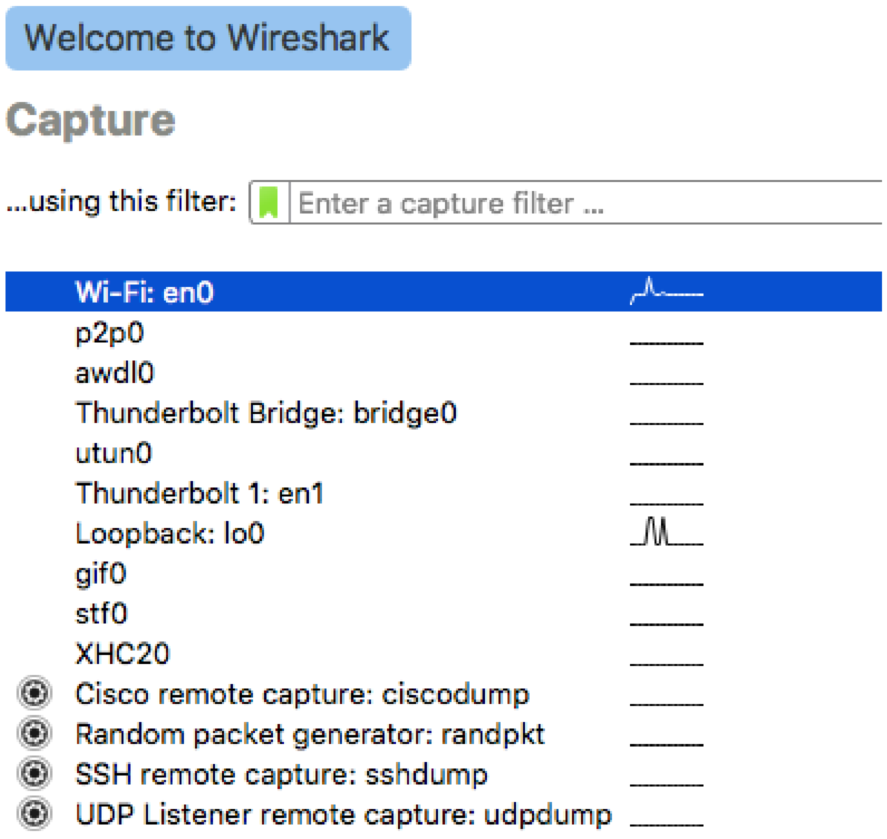
CSP334 : Computer Networks
Lab Assignment No 1
Assignment on Wireshark

Sahil
2016UCS0008

September 10, 2018

1 Network Interface:

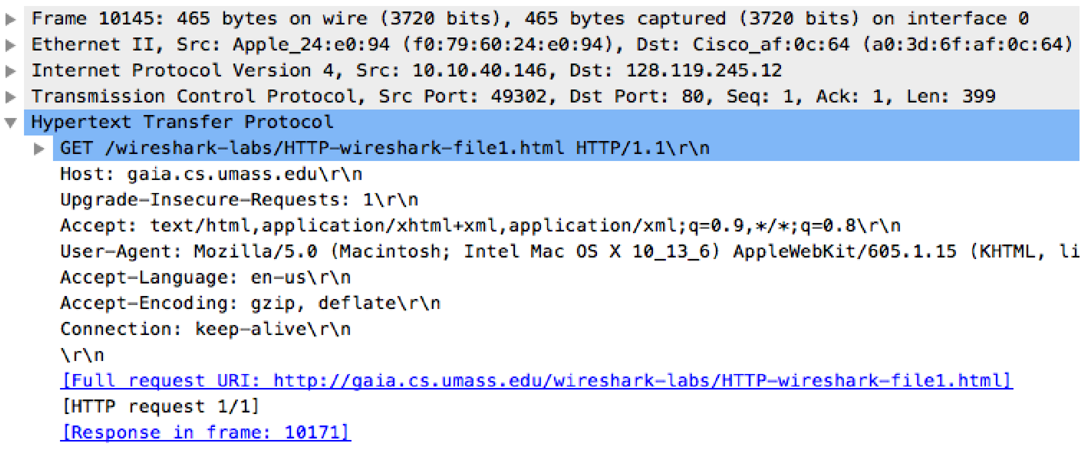
The network interfaces available on the computer are shown in the snapshot below. They include **Wi-Fi**, virtual wireless interface **p2p0**, Thunderbolt bridge, Thunderbolt 1, Software Network interface (gif0) and tunnel interface (stf0).



```
Wi-Fi: en0
p2p0
awdl0
Thunderbolt Bridge: bridge0
utun0
Thunderbolt 1: en1
Loopback: lo0
gif0
stf0
XHC20
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
```

Wi-Fi network interface was eventually selected.

