HW4. Using Wireshark

Description

Steps:

- Download & Install Wireshark
- Capture all packages when you access http://www.zju.edu.cn (using filter)
- Using Wireshark to analysis the packets
- Write a report (in pdf) to describe the procedure & results of the capture & analysis process

一、实验目的

1. 学习试用网络数据抓包软件 Wireshark，对互联网进行数据抓包，巩固对所学知识的理解。

二、实验内容

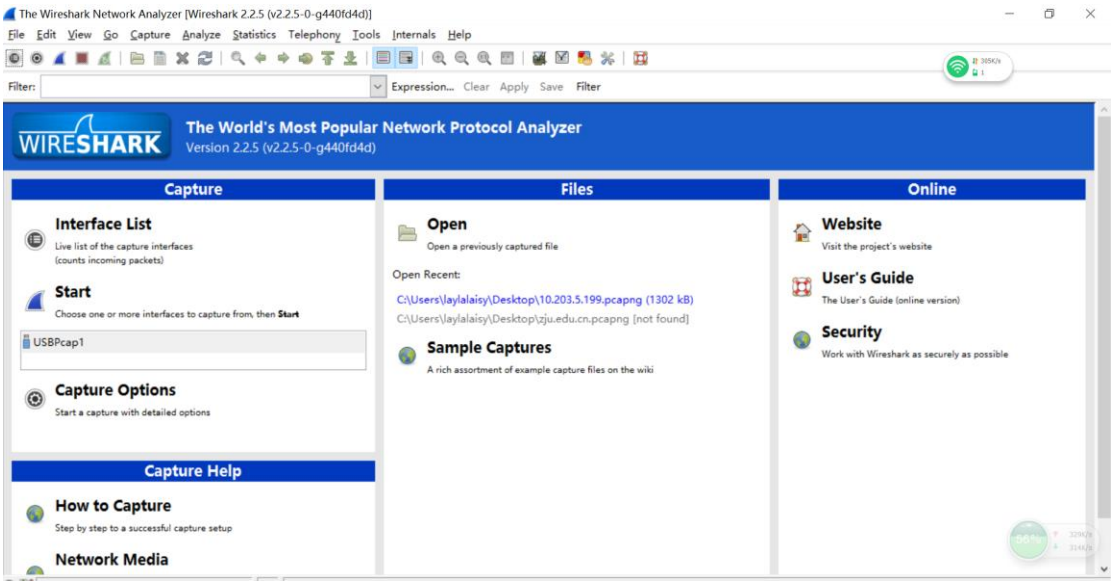
1. 分析 http 协议请求的响应过程；
