北京邮电大学软件学院

2017－2018 学年第一学期实验报告

课程名称： 计算机网络

项目名称： 数据链路层实验

项目完成人：

姓名： 黄莹 **学号：** 2017211953

指导教师： 王文东 雷友珣

日　　期： 2019 年　10 月　27 日

一、 实验目的

通过本实验使学生理解数据链路层协议数据单元（PDU）的定义和数据链路层功能。

二、 实验内容

1) 在可以访问互联网的主机上下载并安装网络抓包软件 Wireshark。
2) 运行 Wireshark 软件，启动 Wireshark 软件的抓包功能抓取本主机访问互联网中某网站过程中发送和接收的数据包。
3) 对所抓取的数据包进行分析，分析所发送和接收的数据包的以太网帧结构中的源 MAC 地址、目的 MAC 地址和类型（type）字段的使用方法；理解各字段的含义和功能。
4) 选做部分：分析所抓取的数据包中的 DNS（Domain Name System）消息、TCP 报文、IP 分组、HTTP 协议消息的字段组成及作用。

三、 实验环境

1) Windows 系统主机或 Linux 系统主机；
2) Wireshark 软件，软件下载网址：https://www.wireshark.org/

四、 实验过程及结果

1） 启动 Wireshark 程序。
2） 点击 Wireshark 程序主窗口的“Capture ”菜单项，选中该下拉菜单中的“Options”菜单项，通过出现的“Capture Options”窗口中的“Interface”选择框设置需要抓取哪个网卡发送/接收的数据包。
3） 可通过“Capture Options”窗口中的“Capture Filter”选择框设置需要抓取的数据包的类型，比如选择“IP only”。
4） 点击“Capture Options”窗口中的“start”按钮，启动抓包工作。此时可看见 Wireshark 程序出现了一个新的窗口：“Capturing”窗口。
5） 启动本计算机中的浏览器程序（IE 浏览器或 Firefox 浏览器），在浏览器的地址栏中输入所要访问的网站的网址（例如北京邮电大学的网址 http://www.bupt.edu.cn）后按回车键，可看到浏览器中出现该网站的主页。

打开网址 http://www.bupt.edu.cn 对传输的数据包分析

6） 在 Wireshark 程序的“Capturing”窗口中观察 Wireshark 程序抓取的数据包；该窗口中的每一条记录为本机发送或接收到的一个数据包；

观察从本机发往北邮官网网址的数据包，协议为 TCP,HTTP 等。

927	39.711775	10.128.235.166	123.151.72.63	TCP	66 13917 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2 SACK_PERM=1
928	39.719963	123.151.72.63	10.128.235.166	TCP	66 80 → 13917 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1440 SACK_PERM=1 WS=128
929	39.720155	10.128.235.166	123.151.72.63	TCP	54 13917 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
930	39.720442	10.128.235.166	123.151.72.63	HTTP	672 GET /gchatpic_new/B69B17D44BE29FB85549174D9FD8E64E6C2350B097ACA38B9A208FEE16AB1906D7D547CA29E47B8C8666572C6
931	39.727406	123.151.72.63	10.128.235.166	TCP	60 80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=0
932	39.728007	123.151.72.63	10.128.235.166	TCP	1494 80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=1440 [TCP segment of a reassembled PDU]
933	39.729220	123.151.72.63	10.128.235.166	HTTP	889 HTTP/1.1 200 OK (image/jpeg)
934	39.729291	10.128.235.166	123.151.72.63	TCP	54 13917 → 80 [ACK] Seq=619 Ack=2276 Win=65536 Len=0

7） 用 ipconfig 命令查看本机 ip 地址与 MAC 地址

本机 ip 地址如下：

```
IPv6 地址 . . . . . . . . . . . . : 2001:da8:215:3c01::aab6
IPv6 地址 . . . . . . . . . . . . : 2001:da8:215:3c01:f8fe:442c:f3d8:3993
临时 IPv6 地址. . . . . . . . . . : 2001:da8:215:3c01:4807:9f00:2a54:cb99
本地链接 IPv6 地址. . . . . . . . : fe80::f8fe:442c:f3d8:3993%19
IPv4 地址 . . . . . . . . . . . . : 10.128.235.166
```

8） 鼠标双击“Capturing”窗口中的一条记录，出现展示该数据包详细信息的窗口。在该窗口下面部分的子窗口中有该数据包（数据链路层帧）的二进制数据表示；在该窗口上面部分的子窗口中有 Wireshark 程序对该帧的分析，详细列出了该帧的字段组成以及各字段的取值。观察该帧的“Destination”字段、“Source”字段、“Type”字段的取值，以及该帧携带的数据。在实验报告中分析数据链路层帧的字段组成和作用。