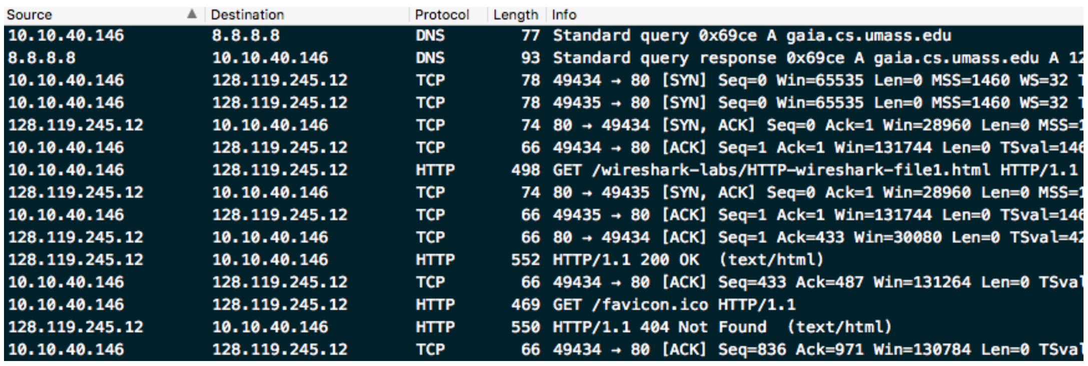
2 Application Layer protocol used:



```
▶ Frame 10145: 465 bytes on wire (3720 bits), 465 bytes captured (3720 bits) on interface 0
▶ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
▶ Internet Protocol Version 4, Src: 10.10.40.146, Dst: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 49302, Dst Port: 80, Seq: 1, Ack: 1, Len: 399
▼ Hypertext Transfer Protocol
  ▶ GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/605.1.15 (KHTML, li
    Accept-Language: en-us\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
    [HTTP request 1/1]
    [Response in frame: 10171]
```

The application layer protocol used is **HTTP**, i.e. HyperText Transfer Protocol, as highlighted in the frame captured.

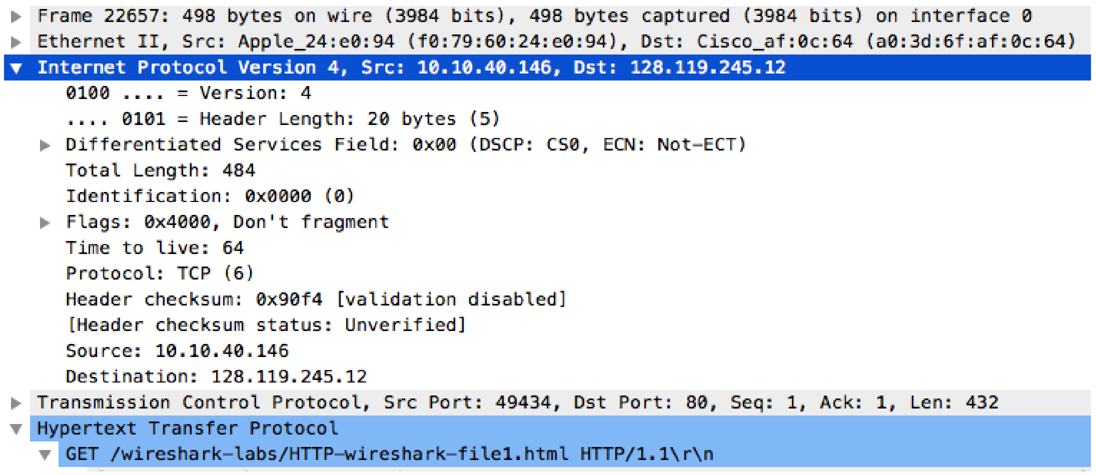
3 Other protocols used:



```
Source           Destination      Protocol  Length  Info
10.10.40.146     8.8.8.8          DNS           77  Standard query 0x69ce A gaia.cs.umass.edu
8.8.8.8          10.10.40.146     DNS           93  Standard query response 0x69ce A gaia.cs.umass.edu A 12
10.10.40.146     128.119.245.12   TCP           78  49434 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 T
10.10.40.146     128.119.245.12   TCP           78  49435 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 T
128.119.245.12   10.10.40.146     TCP           74  80 → 49434 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1
10.10.40.146     128.119.245.12   TCP           66  49434 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=146
10.10.40.146     128.119.245.12   HTTP         498  GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
128.119.245.12   10.10.40.146     TCP           74  80 → 49435 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1
10.10.40.146     128.119.245.12   TCP           66  49435 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=146
128.119.245.12   10.10.40.146     TCP           66  80 → 49434 [ACK] Seq=1 Ack=433 Win=30080 Len=0 TSval=42
128.119.245.12   10.10.40.146     HTTP         552  HTTP/1.1 200 OK  (text/html)
10.10.40.146     128.119.245.12   TCP           66  49434 → 80 [ACK] Seq=433 Ack=487 Win=131264 Len=0 TSval
10.10.40.146     128.119.245.12   HTTP         469  GET /favicon.ico HTTP/1.1
128.119.245.12   10.10.40.146     HTTP         550  HTTP/1.1 404 Not Found  (text/html)
10.10.40.146     128.119.245.12   TCP           66  49434 → 80 [ACK] Seq=836 Ack=971 Win=130784 Len=0 TSval
```

The other protocols used are **DNS** which in turn used UDP and **TCP**.
IP is not displayed in the packet listing window since it is always used.