2. 分析 TCP 的处理过程。

三、实验工具

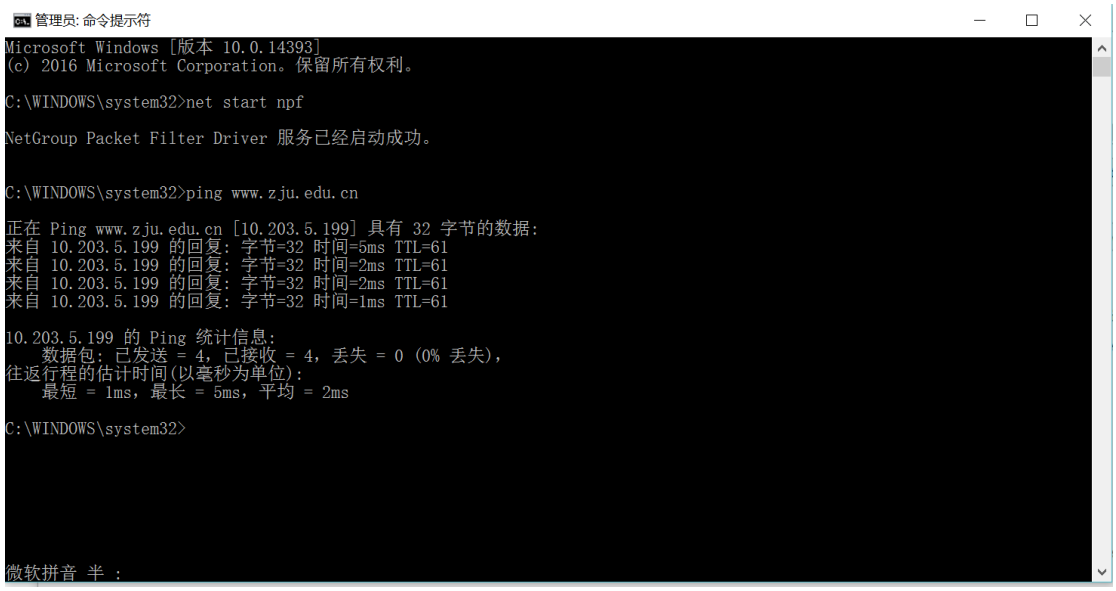
1. Wireshark 抓包工具。

四、实验步骤

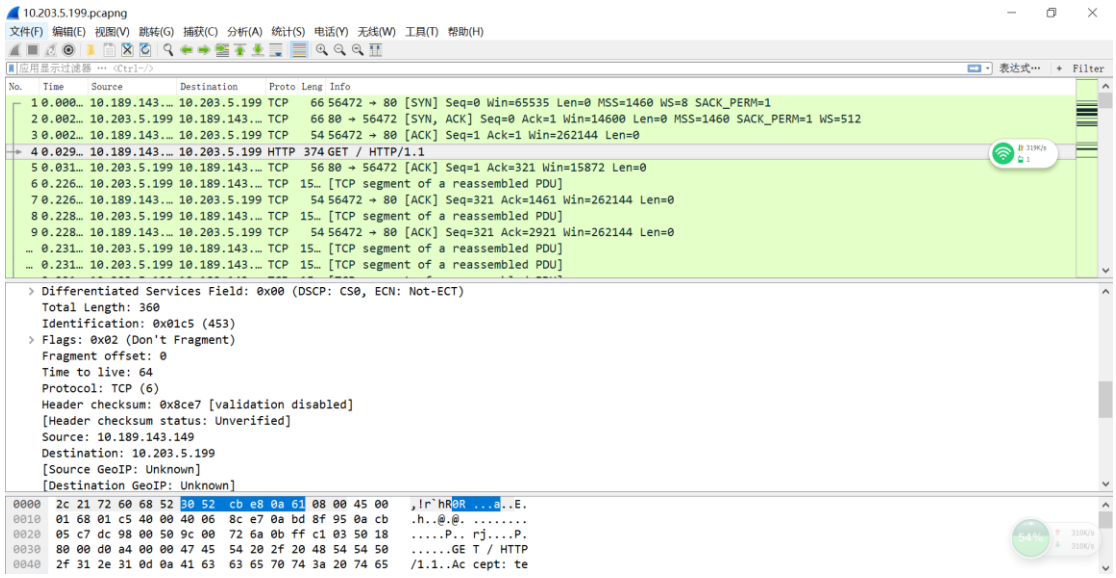
1. 安装 Wireshark 抓包工具；



2. 开始抓包前先重启网卡服务，在 cmd 中输入 net start npf；
3. 在 cmd 中输入 C:\WINDOWS\system32>ping www.zju.edu.cn，获得需要进行抓包的网站的服务器 10.203.5.19



4. 打开 Wireshark，单击"Capture"， 配置"option"选项，输入 **host** 10.203.5.199；
5. 设置完成，点集"start"开始抓包，显示结果；
6. 导出抓包结果并进行分析。