本机发送的数据包

```
> Frame 927: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
> Ethernet II, Src: IntelCor_80:04:8f (58:fb:84:80:04:8f), Dst: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
> Internet Protocol Version 4, Src: 10.128.235.166, Dst: 123.151.72.63
> Transmission Control Protocol, Src Port: 13917, Dst Port: 80, Seq: 0, Len: 0
```

第一行：说明这是 927 号帧，线路 66 字节，实际捕获 66 字节。

第二行：包含了源 MAC 地址字段、目的 MAC 地址字段、类型字段的取值。

```
v Ethernet II, Src: IntelCor_80:04:8f (58:fb:84:80:04:8f), Dst: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
  v Destination: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
      Address: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: IntelCor_80:04:8f (58:fb:84:80:04:8f)
      Address: IntelCor_80:04:8f (58:fb:84:80:04:8f)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

```
0000  74 85 c4 11 20 01 58 fb  84 80 04 8f 08 00 45 00   t··· ·X· ·····E·
0010  00 34 01 98 40 00 80 06  3f 2f 0a 80 eb a6 7b 97   ·4··@··· ?/····{·
0020  48 3f 36 5d 00 50 b4 81  21 e4 00 00 00 00 80 02   H?6]·P·· !·······
0030  ff ff a8 07 00 00 02 04  05 b4 01 03 03 01 01 01   ···· ···· ····
0040  04 02                                             ··
```

数据链路层帧头的 14Bytes 由目的地址（6Byte)、源地址（6Byte）和类型（2Byte）组成，前六个 Bytes（74:85:c4:11:20:01)为目的地址，后六个 Bytes（58:fb:84:80:04:8f)为源地址，最后两个 Bytes(0x0800)表示 ipv4 协议。(0806 ARP 协议，08dd ipv6 协议）。

第三行：包含了互联网协议 IPv4，具体分析见后面 IP 分组字段组成分析；

第四行：包含了 TCP 文字报，具体分析见后面 TCP 报文字段组成分析。

9） 观察“Capturing”窗口显示的所抓取的数据包，分析哪些数据包是发送出去的数据包，哪些数据包是接收到的数据包。

本机 ip 地址为 10.128.235.166

927、929、930、934 均为本机**发送**的数据包；

928、931、932、933 均为本机**接收**的数据包；

926	39.711059	10.128.235.166	183.201.224.69	TCP	54	13880 → 80 [FIN, ACK] Seq=1 Ack=2 Win=32446 Len=0
927	39.711775	10.128.235.166	123.151.72.63	TCP	66	13917 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2 SACK_PERM=1
928	39.719963	123.151.72.63	10.128.235.166	TCP	66	80 → 13917 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1440 SACK_PERM=1 WS=128
929	39.720155	10.128.235.166	123.151.72.63	TCP	54	13917 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
930	39.720442	10.128.235.166	123.151.72.63	HTTP	672	GET /gchatpic_new/B69B17D44BE29FB85549174D9FD8E64E6C2350B097ACA38B9A208FEE16AB1906D7D547CA29E47B8C8666572C6
931	39.727406	123.151.72.63	10.128.235.166	TCP	60	80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=0
932	39.728007	123.151.72.63	10.128.235.166	TCP	1494	80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=1440 [TCP segment of a reassembled PDU]
933	39.729220	123.151.72.63	10.128.235.166	HTTP	889	HTTP/1.1 200 OK (image/jpeg)
934	39.729291	10.128.235.166	123.151.72.63	TCP	54	13917 → 80 [ACK] Seq=619 Ack=2276 Win=65536 Len=0

在抓取的数据中，source 为 123.151.72.63 的为接收到的数据包，destination 为 123.151.72.63 的为发送出去的数据包。

10） 分析访问互联网网站的协议过程，包括 DNS 域名解析过程、TCP 连接建立过程、HTTP 协议过程；分析 DNS 协议消息、IP 协议消息、TCP 协议消息、HTTP 协议消息格式。

1. DNS 域名解析过程

14	1.195727	10.128.235.166	10.3.9.5	DNS	73	Standard query 0x01b9 AAAA www.baidu.com
15	1.198720	10.3.9.5	10.128.235.166	DNS	157	Standard query response 0x01b9 AAAA www.baidu.com CNAME www.a.shifen.com SOA ns1.a.shifen.com
16	1.228228	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=78/19968, ttl=64 (reply in 17)
17	1.234469	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=78/19968, ttl=48 (request in 16)
18	2.242664	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=79/20224, ttl=64 (reply in 19)
19	2.249578	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=79/20224, ttl=48 (request in 18)
20	3.249964	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=80/20480, ttl=64 (reply in 21)
21	3.257290	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=80/20480, ttl=48 (request in 20)
23	4.257345	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=81/20736, ttl=64 (reply in 24)
24	4.264495	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=81/20736, ttl=48 (request in 23)

这是 ping www.baidu.com 抓取到的数据包

