COMPUTER NETWORKS

SAKSHAM KUMAR

COMPUTER SCIENCE AND ENGINEERING
ID - 2022UCP1700
SECTION- A4

ASSIGNMENT – 10

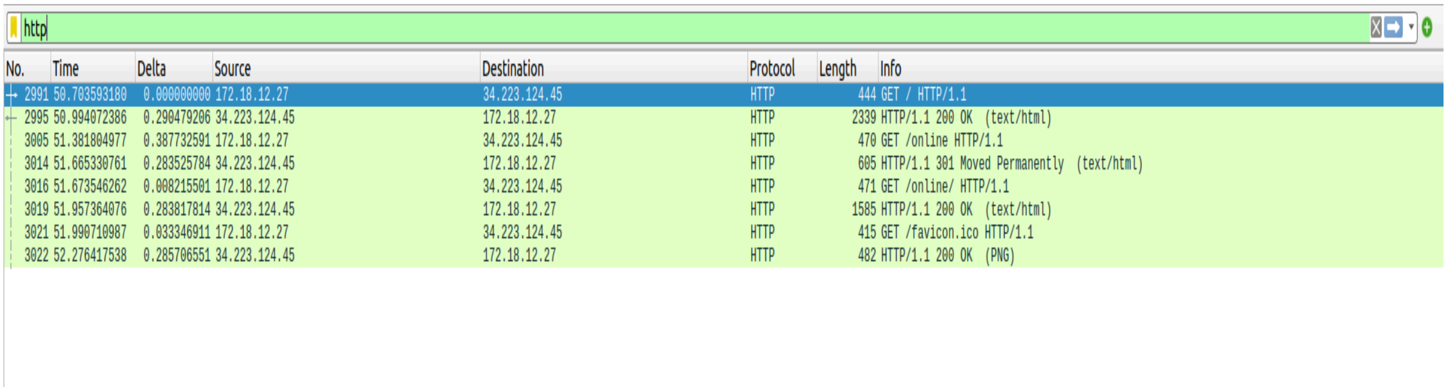
1. List 5 different protocols that appear in the protocol column in the packet-listing window. Also describe each protocol in short.

1. HTTP (Hypertext Transfer Protocol):
   - ○ HTTP is the foundation of data communication on the World Wide Web. It is a protocol used for transferring hypertext requests and information between servers and browsers.
2. TCP (Transmission Control Protocol):
   - ○ TCP is a standard protocol that establishes a connection between two hosts and ensures reliable data delivery by managing the sequencing, acknowledgments, and error checking of packets.
3. UDP (User Datagram Protocol):
   - ○ UDP is a connectionless protocol that provides a simple way for applications to send datagrams across a network. It is often used for time-sensitive applications where loss of individual packets is acceptable, such as streaming media or online gaming.
4. DNS (Domain Name System):
   - ○ DNS is a hierarchical decentralized naming system for computers, services, or any resource connected to the Internet or a private network. It translates domain names into IP addresses, facilitating the retrieval of resources requested by a user.
5. SMTP (Simple Mail Transfer Protocol):
   - ○ SMTP is an internet standard protocol used for email transmission. It is responsible for sending, receiving, and relaying email messages between email servers. SMTP operates on TCP port 25.

2. Visit a http site and examine how long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet listing in seconds)

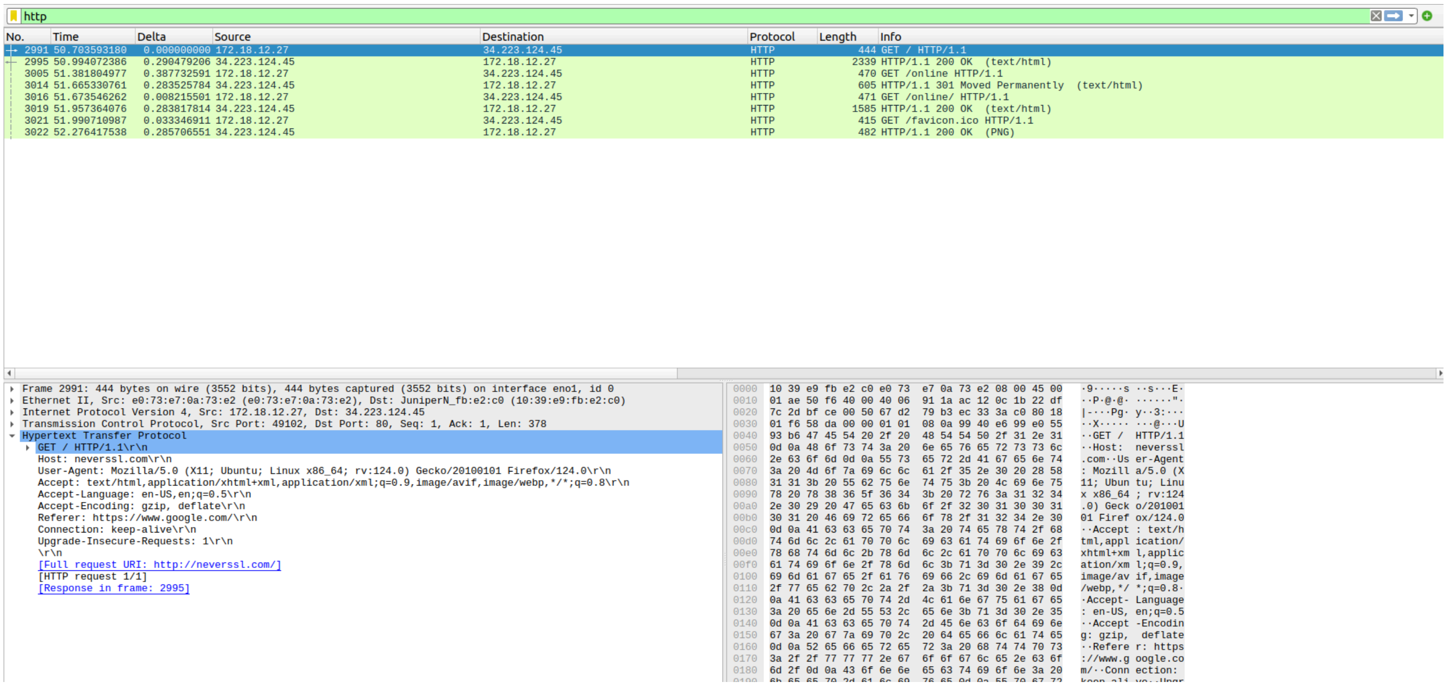
(Hint: Timestamp of http GET packet=10.25478, Timestamp of http OK

packet=10.41478, Required answer = 10.41478-10.25478=0.16 s)



http

No.	Time	Delta	Source	Destination	Protocol	Length	Info
2991	50.703593180	0.000000000	172.18.12.27	34.223.124.45	HTTP	444	GET / HTTP/1.1
2995	50.994072386	0.290479206	34.223.124.45	172.18.12.27	HTTP	2339	HTTP/1.1 200 OK (text/html)
3005	51.381804977	0.387732591	172.18.12.27	34.223.124.45	HTTP	470	GET /online HTTP/1.1
3014	51.665330761	0.283525784	34.223.124.45	172.18.12.27	HTTP	605	HTTP/1.1 301 Moved Permanently (text/html)
3016	51.673546262	0.008215501	172.18.12.27	34.223.124.45	HTTP	471	GET /online/ HTTP/1.1
3019	51.957364076	0.283817814	34.223.124.45	172.18.12.27	HTTP	1585	HTTP/1.1 200 OK (text/html)
3021	51.990710987	0.033346911	172.18.12.27	34.223.124.45	HTTP	415	GET /favicon.ico HTTP/1.1
3022	52.276417538	0.285706551	34.223.124.45	172.18.12.27	HTTP	482	HTTP/1.1 200 OK (PNG)

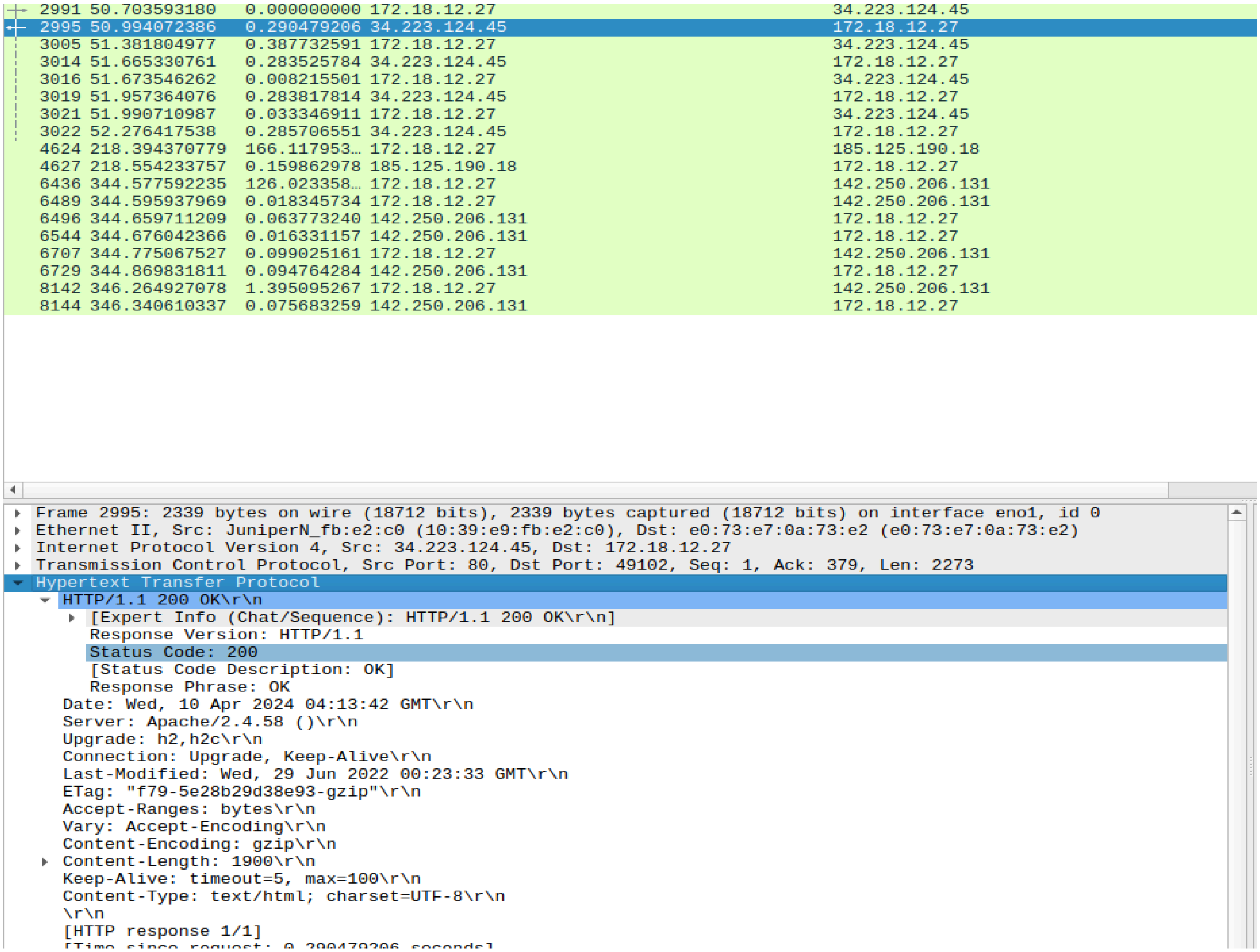


```
http

No.  Time         Delta       Source        Destination   Protocol Length Info
2991 50.703593180 0.000000000 172.18.12.27  34.223.124.45 HTTP      444 GET / HTTP/1.1
2995 50.994072386 0.290479206 34.223.124.45 172.18.12.27  HTTP     2339 HTTP/1.1 200 OK  (text/html)
3005 51.381804977 0.387732591 172.18.12.27  34.223.124.45 HTTP      470 GET /online HTTP/1.1
3014 51.665330761 0.283525784 34.223.124.45 172.18.12.27  HTTP      605 HTTP/1.1 301 Moved Permanently  (text/html)
3016 51.673546262 0.008215501 172.18.12.27  34.223.124.45 HTTP      471 GET /online/ HTTP/1.1
3019 51.957364076 0.283817814 34.223.124.45 172.18.12.27  HTTP     1585 HTTP/1.1 200 OK  (text/html)
3021 51.990710987 0.033346911 172.18.12.27  34.223.124.45 HTTP      415 GET /favicon.ico HTTP/1.1
3022 52.276417538 0.285706551 34.223.124.45 172.18.12.27  HTTP      482 HTTP/1.1 200 OK  (PNG)
```

```
▸ Frame 2991: 444 bytes on wire (3552 bits), 444 bytes captured (3552 bits) on interface eno1, id 0
▸ Ethernet II, Src: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2), Dst: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0)
▸ Internet Protocol Version 4, Src: 172.18.12.27, Dst: 34.223.124.45
▸ Transmission Control Protocol, Src Port: 49102, Dst Port: 80, Seq: 1, Ack: 1, Len: 378
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Host: neverssl.com\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: https://www.google.com/\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://neverssl.com/]
    [HTTP request 1/1]
    [Response in frame: 2995]
```