五、数据分析

1. http://www.zju.edu.cn/ HTTP 是基于 TCP 的连接，因此建立 HTTP 连接必须经过 TCP 的过程。

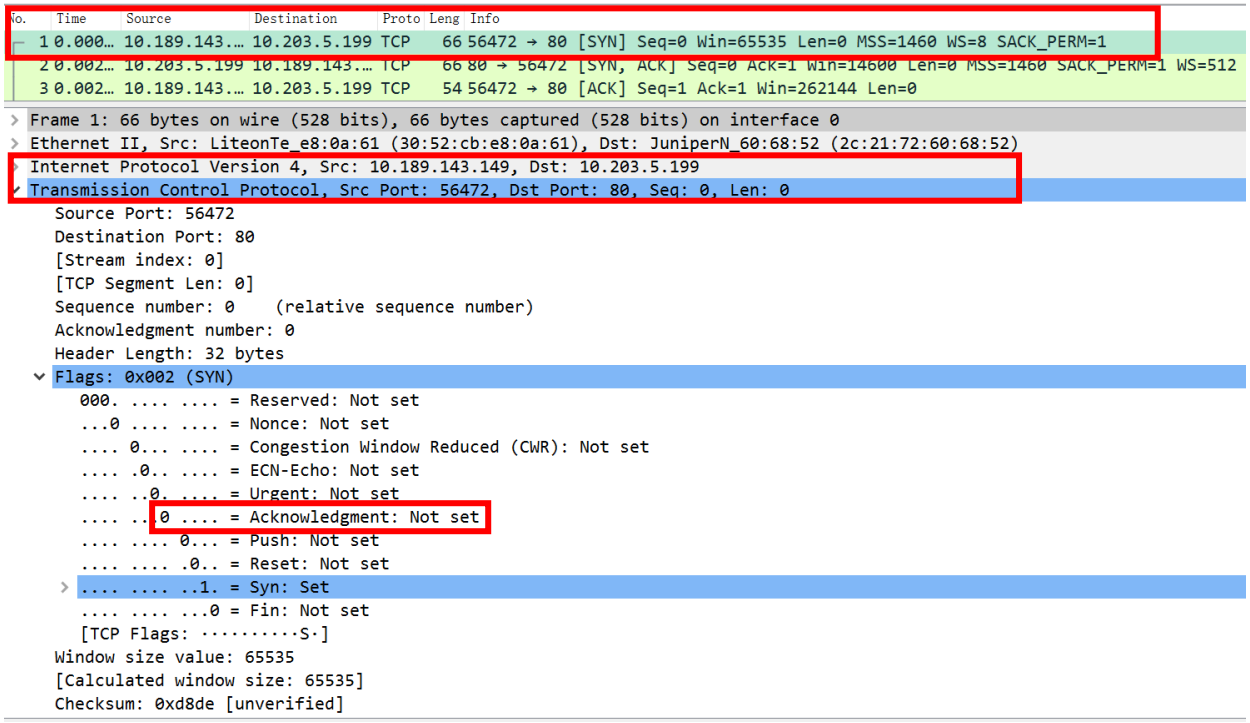
（1）TCP 的建立过程是 3 次握手的过程；

（2）HTTP 有请求和应答报文两种报文；

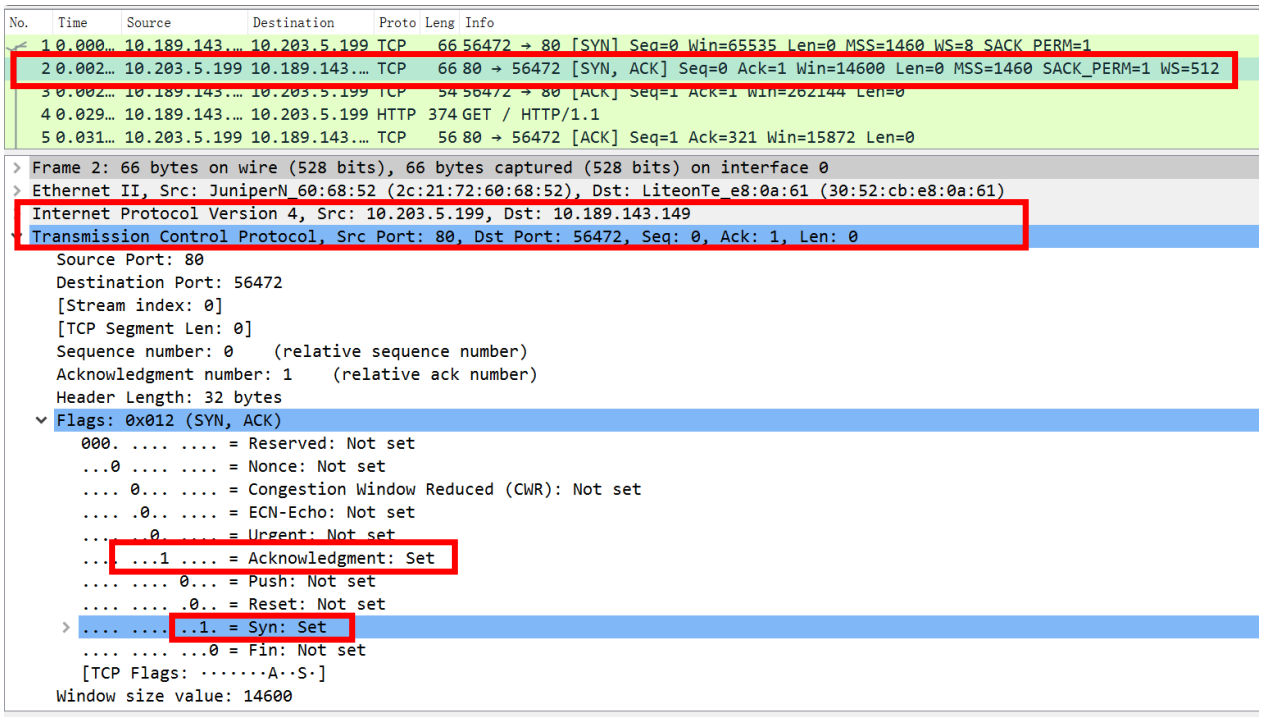
（3）完成 HTTP 过程后，三次断开 TCP 连接。

