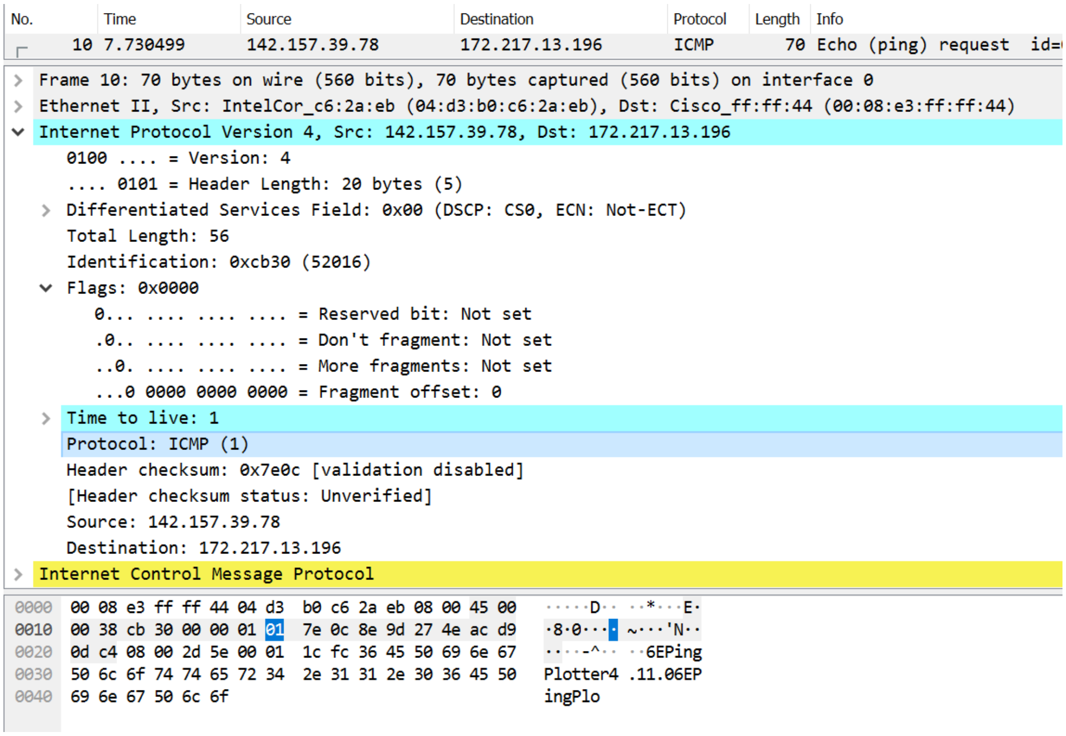
COMP 535 Assignment 2

Jingyuan Wang 260860682

March 20, 2019

Excercise 1

1. My IP address is 142.157.39.78.



```
No.     Time        Source            Destination       Protocol  Length Info
     10 7.730499    142.157.39.78     172.217.13.196    ICMP          70 Echo (ping) request  id=

  Frame 10: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
  Ethernet II, Src: IntelCor_c6:2a:eb (04:d3:b0:c6:2a:eb), Dst: Cisco_ff:ff:44 (00:08:e3:ff:ff:44)
  Internet Protocol Version 4, Src: 142.157.39.78, Dst: 172.217.13.196
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
    Identification: 0xcb30 (52016)
    Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x7e0c [validation disabled]
    [Header checksum status: Unverified]
    Source: 142.157.39.78
    Destination: 172.217.13.196
  Internet Control Message Protocol

0000  00 08 e3 ff ff 44 04 d3  b0 c6 2a eb 08 00 45 00   ·····D·· ··*···E·
0010  00 38 cb 30 00 00 01 01  7e 0c 8e 9d 27 4e ac d9   ·8·0···· ~···'N··
0020  0d c4 08 00 2d 5e 00 01  1c fc 36 45 50 69 6e 67   ····-^·· ··6EPing
0030  50 6c 6f 74 74 65 72 34  2e 31 31 2e 30 36 45 50   Plotter4 .11.06EP
0040  69 6e 67 50 6c 6f                                  ingPlo
```

2. 01 (ICMP Protocol)

3. 20 bytes in IP header. 36 bytes for payload. The field *Total Length* 56 indicates the total length of header and IP payload. Subtract the header length 20 from 56, we get payload length, which is 36 bytes. It can also be counted from the rest data contents of IP datagram(i.e., Internet Control Message Protocol section).