```
C:\Users\Administrator>ping www.baidu.com

正在 Ping www.a.shifen.com [39.156.66.14] 具有 32 字节的数据:
来自 39.156.66.14 的回复: 字节=32 时间=6ms TTL=48
来自 39.156.66.14 的回复: 字节=32 时间=7ms TTL=48
来自 39.156.66.14 的回复: 字节=32 时间=7ms TTL=48
来自 39.156.66.14 的回复: 字节=32 时间=7ms TTL=48

39.156.66.14 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 6ms, 最长 = 7ms, 平均 = 6ms
```

总体来看有两个 DNS 包(一次域名解析),和 8 个 ICMP 包(四次 ping)

DNS 域名解析的过程可分为下面六个过程：

1.先向本地 dns 缓存查看有没有该域名对应的 ip 地址，有直接跳出，没有接着往下

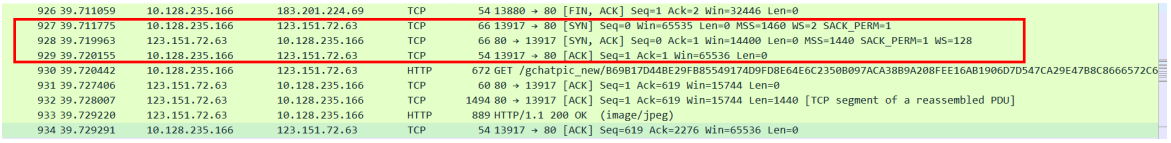
2.向根 dns 服务器询问域名对应的 ip 地址(根 dns 服务器会让他去查询顶级 dns 服务器)

3.向顶级 dns 服务器询问域名对应的 ip 地址(顶级 dns 服务器会让他去查询权威 dns 服务器)

4.向权威 dns 服务器询问域名对应的 ip 地址(权威 dns 服务器会让他去查询二级 dns 服务器)
5.向二级 dns 服务器询问域名对应的 ip 地址(二级 dns 服务器会返回对应的 ip 地址)
6.接收到 ip 地址后，会先把 ip 和域名对应关系保存到本地 dns 缓存，以便下次方便访问

2. TCP 建立连接过程（三次握手）

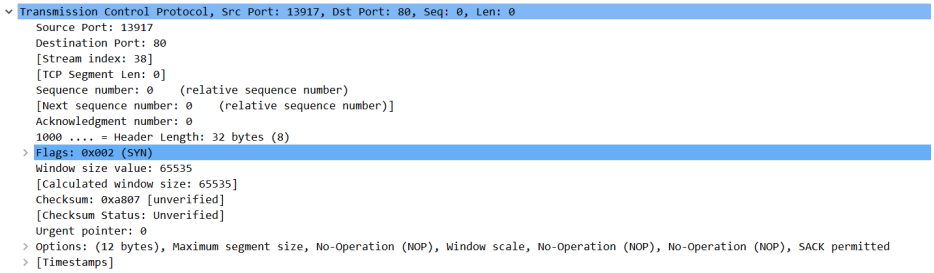
如图红框部分为 TCP 建立连接的过程。



926	39.711059	10.128.235.166	183.201.224.69	TCP	54	13880 → 80 [FIN, ACK] Seq=1 Ack=2 Win=32446 Len=0
927	39.711775	10.128.235.166	123.151.72.63	TCP	66	13917 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2 SACK_PERM=1
928	39.719963	123.151.72.63	10.128.235.166	TCP	66	80 → 13917 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1440 SACK_PERM=1 WS=128
929	39.720155	10.128.235.166	123.151.72.63	TCP	54	13917 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
930	39.720442	10.128.235.166	123.151.72.63	HTTP	672	GET /gchatpic_new/B69B17D44BE29FB85549174D9FD8E64E6C2350B097ACA38B9A208FEE16AB1906D7D547CA29E47B8C8666572C6
931	39.727406	123.151.72.63	10.128.235.166	TCP	60	80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=0
932	39.728007	123.151.72.63	10.128.235.166	TCP	1494	80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=1440 [TCP segment of a reassembled PDU]
933	39.729220	123.151.72.63	10.128.235.166	HTTP	889	HTTP/1.1 200 OK (image/jpeg)
934	39.729291	10.128.235.166	123.151.72.63	TCP	54	13917 → 80 [ACK] Seq=619 Ack=2276 Win=65536 Len=0

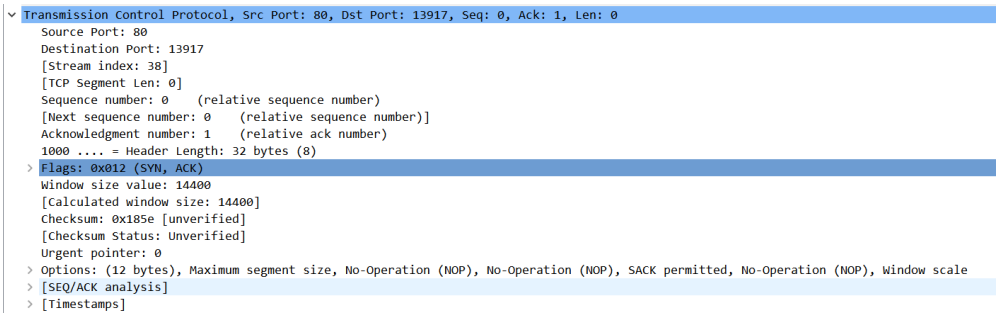
HTTP 开始之前先**三次握手**：

第一阶段就是客户向服务器发送同步请求(第 927 行)，Seq=0，标志位 SYN；



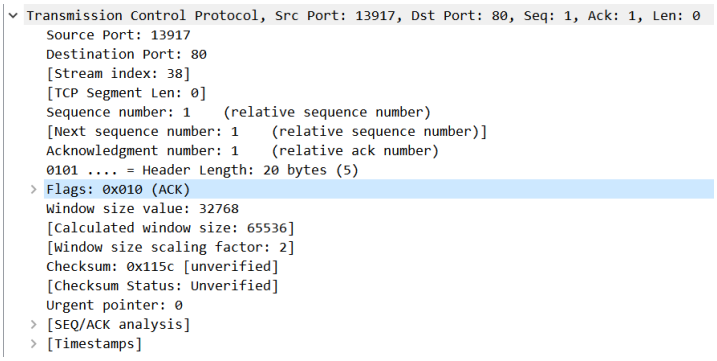
```
Transmission Control Protocol, Src Port: 13917, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 13917
    Destination Port: 80
    [Stream index: 38]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 65535
    [Calculated window size: 65535]
    Checksum: 0xa807 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

