哈尔滨工业大学

<<计算机网络>>

实验报告

(2019 年度秋季学期**)**

姓名:	易亚玲
学号:	**1170300511**
学院:	计算机科学与技术学院
教师:	聂兰顺

实验四 HTTP 代理服务器的设计与实现

一、实验目的

熟悉并掌握 Wireshark 的基本操作，了解网络协议实体间进行交互以 及报文交换的情况。

二、实验内容

- Windows 9x/NT/2000/XP/2003
- 与因特网连接的计算机网络系统
- Wireshark

三、实验过程及结果

1) 学习 Wireshark 的使用

2) 利用 Wireshark 分析 HTTP 协议

3) 利用 Wireshark 分析 TCP 协议

4) 利用 Wireshark 分析 IP 协议

5) 利用 Wireshark 分析 Ethernet 数据帧 选做内容:

a) 利用 Wireshark 分析 DNS 协议

b) 利用 Wireshark 分析 UDP 协议

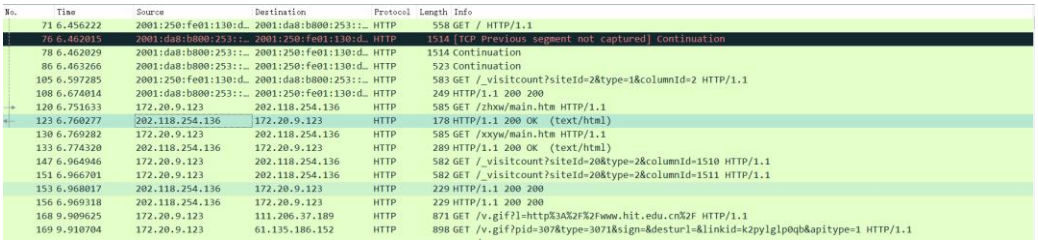
c) 利用 Wireshark 分析 ARP 协议

四、实验心得

4.1 HTTP 分析

4.1.1 HTTP GET/reponse 交互

✧ 由下图可以看出，我的浏览器运行的是 HTTP1.1，我所访问的服务器运行的也是 HTTP1.1。我的浏览器的 ip 地址是 172.20.9.123，www.hit.edu.cn 服务器的 ip 地址是 202.118.254.136



No.	Time	Source	Destination	Protocol	Length	Info
71	6.456222	2001:250:fe01:130:d…	2001:da8:b800:253::…	HTTP	558	GET / HTTP/1.1
76	6.462015	2001:da8:b800:253::…	2001:250:fe01:130:d…	HTTP	1514	[TCP Previous segment not captured] Continuation
78	6.462029	2001:da8:b800:253::…	2001:250:fe01:130:d…	HTTP	1514	Continuation
86	6.463266	2001:da8:b800:253::…	2001:250:fe01:130:d…	HTTP	523	Continuation
105	6.597285	2001:250:fe01:130:d…	2001:da8:b800:253::…	HTTP	583	GET /_visitcount?siteId=2&type=1&columnId=2 HTTP/1.1
108	6.674014	2001:da8:b800:253::…	2001:250:fe01:130:d…	HTTP	249	HTTP/1.1 200 200
120	6.751633	172.20.9.123	202.118.254.136	HTTP	585	GET /zhxw/main.htm HTTP/1.1
123	6.760277	202.118.254.136	172.20.9.123	HTTP	178	HTTP/1.1 200 OK (text/html)
130	6.769282	172.20.9.123	202.118.254.136	HTTP	585	GET /xxyw/main.htm HTTP/1.1
133	6.774320	202.118.254.136	172.20.9.123	HTTP	289	HTTP/1.1 200 OK (text/html)
147	6.964946	172.20.9.123	202.118.254.136	HTTP	582	GET /_visitcount?siteId=20&type=2&columnId=1510 HTTP/1.1
151	6.966701	172.20.9.123	202.118.254.136	HTTP	582	GET /_visitcount?siteId=20&type=2&columnId=1511 HTTP/1.1
153	6.968017	202.118.254.136	172.20.9.123	HTTP	229	HTTP/1.1 200 200
156	6.969318	202.118.254.136	172.20.9.123	HTTP	229	HTTP/1.1 200 200
168	9.909625	172.20.9.123	111.206.37.189	HTTP	871	GET /v.gif?l=http%3A%2F%2Fwww.hit.edu.cn%2F HTTP/1.1
169	9.910704	172.20.9.123	61.135.186.152	HTTP	898	GET /v.gif?pid=307&type=307&sign=&desturl=&linkid=k2pylglp0qb&apitype=1 HTTP/1.1

