses, as well as other Ethernet-specific information such as the protocol type (in this case, HTTP). As a result, when analyzing packet details in Wireshark, the Ethernet II header serves as the initial segment that represents the frame header within an HTTP frame.

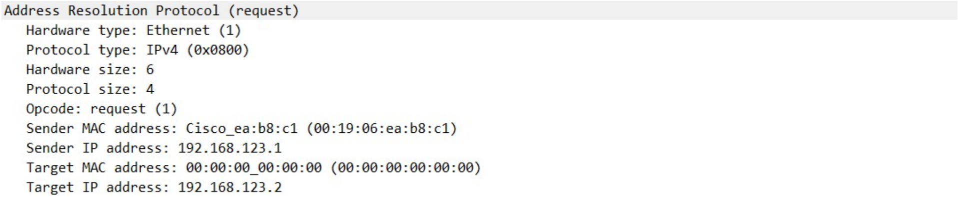
Question 2)

Open file `'ICMP_across_dot1q.cap'`

- A) In packet list, find packet from `'ICMP_across_dot1q.cap'`. What is the source and destination MAC address of this request and the corresponding reply?
- B) In packet list, how many ICMP request packets do you see?

Answer:

- A) As you can see in Figre 1 below, the MAC address of the source and destination of this request is as follows:



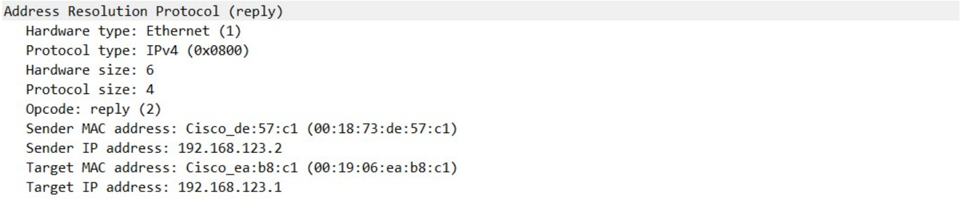
```
Address Resolution Protocol (request)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: request (1)
   Sender MAC address: Cisco_ea:b8:c1 (00:19:06:ea:b8:c1)
   Sender IP address: 192.168.123.1
   Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
   Target IP address: 192.168.123.2
```

Figure 1: ICMP Request

Source MAC address: (00:19:06:ea:b8:c1)

Destination MAC address: (00:00:00:00:00:00)

Also you can see source and destination MAC address for the corresponding reply in Figure 2.



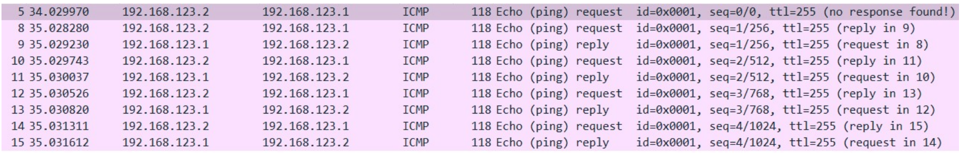
```
Address Resolution Protocol (reply)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: reply (2)
   Sender MAC address: Cisco_de:57:c1 (00:18:73:de:57:c1)
   Sender IP address: 192.168.123.2
   Target MAC address: Cisco_ea:b8:c1 (00:19:06:ea:b8:c1)
   Target IP address: 192.168.123.1
```

Figure 2: ICMP Reply

Source MAC address: (00:18:73:de:57:c1)

Destination MAC address: (00:19:06:ea:b8:c1)

- B) By filtering ICMP packets, we find that the number of ICMP request packets is 5. (Figure 3)



```
 5 34.029970   192.168.123.2   192.168.123.1   ICMP   118 Echo (ping) request  id=0x0001, seq=0/0, ttl=255 (no response found!)
 8 35.028280   192.168.123.2   192.168.123.1   ICMP   118 Echo (ping) request  id=0x0001, seq=1/256, ttl=255 (reply in 9)
 9 35.029230   192.168.123.1   192.168.123.2   ICMP   118 Echo (ping) reply    id=0x0001, seq=1/256, ttl=255 (request in 8)
10 35.029743   192.168.123.2   192.168.123.1   ICMP   118 Echo (ping) request  id=0x0001, seq=2/512, ttl=255 (reply in 11)
11 35.030037   192.168.123.1   192.168.123.2   ICMP   118 Echo (ping) reply    id=0x0001, seq=2/512, ttl=255 (request in 10)
12 35.030526   192.168.123.2   192.168.123.1   ICMP   118 Echo (ping) request  id=0x0001, seq=3/768, ttl=255 (reply in 13)
13 35.030820   192.168.123.1   192.168.123.2   ICMP   118 Echo (ping) reply    id=0x0001, seq=3/768, ttl=255 (request in 12)
14 35.031311   192.168.123.2   192.168.123.1   ICMP   118 Echo (ping) request  id=0x0001, seq=4/1024, ttl=255 (reply in 15)
15 35.031612   192.168.123.1   192.168.123.2   ICMP   118 Echo (ping) reply    id=0x0001, seq=4/1024, ttl=255 (request in 14)
```

