# CHEN Le 59039379

# Problem 1: Capturing Web Browser HTTP Traffic

## What languages (if any) does your browser indicate that it can accept to the server in HTTP Get message?

Accept-Language: en,zh-CN;q=0.9,zh;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n

English and Chinese.

## What application layer protocol does your web browser access above website?

Hypertext Transfer Protocol

HTTP

## What transport layer protocol does this HTTP GET packet use?

TCP

## What is the destination IP address and port of the HTTP GET message?

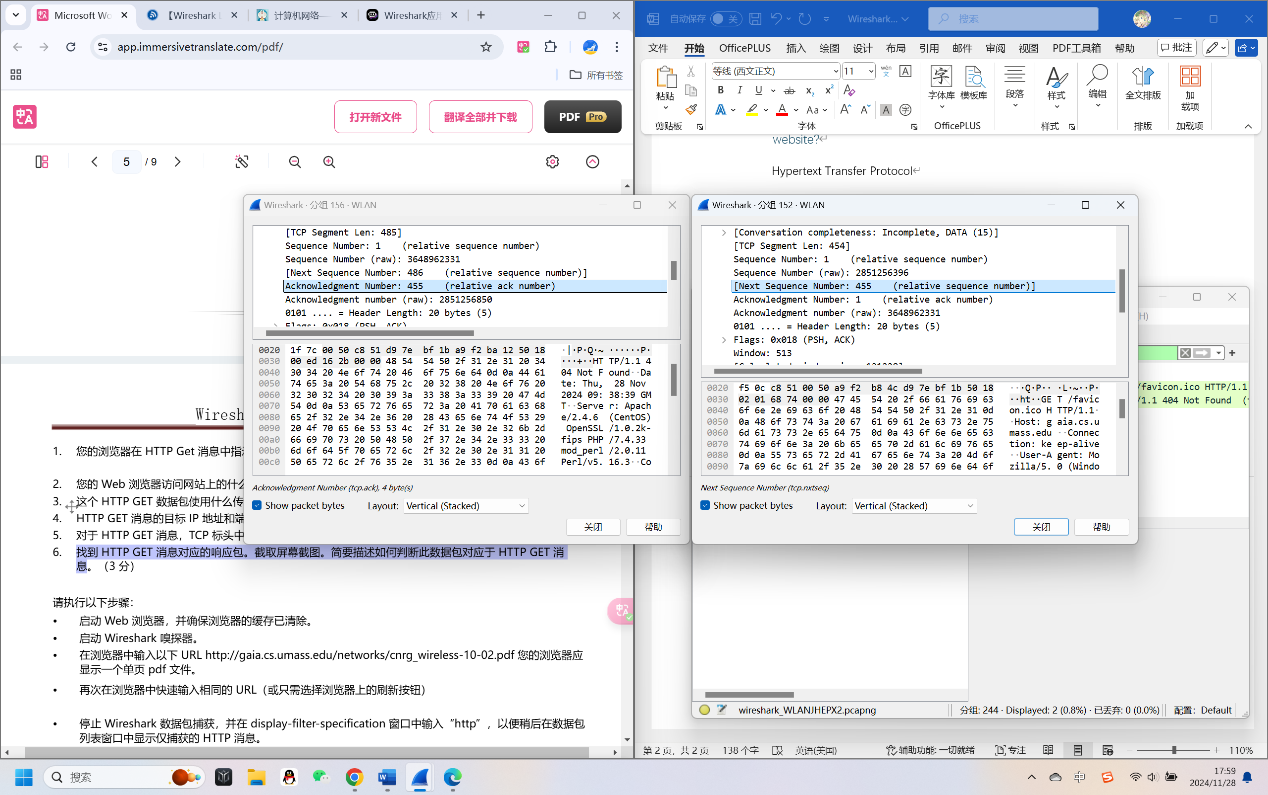
Dst: 128.119.245.12

Destination Port: 80

## For the HTTP GET message, what is the next sequence number in the TCP header?

[Next Sequence Number: 455 (relative sequence number)]

## Find the response packet corresponding to the HTTP GET message. Take a screenshot. Briefly describe how you are able to tell this packet corresponds to the HTTP GET message.



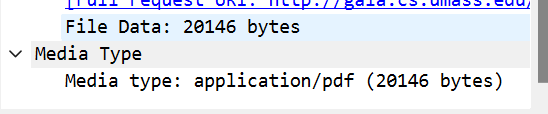
Next Seq and ACK correspond.

## Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

There is no “IF-MODIFIED-SINCE”.

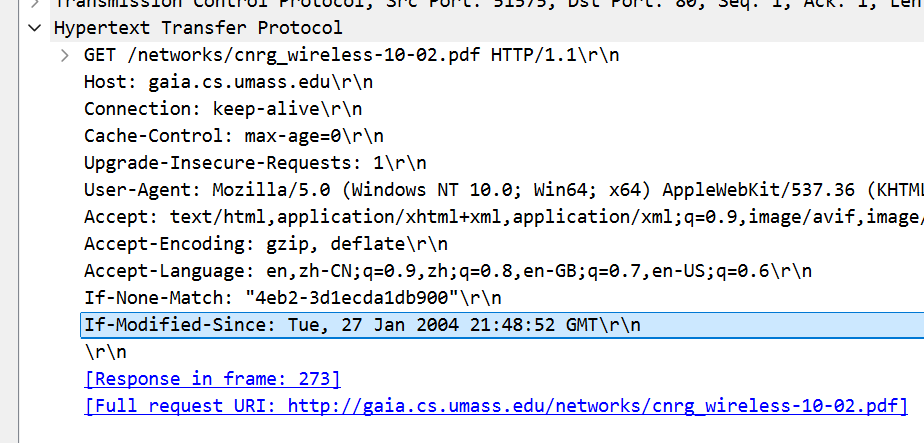
## Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

No.



## Now inspect the contents of the second HTTP GET request that requests the pdf file. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:”header?