✧ 根据捕获到的报文，可以看出我的浏览器可以接受 html 或者 text 文件，该文件可以是由 gzip 编码的字节码,而且由注明接受语言为 zh-CN(即中文简体).同时可以看到服务器在成功返回网页后返回了状态码 200，并且将页面内容明确返回。

```
HTTP/1.1 200 OK\r\n
Date: Fri, 08 Nov 2019 09:48:27 GMT\r\n
Server: Apache/2.4.33 (Unix) mod_jk/1.2.43\r\n
Accept-Ranges: bytes\r\n
Vary: Accept-Encoding\r\n
Content-Encoding: gzip\r\n
Content-Length: 1349\r\n
Connection: close\r\n
Content-Type: text/html\r\n
\r\n
```

4.1.2 HTTP 条件 GET/reponse 交互

第一个 HTTP GET 请求的请求报文中并没有 IF-MODIFIED-SINCE，然后服务器在收到浏览器的请求后，返回了浏览器请求内容的最后修改时间，即 Last-Modified,浏览器，如果浏览器要访问的对象并未更新，浏览器可以再向服务器发送请求报文，然后服务器会返回浏览器想要的内容；加入服务器缓存中已经是最新的内容，那么浏览器将不再发这个内容的请求报文。当再次刷新时，

GET 报文中增加了 IF-Modified-Since 字段（如下图），这句话的意识是告诉服务器，如果服务器的更新时间在这个事件之后，则不用返回页面的内容；如服务器的更新时间在这个时间之前，那么需要返回这个页面的内容，服务器就不需要返回自己的更新时间，直接做出判断。

```
If-Modified-Since: Thu, 07 Sep 2017 01:15:30 GMT\r\n
If-None-Match: "69a6-5588f323d5080-gzip"\r\n
```

较晚发送的 GET 请求，服务求返回的状态码仍然是 200，但是由于有缓存，所以服务器并没有明确返回页面的内容，需要浏览器从缓存中读取。

4.2 TCP 分析

向服务器传输文件的客户机的 ip 是 172.20.9.123，端口号是 54078；服务器的 ip 是 128.119.245.12，端口号是 80

```
Transmission Control Protocol, Src Port: 54078, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
  Source Port: 54078
  Destination Port: 80
```

客户机之间用于初始化 SYN 报文段的序号是 1，使用 flags 来表示报文段

```
Flags: 0x002 (SYN)
Window size value: 65535
```

服务器向客户机发送 SYNACK 报文段的序列号为 0，该报文中，Acknowledge 字段为 1，这是由于服务器已经收到了来自客户机序列号为 0 的 SYN 报文，期望下次收到的序列号就为（0+1）。因为只有 SYNACK 报文段才能 SYN 和 ACK 不同时为 0，所以可以区分 SYNACK 报文段。

```
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
```

其中三次握手如图，，初始客户机发送 SYN=1 到服务器，服务器发送 SYNACK 到客户机，客户机再发送 SYN=0

```
[FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
[SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
[SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
[SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
[SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
[ACK] Seq=1 Ack=1 Win=262144 Len=0
[ACK] Seq=1 Ack=1 Win=262144 Len=0
```

