Wireshark 抓包实验-5 实验报告

2. 实验题目及指导

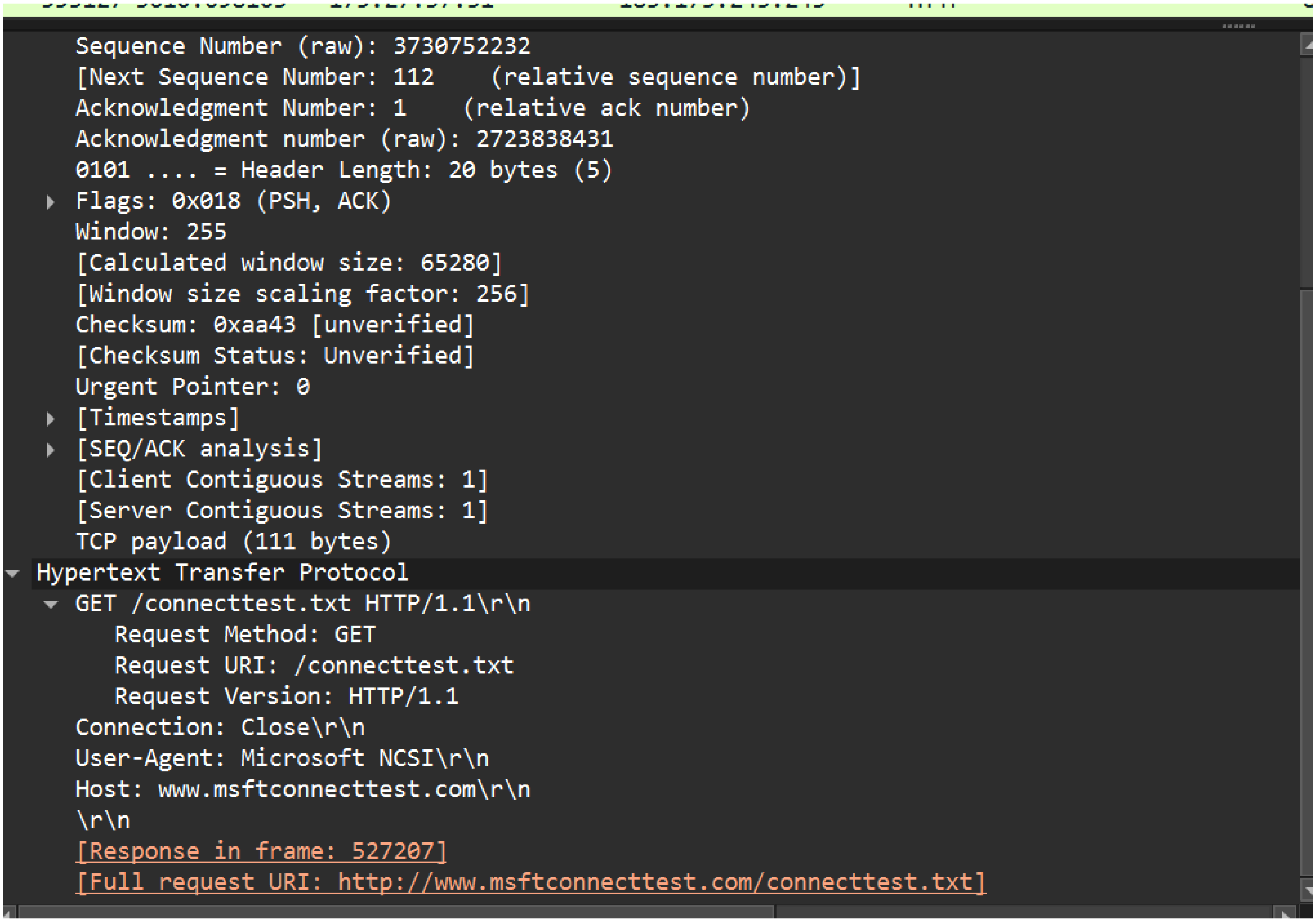
抓包实验 1：观察 HTTP 数据包

(1) HTTP 使用的传输层协议是什么?