4. No. With field *More Fragments* equal to 0, we know this datagram is the last fragment. On the other hand, *Fragment offset* indicates the starting byte position of the original IP datagram's data this fragment provides. So the fragment offset 0 in our case means that this datagram contains the first byte of original datagram's data and so is the first fragment. As a result, this datagram is not fragmented.

5. *Identification*, *Time to live* and *Header checksum.*

6. constant: *src IP*, *dest IP*, *Version*, *Type of Service*, *Header Length*, *Total Length*, *Upper Layer Protocol* and *Flags.*

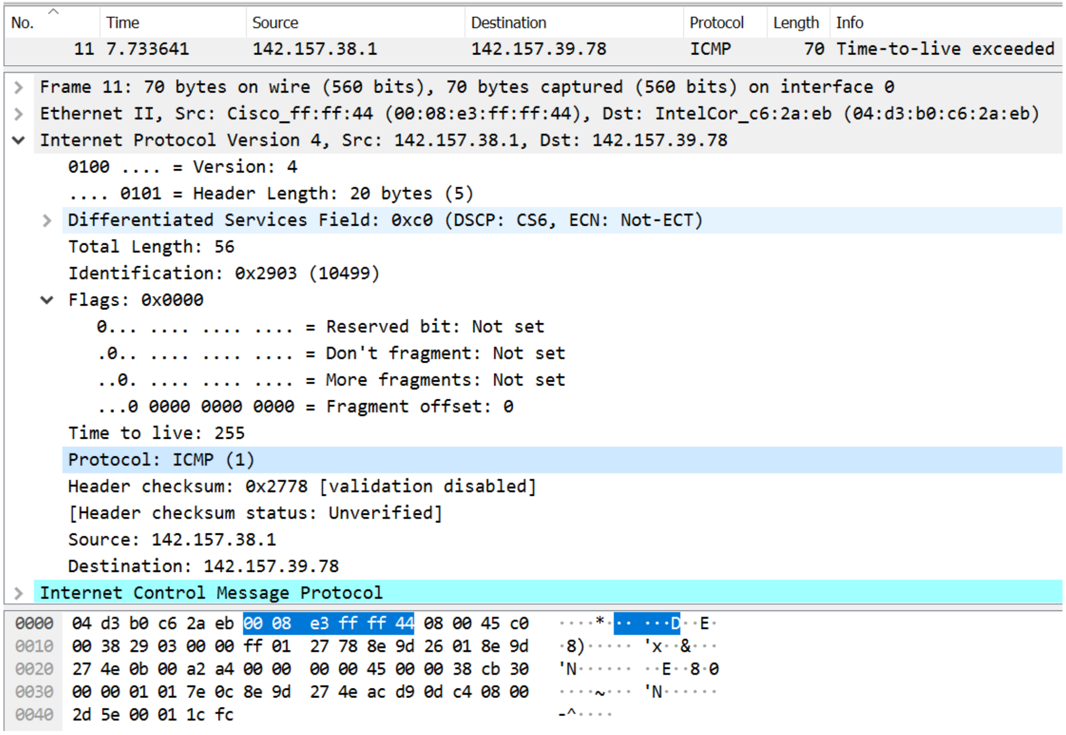
must constant: *src IP*, *dest IP*, *Version*, *Type of Service*, *Header Length*, *Upper Layer Protocol.*

must change: *Identification*, *Header checksum.*

reason: The fields in 'must constant' are all related to IP protocol configurations and host IP addresses so they must remain the same through the whole traceroute process. However, our attempt to change traceroute packet size leads to inconsistency of *Total Length* field and packet fragmentation makes *Fragment offset* in *Flags* section to change. Moreover, *Identification* is a unique number assigned to each IP datagram so it changes from on packet to another. *Header checksum*, which is used for data corruption checking, is calculated by all bits from datagram header, so it would change if any section of the header changes.