2. TCP 的建立

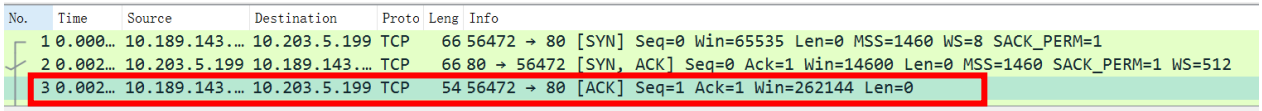
（1）TCP 第一阶段：客户向服务器发送同步请求,将获得包括 Src，Dst，Port 等信息，并将其中 Flags 字段的 Syn 位置为 Set。



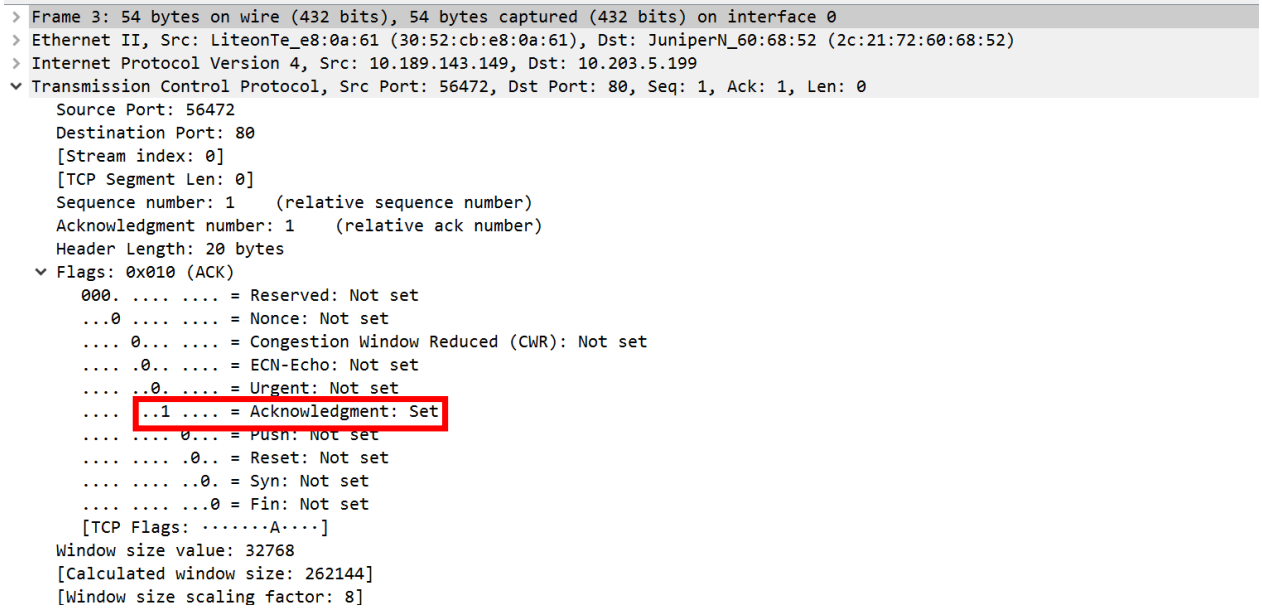
(2) TCP 第二阶段：服务器向客户回复一个 ACK 包，其中 Flag 字段的 Syn 和 Acknowledgment 字段置为 Set.