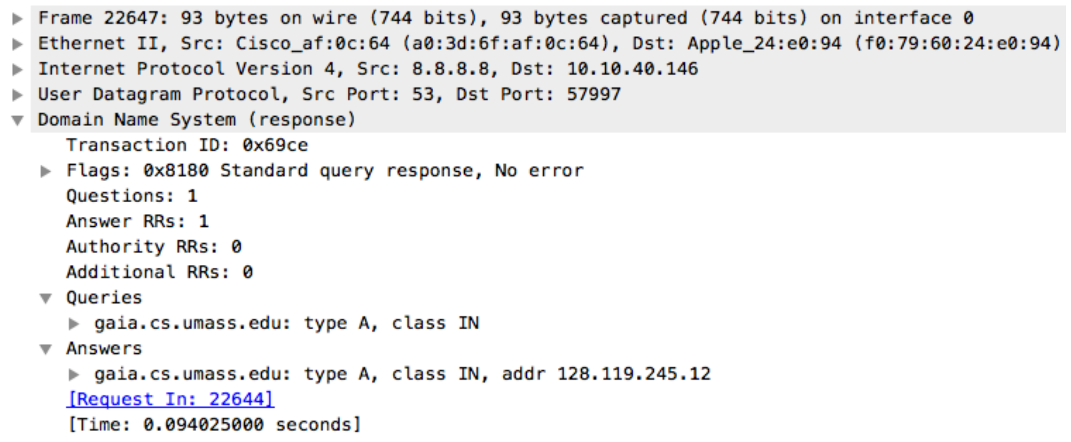
4 IPA of source and destination:



```
▶ Frame 22657: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface 0
▶ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
▼ Internet Protocol Version 4, Src: 10.10.40.146, Dst: 128.119.245.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 484
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x90f4 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.10.40.146
    Destination: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 49434, Dst Port: 80, Seq: 1, Ack: 1, Len: 432
▼ Hypertext Transfer Protocol
  ▼ GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
```

The IPA of the **source machine** is: 10.10.40.146
The IPA of the **destination machine** is: 128.119.245.12

We can ascertain that the IPA of the destination is indeed the same as that observed in the wireshark by either entering the IPA in the web browser since its an HTTP request, or we can do ping to the web address requested to get its IPA.
Another alternative way is to look at the following **DNS** packet captured, which clearly mentions the resolved IPA of the requested website, which is the destination.



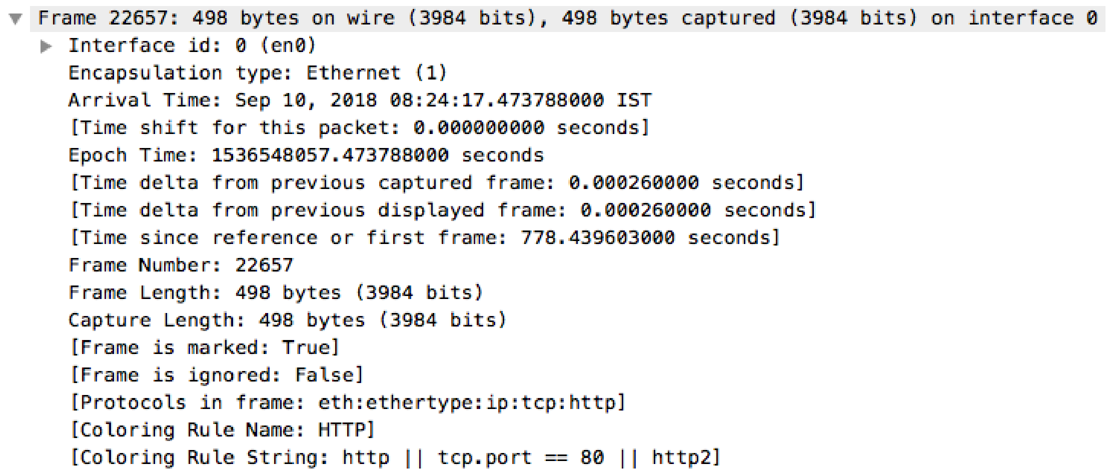
```
▶ Frame 22647: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface 0
▶ Ethernet II, Src: Cisco_af:0c:64 (a0:3d:6f:af:0c:64), Dst: Apple_24:e0:94 (f0:79:60:24:e0:94)
▶ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.10.40.146
▶ User Datagram Protocol, Src Port: 53, Dst Port: 57997
▼ Domain Name System (response)
    Transaction ID: 0x69ce
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▶ gaia.cs.umass.edu: type A, class IN
  ▼ Answers
    ▶ gaia.cs.umass.edu: type A, class IN, addr 128.119.245.12
    [Request In: 22644]
    [Time: 0.094025000 seconds]
```

5 Class of IPA:

IPA of **source** belongs to **class A** since class A contains IPA from 0.0.0.0 to 127.255.255.255 whereas the IPA of **destination** belongs to **class B** since class B contains IPA from 128.0.0.0 to 191.255.255.255.