HTTP POST 段的序列号是 165592，如果这是第一个报文段，那么该 TCP 连接上的第六个报文段的序列号是 165593（之前以序列号 0 开始，第六个序列号是 1），第六个报文段的发送时间是 2019.11.8 ,20:50:44.426274000

```
Transmission Control Protocol, Src Port: 54109, Dst Port: 80, Seq: 165592, Ack: 1, Len: 1000
    Source Port: 54109
    Destination Port: 80
    [Stream index: 2]
    [TCP Segment Len: 1000]
    Sequence number: 165592    (relative sequence number)
    [Next sequence number: 166592    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
```

```
Encapsulation type: Ethernet (1)
Arrival Time: Nov  8, 2019 20:50:44.426274000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
```

一直到 2019.11.8，　20：50：44.670888000 才被接收

```
Interface id: 0 (\Device\NPF_{E1907065-5A77-4BA4-AF0A-28E89468112
Encapsulation type: Ethernet (1)
Arrival Time: Nov  8, 2019 20:50:44.670888000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1573217444.670888000 seconds
```

前 6 个 TCP 报文段的长度依次是 0，0，0，0，0，0，因为要进行三次握手

```
54 54078 → 80 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
66 54108 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
66 54109 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
66 80 → 54108 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
66 80 → 54109 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
54 54108 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
```

接收端在整个跟踪过程中公示的最小的可用缓存空间是512，限制发送端发送后，接收端的缓存就够用了。另外，在跟踪过程中有重传的报文，重传的报文带有 Retransmission 字样

```
514 54109 → 80 [ACK] Seq=12292 Ack=1 Win=262144 Len=1460 [TCP segment of a reassembl
 54 [TCP Retransmission] 54078 → 80 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
 60 80 → 54109 [ACK] Seq=1 Ack=612 Win=30464 Len=0
```

最后，根据第一个和最后一个报文段 arrival time 可以计算总时间约为 31s.

```
Encapsulation type: Ethernet (1)
Arrival Time: Nov  8, 2019 20:50:43.723130000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
Encapsulation type: Ethernet (1)
Arrival Time: Nov  8, 2019 20:51:14.660347000 中国标准时间
[Time shift for this packet: 0.000000000 seconds]
```

4.3 IP 分析

（1）主机的 IP 地址是 172.20.9.123，在 IP 数据包的头部，上层协议字段的值是 1，指的是 ICMP 协议。IP 头一共有 20 个字节

```
Internet Protocol Version 4, Src: 61.167.60.70, Dst: 172.20.9.123
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 56
```

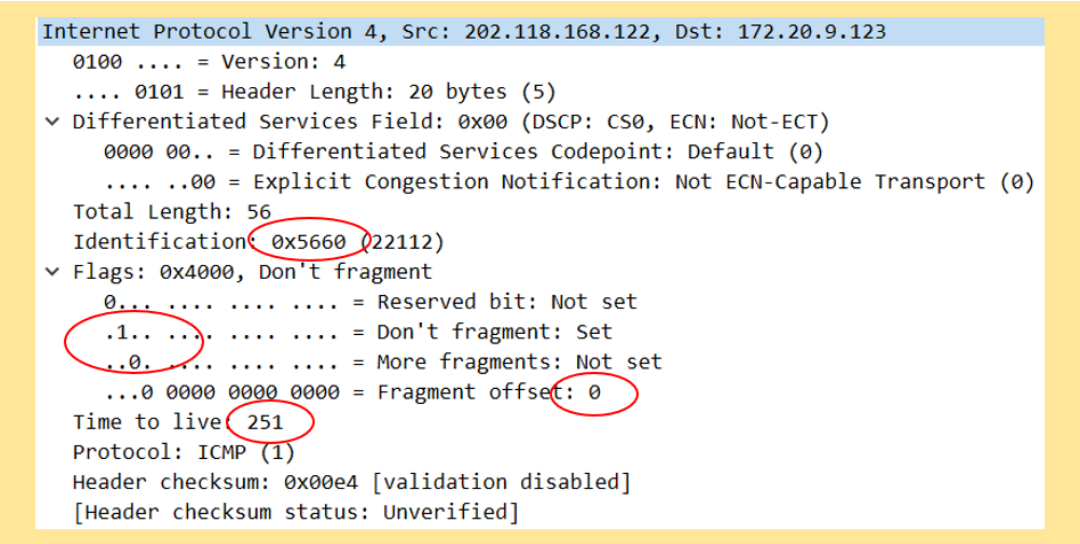
数据包净载为 28 字节

```
    [Response time: 2.457 ms]
  v Data (28 bytes)
      Data: 314550696e67506c6f7474657250726f332e3
      [Length: 28]
```