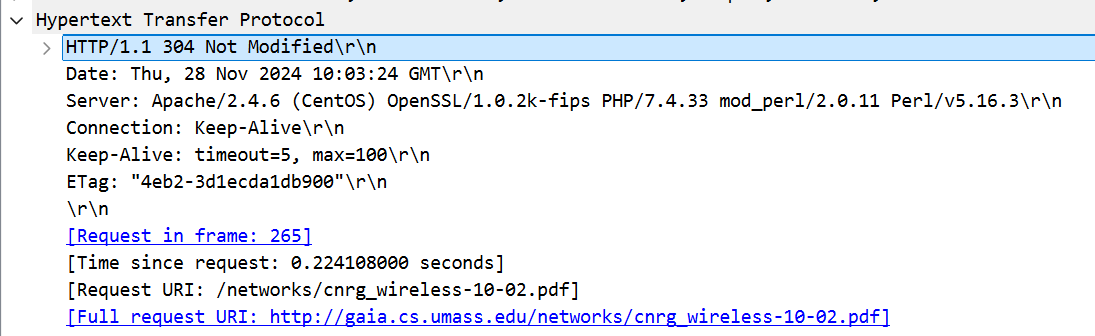
Yes.



## What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

HTTP/1.1 304 Not Modified\r\n.

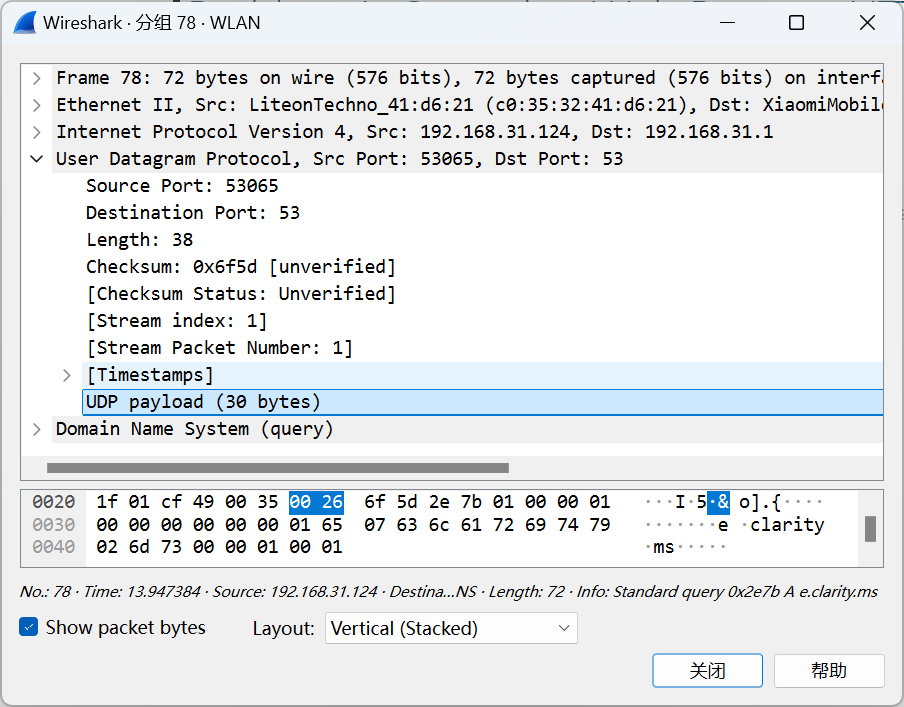
No. No file data.



# Problem 2: Capturing DNS Traffic

## Are they sent over UDP or TCP? (2 marks) How can you tell?

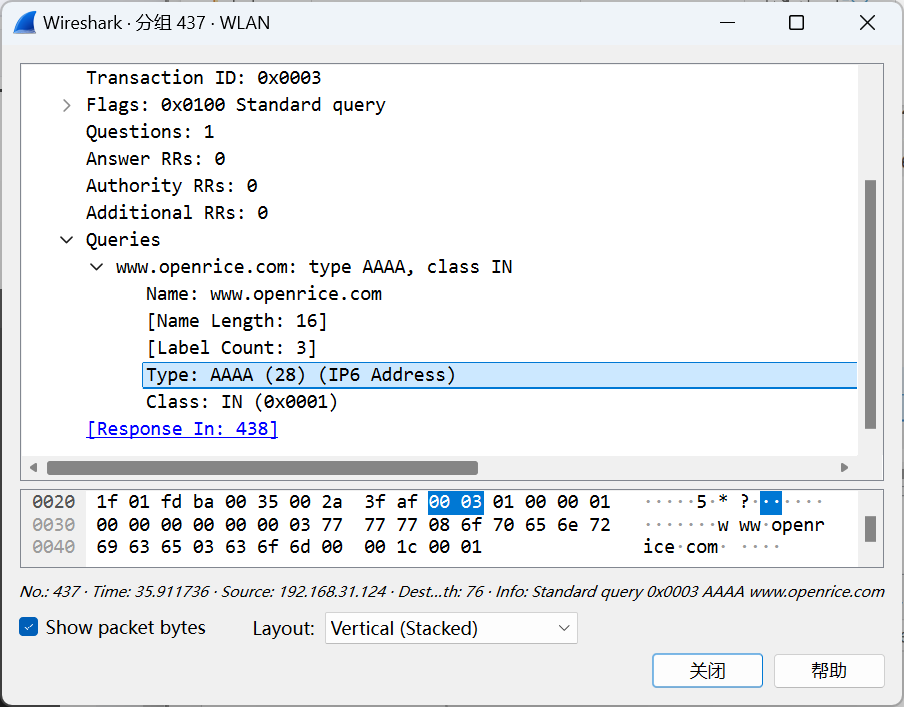
UDP.



## In the Packet List panel, look for the DNS query packet (Standard query 0x???? A www.openrice.com) from the Info column. What “Type” of DNS query is it? Does the query message contain any “answers”?