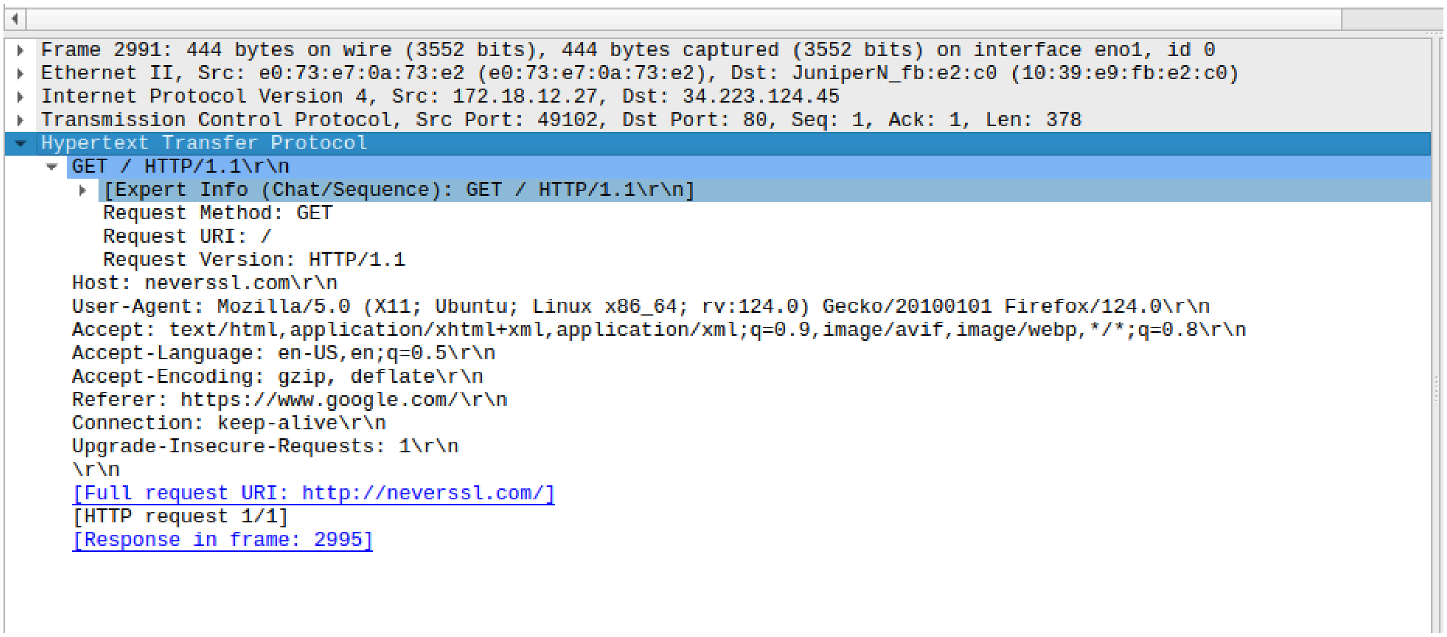
```
0000  10 39 e9 fb e2 c0 e0 73  e7 0a 73 e2 08 00 45 00   ·9·····s ··s···E·
0010  01 ae 50 f6 40 00 40 06  91 1a ac 12 0c 1b 22 df   ··P·@·@· ······"·
0020  7c 2d bf ce 00 50 67 d2  79 b3 ec 33 3a c0 80 18   |-···Pg· y··3:···
0030  01 f6 58 da 00 00 01 01  08 0a 99 40 e6 99 e0 55   ··X····· ···@···U
0040  93 b6 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 31   ··GET /  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  6e 65 76 65 72 73 73 6c   ··Host:  neverssl
0060  2e 63 6f 6d 0d 0a 55 73  65 72 2d 41 67 65 6e 74   .com··Us er-Agent
0070  3a 20 4d 6f 7a 69 6c 6c  61 2f 35 2e 30 20 28 58   : Mozill a/5.0 (X
0080  31 31 3b 20 55 62 75 6e  74 75 3b 20 4c 69 6e 75   11; Ubun tu; Linu
0090  78 20 78 38 36 5f 36 34  3b 20 72 76 3a 31 32 34   x x86_64 ; rv:124
00a0  2e 30 29 20 47 65 63 6b  6f 2f 32 30 31 30 30 31   .0) Geck o/201001
00b0  30 31 20 46 69 72 65 66  6f 78 2f 31 32 34 2e 30   01 Firef ox/124.0
00c0  0d 0a 41 63 63 65 70 74  3a 20 74 65 78 74 2f 68   ··Accept : text/h
00d0  74 6d 6c 2c 61 70 70 6c  69 63 61 74 69 6f 6e 2f   tml,appl ication/
00e0  78 68 74 6d 6c 2b 78 6d  6c 2c 61 70 70 6c 69 63   xhtml+xm l,applic
00f0  61 74 69 6f 6e 2f 78 6d  6c 3b 71 3d 30 2e 39 2c   ation/xm l;q=0.9,
0100  69 6d 61 67 65 2f 61 76  69 66 2c 69 6d 61 67 65   image/av if,image
0110  2f 77 65 62 70 2c 2a 2f  2a 3b 71 3d 30 2e 38 0d   /webp,*/ *;q=0.8·
0120  0a 41 63 63 65 70 74 2d  4c 61 6e 67 75 61 67 65   ·Accept- Language
0130  3a 20 65 6e 2d 55 53 2c  65 6e 3b 71 3d 30 2e 35   : en-US, en;q=0.5
0140  0d 0a 41 63 63 65 70 74  2d 45 6e 63 6f 64 69 6e   ··Accept -Encodin
0150  67 3a 20 67 7a 69 70 2c  20 64 65 66 6c 61 74 65   g: gzip,  deflate
0160  0d 0a 52 65 66 65 72 65  72 3a 20 68 74 74 70 73   ··Refere r: https
0170  3a 2f 2f 77 77 77 2e 67  6f 6f 67 6c 65 2e 63 6f   ://www.g oogle.co
0180  6d 2f 0d 0a 43 6f 6e 6e  65 63 74 69 6f 6e 3a 20   m/··Conn ection: 
0190  6b 65 65 70 2d 61 6c 69  76 65 0d 0a 55 70 67 72   keep-ali ve··Upgr
```



```
2991  50.703593180   0.000000000 172.18.12.27        34.223.124.45
2995  50.994072386   0.290479206 34.223.124.45       172.18.12.27
3005  51.381804977   0.387732591 172.18.12.27        34.223.124.45
3014  51.665330761   0.283525784 34.223.124.45       172.18.12.27
3016  51.673546262   0.008215501 172.18.12.27        34.223.124.45
3019  51.957364076   0.283817814 34.223.124.45       172.18.12.27
3021  51.990710987   0.033346911 172.18.12.27        34.223.124.45
3022  52.276417538   0.285706551 34.223.124.45       172.18.12.27
4624  218.394370779  166.117953… 172.18.12.27        185.125.190.18
4627  218.554233757  0.159862978 185.125.190.18      172.18.12.27
6436  344.577592235  126.023358… 172.18.12.27        142.250.206.131
6489  344.595937969  0.018345734 172.18.12.27        142.250.206.131
6496  344.659711209  0.063773240 142.250.206.131     172.18.12.27
6544  344.676042366  0.016331157 142.250.206.131     172.18.12.27
6707  344.775067527  0.099025161 172.18.12.27        142.250.206.131
6729  344.869831811  0.094764284 142.250.206.131     172.18.12.27
8142  346.264927078  1.395095267 172.18.12.27        142.250.206.131
8144  346.340610337  0.075683259 142.250.206.131     172.18.12.27
```

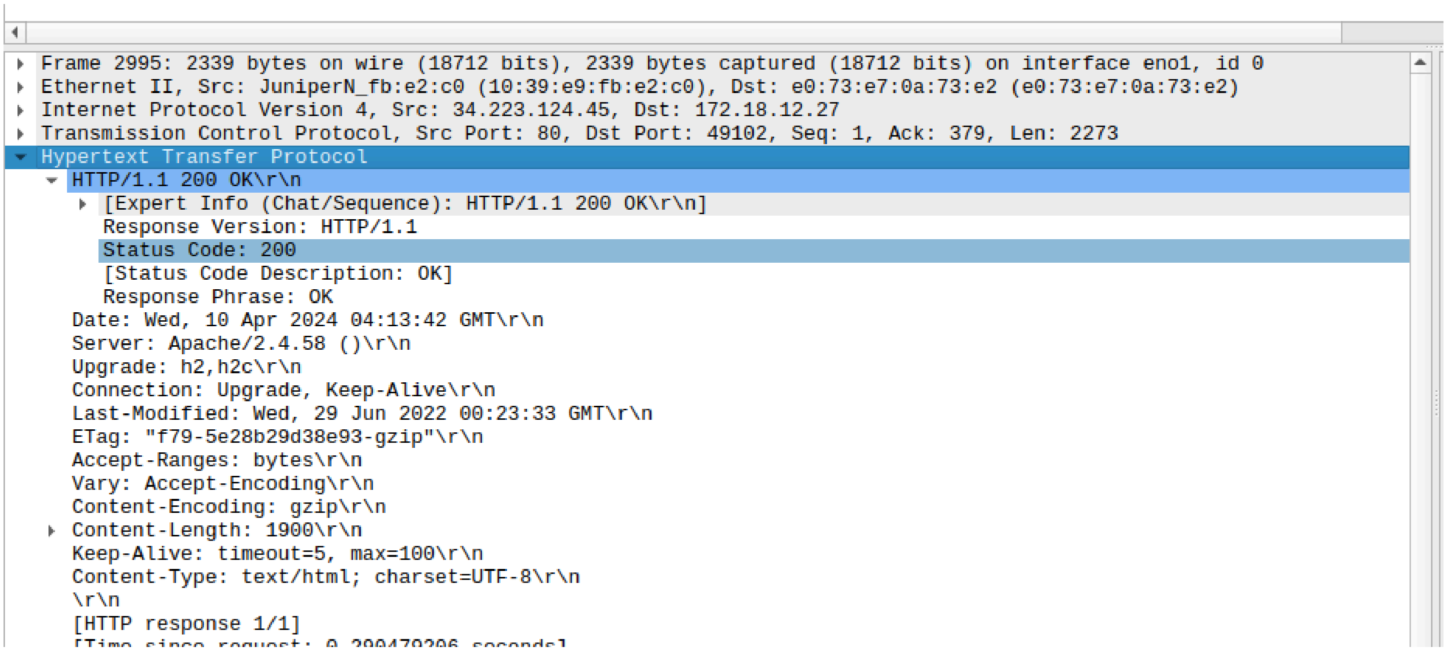
```
▸ Frame 2995: 2339 bytes on wire (18712 bits), 2339 bytes captured (18712 bits) on interface eno1, id 0
▸ Ethernet II, Src: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0), Dst: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2)
▸ Internet Protocol Version 4, Src: 34.223.124.45, Dst: 172.18.12.27
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 49102, Seq: 1, Ack: 379, Len: 2273
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Wed, 10 Apr 2024 04:13:42 GMT\r\n
    Server: Apache/2.4.58 ()\r\n
    Upgrade: h2,h2c\r\n
    Connection: Upgrade, Keep-Alive\r\n
    Last-Modified: Wed, 29 Jun 2022 00:23:33 GMT\r\n
    ETag: "f79-5e28b29d38e93-gzip"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
  ▸ Content-Length: 1900\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.290479206 seconds]
```

3. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running? (Hint: First to know the http version of your browser: Select GET packet, Expand the http protocol, then expand the GET option, HTTP version information is listed in the item 'Request Version'. To know the http version of server, follow same steps with http OK packet)



```
▸ Frame 2991: 444 bytes on wire (3552 bits), 444 bytes captured (3552 bits) on interface eno1, id 0
▸ Ethernet II, Src: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2), Dst: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0)
▸ Internet Protocol Version 4, Src: 172.18.12.27, Dst: 34.223.124.45
▸ Transmission Control Protocol, Src Port: 49102, Dst Port: 80, Seq: 1, Ack: 1, Len: 378
▾ Hypertext Transfer Protocol
  ▾ GET / HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: neverssl.com\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: https://www.google.com/\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://neverssl.com/]
    [HTTP request 1/1]
    [Response in frame: 2995]
```