(3) TCP 第三阶段：客户向服务器发送一个 ACK 包。



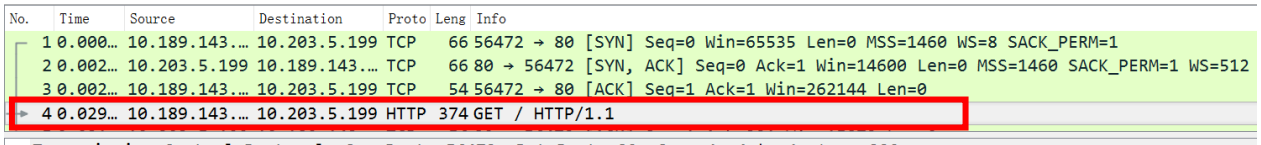
```
No.  Time     Source          Destination     Proto Leng Info
   1 0.000… 10.189.143.… 10.203.5.199 TCP    66 56472 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 SACK_PERM=1
   2 0.002… 10.203.5.199 10.189.143.… TCP    66 80 → 56472 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=512
   3 0.002… 10.189.143.… 10.203.5.199 TCP    54 56472 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
```



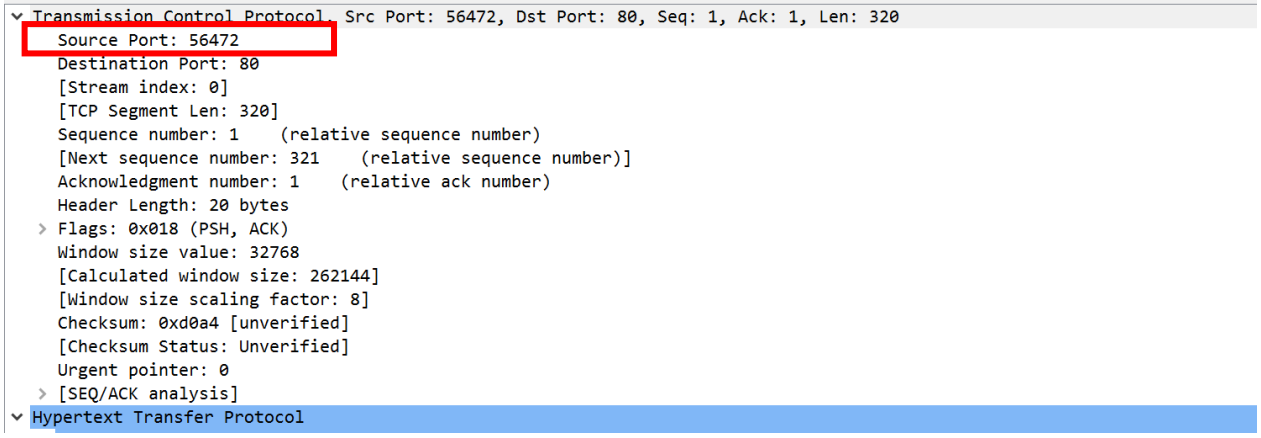
```
> Frame 3: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: LiteonTe_e8:0a:61 (30:52:cb:e8:0a:61), Dst: JuniperN_60:68:52 (2c:21:72:60:68:52)
> Internet Protocol Version 4, Src: 10.189.143.149, Dst: 10.203.5.199
v Transmission Control Protocol, Src Port: 56472, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 56472
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  v Flags: 0x010 (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
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····A····]
    Window size value: 32768
    [Calculated window size: 262144]
    [Window size scaling factor: 8]
```