HTTP 协议使用的传输层协议是 TCP。它通过 TCP 建立可靠的连接后进行数据交换。

(2) HTTP 请求包信息

请求方法：GET
Host：example.com
URL：/ （根目录）
HTTP 版本：HTTP/1.1

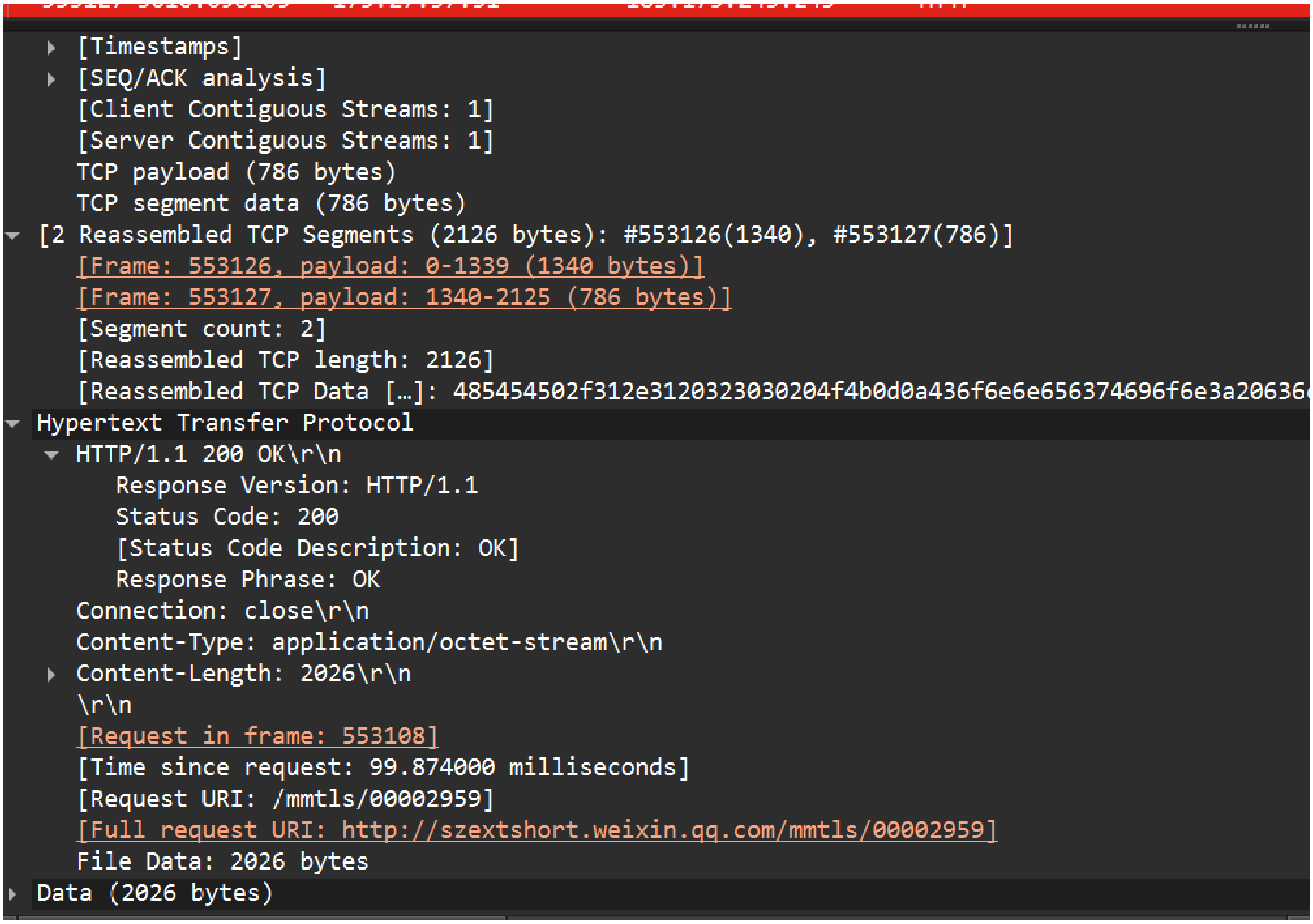


```
    Sequence Number (raw): 3730752232
    [Next Sequence Number: 112    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2723838431
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x018 (PSH, ACK)
    Window: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
    Checksum: 0xaa43 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▸ [Timestamps]
  ▸ [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (111 bytes)
▾ Hypertext Transfer Protocol
  ▾ GET /connecttest.txt HTTP/1.1\r\n
      Request Method: GET
      Request URI: /connecttest.txt
      Request Version: HTTP/1.1
    Connection: Close\r\n
    User-Agent: Microsoft NCSI\r\n
    Host: www.msftconnecttest.com\r\n
    \r\n
    [Response in frame: 527207]
    [Full request URI: http://www.msftconnecttest.com/connecttest.txt]
```