第二阶段就是服务器向客户回复一个 ACK 包（第 928 行），ACK=X+1，X=0，所以 ACK=1，seq=Y=0；标志位为 SYN，ACK;



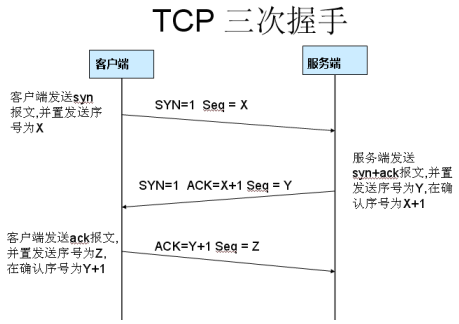
```
Transmission Control Protocol, Src Port: 80, Dst Port: 13917, Seq: 0, Ack: 1, Len: 0
    Source Port: 80
    Destination Port: 13917
    [Stream index: 38]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x012 (SYN, ACK)
    Window size value: 14400
    [Calculated window size: 14400]
    Checksum: 0x185e [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Operation (NOP), SACK permitted, No-Operation (NOP), Window scale
  > [SEQ/ACK analysis]
  > [Timestamps]
```

第三阶段是客户向服务器发送 ACK（第 929 行），其中 Seq=1，ACK=1，标志位为 ACK。



```
Transmission Control Protocol, Src Port: 13917, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
    Source Port: 13917
    Destination Port: 80
    [Stream index: 38]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window size value: 32768
    [Calculated window size: 65536]
    [Window size scaling factor: 2]
    Checksum: 0x115c [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > [SEQ/ACK analysis]
  > [Timestamps]
```

至此，TCP 的三次握手结束。



SYN 报文：图中显示从客户端发送至服务器端的 SYN 报文，此报文用于与服务器建立同步，确保客户端和服务器端的通信按次序传输。SYN 报文的头部有一个 32 bit 序列号。底端对话框显示了报文一些有用信息如报文类型，序列号。

SYN/ACK 报文：服务器的响应。一旦服务器接收到客户端的 SYN 报文，就读取报文的序列号并且使用此编号作为响应，也就是说它告知客户机，服务器接收到了 SYN 报文，通过对原 SYN 报文序列号加一并且作为响应编号来实现，之后客户端就知道服务器能够接收通信。

ACK 报文：客户端对服务器发送的确认报文，告诉服务器客户端接收到了 SYN/ACK 报文，并且与前一步一样客户端也将序列号加一，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