HTTP version :- 1.1

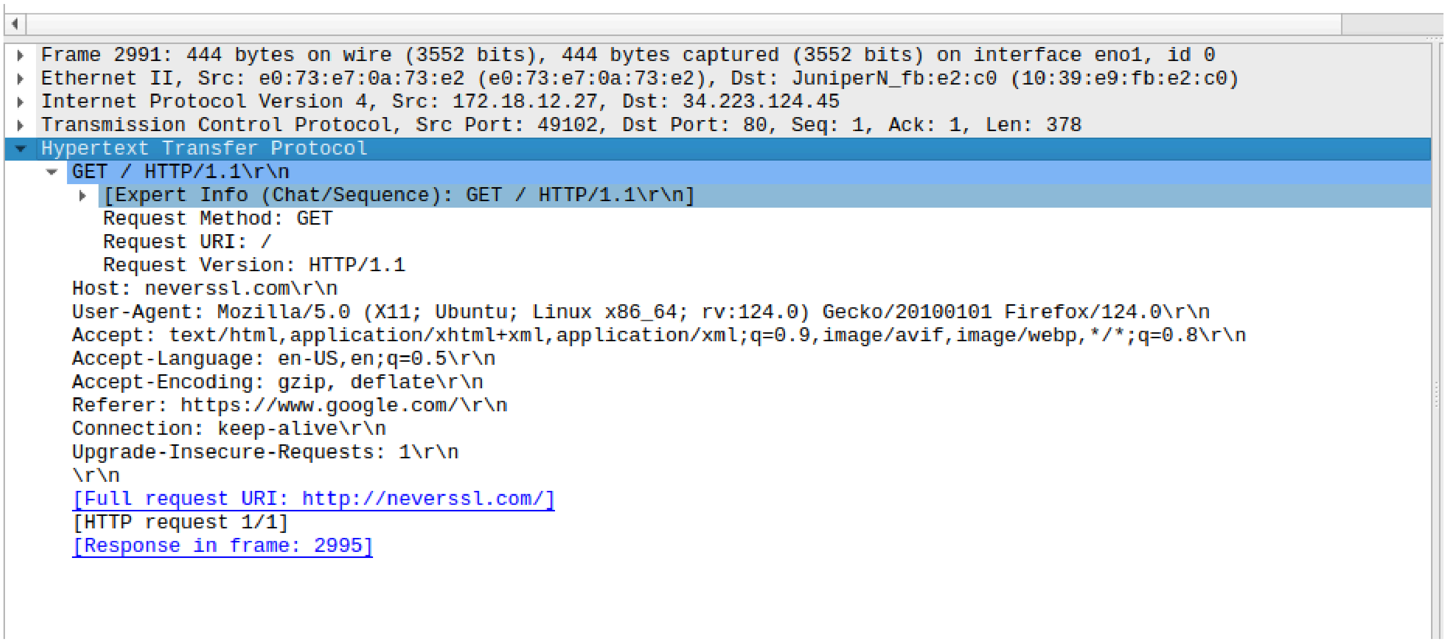


```
▸ Frame 2995: 2339 bytes on wire (18712 bits), 2339 bytes captured (18712 bits) on interface eno1, id 0
▸ Ethernet II, Src: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0), Dst: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2)
▸ Internet Protocol Version 4, Src: 34.223.124.45, Dst: 172.18.12.27
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 49102, Seq: 1, Ack: 379, Len: 2273
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Wed, 10 Apr 2024 04:13:42 GMT\r\n
    Server: Apache/2.4.58 ()\r\n
    Upgrade: h2,h2c\r\n
    Connection: Upgrade, Keep-Alive\r\n
    Last-Modified: Wed, 29 Jun 2022 00:23:33 GMT\r\n
    ETag: "f79-5e28b29d38e93-gzip"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
  ▸ Content-Length: 1900\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.290479206 seconds]
```

HTTP is the server running :- 1.1

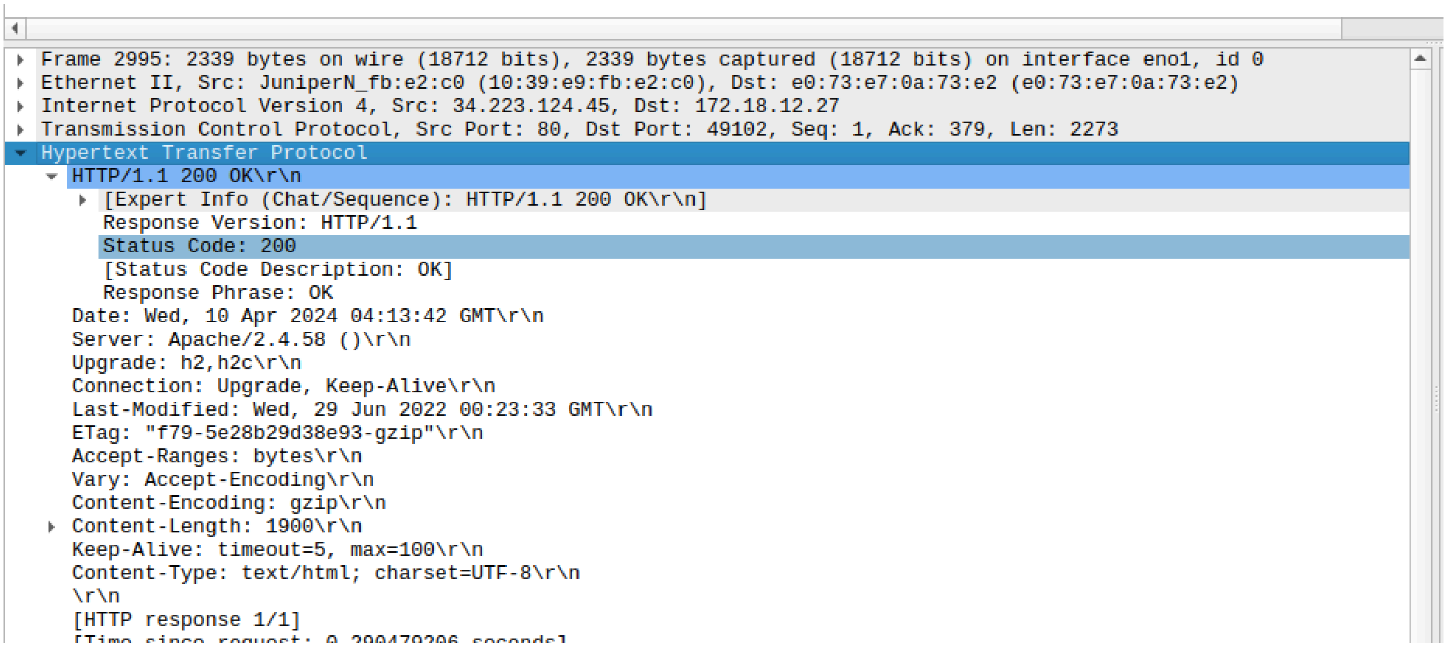
4. What languages does your browser indicate that it can accept to the server? (Hint: languages information is listed in the item ‘Accept-Language’ in the HTTP GET message).

Accept-Language:- en-US, en, q=0.5\r\n



```
Frame 2991: 444 bytes on wire (3552 bits), 444 bytes captured (3552 bits) on interface eno1, id 0
Ethernet II, Src: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2), Dst: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0)
Internet Protocol Version 4, Src: 172.18.12.27, Dst: 34.223.124.45
Transmission Control Protocol, Src Port: 49102, Dst Port: 80, Seq: 1, Ack: 1, Len: 378
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Host: neverssl.com\r\n
  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Referer: https://www.google.com/\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  \r\n
  [Full request URI: http://neverssl.com/]
  [HTTP request 1/1]
  [Response in frame: 2995]
```

5. What is the status code returned from the server to your browser? When was the HTML file that you are retrieving last modified at the server? (Hint: Analyse the http OK packet)

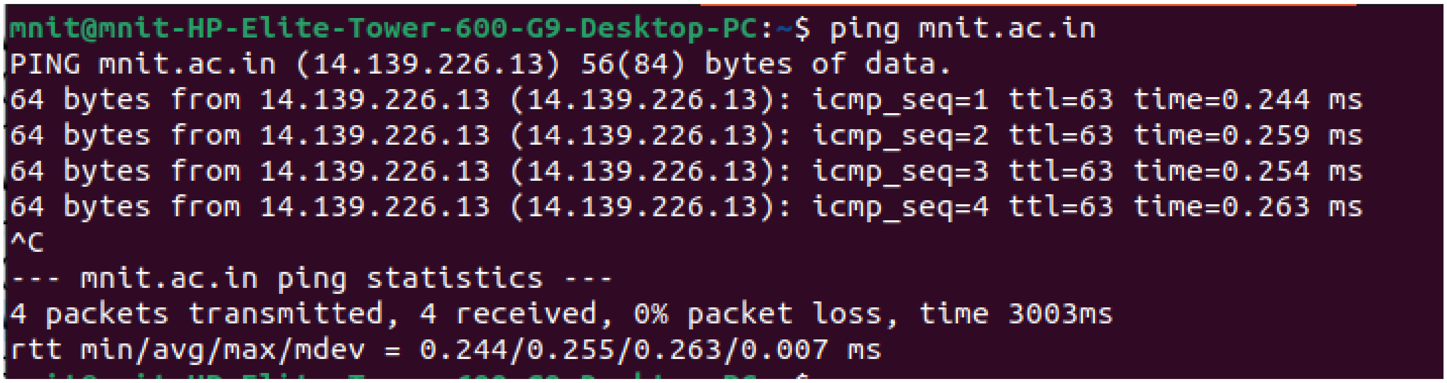


```
Frame 2995: 2339 bytes on wire (18712 bits), 2339 bytes captured (18712 bits) on interface eno1, id 0
Ethernet II, Src: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0), Dst: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2)
Internet Protocol Version 4, Src: 34.223.124.45, Dst: 172.18.12.27
Transmission Control Protocol, Src Port: 80, Dst Port: 49102, Seq: 1, Ack: 379, Len: 2273
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Response Version: HTTP/1.1
    Status Code: 200
    [Status Code Description: OK]
    Response Phrase: OK
  Date: Wed, 10 Apr 2024 04:13:42 GMT\r\n
  Server: Apache/2.4.58 ()\r\n
  Upgrade: h2,h2c\r\n
  Connection: Upgrade, Keep-Alive\r\n
  Last-Modified: Wed, 29 Jun 2022 00:23:33 GMT\r\n
  ETag: "f79-5e28b29d38e93-gzip"\r\n
  Accept-Ranges: bytes\r\n
  Vary: Accept-Encoding\r\n
  Content-Encoding: gzip\r\n
  Content-Length: 1900\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.200479206 seconds]
```

status code :- 200

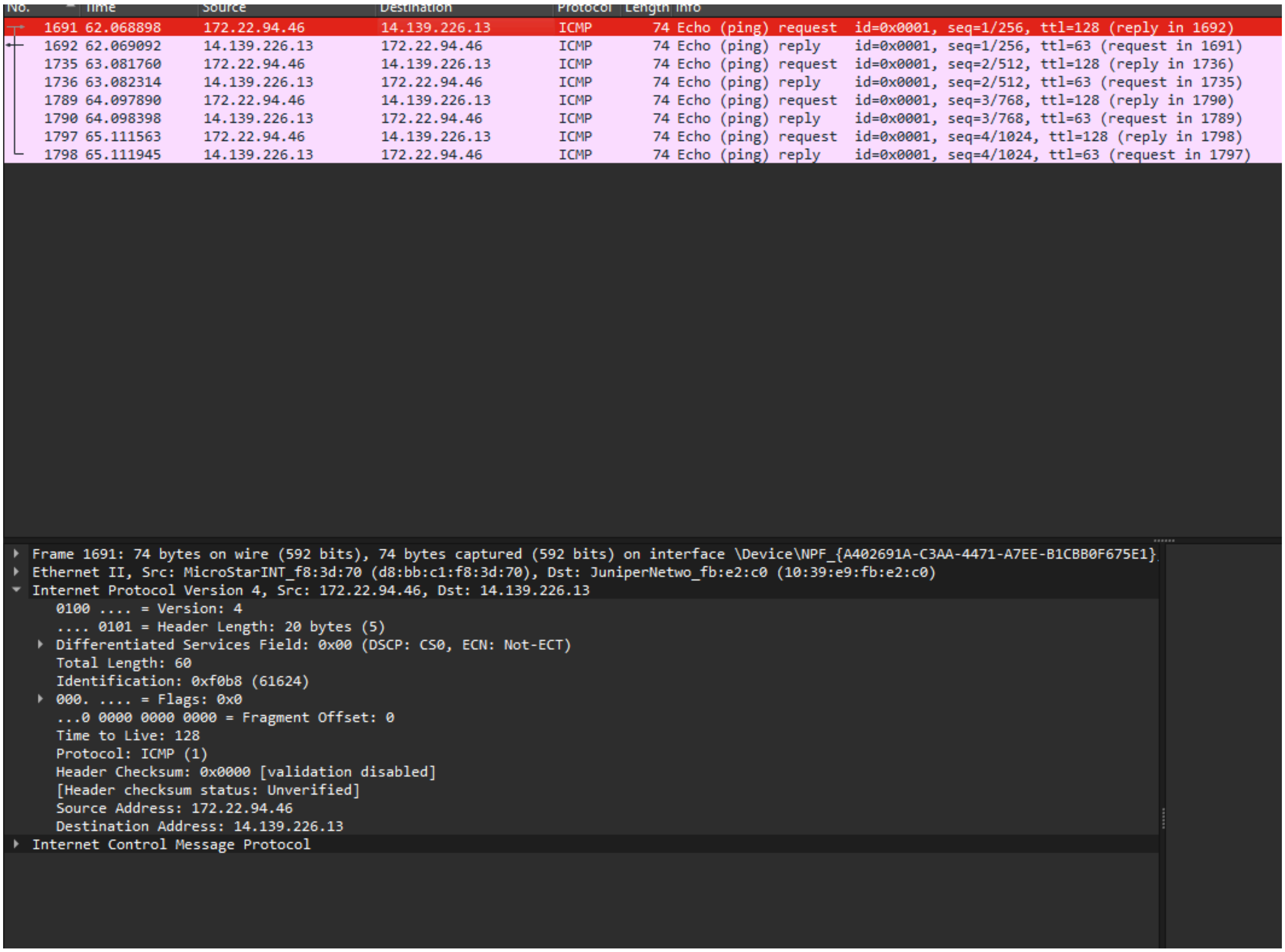
Last-Modified :- Wed, 29 Jun 2022

6. What is the IP address of the mnit.ac.in? What is the IP of your computer? What is the length of these IP addresses in bits? How many bytes are in the IP header? How many bytes are in the payload of the IP datagram? Explain how you determined the number of payload bytes.



```
mnit@mnit-HP-Elite-Tower-600-G9-Desktop-PC:~$ ping mnit.ac.in
PING mnit.ac.in (14.139.226.13) 56(84) bytes of data.
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=1 ttl=63 time=0.244 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=2 ttl=63 time=0.259 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=3 ttl=63 time=0.254 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=4 ttl=63 time=0.263 ms
^C
--- mnit.ac.in ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.244/0.255/0.263/0.007 ms
```