(3) HTTP 响应包信息

状态码：200 OK

Content-Type：text/html; charset=UTF-8

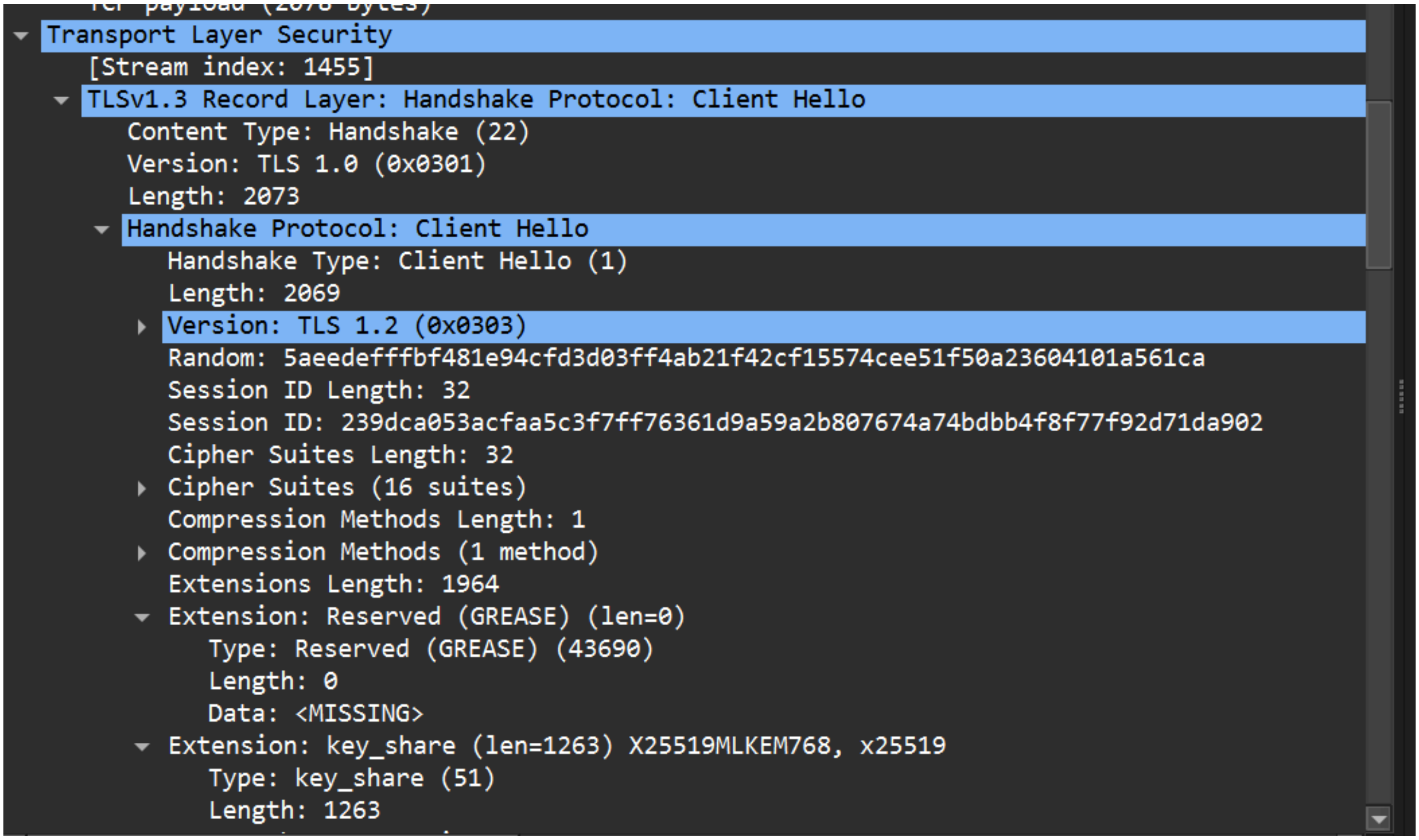


```
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (786 bytes)
    TCP segment data (786 bytes)
[2 Reassembled TCP Segments (2126 bytes): #553126(1340), #553127(786)]
    [Frame: 553126, payload: 0-1339 (1340 bytes)]
    [Frame: 553127, payload: 1340-2125 (786 bytes)]
    [Segment count: 2]
    [Reassembled TCP length: 2126]
    [Reassembled TCP Data […]: 485454502f312e3120323030204f4b0d0a436f6e6e656374696f6e3a20636
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
        Response Version: HTTP/1.1
        Status Code: 200
        [Status Code Description: OK]
        Response Phrase: OK
    Connection: close\r\n
    Content-Type: application/octet-stream\r\n
    Content-Length: 2026\r\n
    \r\n
    [Request in frame: 553108]
    [Time since request: 99.874000 milliseconds]
    [Request URI: /mmtls/00002959]
    [Full request URI: http://szextshort.weixin.qq.com/mmtls/00002959]
    File Data: 2026 bytes
Data (2026 bytes)
```