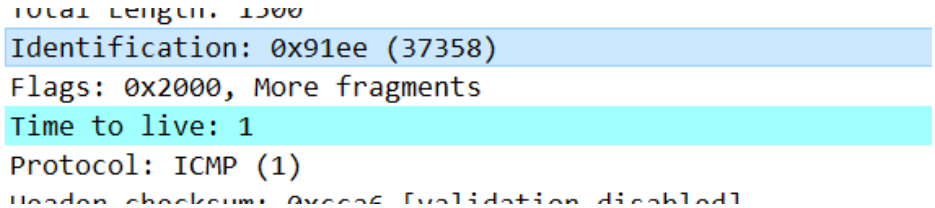
这个 ICMP 数据包没有分片，因为片偏移量为 0，且为最后一片或者未分片

```
  v Flags: 0x0000
      0... .... .... .... = Reserved bit: Not set
      .0.. .... .... .... = Don't fragment: Not set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
```

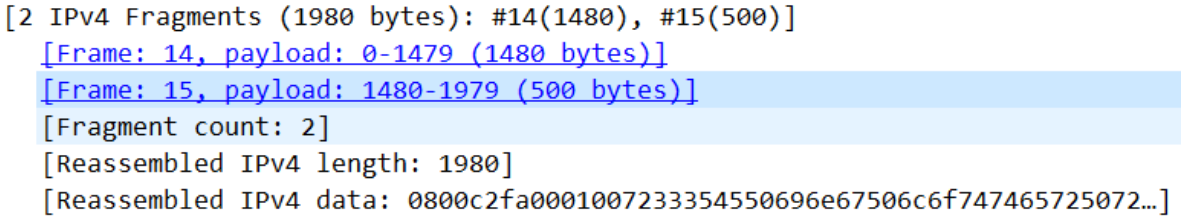
（2）在 IP 数据报中，Identification,DF,MF,片偏移和头部校验和总变，但其余的都不变。其中 Identification 必须要改变，因为这个是每个 IP 数据包的标识；checksum 往往也不相同。但是头部的版本号必须保持常量 4，这表示 IPv4 版本，头部的长度也是固定的 20 字节，这个也不能变。上一级的协议是 ICMP，这个也不可以改变。Identification 在同一主机发送的 ICMP 数据报中存在连续的现象，但不是一直都连续。



（3）Identification 是 IP 数据报的标识，它标识发生 Time-to-live exceeded 的数据报的 ID 号，TTL 是 IP 数据报最多可以经过的路由器的数量（也称为最大生命周期）。TTL 的值仅为 1，因为它只需将消息传给临近的路由器就可以。



（4）数据包大小改为 2000 后主机发送的第一个 ICMP Echo Request 消息被分解为 2 个 IP 数据报，可以由 Fragment count 为 2 看出



如果 IP 数据报的片偏移量不为 0，那么这个一定被分片了。此外，若 IP 数据报片偏移量为 0，但 MF=1,说明 IP 数据报是分片的第一片；若 IP 数据报的 MF=0，片偏移量为 0，则说明 IP 数据报未分片；若 IP 数据报的 MF=0,片偏移量不为 0，说明是 IP 分片的最后一片。