7. *Identification* is decremented by 1 as we move ICMP messages downwards.

8. *Identification*: 0x2903(10499)
TTL:255



No.	Time	Source	Destination	Protocol	Length	Info
11	7.733641	142.157.38.1	142.157.39.78	ICMP	70	Time-to-live exceeded

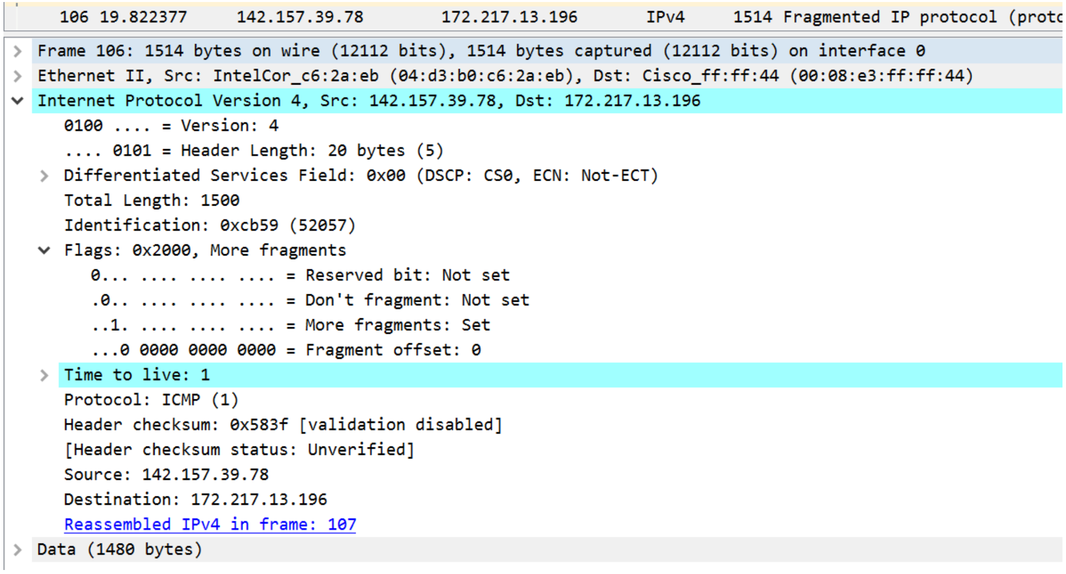
```
> Frame 11: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Cisco_ff:ff:44 (00:08:e3:ff:ff:44), Dst: IntelCor_c6:2a:eb (04:d3:b0:c6:2a:eb)
v Internet Protocol Version 4, Src: 142.157.38.1, Dst: 142.157.39.78
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xc0 (DSCP: CS6, ECN: Not-ECT)
    Total Length: 56
    Identification: 0x2903 (10499)
  v Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 255
    Protocol: ICMP (1)
    Header checksum: 0x2778 [validation disabled]
    [Header checksum status: Unverified]
    Source: 142.157.38.1
    Destination: 142.157.39.78
> Internet Control Message Protocol

0000  04 d3 b0 c6 2a eb 00 08  e3 ff ff 44 08 00 45 c0   ····*·······D··E·
0010  00 38 29 03 00 00 ff 01  27 78 8e 9d 26 01 8e 9d   ·8)·····'x··&···
0020  27 4e 0b 00 a2 a4 00 00  00 00 45 00 00 38 cb 30   'N········E··8·0
0030  00 00 01 01 7e 0c 8e 9d  27 4e ac d9 0d c4 08 00   ····~···'N······
0040  2d 5e 00 01 1c fc                                  -^····
```

9. *Identification* field will change since it is unique to each IP datagram. On the contrary, *TTL* does not change. The default TTL of IP packet is set to 255, and packets from my nearest router to me always take one hop and the header shows 255 in this field.