Figure 3: ICMP Request

Question 3)

According to the previous question, find ICMP request/reply packets.
According to this packet list, obtain the following parameters:
According to this log, obtain the following parameters:

- A) What is minimum round-trip time in milliseconds?

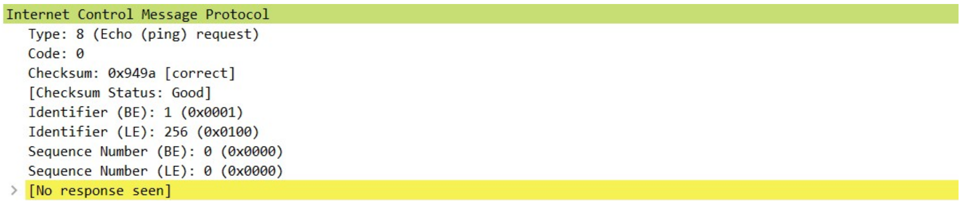
- B) What is maximum round-trip time in milliseconds?

- C) What is average round trip time in milliseconds?

Answer:

To calculate the round trip time, it is enough to subtract the arrival time of the request packet and the reply packet.

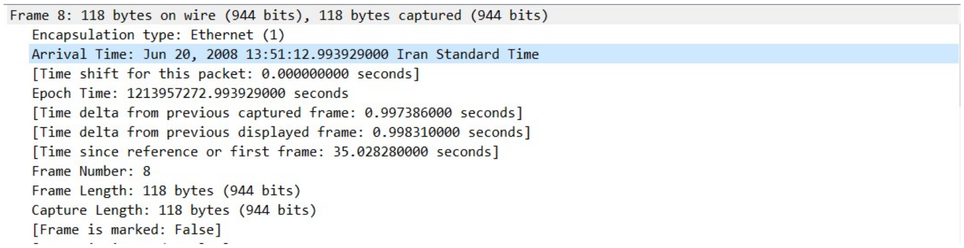
- Frame number 5 does not have any response, so we can not calculate round-trip time for this frame.



```
Internet Control Message Protocol
   Type: 8 (Echo (ping) request)
   Code: 0
   Checksum: 0x949a [correct]
   [Checksum Status: Good]
   Identifier (BE): 1 (0x0001)
   Identifier (LE): 256 (0x0100)
   Sequence Number (BE): 0 (0x0000)
   Sequence Number (LE): 0 (0x0000)
 > [No response seen]
```

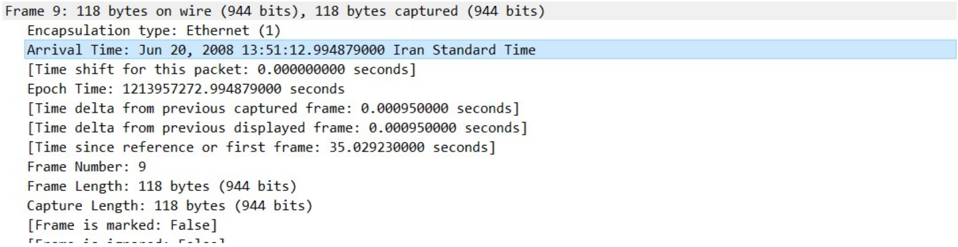
Figure 4: Frame 5

- Frame 8: Arrival time: 12.993929 , Frame 9: Arrival time: 12.994879



```
Frame 8: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.993929000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.993929000 seconds
   [Time delta from previous captured frame: 0.997386000 seconds]
   [Time delta from previous displayed frame: 0.998310000 seconds]
   [Time since reference or first frame: 35.028280000 seconds]
   Frame Number: 8
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
```

Figure 5: Frame 8 (request)