ip of mnit.ac.in :- 14.139.226.13



```
No.     Time       Source          Destination     Protocol Length Info
1691 62.068898   172.22.94.46    14.139.226.13   ICMP        74 Echo (ping) request  id=0x0001, seq=1/256, ttl=128 (reply in 1692)
1692 62.069092   14.139.226.13   172.22.94.46    ICMP        74 Echo (ping) reply    id=0x0001, seq=1/256, ttl=63 (request in 1691)
1735 63.081760   172.22.94.46    14.139.226.13   ICMP        74 Echo (ping) request  id=0x0001, seq=2/512, ttl=128 (reply in 1736)
1736 63.082314   14.139.226.13   172.22.94.46    ICMP        74 Echo (ping) reply    id=0x0001, seq=2/512, ttl=63 (request in 1735)
1789 64.097890   172.22.94.46    14.139.226.13   ICMP        74 Echo (ping) request  id=0x0001, seq=3/768, ttl=128 (reply in 1790)
1790 64.098398   14.139.226.13   172.22.94.46    ICMP        74 Echo (ping) reply    id=0x0001, seq=3/768, ttl=63 (request in 1789)
1797 65.111563   172.22.94.46    14.139.226.13   ICMP        74 Echo (ping) request  id=0x0001, seq=4/1024, ttl=128 (reply in 1798)
1798 65.111945   14.139.226.13   172.22.94.46    ICMP        74 Echo (ping) reply    id=0x0001, seq=4/1024, ttl=63 (request in 1797)
```

```
> Frame 1691: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{A402691A-C3AA-4471-A7EE-B1CBB0F675E1}
> Ethernet II, Src: MicroStarINT_f8:3d:70 (d8:bb:c1:f8:3d:70), Dst: JuniperNetwo_fb:e2:c0 (10:39:e9:fb:e2:c0)
v Internet Protocol Version 4, Src: 172.22.94.46, Dst: 14.139.226.13
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xf0b8 (61624)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.22.94.46
    Destination Address: 14.139.226.13
> Internet Control Message Protocol
```

ip of computer:- 172.18.12.27

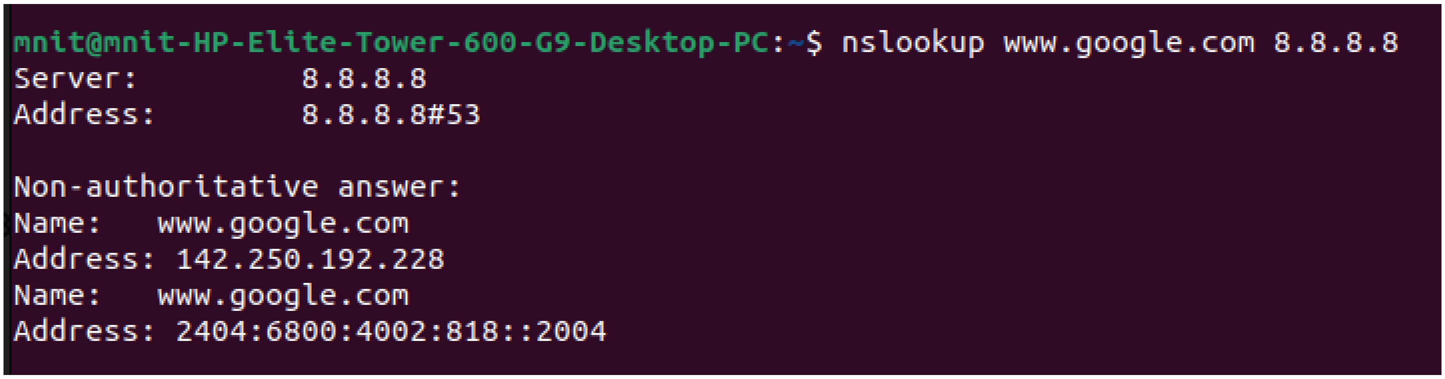
length of this ip address = 32 bits

bytes are in the IP header = 20 bytes

bytes are in the payload of the IP datagram = 60 -20 = 40

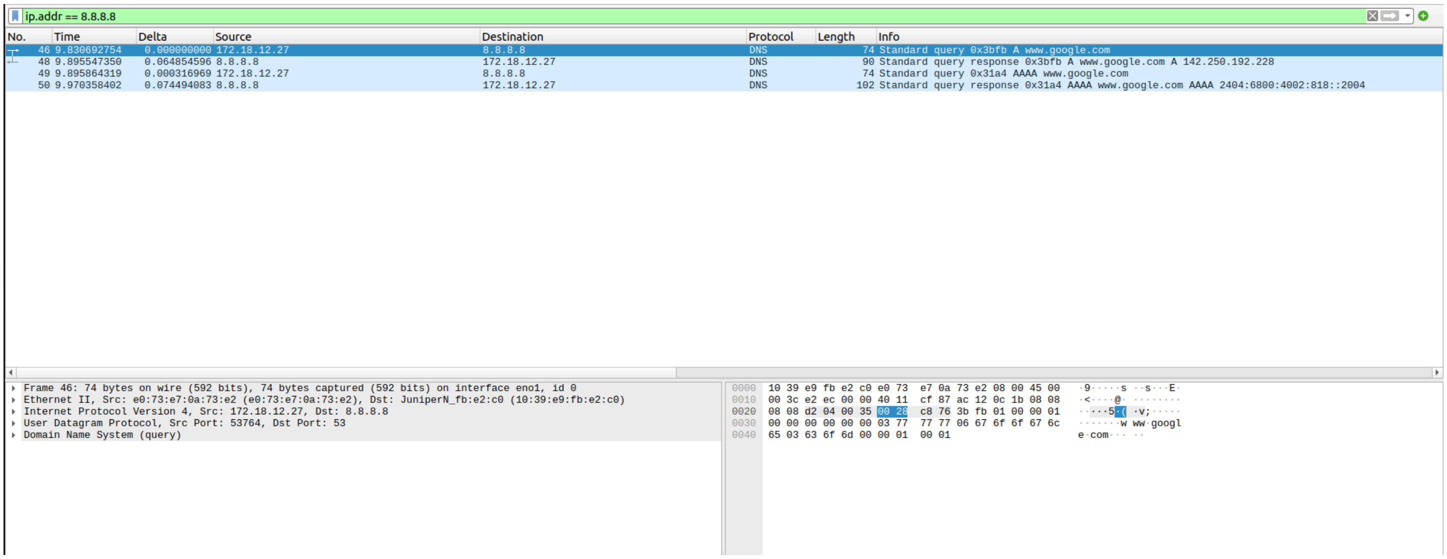
7. Open a terminal on your PC, execute this command:

“nslookup www.google.com 8.8.8.8” While capturing the packet in background, set your filter to “ip.addr == 8.8.8.8”.



```
mnit@mnit-HP-Elite-Tower-600-G9-Desktop-PC:~$ nslookup www.google.com 8.8.8.8
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.192.228
Name:   www.google.com
Address: 2404:6800:4002:818::2004
```

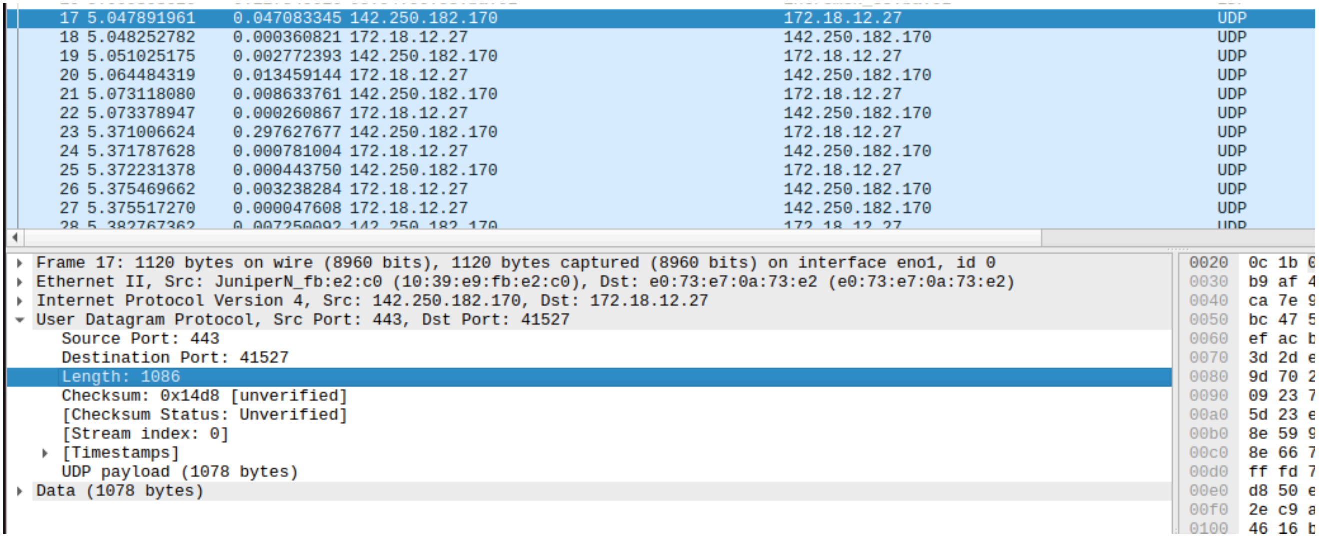


```
ip.addr == 8.8.8.8
No.  Time         Delta       Source        Destination   Protocol  Length Info
  46 9.830692754  0.000000000 172.18.12.27  8.8.8.8       DNS          74 Standard query 0x3bfb A www.google.com
  48 9.895547350  0.064854596 8.8.8.8       172.18.12.27  DNS          90 Standard query response 0x3bfb A www.google.com A 142.250.192.228
  49 9.895864319  0.000316969 172.18.12.27  8.8.8.8       DNS          74 Standard query 0x31a4 AAAA www.google.com
  50 9.970358402  0.074494083 8.8.8.8       172.18.12.27  DNS         102 Standard query response 0x31a4 AAAA www.google.com AAAA 2404:6800:4002:818::2004
```

```
Frame 46: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eno1, id 0
Ethernet II, Src: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2), Dst: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0)
Internet Protocol Version 4, Src: 172.18.12.27, Dst: 8.8.8.8
User Datagram Protocol, Src Port: 53764, Dst Port: 53
Domain Name System (query)
```

```
0000  10 39 e9 fb e2 c0 e0 73  e7 0a 73 e2 08 00 45 00   ·9·····s ··s···E·
0010  00 3c e2 ec 00 00 40 11  cf 87 ac 12 0c 1b 08 08   ·<····@· ········
0020  08 08 d2 04 00 35 00 28  c8 76 3b fb 01 00 00 01   ·····5·( ·v;·····
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   ·······w ww·googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e·com··· ··
```

8. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header.



```
 17 5.047891961    0.047083345 142.250.182.170                              172.18.12.27                           UDP
 18 5.048252782    0.000360821 172.18.12.27                                 142.250.182.170                        UDP
 19 5.051025175    0.002772393 142.250.182.170                              172.18.12.27                           UDP
 20 5.064484319    0.013459144 172.18.12.27                                 142.250.182.170                        UDP
 21 5.073118080    0.008633761 142.250.182.170                              172.18.12.27                           UDP
 22 5.073378947    0.000260867 172.18.12.27                                 142.250.182.170                        UDP
 23 5.371006624    0.297627677 142.250.182.170                              172.18.12.27                           UDP
 24 5.371787628    0.000781004 172.18.12.27                                 142.250.182.170                        UDP
 25 5.372231378    0.000443750 142.250.182.170                              172.18.12.27                           UDP
 26 5.375469662    0.003238284 172.18.12.27                                 142.250.182.170                        UDP
 27 5.375517270    0.000047608 172.18.12.27                                 142.250.182.170                        UDP
 28 5.382767362    0.007250092 142.250.182.170                              172.18.12.27                           UDP
```