6 Frame: Information about packet

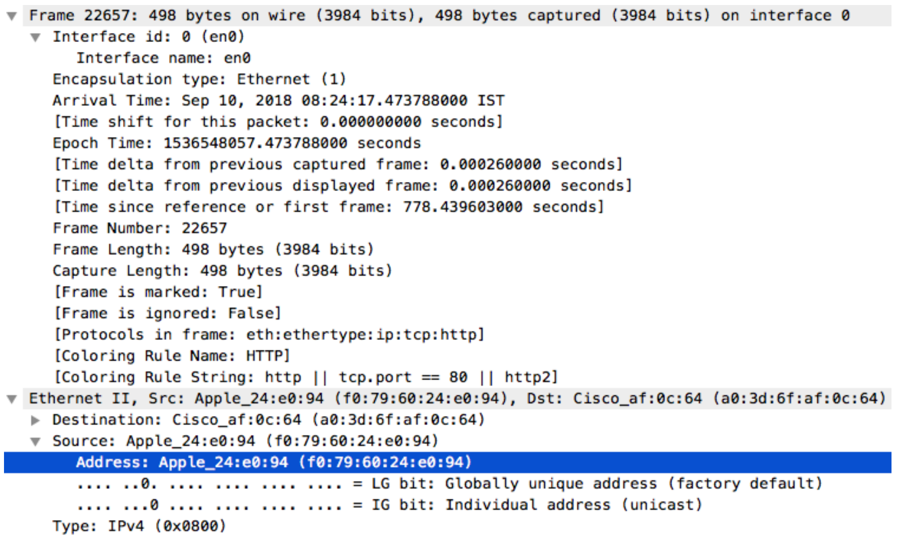
The no. of bits captured in the HTTP packet: 3984
The time at which the packet was captured: Sep 10, 2018 08 : 24 : 17.473788000 IST



```
▼ Frame 22657: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface 0
  ▶ Interface id: 0 (en0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 10, 2018 08:24:17.473788000 IST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1536548057.473788000 seconds
    [Time delta from previous captured frame: 0.000260000 seconds]
    [Time delta from previous displayed frame: 0.000260000 seconds]
    [Time since reference or first frame: 778.439603000 seconds]
    Frame Number: 22657
    Frame Length: 498 bytes (3984 bits)
    Capture Length: 498 bytes (3984 bits)
    [Frame is marked: True]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

7 Interface ID and address of interface:

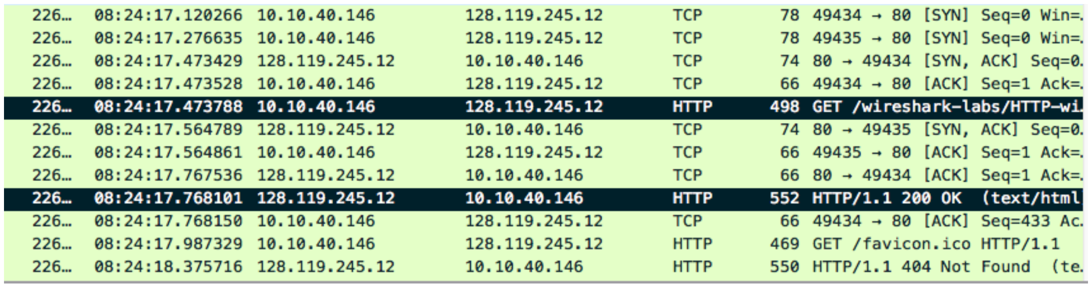
The interface ID used is: 0 (en0)
The address of the interface is: f0:79:60:24:e0:94.



```
▼ Frame 22657: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface 0
  ▼ Interface id: 0 (en0)
      Interface name: en0
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 10, 2018 08:24:17.473788000 IST
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1536548057.473788000 seconds
    [Time delta from previous captured frame: 0.000260000 seconds]
    [Time delta from previous displayed frame: 0.000260000 seconds]
    [Time since reference or first frame: 778.439603000 seconds]
    Frame Number: 22657
    Frame Length: 498 bytes (3984 bits)
    Capture Length: 498 bytes (3984 bits)
    [Frame is marked: True]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
▼ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
  ▶ Destination: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
  ▼ Source: Apple_24:e0:94 (f0:79:60:24:e0:94)
      Address: Apple_24:e0:94 (f0:79:60:24:e0:94)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

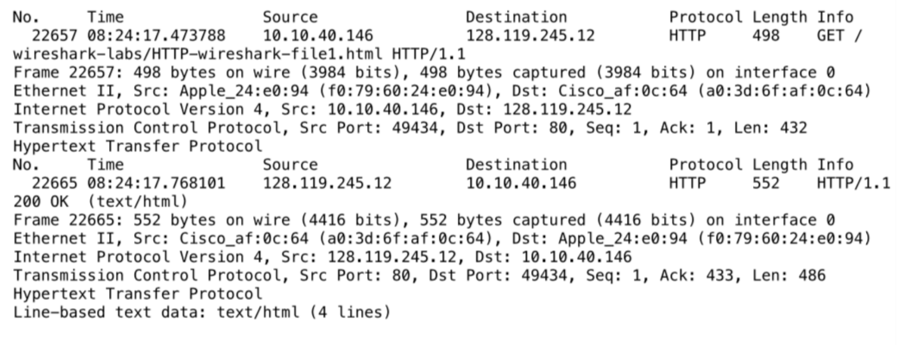
8 Time taken between HTTP GET and HTTP OK reply:

The HTTP GET message sent and the HTTP OK reply are highlighted, the time taken = 17.768101 − 17.473788 = 0.294313 seconds.



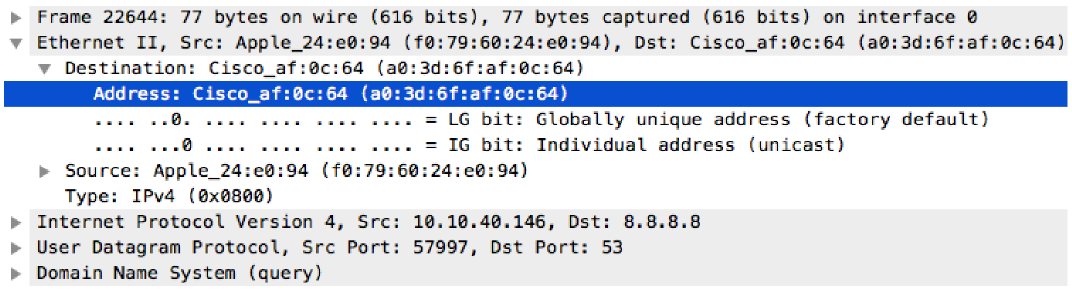
```
226…  08:24:17.120266  10.10.40.146    128.119.245.12  TCP    78  49434 → 80 [SYN] Seq=0 Win=…
226…  08:24:17.276635  10.10.40.146    128.119.245.12  TCP    78  49435 → 80 [SYN] Seq=0 Win=…
226…  08:24:17.473429  128.119.245.12  10.10.40.146    TCP    74  80 → 49434 [SYN, ACK] Seq=0…
226…  08:24:17.473528  10.10.40.146    128.119.245.12  TCP    66  49434 → 80 [ACK] Seq=1 Ack=…
226…  08:24:17.473788  10.10.40.146    128.119.245.12  HTTP  498  GET /wireshark-labs/HTTP-wi…
226…  08:24:17.564789  128.119.245.12  10.10.40.146    TCP    74  80 → 49435 [SYN, ACK] Seq=0…
226…  08:24:17.564861  10.10.40.146    128.119.245.12  TCP    66  49435 → 80 [ACK] Seq=1 Ack=…
226…  08:24:17.767536  128.119.245.12  10.10.40.146    TCP    66  80 → 49434 [ACK] Seq=1 Ack=…
226…  08:24:17.768101  128.119.245.12  10.10.40.146    HTTP  552  HTTP/1.1 200 OK  (text/html
226…  08:24:17.768150  10.10.40.146    128.119.245.12  TCP    66  49434 → 80 [ACK] Seq=433 Ac…
226…  08:24:17.987329  10.10.40.146    128.119.245.12  HTTP  469  GET /favicon.ico HTTP/1.1
226…  08:24:18.375716  128.119.245.12  10.10.40.146    HTTP  550  HTTP/1.1 404 Not Found  (te…
```