Type: AAAA (28) (IP6 Address)

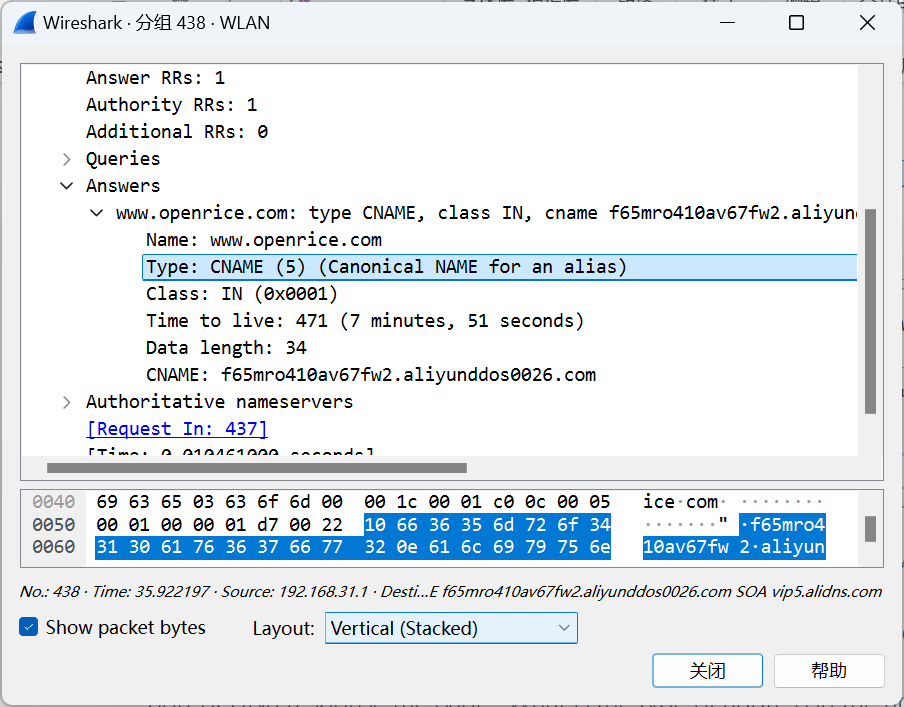
No “answers”.



## Look for the DNS response packet (Standard query response 0x???? A www.openrice.com). Examine the DNS response message. What is the source port of DNS response message? What is the type of address in the answer of the response message?

Source Port: 53.

Type: CNAME (5) (Canonical NAME for an alias)



## List the IP address of [www.openrice.com](http://www.openrice.com).

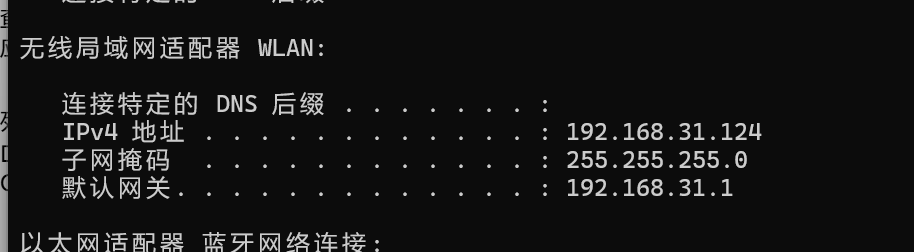
170.33.8.214

CNAME: f65mro410av67fw2.aliyunddos0026.com

Mname: vip5.alidns.com

## To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server (If you are using OSX, you may check it in “System preferences/ Network/ Advanced/DNS”). Are these two IP addresses the same?

192.168.31.1



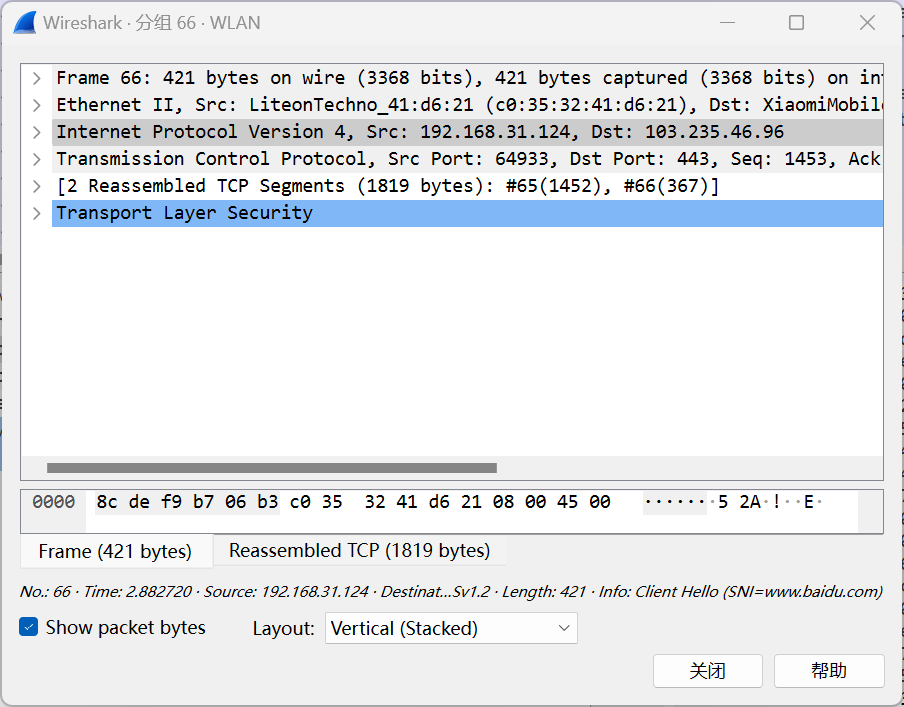
It’ s the same.