3. HTTP 协议过程:

926	39.711059	10.128.235.166	183.201.224.69	TCP	54	13880 → 80 [FIN, ACK] Seq=1 Ack=2 Win=32446 Len=0
927	39.711775	10.128.235.166	123.151.72.63	TCP	66	13917 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=2 SACK_PERM=1
928	39.719963	123.151.72.63	10.128.235.166	TCP	66	80 → 13917 [SYN, ACK] Seq=0 Ack=1 Win=14400 Len=0 MSS=1440 SACK_PERM=1 WS=128
929	39.720155	10.128.235.166	123.151.72.63	TCP	54	13917 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
930	39.720442	10.128.235.166	123.151.72.63	HTTP	672	GET /gchatpic_new/B69B17D44BE29FB85549174D9FD8E64E6C2350B097ACA38B9A208FEE16AB1906D7D547CA29E47B8C8666572C6
931	39.727406	123.151.72.63	10.128.235.166	TCP	60	80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=0
932	39.728007	123.151.72.63	10.128.235.166	TCP	1494	80 → 13917 [ACK] Seq=1 Ack=619 Win=15744 Len=1440 [TCP segment of a reassembled PDU]
933	39.729220	123.151.72.63	10.128.235.166	HTTP	889	HTTP/1.1 200 OK (image/jpeg)
934	39.729291	10.128.235.166	123.151.72.63	TCP	54	13917 → 80 [ACK] Seq=619 Ack=2276 Win=65536 Len=0

TCP 三次握手建立连接（927-929 行）；
TCP 三次握手结束之后就是 HTTP 请求(第 930 行)；
客户发出 HTTP 请求之后，服务器收到请求发送 ACK（第 931 行）；
服务器发送报文（第 933 行）；
客户收到保文后发送应答报文（第 934 行）

4. DNS 协议消息分析

10	3.044738	10.128.235.166	10.3.9.5	DNS	73	Standard query 0xc8a6 AAAA www.baidu.com
11	3.046766	10.3.9.5	10.128.235.166	DNS	157	Standard query response 0xc8a6 AAAA www.baidu.com CNAME www.a.shifen.com SOA ns1.a.shifen.com
12	3.070627	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=42/10752, ttl=64 (reply in 13)
13	3.076659	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=42/10752, ttl=48 (request in 12)
15	4.095120	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=43/11008, ttl=64 (reply in 16)
16	4.101986	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=43/11008, ttl=48 (request in 15)
25	5.109340	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=44/11264, ttl=64 (reply in 26)
26	5.116016	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=44/11264, ttl=48 (request in 25)
29	6.129018	10.128.235.166	39.156.66.14	ICMP	74	Echo (ping) request id=0x0001, seq=45/11520, ttl=64 (reply in 30)
30	6.135720	39.156.66.14	10.128.235.166	ICMP	74	Echo (ping) reply id=0x0001, seq=45/11520, ttl=48 (request in 29)

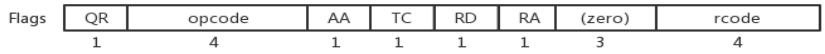
打开第一个 DNS，如下图所示：

```
> Frame 14: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
> Ethernet II, Src: IntelCor_80:04:8f (58:fb:84:80:04:8f), Dst: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
> Internet Protocol Version 4, Src: 10.128.235.166, Dst: 10.3.9.5
> User Datagram Protocol, Src Port: 58064, Dst Port: 53
> Domain Name System (query)
```

DNS 为应用层，下层传输层采用 UDP，再下层网络层为 IPv4，然后是数据链路层以太网帧，需要关注的应用层实现也是 DNS 本身，UDP 中目的端口为 53，Ip 协议中目的地址为 10.3.9.5，**第一个包是请求包。**

```
Domain Name System (query)
    Transaction ID: 0x01b9
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
    [Response In: 15]
```

第二行：会话标识（2 字节）0x01b9，是 DNS 报文的 ID 标识，对于请求报文和其对应的应答报文，这个字段是相同的，通过它可以区分 DNS 应答报文是哪个请求的响应。
第三行：标志（2 字节），含义如下：



QR（1bit）：查询/响应标志，0 为查询，1 为响应
opcode（4bit）：0 表示标准查询，1 表示反向查询，2 表示服务器状态请求
AA（1bit）：表示授权回答
TC（1bit）：表示可截断的
RD（1bit）：表示期望递归
RA（1bit）：表示可用递归
rcode（4bit）：表示返回码，0 表示没有差错，3 表示名字差错，2 表示服务器错误（Server Failure）
也就是说，当前是一个请求标志，标准查询，不授权回答，不可截断，期望递归，不可用递归，没有差错。
第四行、第五行、第六行、第七行：数量字段（总共 8 字节）：Questions、Answer RRs、Authority RRs、Additional RRs 各自表示后面的四个区域的数目。Questions 表示查询问题区域节的数量，Answers 表示回答区域的数量，Authoritative namesversers 表示授权区域的数量，Additional recoreds 表示附加区域的数量。