9 HTTP GET and OK messages:


```
No.     Time               Source                Destination           Protocol Length Info
  22657 08:24:17.473788    10.10.40.146          128.119.245.12        HTTP     498    GET /
wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
Frame 22657: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface 0
Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
Internet Protocol Version 4, Src: 10.10.40.146, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 49434, Dst Port: 80, Seq: 1, Ack: 1, Len: 432
Hypertext Transfer Protocol
No.     Time               Source                Destination           Protocol Length Info
  22665 08:24:17.768101    128.119.245.12        10.10.40.146          HTTP     552    HTTP/1.1
200 OK  (text/html)
Frame 22665: 552 bytes on wire (4416 bits), 552 bytes captured (4416 bits) on interface 0
Ethernet II, Src: Cisco_af:0c:64 (a0:3d:6f:af:0c:64), Dst: Apple_24:e0:94 (f0:79:60:24:e0:94)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.10.40.146
Transmission Control Protocol, Src Port: 80, Dst Port: 49434, Seq: 1, Ack: 433, Len: 486
Hypertext Transfer Protocol
Line-based text data: text/html (4 lines)
```

The HTTP GET and OK messages are as shown above.

10 Destination physical address of the first packet captured and device it belongs to:

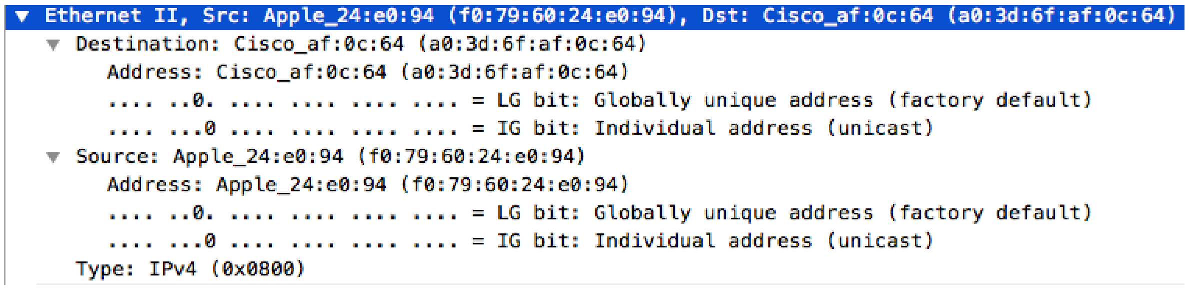

```
▶ Frame 22644: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
▼ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
  ▼ Destination: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
      Address: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▶ Source: Apple_24:e0:94 (f0:79:60:24:e0:94)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.10.40.146, Dst: 8.8.8.8
▶ User Datagram Protocol, Src Port: 57997, Dst Port: 53
▶ Domain Name System (query)
```

The destination physical address of the first packet (HTTP) captured is **a0:3d:6f:af:0c:64** and it belongs to the device Cisco.

11 Bytes of header in the first frame:

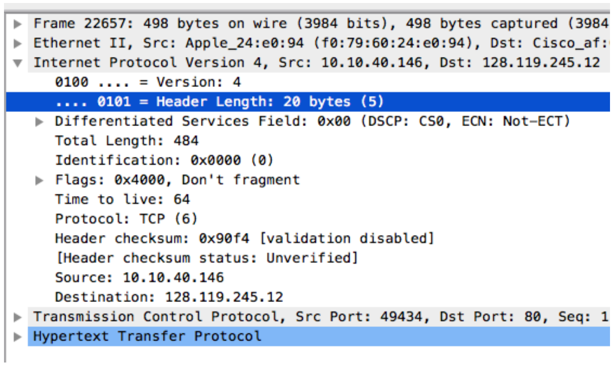
The bytes of header in the first frame is the sum of bytes of header at the different layers. This is not directly visible, but we have to add the **Header length** field values for the Ethernet, IP and transport layers.

Ethernet Header: Size = 6 + 6 + 2 = 14 bytes for the 3 fields shown.