# Problem 3: Capturing SSL Traffic

## Find a pair of client and server. Find the first 8 Ethernet frames in their session. For each frame, specify the source address and destination address, determine the number of SSL records that are included in the frame, and list the SSL record types that are included in the frame. Draw a table to show the packet exchange between client and server. An example of the table is shown as follows. (There might be segments missed by Wireshark, e.g., “TCP Previous segment not captured, Ignored unknown Record”. Find a session that has no missing segment.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frame | Source address | Des. address | SSL count | SSL Type |
| 53 | c0:35:32:41:d6:21 | 8c:de:f9:b7:06:b3 | 1 | Application Data |
| 66 | c0:35:32:41:d6:21 | 8c:de:f9:b7:06:b3 | 1 | Handshake |
| 68 | 8c:de:f9:b7:06:b3 | c0:35:32:41:d6:21 | 1 | Handshake |
| 71 | 8c:de:f9:b7:06:b3 | c0:35:32:41:d6:21 | 3 | Handshake |
| 73 | c0:35:32:41:d6:21 | 8c:de:f9:b7:06:b3 | 3 | Change Cipher Spec & Handshake |
| 74 | 8c:de:f9:b7:06:b3 | c0:35:32:41:d6:21 | 1 | Application Data |
| 78 | 8c:de:f9:b7:06:b3 | c0:35:32:41:d6:21 | 3 | Change Cipher Spec & Handshake |
| 89 | c0:35:32:41:d6:21 | 8c:de:f9:b7:06:b3 | 1 | Application Data |

Specify the IP addresses of client and server respectively. How can you tell?

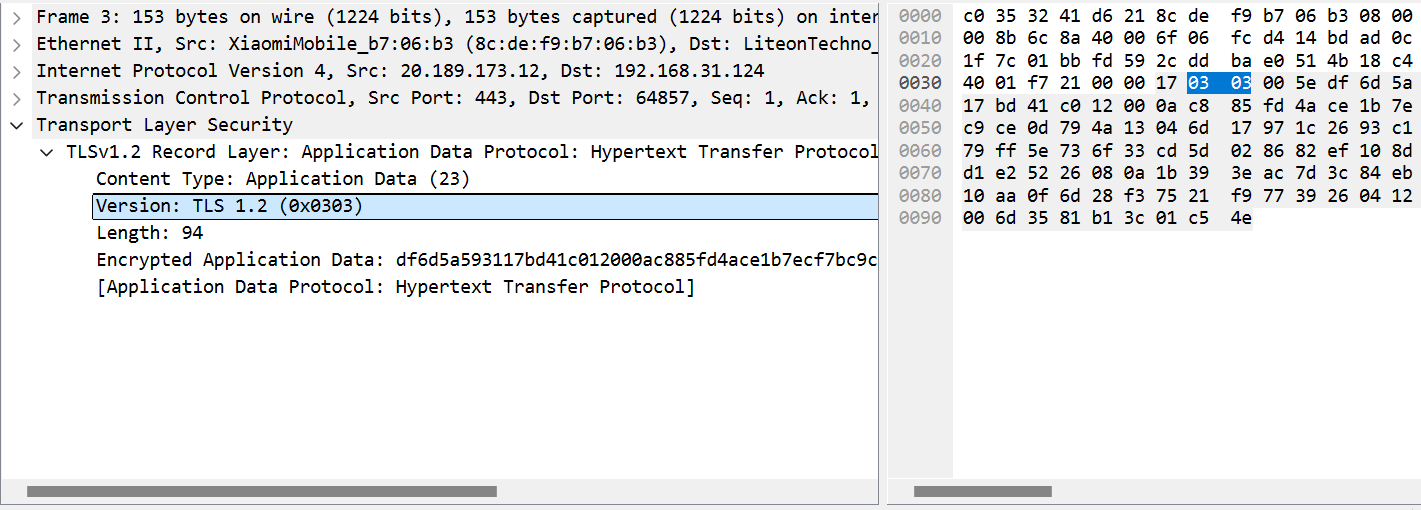


## Each of the SSL records begins with the same three fields (with possibly different values). One of these fields is “content type” and has length of one byte. List all three fields and their lengths.