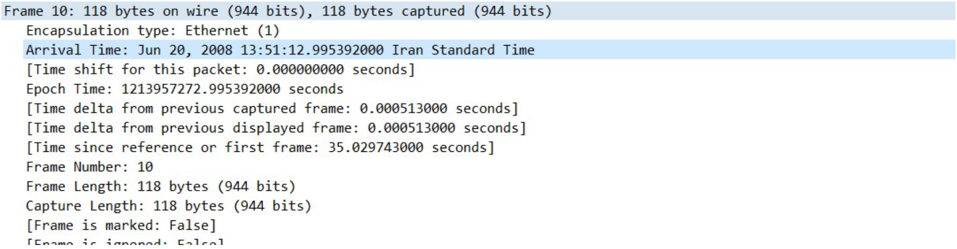


```
Frame 9: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.994879000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.994879000 seconds
   [Time delta from previous captured frame: 0.000950000 seconds]
   [Time delta from previous displayed frame: 0.000950000 seconds]
   [Time since reference or first frame: 35.029230000 seconds]
   Frame Number: 9
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
```

Figure 6: Frame 9 (reply)

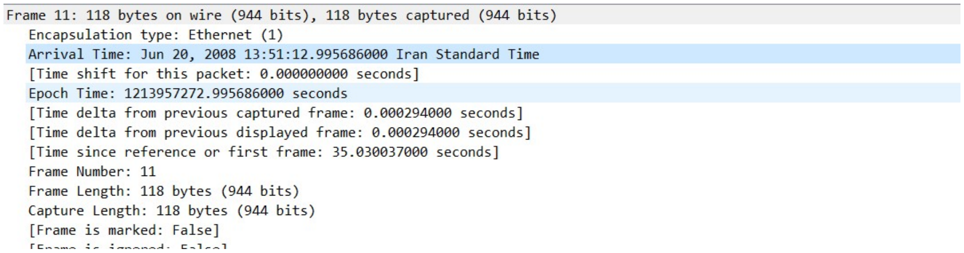
round trip time = 12.994879 - 12.993929 = 0.00095

- Frame 10: Arrival time: 12.995392 , Frame 11: Arrival time: 12.995686



```
Frame 10: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.995392000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.995392000 seconds
   [Time delta from previous captured frame: 0.000513000 seconds]
   [Time delta from previous displayed frame: 0.000513000 seconds]
   [Time since reference or first frame: 35.029743000 seconds]
   Frame Number: 10
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
```

Figure 7: Frame 10 (request)

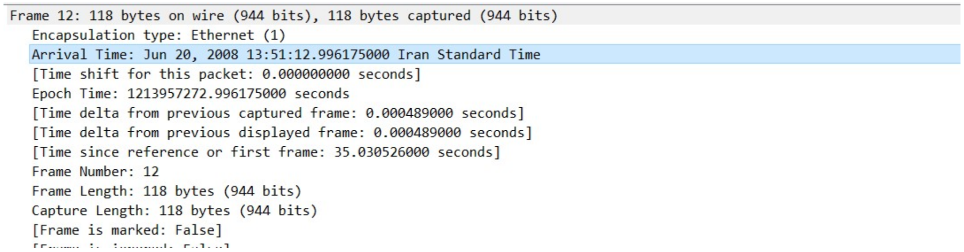


```
Frame 11: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.995686000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.995686000 seconds
   [Time delta from previous captured frame: 0.000294000 seconds]
   [Time delta from previous displayed frame: 0.000294000 seconds]
   [Time since reference or first frame: 35.030037000 seconds]
   Frame Number: 11
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
```

Figure 8: Frame 11 (reply)

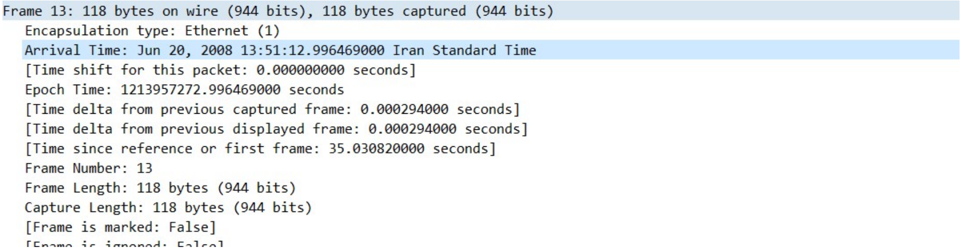
round trip time = 12.995686 - 12.995392 = 0.000294

- Frame 12: Arrival time: 12.996175 , Frame 13: Arrival time: 12.996469



```
Frame 12: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.996175000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.996175000 seconds
   [Time delta from previous captured frame: 0.000489000 seconds]
   [Time delta from previous displayed frame: 0.000489000 seconds]
   [Time since reference or first frame: 35.030526000 seconds]
   Frame Number: 12
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
```