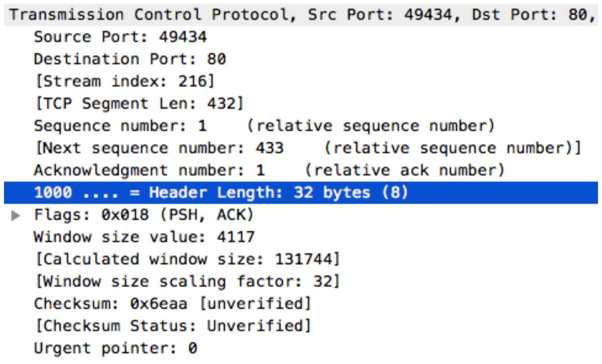
```
▼ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
  ▼ Destination: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
      Address: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▼ Source: Apple_24:e0:94 (f0:79:60:24:e0:94)
      Address: Apple_24:e0:94 (f0:79:60:24:e0:94)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

IP Header: Size = 20 bytes



```
▶ Frame 22657: 498 bytes on wire (3984 bits), 498 bytes captured (3984
▶ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:
▼ Internet Protocol Version 4, Src: 10.10.40.146, Dst: 128.119.245.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 484
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x90f4 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.10.40.146
    Destination: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 49434, Dst Port: 80, Seq: 1
▶ Hypertext Transfer Protocol
```

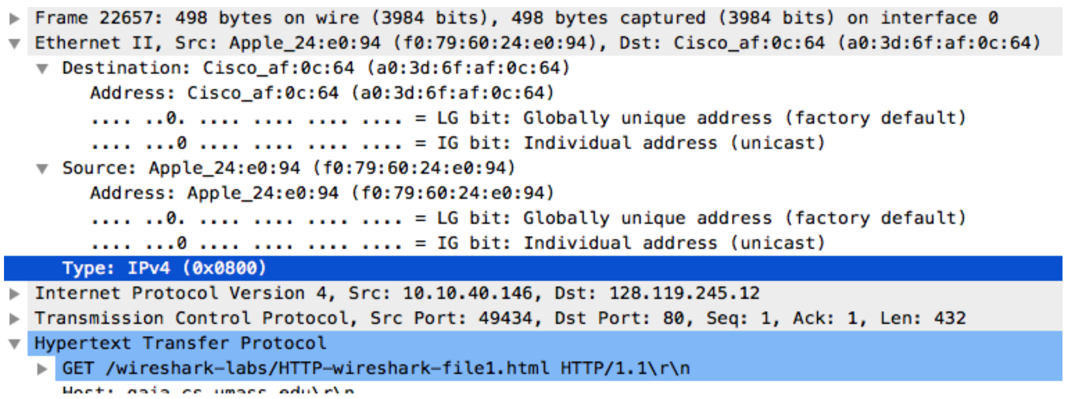
TCP Header: Size = 32 bytes



```
Transmission Control Protocol, Src Port: 49434, Dst Port: 80,
   Source Port: 49434
   Destination Port: 80
   [Stream index: 216]
   [TCP Segment Len: 432]
   Sequence number: 1    (relative sequence number)
   [Next sequence number: 433    (relative sequence number)]
   Acknowledgment number: 1    (relative ack number)
   1000 .... = Header Length: 32 bytes (8)
▶  Flags: 0x018 (PSH, ACK)
   Window size value: 4117
   [Calculated window size: 131744]
   [Window size scaling factor: 32]
   Checksum: 0x6eaa [unverified]
   [Checksum Status: Unverified]
   Urgent pointer: 0
```

Thus, total bytes of header = 14 + 20 + 32 = 66 bytes

12 How to know if Ethernet header contains an IP packet?



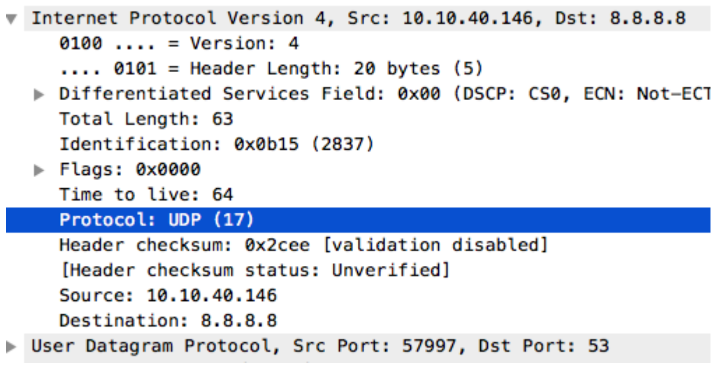
```
▶ Frame 22657: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface 0
▼ Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
  ▼ Destination: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
      Address: Cisco_af:0c:64 (a0:3d:6f:af:0c:64)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▼ Source: Apple_24:e0:94 (f0:79:60:24:e0:94)
      Address: Apple_24:e0:94 (f0:79:60:24:e0:94)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 10.10.40.146, Dst: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 49434, Dst Port: 80, Seq: 1, Ack: 1, Len: 432
▼ Hypertext Transfer Protocol
  ▶ GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
```

We can determine by looking at the Ethernet header of the frame whether it contains an IP packet since the field **Type** contains this detail as highlighted above.

13 How to know if the first packet captured has TCP or UDP as transport protocol by looking at the IP header?

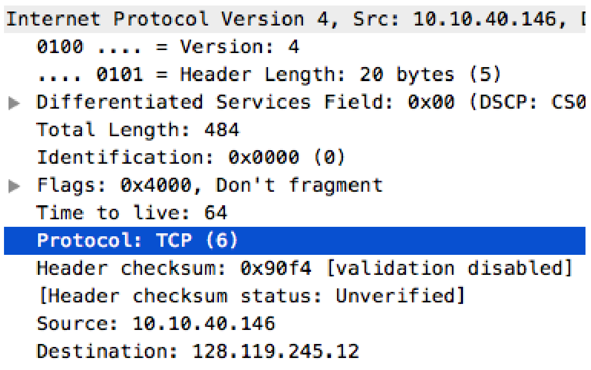
We can know whether the packet captured has TCP or UDP as transport protocol by looking at the **Protocol** field in the IP header. If we consider DNS as the first packet captured, it has UDP whereas considering HTTP as the first packet, it has TCP.

DNS packet:



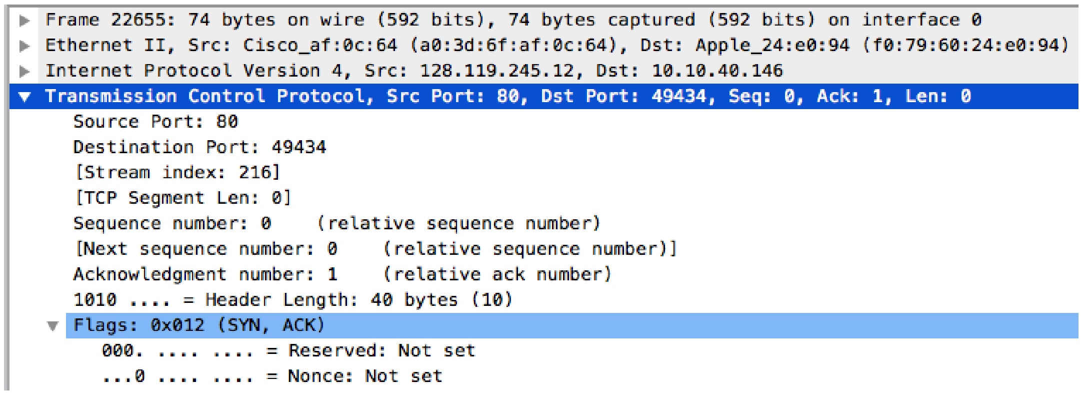
```
▼ Internet Protocol Version 4, Src: 10.10.40.146, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT
    Total Length: 63
    Identification: 0x0b15 (2837)
  ▶ Flags: 0x0000
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0x2cee [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.10.40.146
    Destination: 8.8.8.8
▶ User Datagram Protocol, Src Port: 57997, Dst Port: 53
```

HTTP packet:



```
Internet Protocol Version 4, Src: 10.10.40.146, I
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0
    Total Length: 484
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0x90f4 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.10.40.146
    Destination: 128.119.245.12
```

14 Source and destination ports in the SYN, ACK:

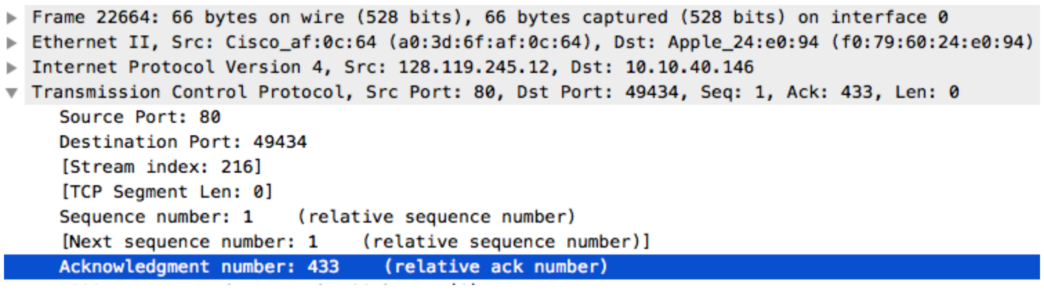


```
▶ Frame 22655: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▶ Ethernet II, Src: Cisco_af:0c:64 (a0:3d:6f:af:0c:64), Dst: Apple_24:e0:94 (f0:79:60:24:e0:94)
▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.10.40.146
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 49434, Seq: 0, Ack: 1, Len: 0
     Source Port: 80
     Destination Port: 49434
     [Stream index: 216]
     [TCP Segment Len: 0]
     Sequence number: 0    (relative sequence number)
     [Next sequence number: 0    (relative sequence number)]
     Acknowledgment number: 1    (relative ack number)
     1010 .... = Header Length: 40 bytes (10)
   ▼ Flags: 0x012 (SYN, ACK)
       000. .... .... = Reserved: Not set
       ...0 .... .... = Nonce: Not set
```

In the SYN, ACK message, the source port is 80 and the destination port is 49434 as shown above. It has to be a well-known port for the server which is source here since client first sent a SYN in response to which server sends a SYN, ACK message. It cannot be the same for the client since client sends a request from ephemeral port number.

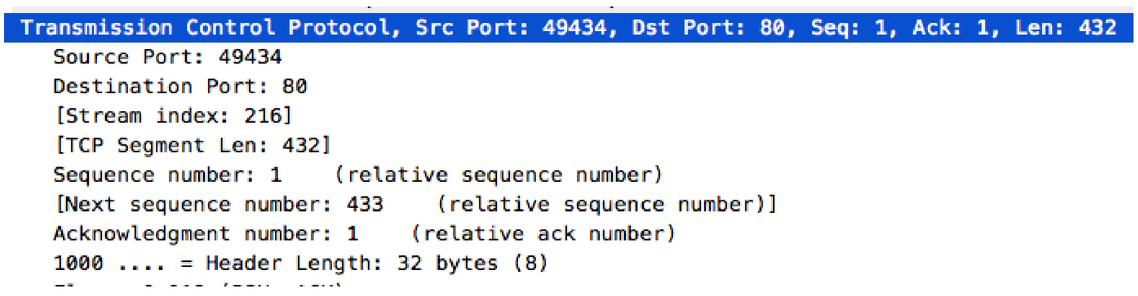
15 Server Hello message has 1 as relative sequence number and 185 as relative acknowledgement number:

Initially, the sequence number starts from 0 when the client requests a connection, and when the connection has been setup, the next packet from client has a relative sequence number of 1, this is incremented each time a new request is made. The acknowledgement number is 433 as shown,



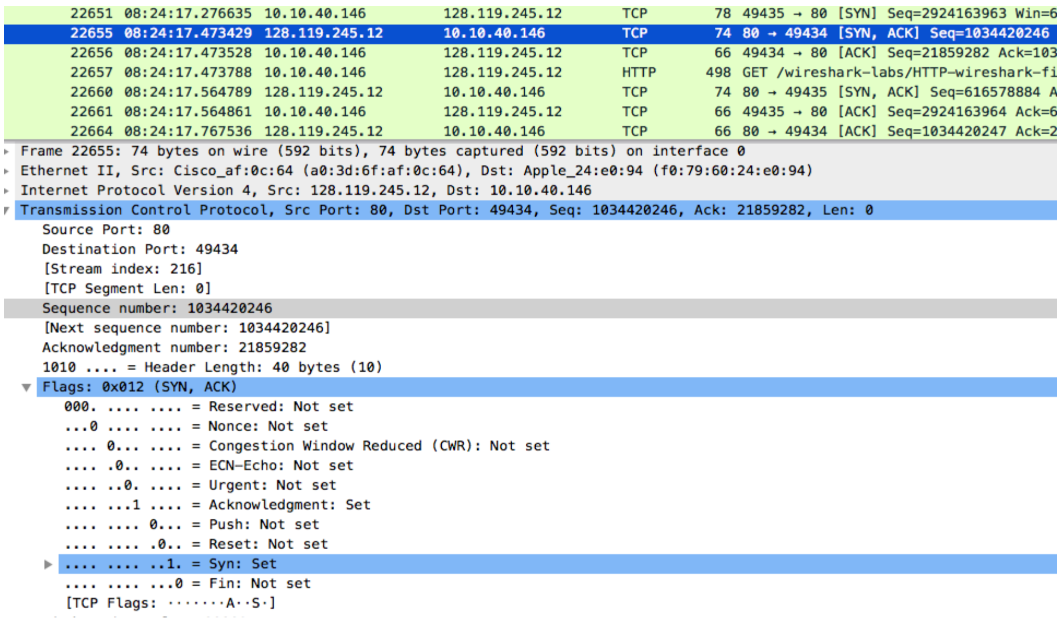
```
▶ Frame 22664: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: Cisco_af:0c:64 (a0:3d:6f:af:0c:64), Dst: Apple_24:e0:94 (f0:79:60:24:e0:94)
▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.10.40.146
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 49434, Seq: 1, Ack: 433, Len: 0
    Source Port: 80
    Destination Port: 49434
    [Stream index: 216]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 433    (relative ack number)