10. Yes.(Screen shot shown below)

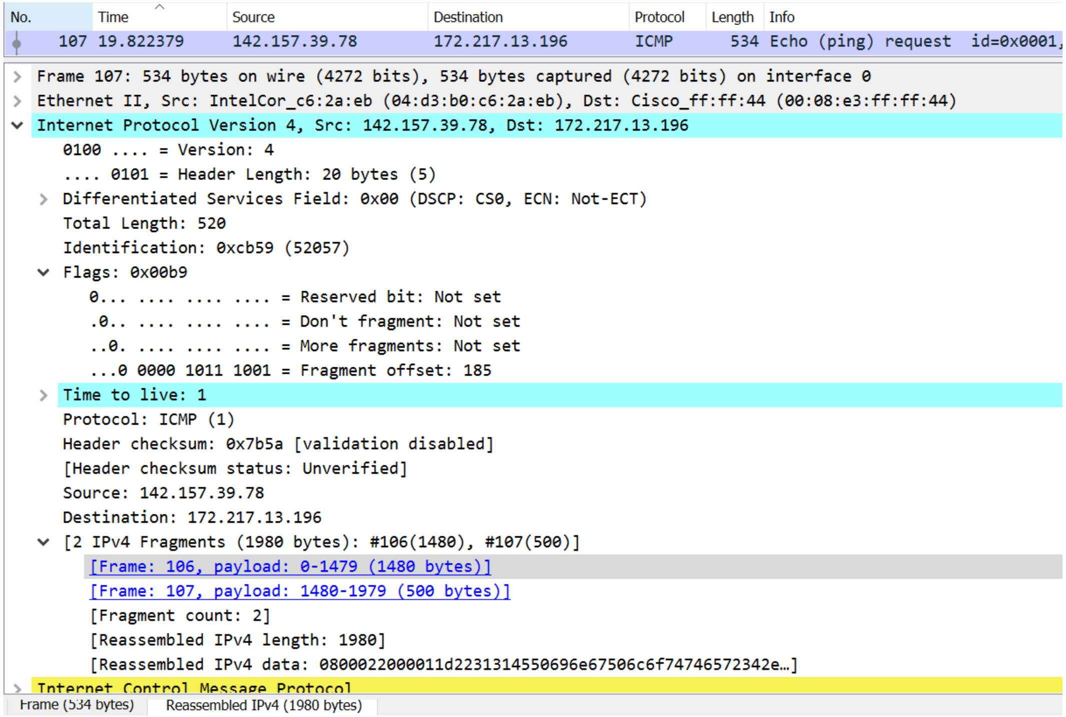
11. *More Fragments* field is set to 1, indicating there is fragmentation with this IP datagram. *Fragment offset* is 0, meaning this is the first fragment. This whole IP datagram has a length of

2,000 bytes.



```
  106 19.822377     142.157.39.78         172.217.13.196        IPv4     1514 Fragmented IP protocol (protc
> Frame 106: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0
> Ethernet II, Src: IntelCor_c6:2a:eb (04:d3:b0:c6:2a:eb), Dst: Cisco_ff:ff:44 (00:08:e3:ff:ff:44)
v Internet Protocol Version 4, Src: 142.157.39.78, Dst: 172.217.13.196
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0xcb59 (52057)
  v Flags: 0x2000, More fragments
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..1. .... .... .... = More fragments: Set
      ...0 0000 0000 0000 = Fragment offset: 0
  > Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x583f [validation disabled]
    [Header checksum status: Unverified]
    Source: 142.157.39.78
    Destination: 172.217.13.196
    Reassembled IPv4 in frame: 107
> Data (1480 bytes)
```