IP 分片的第一片的长度为 1500（1480+20）

（5）原始数据包一共被分为了 3 片,这些分片的 IP 数据包的头部的 MF，片偏移字段会发生改变。

```
[3 IPv4 Fragments (3480 bytes): #25(1480), #26(1480), #27(520)]
   [Frame: 25, payload: 0-1479 (1480 bytes)]
   [Frame: 26, payload: 1480-2959 (1480 bytes)]
   [Frame: 27, payload: 2960-3479 (520 bytes)]
   [Fragment count: 3]
   [Reassembled IPv4 length: 3480]
   [Reassembled IPv4 data: 08005e940001007133324550696e67506c6f747465725072…]
```

4.4 ARP 分析

主机的 arp 缓存表

```
C:\Users\Ww小易>arp -a

接口: 192.168.198.1 --- 0xb
  Internet 地址         物理地址              类型
  192.168.198.254       00-50-56-e9-bb-b3     动态
  192.168.198.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态

接口: 192.168.219.1 --- 0xc
  Internet 地址         物理地址              类型
  192.168.219.254       00-50-56-e1-e1-65     动态
  192.168.219.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态

接口: 192.168.137.1 --- 0xd
  Internet 地址         物理地址              类型
  192.168.137.255       ff-ff-ff-ff-ff-ff     静态
  224.0.0.22            01-00-5e-00-00-16     静态
  224.0.0.251           01-00-5e-00-00-fb     静态
  224.0.0.252           01-00-5e-00-00-fc     静态
  239.255.255.250       01-00-5e-7f-ff-fa     静态
  255.255.255.255       ff-ff-ff-ff-ff-ff     静态
```

ARP 数据包包含数据报格式，协议类型等数据，判断是请求还是应答，直接根据 Opcode 来判断，1 表示请求，2 表示应答。ARP 请求要在广播中传送，而应答在局域网的地址中传送的原因是：由于查询方不知道被查询方的 MAC 地址（这也正是为何要进行 ARP 查询的原因），而所有结点都要处理广播帧，因此通过广播发送给被查询方。而被查询方通过接收到的广播帧的源地址知道查询方的 MAC 地址了，因此可以用该地址进行响应，这样局域网中的除查询方外其它主机就不会接收和处理该 ARP 响应了，避免浪费带宽和其它主机的计算资源。

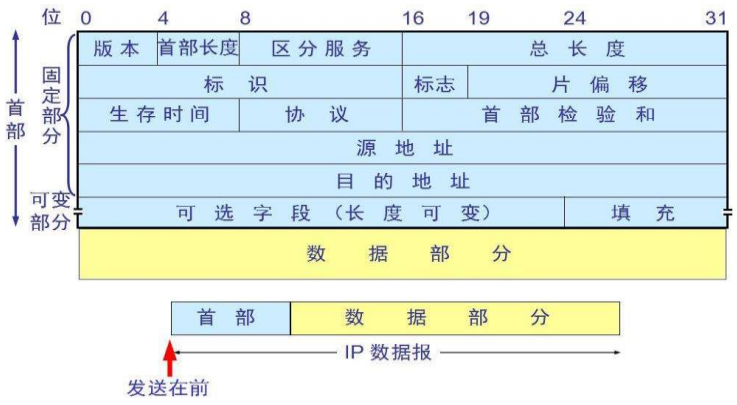
```
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: RuijieNe_a5:e2:d3 (58:69:6c:a5:e2:d3)
Sender IP address: 172.20.0.1
Target MAC address: IntelCor_fc:74:cf (3c:f8:62:fc:74:cf)
Target IP address: 172.20.15.139
```

4.5 UDP 分析

消息是基于 UDP 的，头部的协议序号为 17，主机 IP 为 172.20.15.139，目的主机的 IP 为 140.207.62.151，主机端口号为 4019，目的主机的端口号为 8000