查询区正文：

```
Queries
  www.baidu.com: type AAAA, class IN
      Name: www.baidu.com
      [Name Length: 13]
      [Label Count: 3]
      Type: AAAA (IPv6 Address) (28)
      Class: IN (0x0001)
  [Response In: 15]
```

第三行：查询名 www.baidu.com，长度不固定，且不使用填充字节，一般该字段表示的就是需要查询的域名（如果是反向查询，则为 IP，反向查询即由 IP 地址反查域名）
第四行：查询名长度
第六行：查询类型，各类型如下，图中为域名获得的 IPv6 地址。

1	A	由域名获得 IPv4 地址
2	NS	查询域名服务器
5	CNAME	查询规范名称
6	SOA	开始授权
11	WKS	熟知服务
12	PTR	把 IP 地址转换成域名
13	HINFO	主机信息
15	MX	邮件交换

28　AAAA　由域名获得 IPv6 地址
252 AXFR　传送整个区的请求
255 ANY　对所有记录的请求
第七行：查询类，通常为 1，表明是 Internet 数据

打开第二个 DNS 包，该包为响应包

```
> Frame 15: 157 bytes on wire (1256 bits), 157 bytes captured (1256 bits) on interface 0
> Ethernet II, Src: NewH3CTe_11:20:01 (74:85:c4:11:20:01), Dst: IntelCor_80:04:8f (58:fb:84:80:04:8f)
> Internet Protocol Version 4, Src: 10.3.9.5, Dst: 10.128.235.166
> User Datagram Protocol, Src Port: 53, Dst Port: 58064
> Domain Name System (response)
```

UDP 中目的端口为 58064，Ip 协议中目的地址为 10.128.235.166。

```
v Domain Name System (response)
    Transaction ID: 0x01b9
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 0
  > Queries
  v Answers
    > www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
  v Authoritative nameservers
    > a.shifen.com: type SOA, class IN, mname ns1.a.shifen.com
    [Request In: 14]
    [Time: 0.002993000 seconds]
```

第二行：会话标识（2 字节）0x01b9，DNS 报文的 ID 标识，与它的请求包相同。
第三行：标志，含义是：响应，标准查询，非授权回答，不可截断，期望递归，可用递归，没有差错。
第四行、第五行、第六行、第七行：查询问题区域节的数量为 1，回答区域的数量为 1，授权区域的数量为 1，附加区域的数量为 0。
第八行：查询区域，已经分析过。
第九行：回答区域；
第十行：授权区域。

回答区域正文：

```
v Answers
  v www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
      Name: www.baidu.com
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 882
      Data length: 15
      CNAME: www.a.shifen.com
```

第二行：域名（2 字节或不定长）：它的格式和 Queries 区域的查询名字字段是一样的。有一点不同就是，当报文中域名重复出现的时候，该字段使用 2 个字节的偏移指针来表示。比如，在资源记录中，域名通常是查询问题部分的域名的重复，因此用 2 字节的指针来表示，具体格式是最前面的两个高位是 11，用于识别指针。其余的 14 位从 DNS 报文的开始处计数（从 0 开始），指出该报文中的相应字节数。
第三行：查询类型，表示资源记录的类型，CNAME 为查询规范名称。
第四行：查询类：对于 Internet 信息，总是 IN
第五行：生存时间（TTL）：以秒为单位，表示的是资源记录的生命周期，一般用于当地址解析程序取出资源记录后决定保存及使用缓存数据的时间，它同时也可以表明该资源记录的稳定程度，极为稳定的信息会被分配一个很大的值（比如 86400，这是一天的秒数）。
第六行：资源数据长度

第七行：资源数据：该字段是一个可变长字段，表示按照查询段的要求返回的相关资源记录的数据。可以是 Address（表明查询报文想要的回应是一个 IP 地址）或者 CNAME（表明查询报文想要的回应是一个规范主机名）等。

授权区域正文同回答区域正文格式几乎相同。

```
Authoritative nameservers
  a.shifen.com: type SOA, class IN, mname ns1.a.shifen.com
      Name: a.shifen.com
      Type: SOA (Start Of a zone of Authority) (6)
      Class: IN (0x0001)
      Time to live: 573
      Data length: 45
      Primary name server: ns1.a.shifen.com
      Responsible authority's mailbox: baidu_dns_master.baidu.com
      Serial Number: 1910270002
      Refresh Interval: 5 (5 seconds)
      Retry Interval: 5 (5 seconds)
      Expire limit: 2592000 (30 days)
      Minimum TTL: 3600 (1 hour)
```