3. HTTP 请求

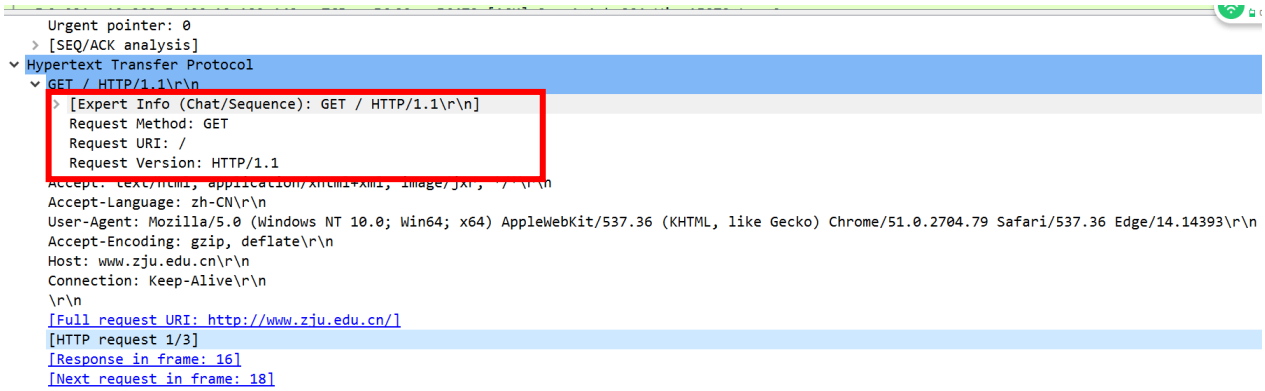
(1)客户发出 HTTP 请求之后，服务器收到请求发送 ACK



```
No.  Time     Source          Destination     Proto Leng Info
   1 0.000… 10.189.143.… 10.203.5.199 TCP    66 56472 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 SACK_PERM=1
   2 0.002… 10.203.5.199 10.189.143.… TCP    66 80 → 56472 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=512
   3 0.002… 10.189.143.… 10.203.5.199 TCP    54 56472 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
   4 0.029… 10.189.143.… 10.203.5.199 HTTP  374 GET / HTTP/1.1
```



```
v Transmission Control Protocol, Src Port: 56472, Dst Port: 80, Seq: 1, Ack: 1, Len: 320
    Source Port: 56472
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 320]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 321    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x018 (PSH, ACK)
    Window size value: 32768
    [Calculated window size: 262144]
    [Window size scaling factor: 8]
    Checksum: 0xd0a4 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
v Hypertext Transfer Protocol
```



```
    Urgent pointer: 0
  > [SEQ/ACK analysis]
v Hypertext Transfer Protocol
  v 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
    Accept: text/html, application/xhtml+xml, image/jxr, */*\r\n
    Accept-Language: zh-CN\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.79 Safari/537.36 Edge/14.14393\r\n
    Accept-Encoding: gzip, deflate\r\n
    Host: www.zju.edu.cn\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://www.zju.edu.cn/]
    [HTTP request 1/3]
    [Response in frame: 16]
    [Next request in frame: 18]
```

(2) 服务器发送应答报文

No.	Time	Source	Destination	Proto	Leng	Info
1	0.000…	10.189.143.…	10.203.5.199	TCP	66	56472 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 SACK_PERM=1
2	0.002…	10.203.5.199	10.189.143.…	TCP	66	80 → 56472 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=512
3	0.002…	10.189.143.…	10.203.5.199	TCP	54	56472 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
4	0.029…	10.189.143.…	10.203.5.199	HTTP	374	GET / HTTP/1.1
5	0.031…	10.203.5.199	10.189.143.…	TCP	56	80 → 56472 [ACK] Seq=1 Ack=321 Win=15872 Len=0
6	0.226…	10.203.5.199	10.189.143.…	TCP	15…	[TCP segment of a reassembled PDU]

```
> Frame 5: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface 0
> Ethernet II, Src: JuniperN_60:68:52 (2c:21:72:60:68:52), Dst: LiteonTe_e8:0a:61 (30:52:cb:e8:0a:61)
> Internet Protocol Version 4, Src: 10.203.5.199, Dst: 10.189.143.149
v Transmission Control Protocol, Src Port: 80, Dst Port: 56472, Seq: 1, Ack: 321, Len: 0
    Source Port: 80
    Destination Port: 56472
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    Acknowledgment number: 321    (relative ack number)
    Header Length: 20 bytes
  > Flags: 0x010 (ACK)
    Window size value: 31
    [Calculated window size: 15872]
    [Window size scaling factor: 512]
    Checksum: 0x4b3b [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
```