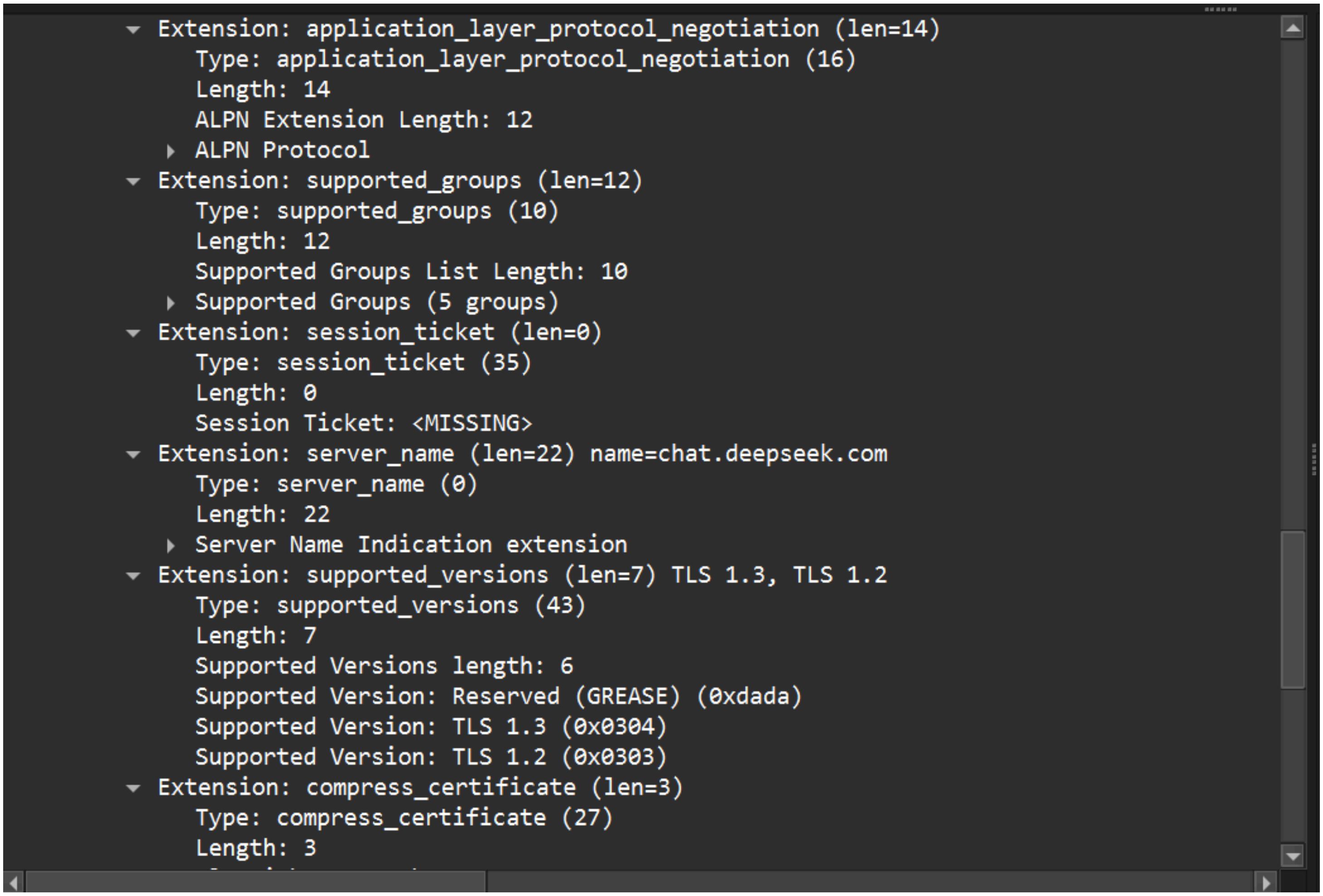
观察 TLS (HTTPS) 数据包

(4) TLS 版本及内容可见性

TLS 版本： TLS 1.3。
内容可见性：无法看到具体的 HTTP 请求或响应内容。
原因：由于 TLS 协议对应用层数据进行了加密传输，在 Wireshark 中只能看到加密后的 Application Data，确保了数据的私密性。



```
    TCP payload (2078 bytes)
Transport Layer Security
    [Stream index: 1455]
    TLSv1.3 Record Layer: Handshake Protocol: Client Hello
        Content Type: Handshake (22)
        Version: TLS 1.0 (0x0301)
        Length: 2073
        Handshake Protocol: Client Hello
            Handshake Type: Client Hello (1)
            Length: 2069
            Version: TLS 1.2 (0x0303)
            Random: 5aeedefffbf481e94cfd3d03ff4ab21f42cf15574cee51f50a23604101a561ca
            Session ID Length: 32
            Session ID: 239dca053acfaa5c3f7ff76361d9a59a2b807674a74bdbb4f8f77f92d71da902
            Cipher Suites Length: 32
            Cipher Suites (16 suites)
            Compression Methods Length: 1
            Compression Methods (1 method)
            Extensions Length: 1964
            Extension: Reserved (GREASE) (len=0)
                Type: Reserved (GREASE) (43690)
                Length: 0
                Data: <MISSING>
            Extension: key_share (len=1263) X25519MLKEM768, x25519
                Type: key_share (51)
                Length: 1263
```