第 8-14 行：均为资源数据。

5. TCP 协议消息分析

1.帧：

```
Frame 927: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
  Interface id: 0 (\Device\NPF_{AC73FB52-CECE-40E0-9A19-6190533DEF6F})
  Encapsulation type: Ethernet (1)
  Arrival Time: Oct 27, 2019 17:56:26.925409000 中国标准时间
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1572170186.925409000 seconds
  [Time delta from previous captured frame: 0.000716000 seconds]
  [Time delta from previous displayed frame: 0.000716000 seconds]
  [Time since reference or first frame: 39.711775000 seconds]
  Frame Number: 927
  Frame Length: 66 bytes (528 bits)
  Capture Length: 66 bytes (528 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:tcp]
  [Coloring Rule Name: HTTP]
  [Coloring Rule String: http || tcp.port == 80 || http2]
```

第一行：927 号帧，线路 66 字节，实际捕获 66 字节；
第二行：接口 id 为 0
第三行：封装类型
第四行：捕获日期和时间：Arrival Time: Oct 27, 2019 17:56:26.925409000 中国标准时间
第六行：信息出现时间：Epoch Time: 1572170186.925409000 seconds
第七行：[Time delta from previous captured frame: 0.141705000 seconds]与前一捕获数据帧时间隔
第八行：[Time delta from previous displayed frame: 0.141705000 seconds]与前一显示帧的时间隔
第九行：[Time since reference or first frame: 28.052351000 seconds]此包与第一帧的时间间隔
第十行：Frame Number: 927 帧的编号
第十一行：Frame Length: 66 bytes (528 bits) 帧的长度
第十二行：Capture Length: 66 bytes (528 bits) 被捕获的帧的长度
第十三行：[Frame is marked: False] 帧被标记：无
第十四行：[Frame is ignored: False] 帧被忽略：无
第十五行：[Protocols in frame: eth:ethertype:ip:tcp] 帧中的协议以太网、ip、tcp
第十六行：[Coloring Rule Name: HTTP] 色彩规则名称

第十七行：[Coloring Rule String: http || tcp.port == 80 || http2] 色彩规则字符串

2.Ethernet 协议分析：

```
Ethernet II, Src: IntelCor_80:04:8f (58:fb:84:80:04:8f), Dst: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
  Destination: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
       Address: NewH3CTe_11:20:01 (74:85:c4:11:20:01)
       .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
       .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: IntelCor_80:04:8f (58:fb:84:80:04:8f)
       Address: IntelCor_80:04:8f (58:fb:84:80:04:8f)
       .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
       .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

Destination: NewH3CTe_11:20:01 (74:85:c4:11:20:01)目的 mac 地址

Source: IntelCor_80:04:8f (58:fb:84:80:04:8f)源 mac 地址

Type: IPv4 (0x0800) 类型是 ipv4 数据包

3.ipv4 协议信息分析

```
Internet Protocol Version 4, Src: 10.128.235.166, Dst: 123.151.72.63
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0198 (408)
  > Flags: 0x4000, Don't fragment
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x3f2f [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.128.235.166
    Destination: 123.151.72.63
```

第一行：Internet Protocol Version 4, Src: 10.128.235.166, Dst: 123.151.72.63 ip 版本为 4

第二行：互联网协议 IPv4

第三行：.... 0101 = Header Length: 20 bytes (5) IP 头部长度为 20 字节

第四行：Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) 差分服务字段

第五行：IP 包的总长度为 52

第六行：标志字段

第七行：Flags: 0x4000, Don't fragment 标记字段，不支持分组

第八行：Time to live: 128 生存期，TTL 通常表示包在被丢弃前最多能经过的路由器个数，当数据包传输到一个路由器之后，TTL 就自动减 1，如果减到 0 了还没有传送到目标主机，那么就自动丢失。

第九行：此包内封装的上层协议为 TCP

第十行：Header checksum: 0x3f2f [validation disabled] 头部数据的校验和

第十一行：头部校验和状态：未验证

第十二行：Source: 10.128.235.166 源 IP 地址

第十三行：Destination: 123.151.72.63 目标 IP 地址

4.transmission control protocol 分析