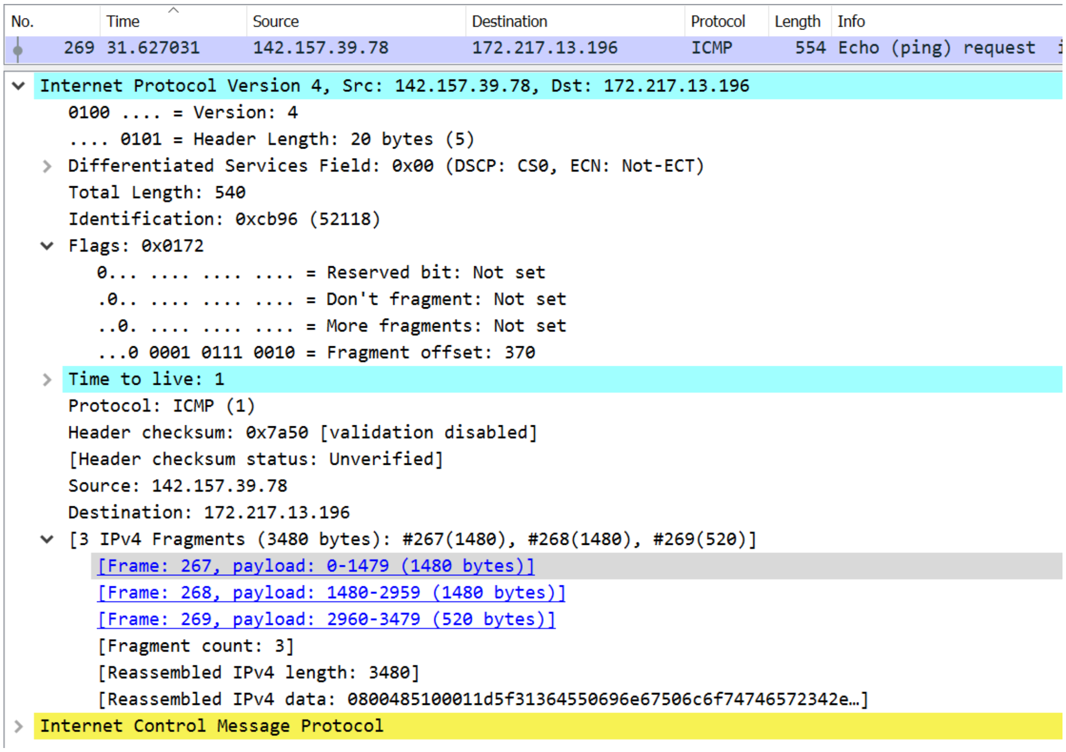
12. The *Fragment offset* is not equal to 0. Instead it is showing position of the first byte in the original IP datagram, indicating it is not the first fragment. With *More Fragments* set to 0, it shows there is no further fragments.



```
No.      Time           Source              Destination          Protocol   Length  Info
    107 19.822379      142.157.39.78       172.217.13.196       ICMP        534 Echo (ping) request  id=0x0001,

> Frame 107: 534 bytes on wire (4272 bits), 534 bytes captured (4272 bits) on interface 0
> Ethernet II, Src: IntelCor_c6:2a:eb (04:d3:b0:c6:2a:eb), Dst: Cisco_ff:ff:44 (00:08:e3:ff:ff:44)
v Internet Protocol Version 4, Src: 142.157.39.78, Dst: 172.217.13.196
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 520
    Identification: 0xcb59 (52057)
  v Flags: 0x00b9
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 1011 1001 = Fragment offset: 185
  > Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x7b5a [validation disabled]
    [Header checksum status: Unverified]
    Source: 142.157.39.78
    Destination: 172.217.13.196
  v [2 IPv4 Fragments (1980 bytes): #106(1480), #107(500)]
      [Frame: 106, payload: 0-1479 (1480 bytes)]
      [Frame: 107, payload: 1480-1979 (500 bytes)]
      [Fragment count: 2]
      [Reassembled IPv4 length: 1980]
      [Reassembled IPv4 data: 0800022000011d2231314550696e67506c6f74746572342e…]
> Internet Control Message Protocol
 Frame (534 bytes)    Reassembled IPv4 (1980 bytes)
```