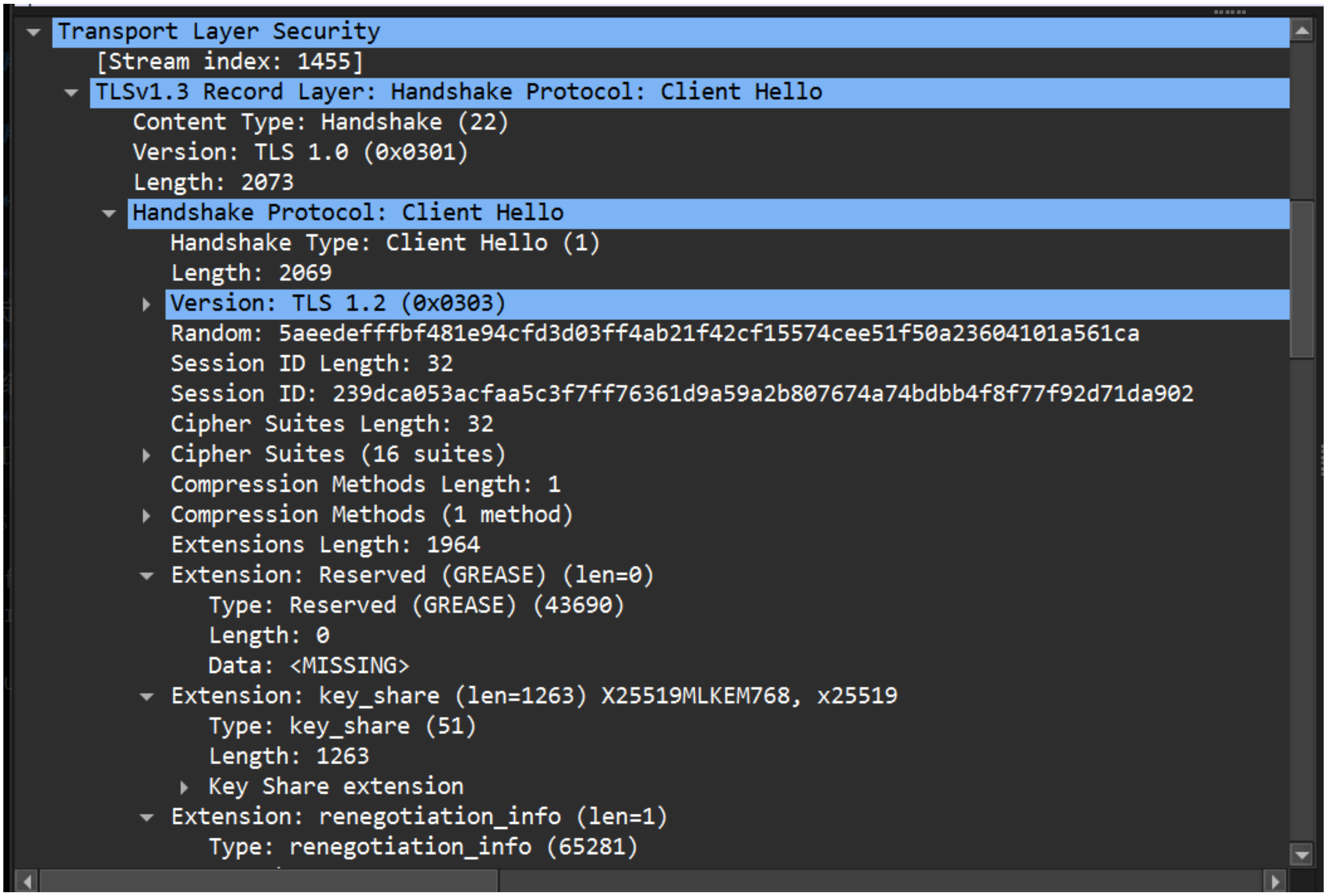


```
            Extension: application_layer_protocol_negotiation (len=14)
                Type: application_layer_protocol_negotiation (16)
                Length: 14
                ALPN Extension Length: 12
                ALPN Protocol
            Extension: supported_groups (len=12)
                Type: supported_groups (10)
                Length: 12
                Supported Groups List Length: 10
                Supported Groups (5 groups)
            Extension: session_ticket (len=0)
                Type: session_ticket (35)
                Length: 0
                Session Ticket: <MISSING>
            Extension: server_name (len=22) name=chat.deepseek.com
                Type: server_name (0)
                Length: 22
                Server Name Indication extension
            Extension: supported_versions (len=7) TLS 1.3, TLS 1.2
                Type: supported_versions (43)
                Length: 7
                Supported Versions length: 6
                Supported Version: Reserved (GREASE) (0xdada)
                Supported Version: TLS 1.3 (0x0304)
                Supported Version: TLS 1.2 (0x0303)
            Extension: compress_certificate (len=3)
                Type: compress_certificate (27)
                Length: 3
```