Figure 9: Frame 12 (request)

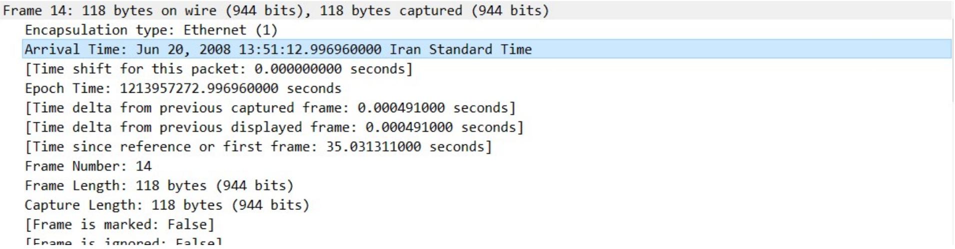


```
Frame 13: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.996469000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.996469000 seconds
   [Time delta from previous captured frame: 0.000294000 seconds]
   [Time delta from previous displayed frame: 0.000294000 seconds]
   [Time since reference or first frame: 35.030820000 seconds]
   Frame Number: 13
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
```

Figure 10: Frame 13 (reply)

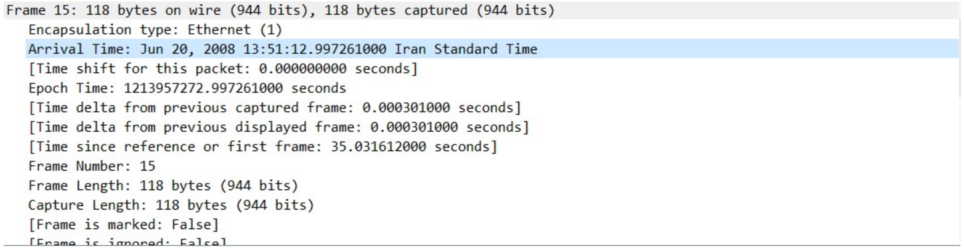
round trip time = 12.996469 - 12.996175 = 0.000294

- Frame 14: Arrival time: 12.996960 , Frame 15: Arrival time: 12.997261



```
Frame 14: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.996960000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.996960000 seconds
   [Time delta from previous captured frame: 0.000491000 seconds]
   [Time delta from previous displayed frame: 0.000491000 seconds]
   [Time since reference or first frame: 35.031311000 seconds]
   Frame Number: 14
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
```

Figure 11: Frame 14 (request)



```
Frame 15: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
   Encapsulation type: Ethernet (1)
   Arrival Time: Jun 20, 2008 13:51:12.997261000 Iran Standard Time
   [Time shift for this packet: 0.000000000 seconds]
   Epoch Time: 1213957272.997261000 seconds
   [Time delta from previous captured frame: 0.000301000 seconds]
   [Time delta from previous displayed frame: 0.000301000 seconds]
   [Time since reference or first frame: 35.031612000 seconds]
   Frame Number: 15
   Frame Length: 118 bytes (944 bits)
   Capture Length: 118 bytes (944 bits)
   [Frame is marked: False]
   [Frame is ignored: False]
```

Figure 12: Frame 15 (reply)

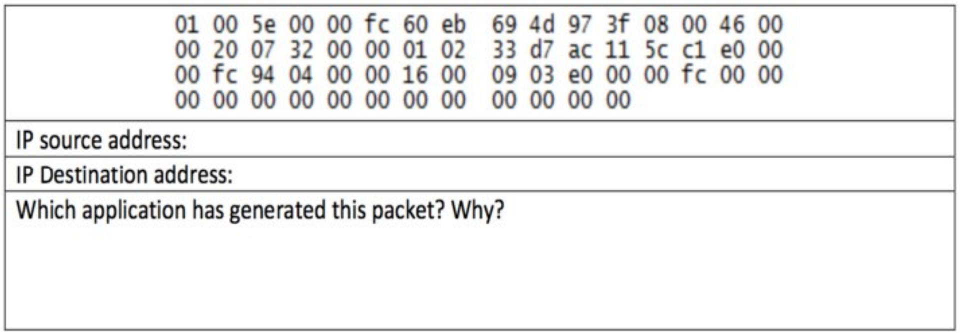
round trip time = 12.9972619 - 12.996960 = 0.000301

- A) Minimum round-trip time: 0.000294
- B) Maximum round-trip time: 0.00095
- C) Average round trip time: 0.000459975

Question 4)

According to the following captured packets, fill requested fields:

Note: for each packet, the first 14 Bytes are the Ethernet header.