13. *Flags*, *Total Length* and *Header checksum*

14. 3 fragments.



```
No.   Time          Source          Destination      Protocol  Length  Info
  269 31.627031     142.157.39.78   172.217.13.196   ICMP         554  Echo (ping) request  i

∨ Internet Protocol Version 4, Src: 142.157.39.78, Dst: 172.217.13.196
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 540
    Identification: 0xcb96 (52118)
  ∨ Flags: 0x0172
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0001 0111 0010 = Fragment offset: 370
  > Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x7a50 [validation disabled]
    [Header checksum status: Unverified]
    Source: 142.157.39.78
    Destination: 172.217.13.196
  ∨ [3 IPv4 Fragments (3480 bytes): #267(1480), #268(1480), #269(520)]
      [Frame: 267, payload: 0-1479 (1480 bytes)]
      [Frame: 268, payload: 1480-2959 (1480 bytes)]
      [Frame: 269, payload: 2960-3479 (520 bytes)]
      [Fragment count: 3]
      [Reassembled IPv4 length: 3480]
      [Reassembled IPv4 data: 0800485100011d5f31364550696e67506c6f74746572342e…]
> Internet Control Message Protocol
```

15. *Flags*, *Total Length* and *Header checksum*

Excercise 2

1. UDP provides communication with port numbers so different user request now can be distinguished such as web browser and e-mails. Another service is data integrity verification using checksums.

2. We want UDP rather than TCP when transmission speed is more important than reliability and tolerance for lantency is low. It is often used in gaming and video communications.

3. Flow Control involves only one TCP communication with one sender and one receiver and prevents the receiver from being overwhelmed by the sender. The sender maintains a *Receiver Window* for detecting how many segments the receiver is able to receive at each moment and makes sure sender itself is transmitting data no larger than the size of *Receiver Window.*

Congestion Control, on the other hand, is a global network controlling method. TCP maintains *Congestion Windows* to limit the total number of packets transmitting in the whole network so as to prevent congestion.

Firstly TCP used a *Slow Start* mechanism to increase the speed of transmission exponentially from 1 MSS per time when communication is initialized or a timeout is detected. If the sending speed is larger than a slow start threshold, the network increases the transmitting speed linearly, which is the Congestion Avoidance phase. If a packet loss is detected(indicated by timeout), the congestion window is set to 1 MSS, threshold is halved and begin another slow start. Some of the TCP implementation provides *Fast Recovery* scheme, it sees receiving 3 duplicate ACKs as congestion signal instead of a timeout. When 3 duplicated ACKs are detected, threshold is still halved, congestion windows is set to threshold plus 3 MSS as *Fast Recovery* and then a *Congestion Avoidance* is carried out.

4. According to $EstimatedRTT_{new} = \alpha \cdot EstimatedRTT + (1 - \alpha) \cdot SampleRTT$,
after 20ms:$0.125 \times 30 + (1 - 0.125) \times 20 = 21.25ms$
after 30ms:$0.125 \times 21.25 + (1 - 0.125) \times 30 = 28.9063ms$
after 25ms:$0.125 \times 28.9063 + (1 - 0.125) \times 25 = 25.4883ms$

Excercise 3

a. 2^{32}bytes. Maximum number of L is the largest number in 4 bytes(32 bits) sequence field.

b. $\frac{\left\lceil \frac{2^{32}}{536} \right\rceil \times 66 + 2^{32} bytes}{155Mbps} = 248.9716s$