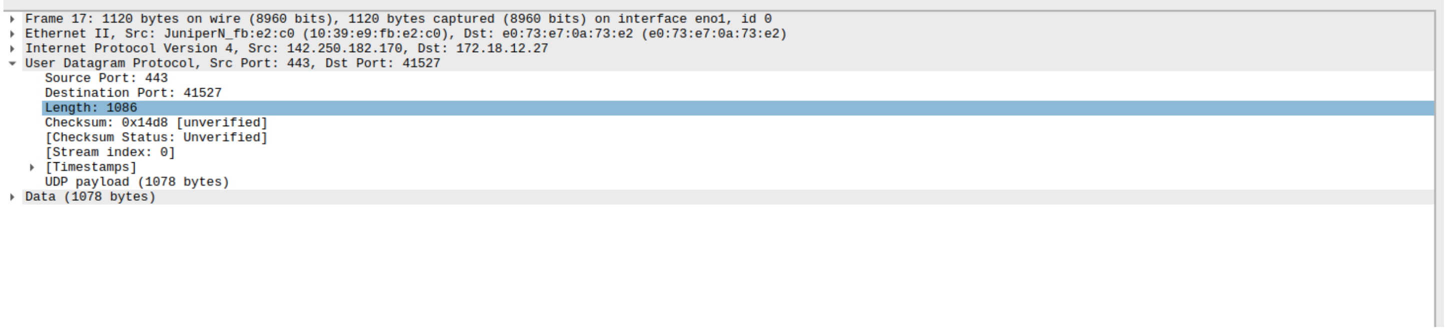
```
▸ Frame 17: 1120 bytes on wire (8960 bits), 1120 bytes captured (8960 bits) on interface eno1, id 0
▸ Ethernet II, Src: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0), Dst: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2)
▸ Internet Protocol Version 4, Src: 142.250.182.170, Dst: 172.18.12.27
▾ User Datagram Protocol, Src Port: 443, Dst Port: 41527
    Source Port: 443
    Destination Port: 41527
    Length: 1086
    Checksum: 0x14d8 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  ▸ [Timestamps]
    UDP payload (1078 bytes)
▸ Data (1078 bytes)
```

```
0020  0c 1b 0
0030  b9 af 4
0040  ca 7e 9
0050  bc 47 5
0060  ef ac b
0070  3d 2d e
0080  9d 70 2
0090  09 23 7
00a0  5d 23 e
00b0  8e 59 9
00c0  8e 66 7
00d0  ff fd 7
00e0  d8 50 e
00f0  2e c9 a
0100  46 16 b
```

There are 4 fields in UDP header :- Source Port, Destination Port, Length, Checksum

9. Determine the length (in bytes) of each of the UDP header fields.

(Hint: See the packet diagram)



```
▸ Frame 17: 1120 bytes on wire (8960 bits), 1120 bytes captured (8960 bits) on interface eno1, id 0
▸ Ethernet II, Src: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0), Dst: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2)
▸ Internet Protocol Version 4, Src: 142.250.182.170, Dst: 172.18.12.27
▾ User Datagram Protocol, Src Port: 443, Dst Port: 41527
    Source Port: 443
    Destination Port: 41527
    Length: 1086
    Checksum: 0x14d8 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  ▸ [Timestamps]
    UDP payload (1078 bytes)
▸ Data (1078 bytes)
```

Source Port:- 2 length

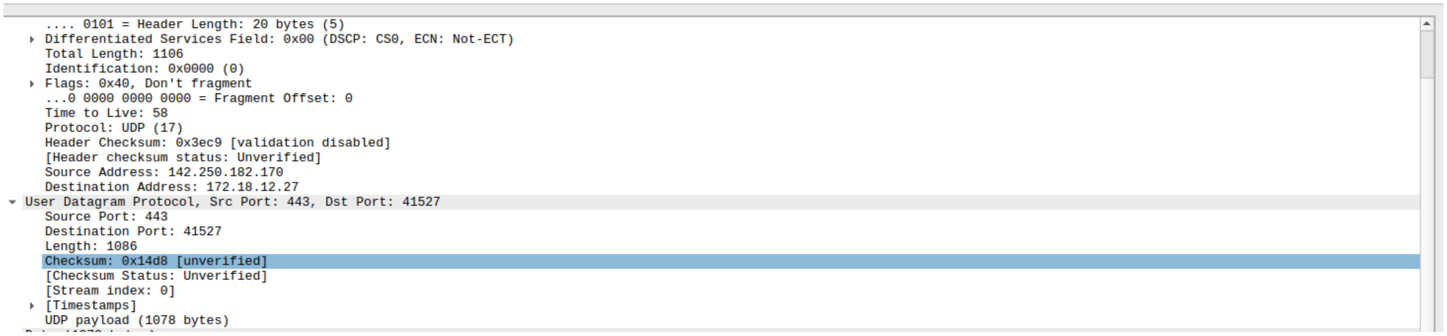
Destination Port:- 2length

Checksum :- 2 length

rem-length of header:- 1078

total length of udp :- 1086

10. The value in the Length field is the length of what? What is the length of UDP payload for your selected packet?



```
      .... 0101 = Header Length: 20 bytes (5)
    ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 1106
      Identification: 0x0000 (0)
    ▸ Flags: 0x40, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 58
      Protocol: UDP (17)
      Header Checksum: 0x3ec9 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 142.250.182.170
      Destination Address: 172.18.12.27
  ▾ User Datagram Protocol, Src Port: 443, Dst Port: 41527
      Source Port: 443
      Destination Port: 41527
      Length: 1086
      Checksum: 0x14d8 [unverified]
      [Checksum Status: Unverified]
      [Stream index: 0]
    ▸ [Timestamps]
      UDP payload (1078 bytes)
```

the value of Length field is the length of of DataGram

Payload length :- 1078

11. What is the largest possible source port number?

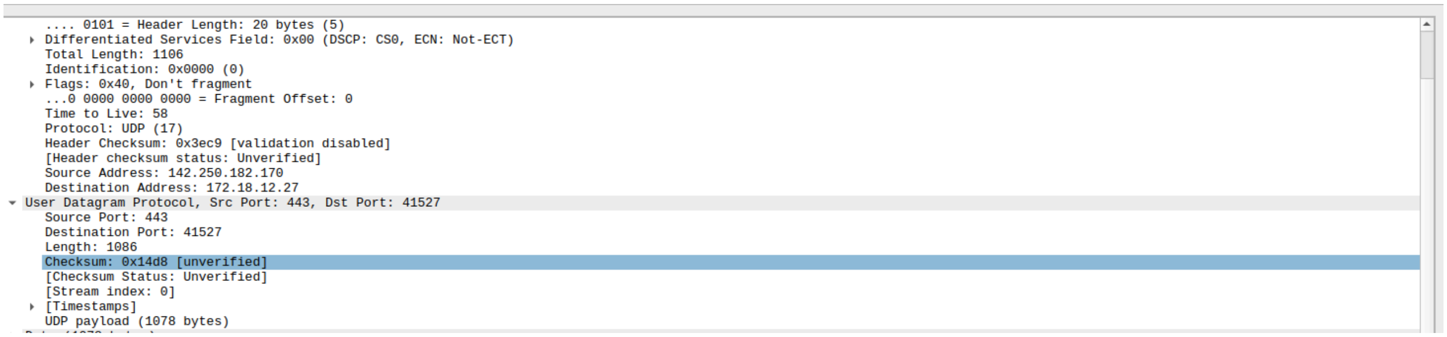
The largest Possible source port number

the Source port length is 2 byte = 16 bit

then the number is 2^16

12. What is the protocol number for UDP?

(Hint: To answer this question, you'll need to look into the IP header.)



```
      .... 0101 = Header Length: 20 bytes (5)
    ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 1106
      Identification: 0x0000 (0)
    ▸ Flags: 0x40, Don't fragment
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 58
      Protocol: UDP (17)
      Header Checksum: 0x3ec9 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 142.250.182.170
      Destination Address: 172.18.12.27
  ▾ User Datagram Protocol, Src Port: 443, Dst Port: 41527
      Source Port: 443
      Destination Port: 41527
      Length: 1086
      Checksum: 0x14d8 [unverified]
      [Checksum Status: Unverified]
      [Stream index: 0]
    ▸ [Timestamps]
      UDP payload (1078 bytes)
```

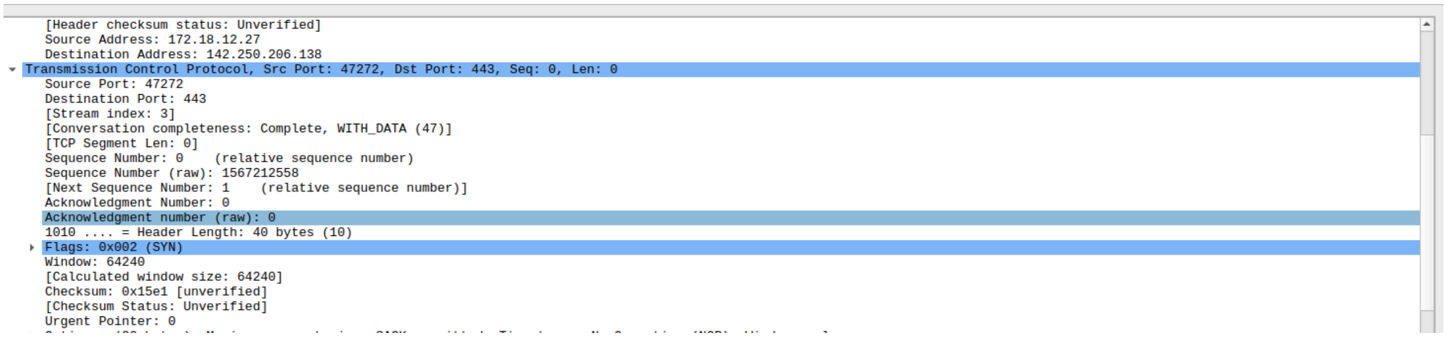
protocol number for UDP :- 17

13. Establish TCP connection and name the 3 packets involved in the connection (TCP handshake). Determine what is the IP address of the client (the initiator of this TCP connection), and what is the server's IP address? From which port the client initiates the connection, and what is the port number used for this connection on the server side?

In a TCP handshake, three packets are typically involved: SYN, SYN-ACK, and ACK. Here's how it works:

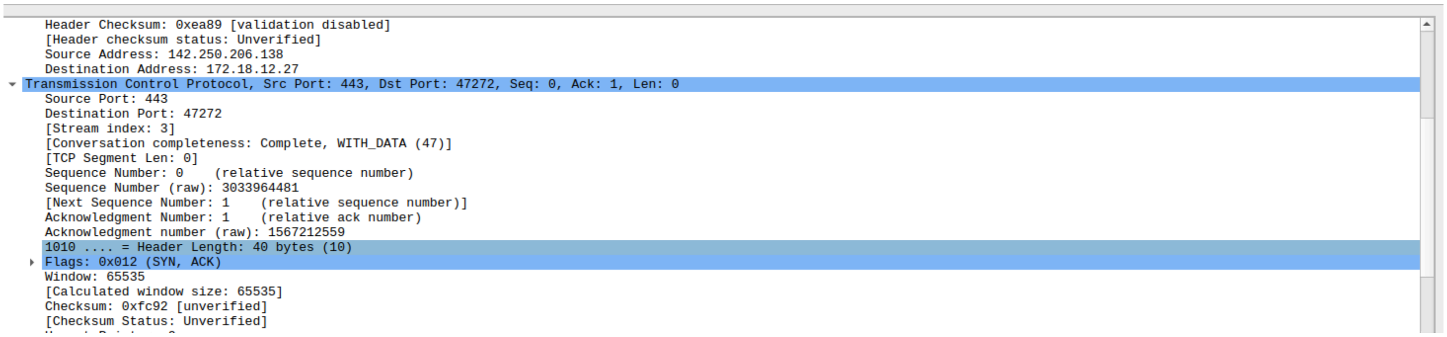
1. SYN (Synchronize): The client sends a packet with the SYN flag set to the server, indicating that it wants to initiate a connection.
2. SYN-ACK (Synchronize-Acknowledge): The server responds with a packet that has both the SYN and ACK flags set, acknowledging the client's request to connect and indicating its own readiness to establish the connection.
3. ACK (Acknowledge): Finally, the client sends a packet back to the server with the ACK flag set, confirming the server's acknowledgment, and completing the three-way handshake.