```
Transmission Control Protocol, Src Port: 13917, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 13917
    Destination Port: 80
    [Stream index: 38]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x002 (SYN)
    Window size value: 65535
    [Calculated window size: 65535]
    Checksum: 0xa807 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  > Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
  > [Timestamps]
```

第一行：TCP 报文主要内容，包括源端口号和目的端口号、Seq 和长度。
第二行：源端口号 13917，数据传输的 16 位源端口号和 16 位目标端口号(用于寻找发端和收端应用进程)；
第三行：目标端口号 80。
第四行：Stream index 是一个内部 Wireshark 映射到：[IP 地址 A，TCP 端口 A，IP 地址 B，TCP 端口 B] ,相同 tcp.stream 值的所有数据包对于这些字段都应该具有相同的值
第五行：TCP 段长度为 0
第六行：相对序列号(此序列号用来确定传送数据的正确位置，且序列号用来侦测丢失的包)；
第七行：下一个序列号
第八行：Acknowledgment number 是 44 位确认序列号，值等于 1 表示数据包收到，确认有效；
第九行：头部字节长度是 32 字节；
第十行：Flags，TCP 标记字段，含 6 种标志；ACK：确认序号有效；SYN：同步序号用来发起一个连接；FIN：发端完成发送任务；RST：重新连接；PSH：接收方应该尽快将这个报文段交给应用层；URG：紧急指针(urgentpointer)有效；
第十一行：window，流量控制的窗口大小，TCP 的流量控制由连接的每一端通过声明的窗口大小来提供。窗口大小为字节数，起始于确认序号字段指明的值，这个值是接收端正期望接收的字节。窗口大小是一个 16bit 字段，因而窗口大小最大为 65536 字节，上面显示窗口大小为 65535 字节；
第十三行：Checksum，TCP 数据段 16 位校验和，检验和覆盖了整个的 TCP 报文段，由发端计算和存储，并由收端进行验证；
第十五行：紧急指针：只有当 URG 标志置 1 时紧急指针才有效。紧急指针是一个正的偏移量，和顺序号字段中的值相加表示紧急数据最后一个字节的序号。 TCP 的紧急方式是发送端向另一端发送紧急数据的一种方式；
第十六行：选项和填充：最常见的可选字段是最长报文大小，又称为 MSS（Maximum Segment Size），每个连接方通常都在通信的第一个报文段（为建立连接而设置 SYN 标志为 1 的那个段）中指明这个选项，它表示本端所能接受的最大报文段的长度。选项长度不一定是 32 位的整数倍，所以要加填充位，即在这个字段中加入额外的零，以保证 TCP 头是 32 的整数倍。

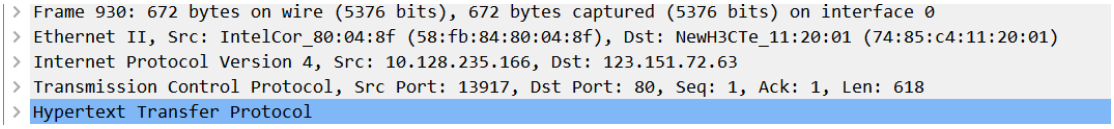
6. IP 协议消息分析

```
Internet Protocol Version 4, Src: 10.128.235.166, Dst: 123.151.72.63
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x0198 (408)
  > Flags: 0x4000, Don't fragment
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x3f2f [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.128.235.166
    Destination: 123.151.72.63
```

第一行：Internet Protocol Version 4, Src: 10.128.235.166, Dst: 123.151.72.63 ip 版本为 4
第二行：互联网协议 IPv4
第三行：.... 0101 = Header Length: 20 bytes (5) IP 头部长度为 20 字节
第四行：Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) 差分服务字段
第五行：IP 包的总长度为 52

第六行：标志字段
第七行：Flags: 0x4000, Don't fragment 标记字段，不支持分组
第八行：Time to live: 128 生存期，TTL 通常表示包在被丢弃前最多能经过的路由器个数，当数据包传输到一个路由器之后，TTL 就自动减 1，如果减到 0 了还没有传送到目标主机，那么就自动丢失。
第九行：此包内封装的上层协议为 TCP
第十行：Header checksum: 0x3f2f [validation disabled] 头部数据的校验和
第十一行：头部校验和状态：未验证
第十二行：Source: 10.128.235.166 源 IP 地址
第十三行：Destination: 123.151.72.63 目标 IP 地址

7. http 协议消息格式



第一行：帧，帧序号为 930，传送的字节数为 672 字节
第二行：以太网，是数据链路层，源 mac 地址为 58:fb:84:80:04:8f，目的 mac 地址为 74:85:c4:11:20:01
第三行：ipv4 协议，是网络层，源 ip 地址为 10.128.235.166，目的 ip 地址为 123.151.72.63
第四行：TCP 协议，传输控制协议，是传输层，源端口为 13917，目标端口为 80，此时三次握手已经完成，Seq=1，Ack=1，长度为 618
第五行：http 协议，超文本传输协议，是应用层。