Content Type: 1 byte

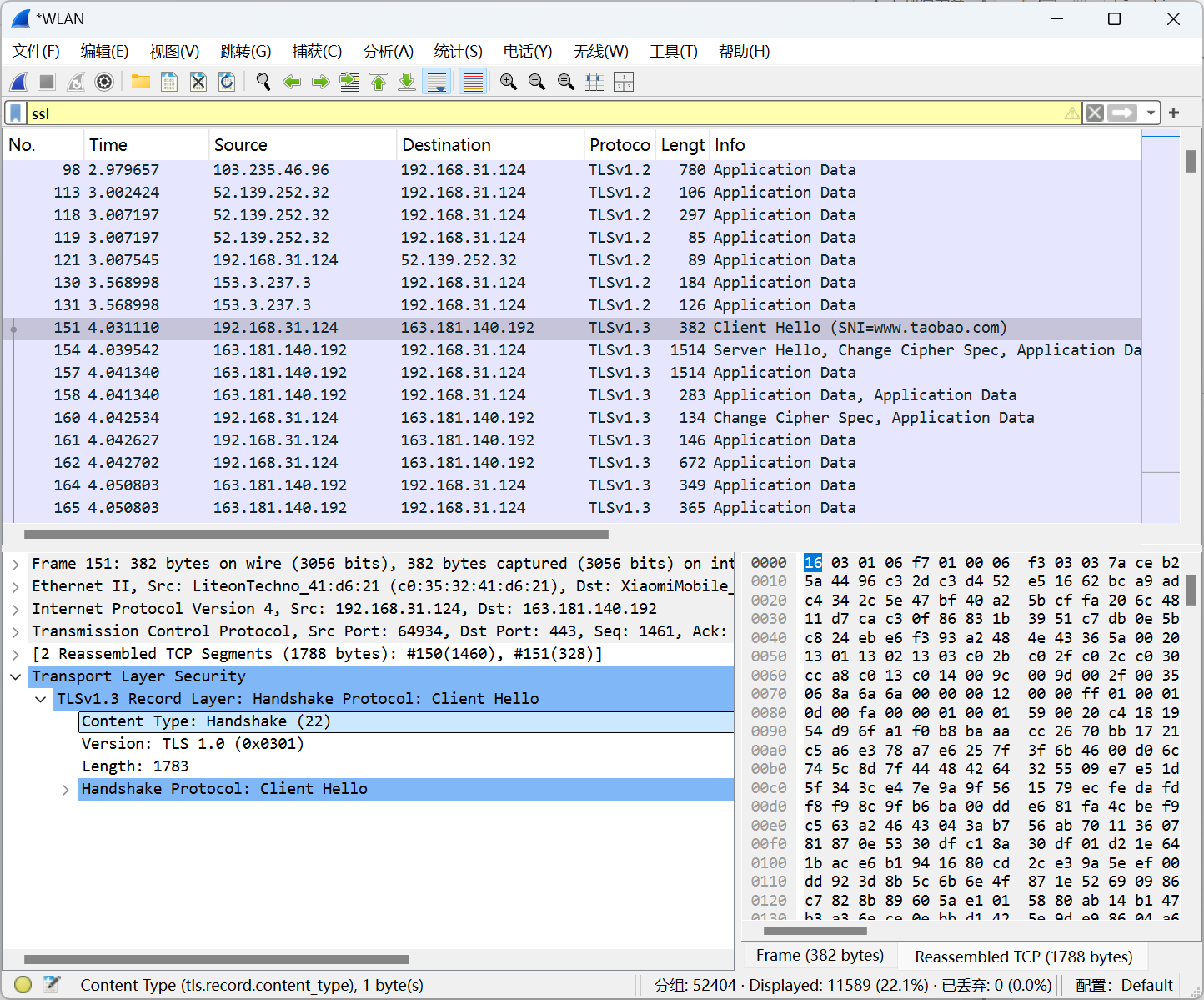
Version: 2 byte

Length: 2 byte



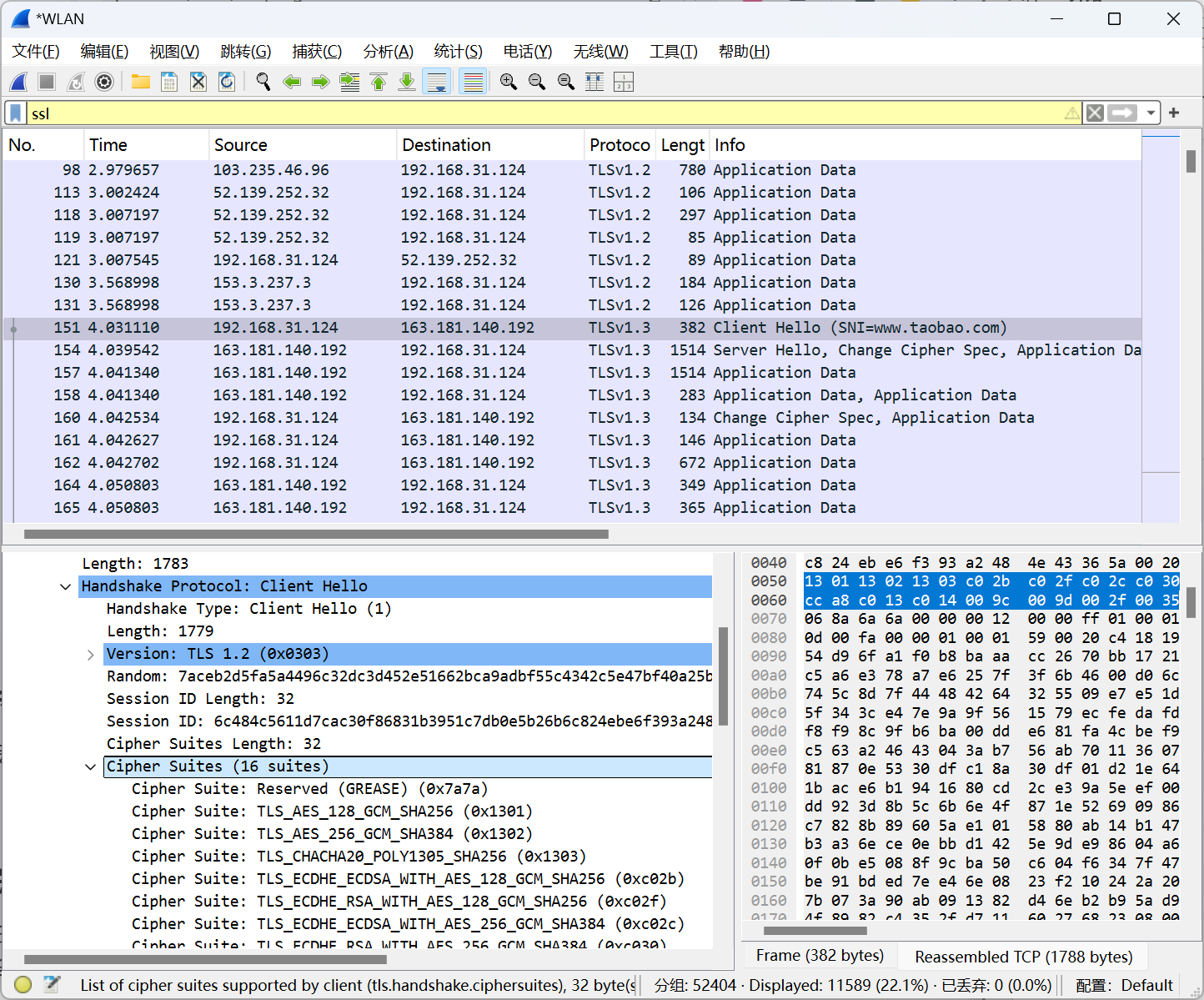
## Expand a ClientHello record. What is the value of the content type? What is it used for?

Handshake, It is used to initialize a secure connection in the SSL/TLS protocol and carries important information such as the protocol version, random number, encryption suite, etc. supported by the client.