SYK



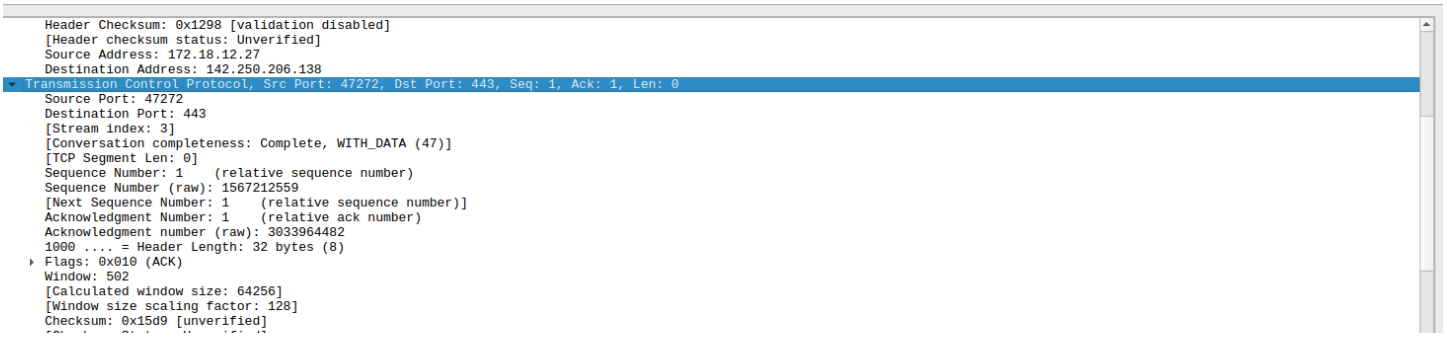
```
        [Header checksum status: Unverified]
        Source Address: 172.18.12.27
        Destination Address: 142.250.206.138
  ▾ Transmission Control Protocol, Src Port: 47272, Dst Port: 443, Seq: 0, Len: 0
        Source Port: 47272
        Destination Port: 443
        [Stream index: 3]
        [Conversation completeness: Complete, WITH_DATA (47)]
        [TCP Segment Len: 0]
        Sequence Number: 0    (relative sequence number)
        Sequence Number (raw): 1567212558
        [Next Sequence Number: 1    (relative sequence number)]
        Acknowledgment Number: 0
        Acknowledgment number (raw): 0
        1010 .... = Header Length: 40 bytes (10)
     ▸ Flags: 0x002 (SYN)
        Window: 64240
        [Calculated window size: 64240]
        Checksum: 0x15e1 [unverified]
        [Checksum Status: Unverified]
        Urgent Pointer: 0
```

SYK_ACK



```
        Header Checksum: 0xea89 [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 142.250.206.138
        Destination Address: 172.18.12.27
  ▾ Transmission Control Protocol, Src Port: 443, Dst Port: 47272, Seq: 0, Ack: 1, Len: 0
        Source Port: 443
        Destination Port: 47272
        [Stream index: 3]
        [Conversation completeness: Complete, WITH_DATA (47)]
        [TCP Segment Len: 0]
        Sequence Number: 0    (relative sequence number)
        Sequence Number (raw): 3033964481
        [Next Sequence Number: 1    (relative sequence number)]
        Acknowledgment Number: 1    (relative ack number)
        Acknowledgment number (raw): 1567212559
        1010 .... = Header Length: 40 bytes (10)
     ▸ Flags: 0x012 (SYN, ACK)
        Window: 65535
        [Calculated window size: 65535]
        Checksum: 0xfc92 [unverified]
        [Checksum Status: Unverified]
```

ACK



```
        Header Checksum: 0x1298 [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 172.18.12.27
        Destination Address: 142.250.206.138
  ▾ Transmission Control Protocol, Src Port: 47272, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
        Source Port: 47272
        Destination Port: 443
        [Stream index: 3]
        [Conversation completeness: Complete, WITH_DATA (47)]
        [TCP Segment Len: 0]
        Sequence Number: 1    (relative sequence number)
        Sequence Number (raw): 1567212559
        [Next Sequence Number: 1    (relative sequence number)]
        Acknowledgment Number: 1    (relative ack number)
        Acknowledgment number (raw): 3033964482
        1000 .... = Header Length: 32 bytes (8)
     ▸ Flags: 0x010 (ACK)
        Window: 502
        [Calculated window size: 64256]
        [Window size scaling factor: 128]
        Checksum: 0x15d9 [unverified]
```

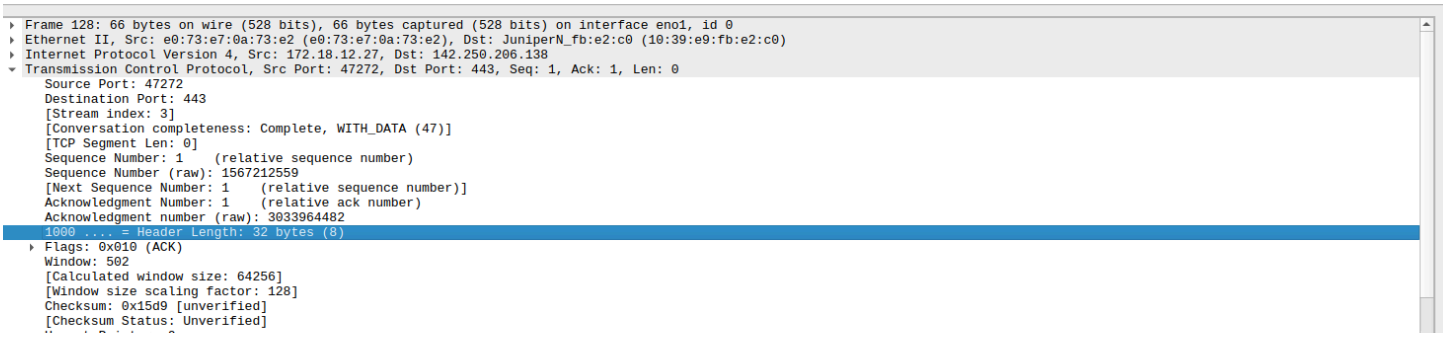
ip of client:- 172.18.12.27

ip of server:- 142.250.206.138

client port:- 47272

server port:- 443

14. During the handshaking of this connection, what is the length of the TCP header? What is the optional field(s) in the TCP header.

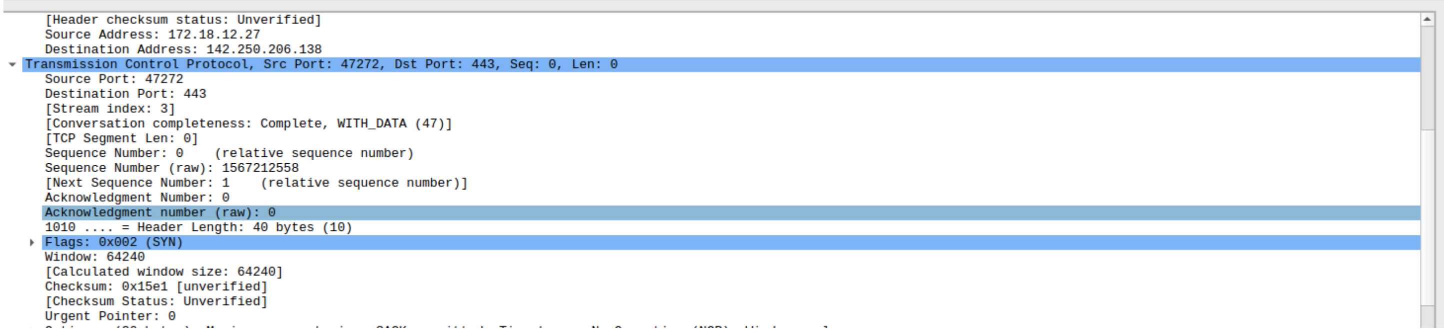


```
▸ Frame 128: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eno1, id 0
▸ Ethernet II, Src: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2), Dst: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0)
▸ Internet Protocol Version 4, Src: 172.18.12.27, Dst: 142.250.206.138
▾ Transmission Control Protocol, Src Port: 47272, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
    Source Port: 47272
    Destination Port: 443
    [Stream index: 3]
    [Conversation completeness: Complete, WITH_DATA (47)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1567212559
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3033964482
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x010 (ACK)
    Window: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0x15d9 [unverified]
    [Checksum Status: Unverified]
```

length of TCP header:- 32 bytes

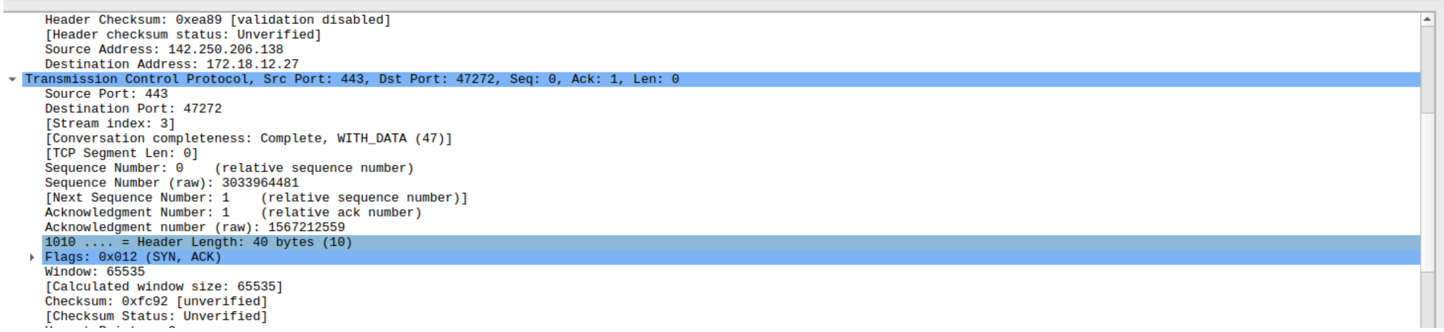
Source Port,Destination Port, Sequence Number, Acknowledgement Number,Flag,Window,Checksum

15. What is the sequence number of the TCP SYN that is used to initiate the TCP connection. What is the sequence number of the SYN-ACK segment? What is the initial buffer size (window size) advertised by the client?



```
    [Header checksum status: Unverified]
    Source Address: 172.18.12.27
    Destination Address: 142.250.206.138
▾ Transmission Control Protocol, Src Port: 47272, Dst Port: 443, Seq: 0, Len: 0
    Source Port: 47272
    Destination Port: 443
    [Stream index: 3]
    [Conversation completeness: Complete, WITH_DATA (47)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 1567212558
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1010 .... = Header Length: 40 bytes (10)
  ▸ Flags: 0x002 (SYN)
    Window: 64240
    [Calculated window size: 64240]
    Checksum: 0x15e1 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
```

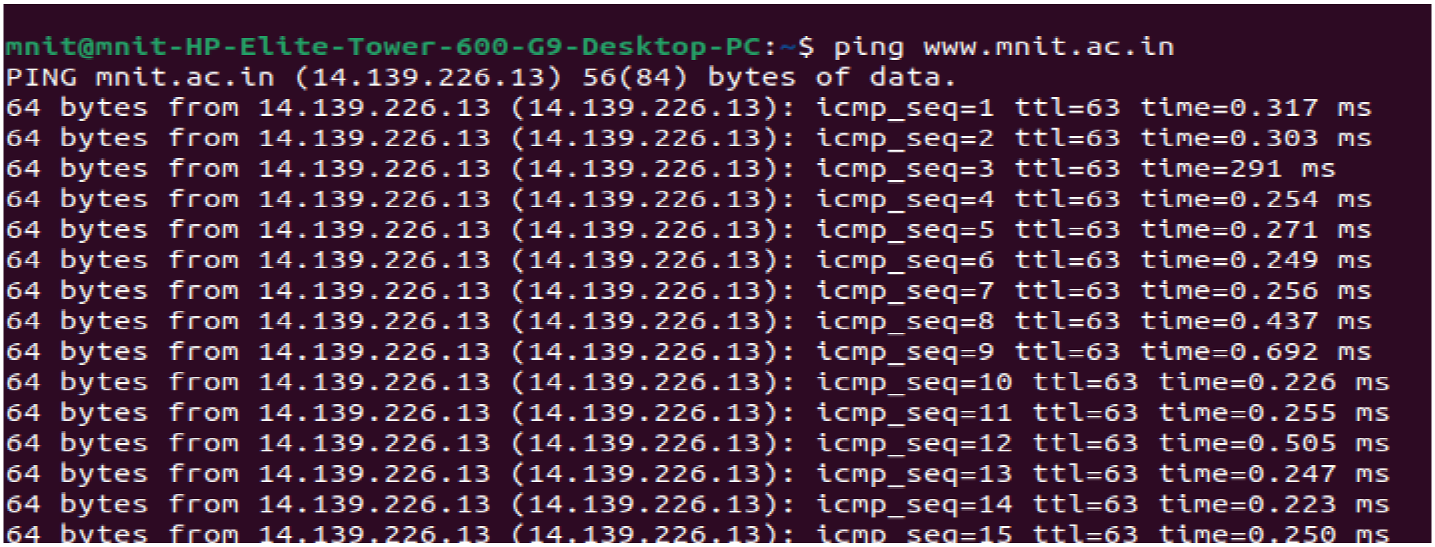
sequence Number of SYN:- 1567212558



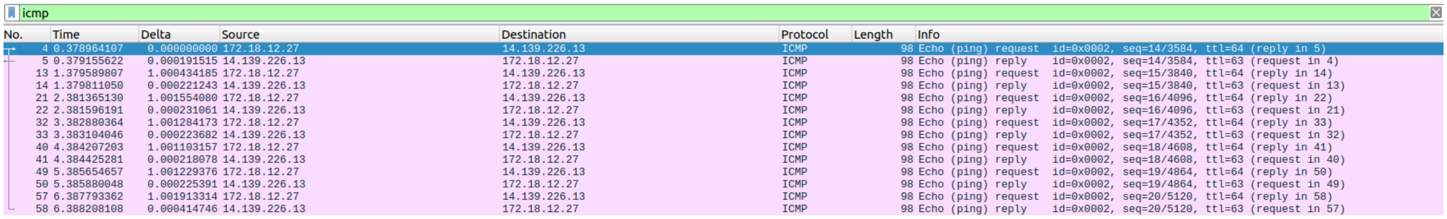
```
    Header Checksum: 0xea89 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 142.250.206.138
    Destination Address: 172.18.12.27
▾ Transmission Control Protocol, Src Port: 443, Dst Port: 47272, Seq: 0, Ack: 1, Len: 0
    Source Port: 443
    Destination Port: 47272
    [Stream index: 3]
    [Conversation completeness: Complete, WITH_DATA (47)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3033964481
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1567212559
    1010 .... = Header Length: 40 bytes (10)
  ▸ Flags: 0x012 (SYN, ACK)
    Window: 65535
    [Calculated window size: 65535]
    Checksum: 0xfc92 [unverified]
    [Checksum Status: Unverified]
```

initial buffer size:- 64240

16. Execute the command “ping www.mnit.ac.in” in terminal, Use WireShark to capture the generated ICMP packet (you can use filter “icmp”) and answer why is it that an ICMP packet does not have source and destination port numbers?