```

This is because the request sent from the client had a TCP message (header + payload) of size 432 bits, so the acknowledgement number for the request is (size of request) + 1.



```
Transmission Control Protocol, Src Port: 49434, Dst Port: 80, Seq: 1, Ack: 1, Len: 432
    Source Port: 49434
    Destination Port: 80
    [Stream index: 216]
    [TCP Segment Len: 432]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 433    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

16 First sequence number sent by the server to the client:



```
22651 08:24:17.276635 10.10.40.146    128.119.245.12  TCP    78 49435 → 80 [SYN] Seq=2924163963 Win=6
22655 08:24:17.473429 128.119.245.12  10.10.40.146    TCP    74 80 → 49434 [SYN, ACK] Seq=1034420246
22656 08:24:17.473528 10.10.40.146    128.119.245.12  TCP    66 49434 → 80 [ACK] Seq=21859282 Ack=103
22657 08:24:17.473788 10.10.40.146    128.119.245.12  HTTP  498 GET /wireshark-labs/HTTP-wireshark-fi
22660 08:24:17.564789 128.119.245.12  10.10.40.146    TCP    74 80 → 49435 [SYN, ACK] Seq=616578884 A
22661 08:24:17.564861 10.10.40.146    128.119.245.12  TCP    66 49435 → 80 [ACK] Seq=2924163964 Ack=6
22664 08:24:17.767536 128.119.245.12  10.10.40.146    TCP    66 80 → 49434 [ACK] Seq=1034420247 Ack=2
▸ Frame 22655: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: Cisco_af:0c:64 (a0:3d:6f:af:0c:64), Dst: Apple_24:e0:94 (f0:79:60:24:e0:94)
▸ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.10.40.146
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 49434, Seq: 1034420246, Ack: 21859282, Len: 0
    Source Port: 80
    Destination Port: 49434
    [Stream index: 216]
    [TCP Segment Len: 0]
    Sequence number: 1034420246
    [Next sequence number: 1034420246]
    Acknowledgment number: 21859282
    1010 .... = Header Length: 40 bytes (10)
  ▾ Flags: 0x012 (SYN, ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ▸ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·······A··S·]
```

The first sequence number sent by the server to the client is not 0 because intially the client sends a random sequence number which then is incremented by the server by 1 and sent to the client along with an acknowledgement.
Also, it would have been shown as 0 if we viewed relative sequence numbers since wireshark starts the sequence numbers from 0 in relative ordering.