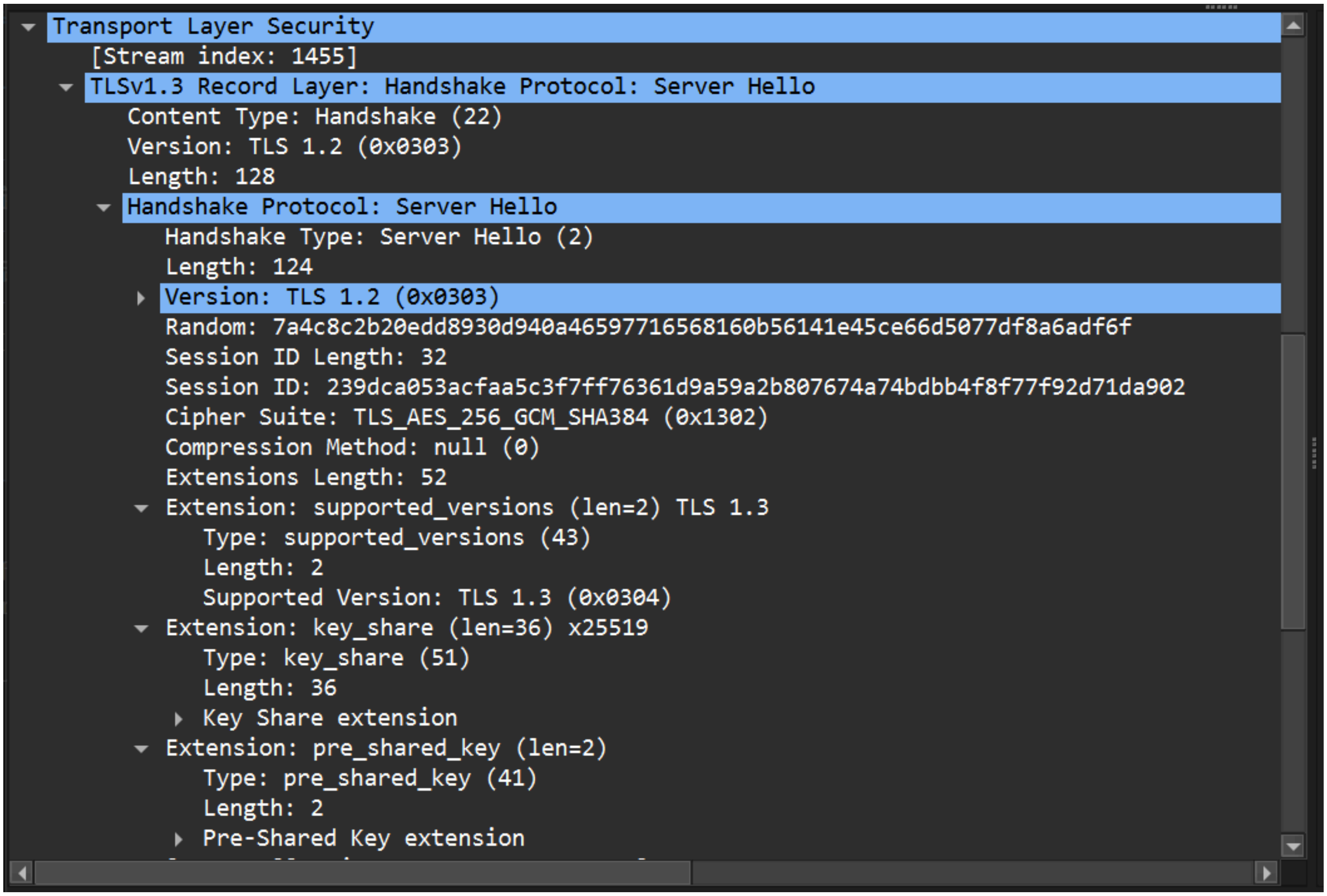
(5) HTTPS 整体交互流程简述

HTTPS 流程主要包括：

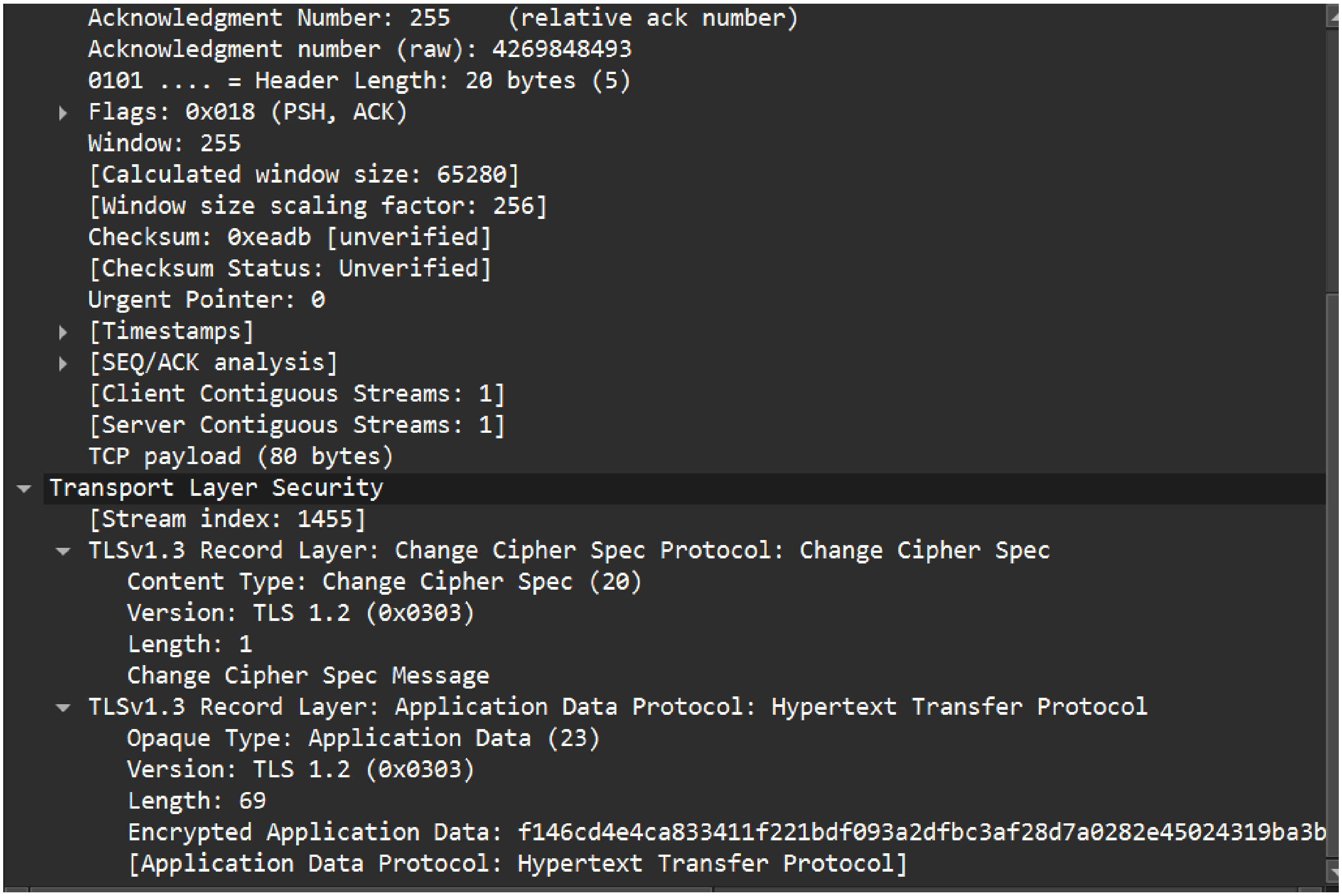
1. TCP 三次握手建立基础连接。
2. TLS 握手：客户端发送 `Client Hello`，服务器回复 `Server Hello`、证书及密钥交换信息。
3. 加密传输：双方协商对称加密密钥后，开始传输加密的 `Application Data`。



```
Transport Layer Security
    [Stream index: 1455]
  TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 2073
    Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 2069
        Version: TLS 1.2 (0x0303)
        Random: 5aeedefffbf481e94cfd3d03ff4ab21f42cf15574cee51f50a23604101a561ca
        Session ID Length: 32
        Session ID: 239dca053acfaa5c3f7ff76361d9a59a2b807674a74bdbb4f8f77f92d71da902
        Cipher Suites Length: 32
        Cipher Suites (16 suites)
        Compression Methods Length: 1
        Compression Methods (1 method)
        Extensions Length: 1964
        Extension: Reserved (GREASE) (len=0)
            Type: Reserved (GREASE) (43690)
            Length: 0
            Data: <MISSING>
        Extension: key_share (len=1263) X25519MLKEM768, x25519
            Type: key_share (51)
            Length: 1263