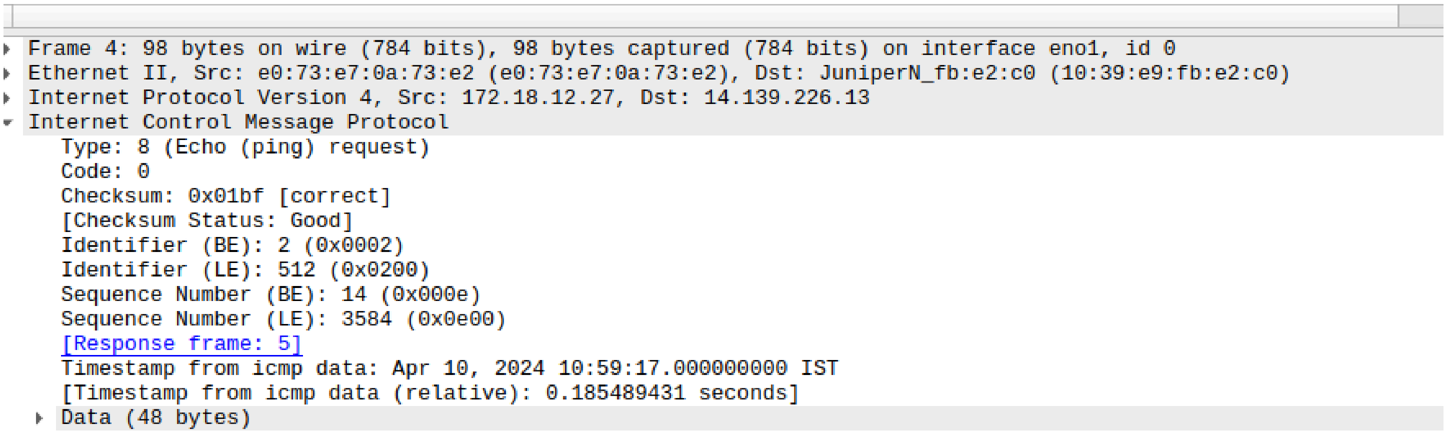


```
mnit@mnit-HP-Elite-Tower-600-G9-Desktop-PC:~$ ping www.mnit.ac.in
PING mnit.ac.in (14.139.226.13) 56(84) bytes of data.
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=1 ttl=63 time=0.317 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=2 ttl=63 time=0.303 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=3 ttl=63 time=291 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=4 ttl=63 time=0.254 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=5 ttl=63 time=0.271 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=6 ttl=63 time=0.249 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=7 ttl=63 time=0.256 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=8 ttl=63 time=0.437 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=9 ttl=63 time=0.692 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=10 ttl=63 time=0.226 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=11 ttl=63 time=0.255 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=12 ttl=63 time=0.505 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=13 ttl=63 time=0.247 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=14 ttl=63 time=0.223 ms
64 bytes from 14.139.226.13 (14.139.226.13): icmp_seq=15 ttl=63 time=0.250 ms
```



icmp

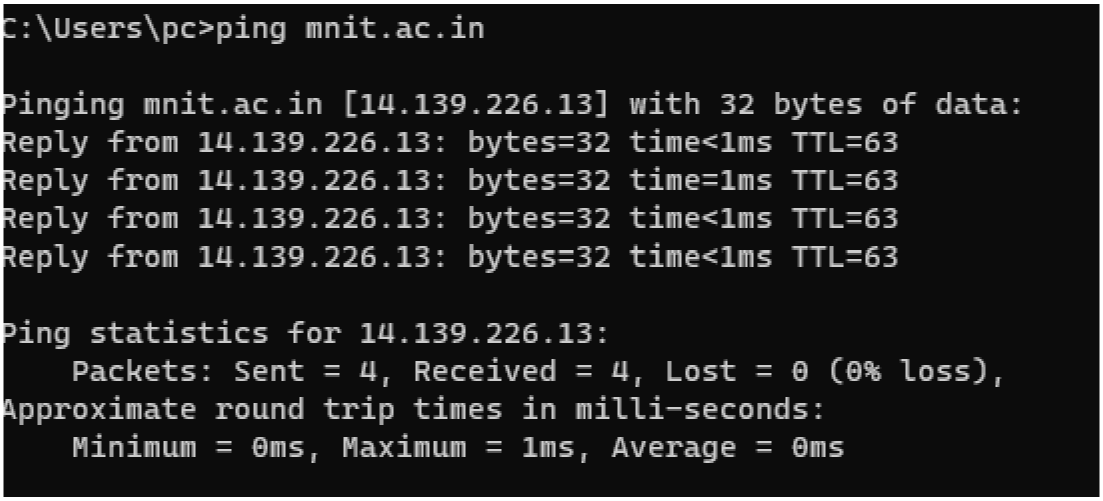
No.	Time	Delta	Source	Destination	Protocol	Length	Info
4	0.378964107	0.000000000	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=14/3584, ttl=64 (reply in 5)
5	0.379155622	0.000191515	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=14/3584, ttl=63 (request in 4)
13	1.379589807	1.000434185	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=15/3840, ttl=64 (reply in 14)
14	1.379811050	0.000221243	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=15/3840, ttl=63 (request in 13)
21	2.381365130	1.001554080	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=16/4096, ttl=64 (reply in 22)
22	2.381596191	0.000231061	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=16/4096, ttl=63 (request in 21)
32	3.382880364	1.001284173	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=17/4352, ttl=64 (reply in 33)
33	3.383104046	0.000223682	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=17/4352, ttl=63 (request in 32)
40	4.384207203	1.001103157	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=18/4608, ttl=64 (reply in 41)
41	4.384425281	0.000218078	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=18/4608, ttl=63 (request in 40)
49	5.385654657	1.001229376	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=19/4864, ttl=64 (reply in 50)
50	5.385880048	0.000225391	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=19/4864, ttl=63 (request in 49)
57	6.387793362	1.001913314	172.18.12.27	14.139.226.13	ICMP	98	Echo (ping) request id=0x0002, seq=20/5120, ttl=64 (reply in 58)
58	6.388208108	0.000414746	14.139.226.13	172.18.12.27	ICMP	98	Echo (ping) reply id=0x0002, seq=20/5120, ttl=63 (request in 57)



```
▸ Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eno1, id 0
▸ Ethernet II, Src: e0:73:e7:0a:73:e2 (e0:73:e7:0a:73:e2), Dst: JuniperN_fb:e2:c0 (10:39:e9:fb:e2:c0)
▸ Internet Protocol Version 4, Src: 172.18.12.27, Dst: 14.139.226.13
▾ Internet Control Message Protocol
     Type: 8 (Echo (ping) request)
     Code: 0
     Checksum: 0x01bf [correct]
     [Checksum Status: Good]
     Identifier (BE): 2 (0x0002)
     Identifier (LE): 512 (0x0200)
     Sequence Number (BE): 14 (0x000e)
     Sequence Number (LE): 3584 (0x0e00)
     [Response frame: 5]
     Timestamp from icmp data: Apr 10, 2024 10:59:17.000000000 IST
     [Timestamp from icmp data (relative): 0.185489431 seconds]
   ▸ Data (48 bytes)
```

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. Each ICMP packet has a "Type" and a "Code". The Type/Code combination identifies the specific message being received.

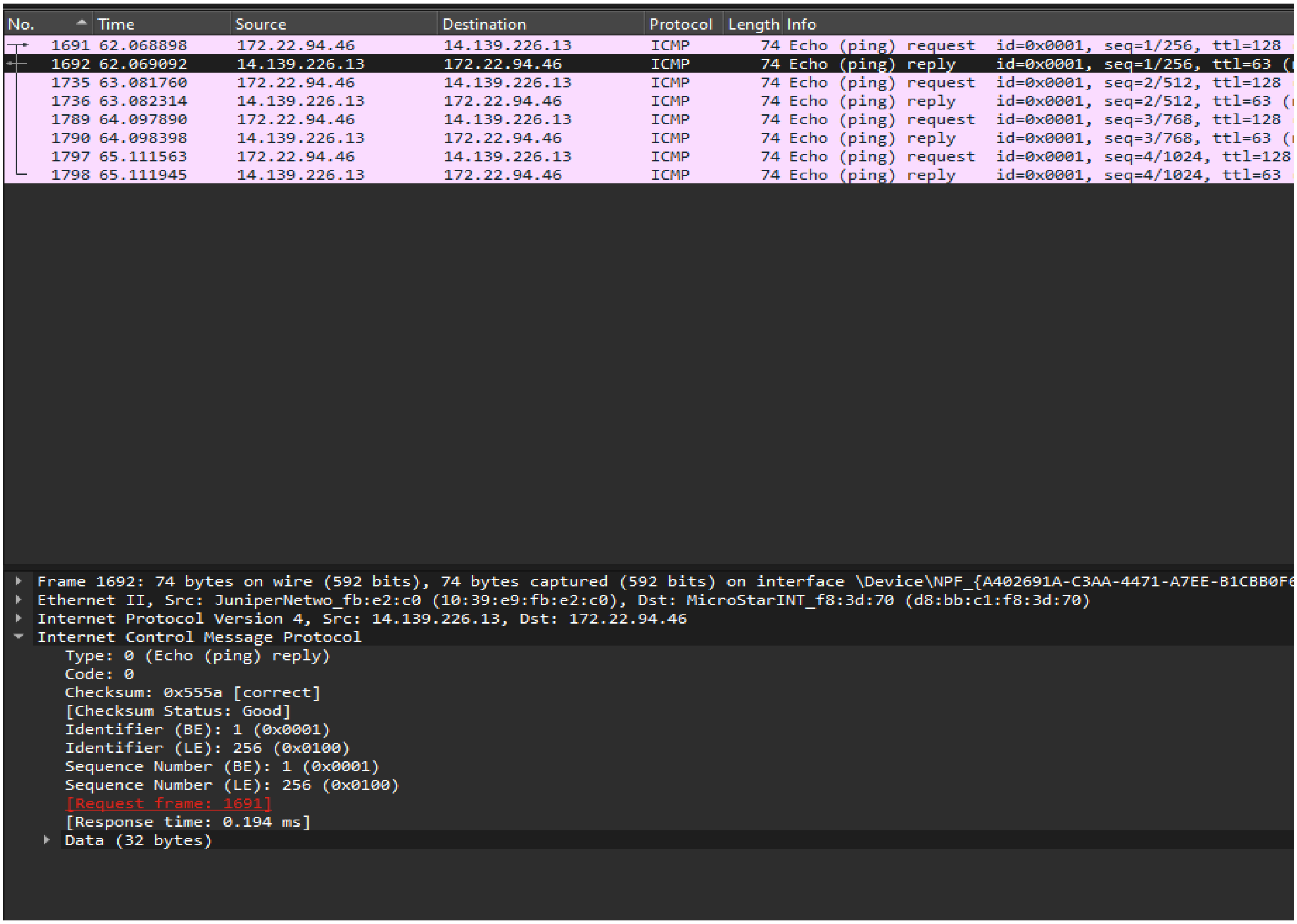
17. Choose one of the ping request packets sent by your host, what are the ICMP type and code numbers? Find the corresponding ping reply, what are the type and code numbers?



```
C:\Users\pc>ping mnit.ac.in

Pinging mnit.ac.in [14.139.226.13] with 32 bytes of data:
Reply from 14.139.226.13: bytes=32 time<1ms TTL=63
Reply from 14.139.226.13: bytes=32 time=1ms TTL=63
Reply from 14.139.226.13: bytes=32 time<1ms TTL=63
Reply from 14.139.226.13: bytes=32 time<1ms TTL=63

Ping statistics for 14.139.226.13:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```



```
No.     Time       Source          Destination     Protocol Length Info
1691 62.068898  172.22.94.46    14.139.226.13   ICMP      74 Echo (ping) request  id=0x0001, seq=1/256, ttl=128
1692 62.069092  14.139.226.13   172.22.94.46    ICMP      74 Echo (ping) reply    id=0x0001, seq=1/256, ttl=63 (
1735 63.081760  172.22.94.46    14.139.226.13   ICMP      74 Echo (ping) request  id=0x0001, seq=2/512, ttl=128
1736 63.082314  14.139.226.13   172.22.94.46    ICMP      74 Echo (ping) reply    id=0x0001, seq=2/512, ttl=63 (
1789 64.097890  172.22.94.46    14.139.226.13   ICMP      74 Echo (ping) request  id=0x0001, seq=3/768, ttl=128
1790 64.098398  14.139.226.13   172.22.94.46    ICMP      74 Echo (ping) reply    id=0x0001, seq=3/768, ttl=63 (
1797 65.111563  172.22.94.46    14.139.226.13   ICMP      74 Echo (ping) request  id=0x0001, seq=4/1024, ttl=128
1798 65.111945  14.139.226.13   172.22.94.46    ICMP      74 Echo (ping) reply    id=0x0001, seq=4/1024, ttl=63
```