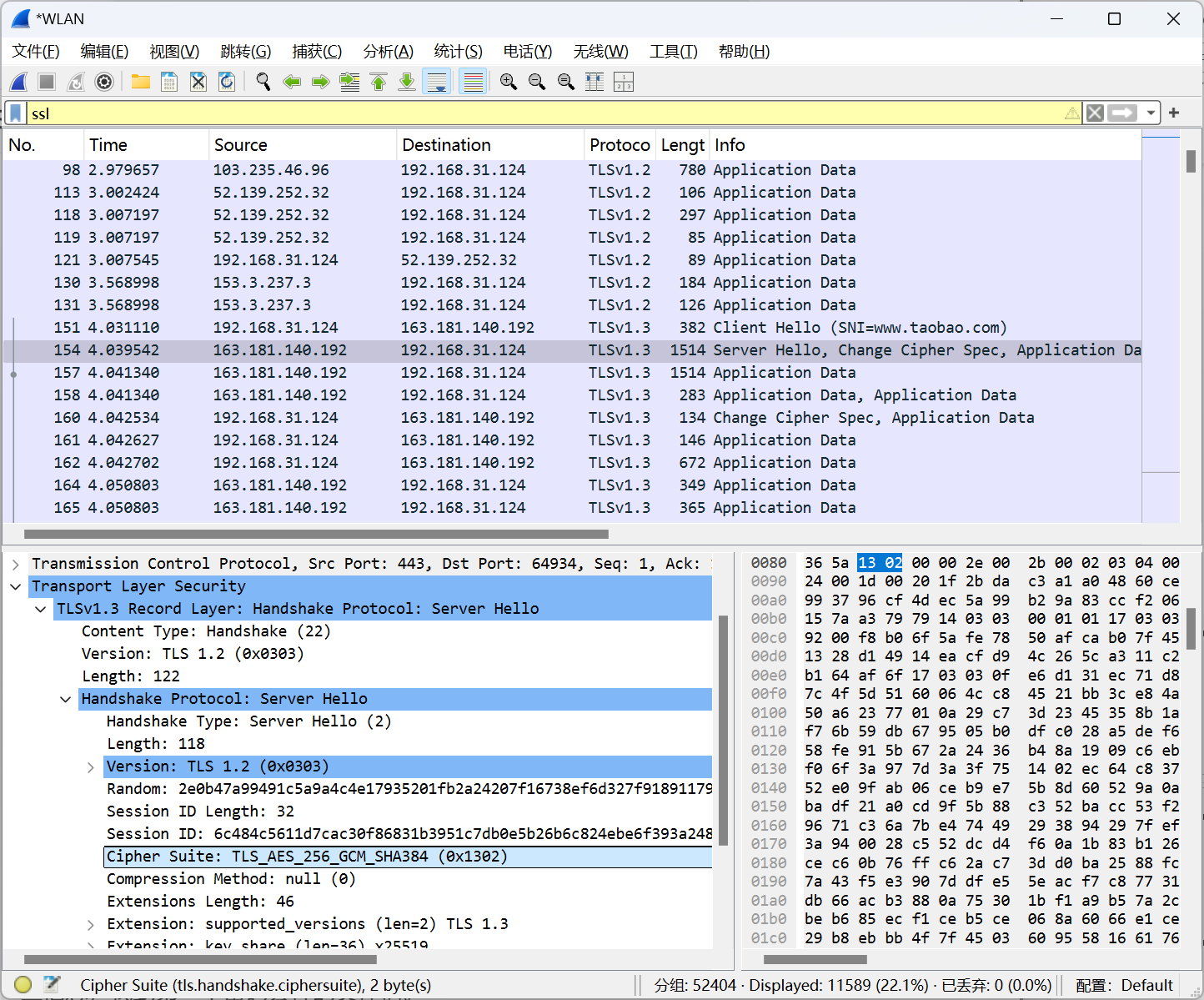
## Does the ClientHello record advertise the cipher suites it supports? Show the first cipher suite.

Yes.



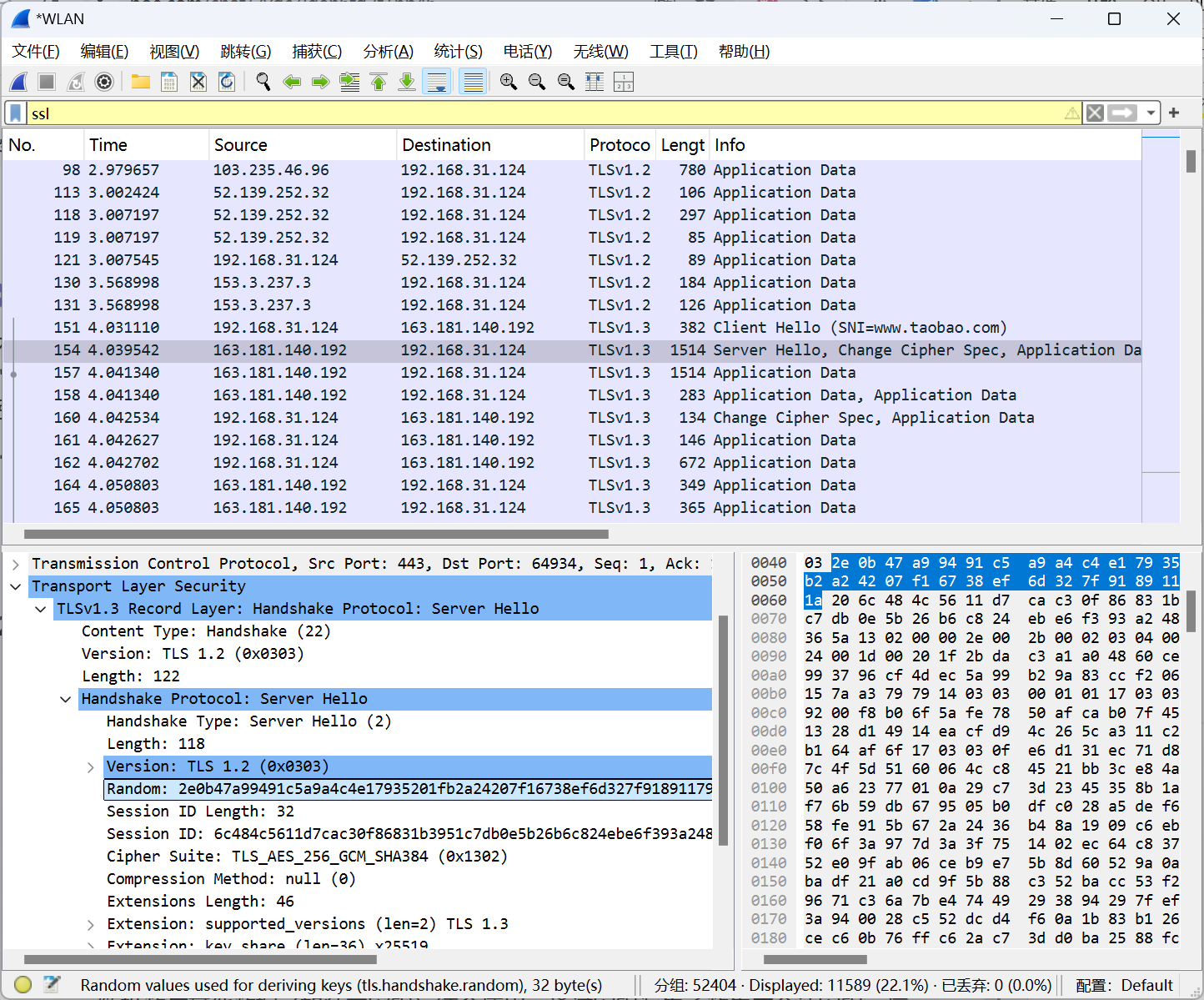
## Find the corresponding ServerHello SSL record. Does this record specify a chosen cipher suite? Show the chosen cipher suite.