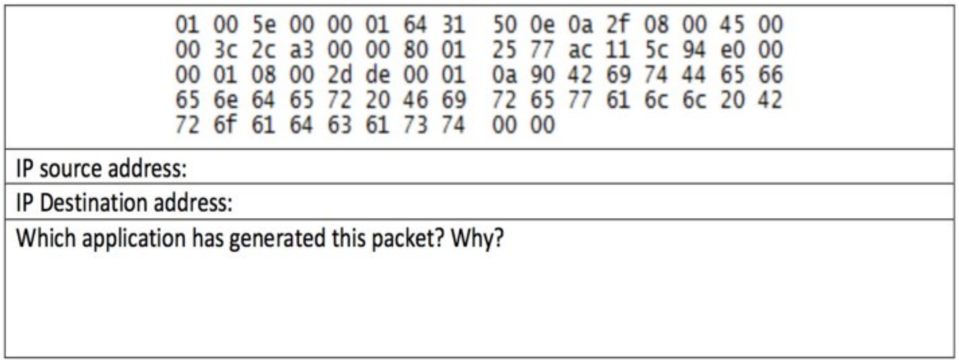


```
01 00 5e 00 00 fc 60 eb  69 4d 97 3f 08 00 46 00
00 20 07 32 00 00 01 02  33 d7 ac 11 5c c1 e0 00
00 fc 94 04 00 00 16 00  09 03 e0 00 00 fc 00 00
00 00 00 00 00 00 00 00  00 00 00 00
```

IP source address:

IP Destination address:

Which application has generated this packet? Why?



```
01 00 5e 00 00 01 64 31  50 0e 0a 2f 08 00 45 00
00 3c 2c a3 00 00 80 01  25 77 ac 11 5c 94 e0 00
00 01 08 00 2d de 00 01  0a 90 42 69 74 44 65 66
65 6e 64 65 72 20 46 69  72 65 77 61 6c 6c 20 42
72 6f 61 64 63 61 73 74  00 00
```

IP source address:

IP Destination address:

Which application has generated this packet? Why?

Answer:

- First image:

  IP Source address: ac 11 5c c1 = 172.17.92.193

  IP Destination address: e0 00 00 fc = 224.0.0.252

  Which appliccation has generated this packet? Why? The value of the protocol field, located at byte 24, is determined to be 2, indicating the utilization of the IGMP protocol (Internet Group Management Protocol).

- Second image:

  IP Source address: ac 11 5c 94 = 172.17.92.148

  IP Destination address: e0 00 00 01 = 224.0.0.1

  Which appliccation has generated this packet? Why? The value of the protocol field, located at byte 24, is determined to be 1, indicating the utilization of the ICMP protocol (Internet Control Message Protocol).

Question 5)

Open your Wireshark program and select ICMP as filter. Start capturing packets on your interface (en0, wlan0, etc.). Now open up your terminal (Linux) or command prompt (windows) and use one of the following commands:

Windows: ping –n 10 8.8.8.8

Linux: ping –c 10 8.8.8.8

- A) Why is it that an ICMP packet does not have source and destination port numbers?
- B) Examine one of the ping request packets sent by your host. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?
- C) Examine the corresponding ping reply packet. What are the ICMP type and code numbers? What other fields does this ICMP packet have? How many bytes are the checksum, sequence number and identifier fields?

Answer:

- A) The ICMP packet does not have source and destination port numbers because it was designed to communicate network-layer information between hosts and routers, not between application layer processes. If this were application layer, it would have source and destination port numbers. However, this is using network layer so those are not needed.

- B) The ICMP type and code numbers for a ping request packet sent by the host are Type 8 (Echo Request) and Code 0. In addition to the type and code fields, an ICMP Echo Request packet contains other fields such as:

  1. Checksum: A 2-byte field used for error detection.
  2. Identifier: A 2-byte field that helps match Echo Request and Echo Reply packets.
  3. Sequence Number: A 2-byte field indicating the order of the Echo Request packets.

  The sizes of the checksum, sequence number, and identifier fields in ICMP packets are 2 bytes each.

- C) The ICMP type and code numbers for a ping reply packet, in response to a ping request, are Type 0 (Echo Reply) and Code 0. Similar to the Echo Request packet, an ICMP Echo Reply packet contains fields such as:

  1. Checksum: A 2-byte field used for error detection.
  2. Identifier: The same 2-byte field from the Echo Request packet.
  3. Sequence Number: The same 2-byte field from the Echo Request packet.

  The sizes of the checksum, sequence number, and identifier fields in ICMP packets are 2 bytes each.