```
Frame 1692: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{A402691A-C3AA-4471-A7EE-B1CBB0F6
Ethernet II, Src: JuniperNetwo_fb:e2:c0 (10:39:e9:fb:e2:c0), Dst: MicroStarINT_f8:3d:70 (d8:bb:c1:f8:3d:70)
Internet Protocol Version 4, Src: 14.139.226.13, Dst: 172.22.94.46
Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x555a [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
    [Request frame: 1691]
    [Response time: 0.194 ms]
  Data (32 bytes)
```

Type :- 0

Code: - 0

18. Apart from the ICMP headers, what is in the data field of these ICMP packets? Which fields in the IP datagram always change from one datagram to the next within this series of ICMP messages sent by your computer? And Which fields stay constant?

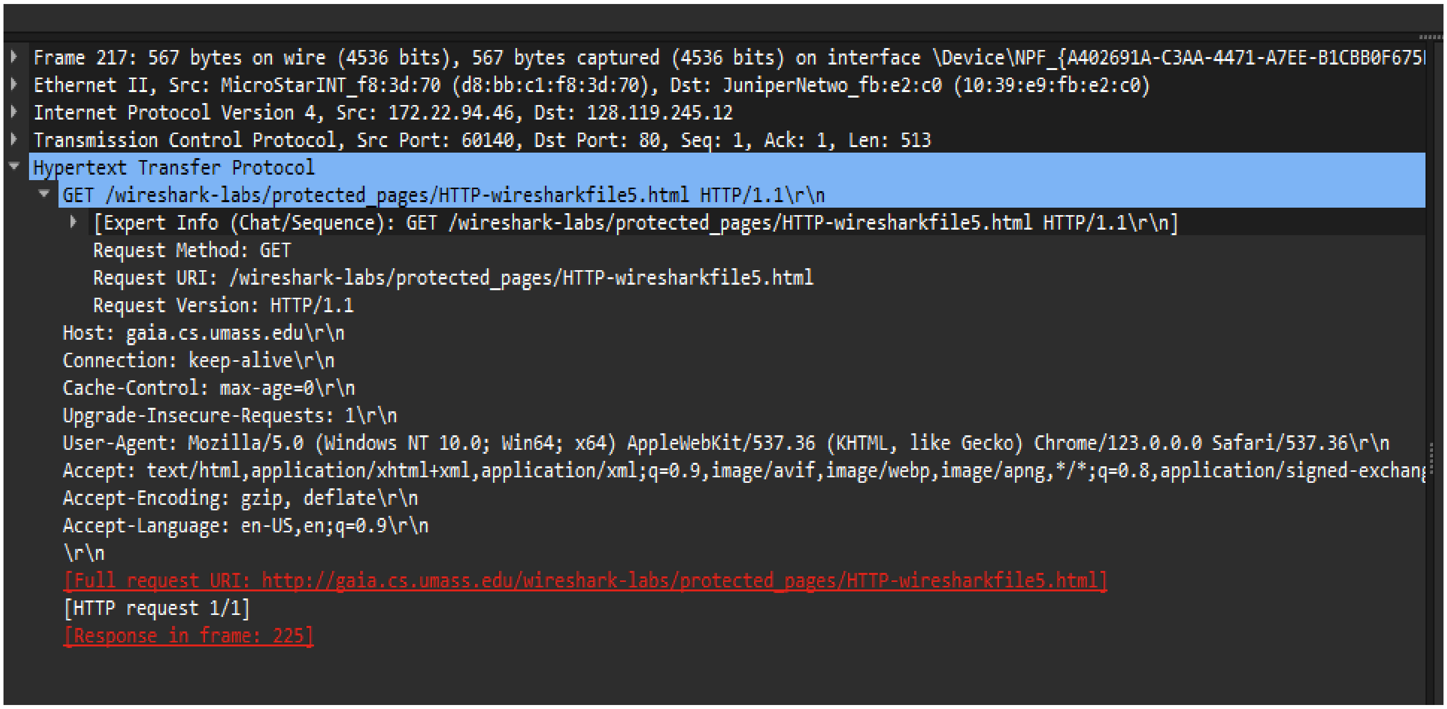
- The data field contains the IP header and first 8 bytes of original datagram's data.
- Identification, Time to live and Header checksum always change.
- version, protocol, header,Type of Service are stay constant

19. Enter the following URL into your browser

http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html

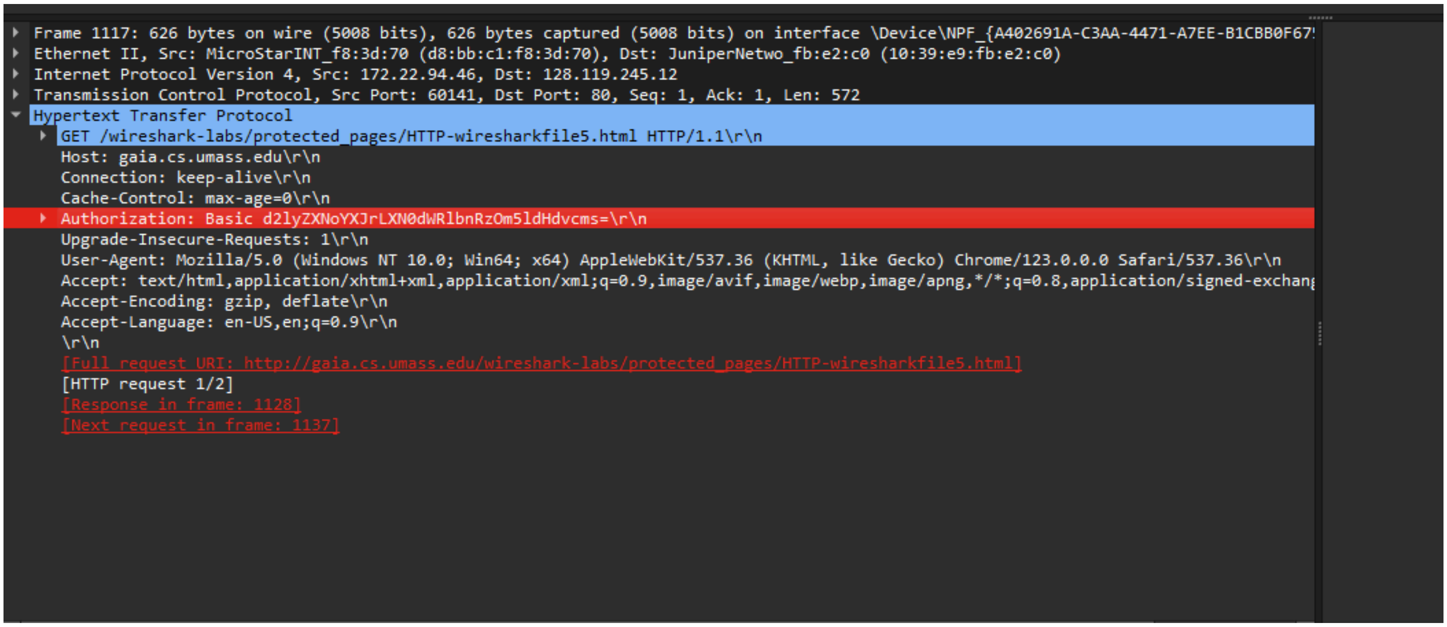
The username is "wireshark-students" (without the quotes), and the password is "network". When your browser sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

first time



```
Frame 217: 567 bytes on wire (4536 bits), 567 bytes captured (4536 bits) on interface \Device\NPF_{A402691A-C3AA-4471-A7EE-B1CBB0F675
Ethernet II, Src: MicroStarINT_f8:3d:70 (d8:bb:c1:f8:3d:70), Dst: JuniperNetwo_fb:e2:c0 (10:39:e9:fb:e2:c0)
Internet Protocol Version 4, Src: 172.22.94.46, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 60140, Dst Port: 80, Seq: 1, Ack: 1, Len: 513
Hypertext Transfer Protocol
  GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html
    Request Version: HTTP/1.1
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html]
  [HTTP request 1/1]
  [Response in frame: 225]
```

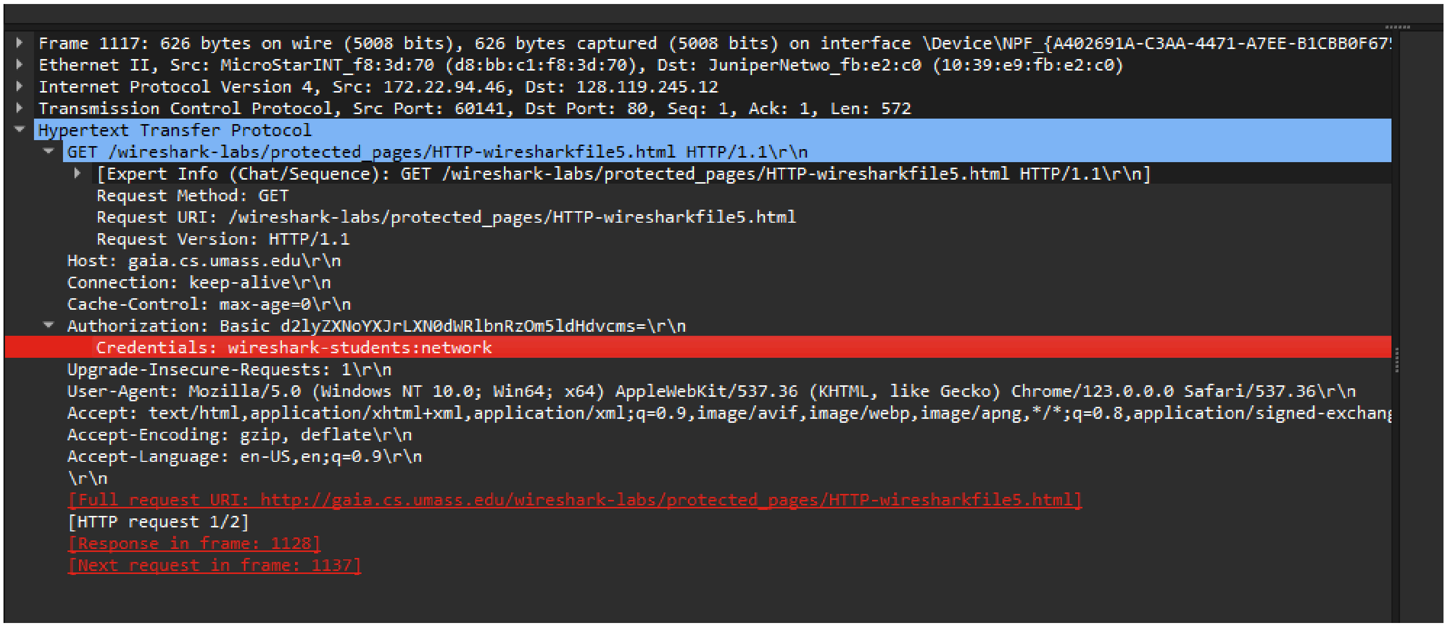
second time



```
Frame 1117: 626 bytes on wire (5008 bits), 626 bytes captured (5008 bits) on interface \Device\NPF_{A402691A-C3AA-4471-A7EE-B1CBB0F67
Ethernet II, Src: MicroStarINT_f8:3d:70 (d8:bb:c1:f8:3d:70), Dst: JuniperNetwo_fb:e2:c0 (10:39:e9:fb:e2:c0)
Internet Protocol Version 4, Src: 172.22.94.46, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 60141, Dst Port: 80, Seq: 1, Ack: 1, Len: 572
Hypertext Transfer Protocol
   GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1\r\n
   Host: gaia.cs.umass.edu\r\n
   Connection: keep-alive\r\n
   Cache-Control: max-age=0\r\n
   Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
   Upgrade-Insecure-Requests: 1\r\n
   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36\r\n
   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
   Accept-Encoding: gzip, deflate\r\n
   Accept-Language: en-US,en;q=0.9\r\n
   \r\n
   [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html]
   [HTTP request 1/2]
   [Response in frame: 1128]
   [Next request in frame: 1137]
```

The HTTP GET includes the Authorization: Basic: field

20. Extract credential from the second GET message.



```
Frame 1117: 626 bytes on wire (5008 bits), 626 bytes captured (5008 bits) on interface \Device\NPF_{A402691A-C3AA-4471-A7EE-B1CBB0F67
Ethernet II, Src: MicroStarINT_f8:3d:70 (d8:bb:c1:f8:3d:70), Dst: JuniperNetwo_fb:e2:c0 (10:39:e9:fb:e2:c0)
Internet Protocol Version 4, Src: 172.22.94.46, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 60141, Dst Port: 80, Seq: 1, Ack: 1, Len: 572
Hypertext Transfer Protocol
   GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1\r\n
      [Expert Info (Chat/Sequence): GET /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /wireshark-labs/protected_pages/HTTP-wiresharkfile5.html
      Request Version: HTTP/1.1
   Host: gaia.cs.umass.edu\r\n
   Connection: keep-alive\r\n
   Cache-Control: max-age=0\r\n
   Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
      Credentials: wireshark-students:network
   Upgrade-Insecure-Requests: 1\r\n
   User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36\r\n
   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
   Accept-Encoding: gzip, deflate\r\n
   Accept-Language: en-US,en;q=0.9\r\n
   \r\n
   [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html]
   [HTTP request 1/2]
   [Response in frame: 1128]
   [Next request in frame: 1137]
```

credentials : wireshark-students:network

THE END

SAKSHAM KUMAR

COMPUTER SCIENCE AND ENGINEERING
ID - 2022UCP1700
SECTION- D