Yes.



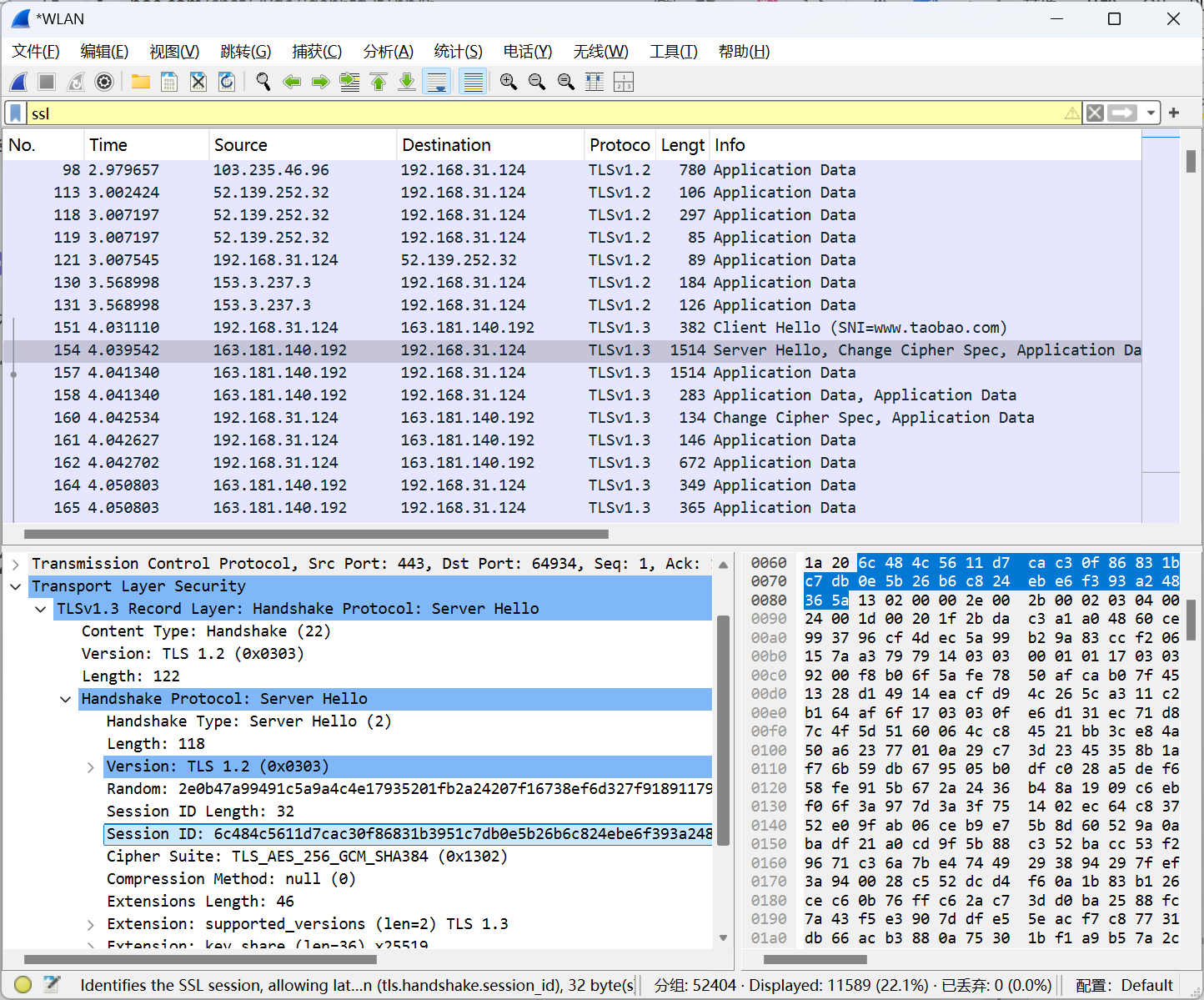
## Does the ServerHello SSL record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in SSL?

Yes. 32 bytes. It is used to enhance security, prevent replay attacks, and play an important role in the key generation process.



## Does the ServerHello SSL record include a session ID? What is the purpose of the session ID?

Yes. Its primary purpose is to support session resumption and improve performance, allowing clients and servers to quickly reestablish secure sessions on subsequent connections without having to redo a full handshake.