            Key Share extension
        Extension: renegotiation_info (len=1)
            Type: renegotiation_info (65281)
```



```
Transport Layer Security
    [Stream index: 1455]
  TLSv1.3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 128
    Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 124
        Version: TLS 1.2 (0x0303)
        Random: 7a4c8c2b20edd8930d940a46597716568160b56141e45ce66d5077df8a6adf6f
        Session ID Length: 32
        Session ID: 239dca053acfaa5c3f7ff76361d9a59a2b807674a74bdbb4f8f77f92d71da902
        Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
        Compression Method: null (0)
        Extensions Length: 52
        Extension: supported_versions (len=2) TLS 1.3
            Type: supported_versions (43)
            Length: 2
            Supported Version: TLS 1.3 (0x0304)
        Extension: key_share (len=36) x25519
            Type: key_share (51)
            Length: 36
            Key Share extension
        Extension: pre_shared_key (len=2)
            Type: pre_shared_key (41)
            Length: 2
            Pre-Shared Key extension
```



```
    Acknowledgment Number: 255    (relative ack number)
    Acknowledgment number (raw): 4269848493
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
    Checksum: 0xeadb [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (80 bytes)
Transport Layer Security
    [Stream index: 1455]
    TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
        Content Type: Change Cipher Spec (20)
        Version: TLS 1.2 (0x0303)
        Length: 1
        Change Cipher Spec Message
    TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
        Opaque Type: Application Data (23)
        Version: TLS 1.2 (0x0303)
        Length: 69
        Encrypted Application Data: f146cd4e4ca833411f221bdf093a2dfbc3af28d7a0282e45024319ba3b
        [Application Data Protocol: Hypertext Transfer Protocol]
```

3. 简述题

(1) 分析 HTTP 头部与 IP/TCP 头的设计思路差异

表现形式：IP/TCP 头部采用二进制定长/偏移设计，字段位置固定（如协议号始终在 IP 头的固定偏移处），旨在提高硬件处理和转发效率。
文本化 vs 二进制：HTTP 头部采用 ASCII 文本形式（Key-Value 结构），每行以回车换行符结束。
扩展性：HTTP 设计思路侧重于灵活性与可读性，允许通过自定义 Header 轻松扩展功能；而 IP/TCP 头部设计更侧重于传输效率与低开销。