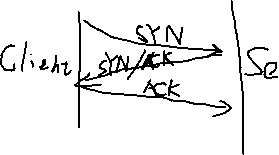


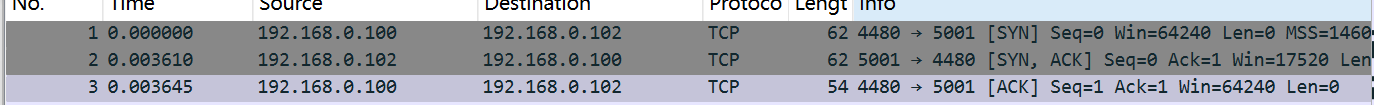
## Does the ServerHello SSL record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?

No. The certificate included in a separate record. Certificates can usually fit into a single Ethernet frame, especially if their size is within the typical range. However, for larger certificates, it may be necessary to split the data into multiple frames for transmission.

# Problem 4: Analyzing TCP Traffic

## Draw a diagram to illustrate the handshake between 192.168.0.100:4480 and 192.168.0.102:5001. List the length of the all packets used to complete TCP handshake.



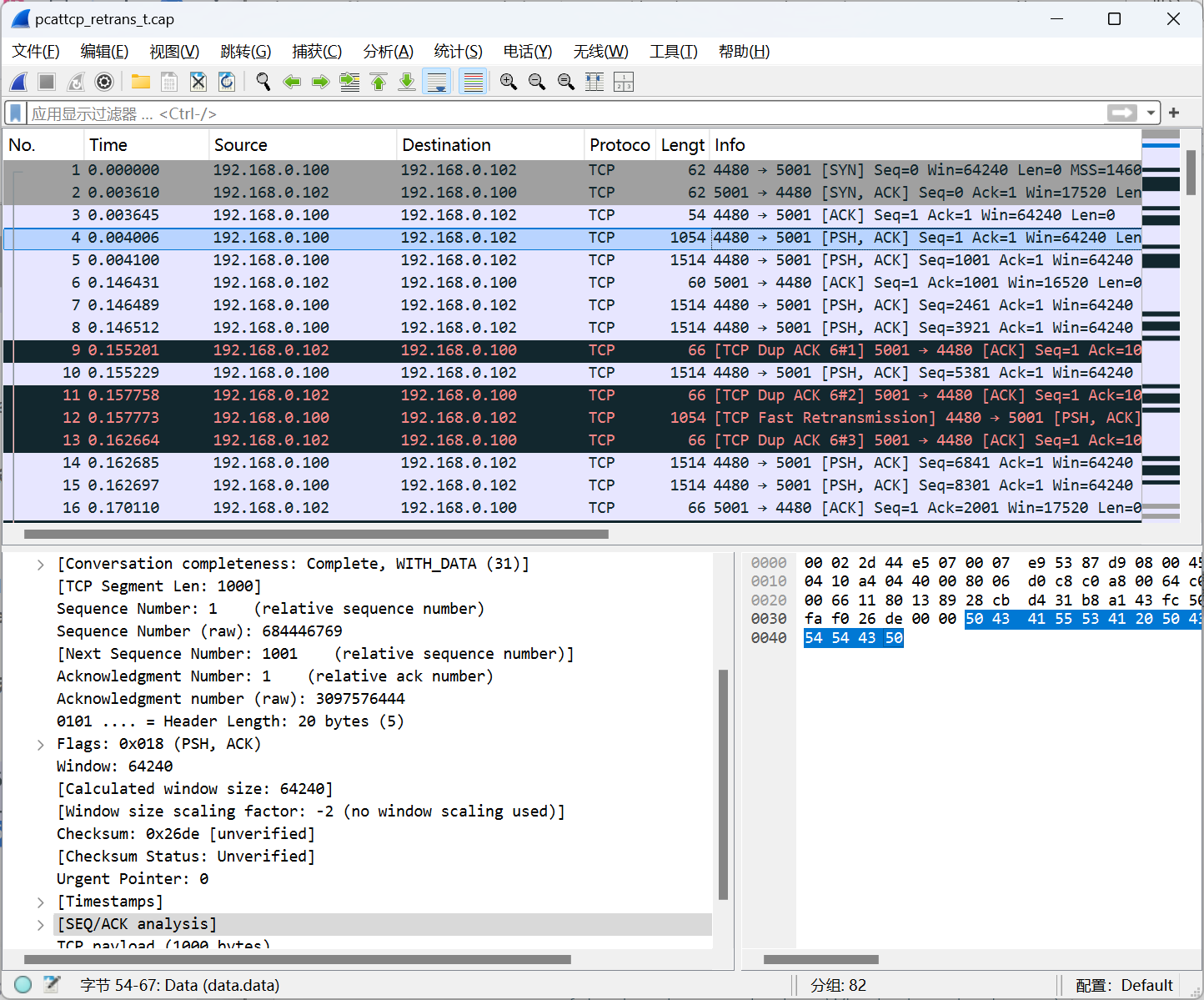


62 62 54

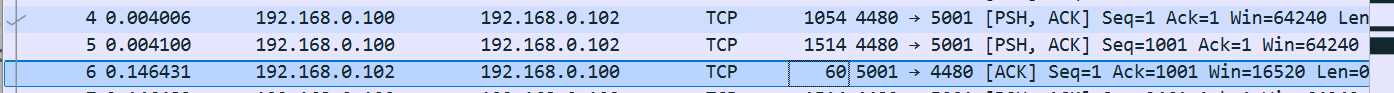
## Finding the first TCP segment sending from 192.168.0.100:4480 after the handshake of TCP. What’s the length of this TCP segment?

1054

[TCP Segment Len: 1000]



## Calculating the RTT value for the TCP segment you found in question 2. (Hint: the value of the Time column in the packet listing window is the amount of the time, in seconds, since Wireshark tracing began.)



RTT = 0.146431s - 0.004006s = 0.142425s