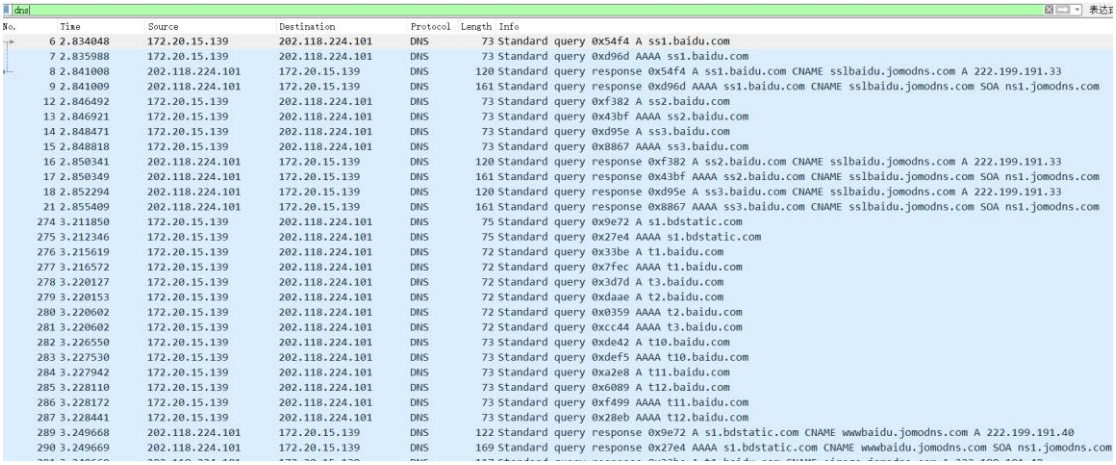
```
Internet Protocol Version 4, Src: 172.20.15.139, Dst: 140.207.62.151
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
   Total Length: 67
   Identification: 0x3eff (16127)
 > Flags: 0x0000
   Time to live: 128
   Protocol: UDP (17)
   Header checksum: 0x74a5 [validation disabled]
   [Header checksum status: Unverified]
   Source: 172.20.15.139
   Destination: 140.207.62.151
User Datagram Protocol, Src Port: 4019, Dst Port: 8000
   Source Port: 4019
   Destination Port: 8000
   Length: 47
   Checksum: 0xcb7e [unverified]
   [Checksum Status: Unverified]
   [Stream index: 0]
 > [Timestamps]
Data (39 bytes)
```

IP 数据报的格式如下



主机发送一个 ICQ 数据包后，服务器返回一个数据包是为了和主机建立临时连接。UDP 相比于 TCP，没有 keep-alive 字段，而且每次发送 UDP 数据报都需要建立新的连接，这和 TCP 完全不一样。

4.6 DNS 分析



No.	Time	Source	Destination	Protocol	Length	Info
6	2.834048	172.20.15.139	202.118.224.101	DNS	73	Standard query 0x54f4 A ss1.baidu.com
7	2.835988	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xd96d AAAA ss1.baidu.com
8	2.841008	202.118.224.101	172.20.15.139	DNS	120	Standard query response 0x54f4 A ss1.baidu.com CNAME sslbaidu.jomodns.com A 222.199.191.33
9	2.841009	202.118.224.101	172.20.15.139	DNS	161	Standard query response 0xd96d AAAA ss1.baidu.com CNAME sslbaidu.jomodns.com SOA ns1.jomodns.com
12	2.846492	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xf382 A ss2.baidu.com
13	2.846921	172.20.15.139	202.118.224.101	DNS	73	Standard query 0x43bf AAAA ss2.baidu.com
14	2.848471	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xd95e A ss3.baidu.com
15	2.848818	172.20.15.139	202.118.224.101	DNS	73	Standard query 0x8867 AAAA ss3.baidu.com
16	2.850341	202.118.224.101	172.20.15.139	DNS	120	Standard query response 0xf382 A ss2.baidu.com CNAME sslbaidu.jomodns.com A 222.199.191.33
17	2.850349	202.118.224.101	172.20.15.139	DNS	161	Standard query response 0x43bf AAAA ss2.baidu.com CNAME sslbaidu.jomodns.com SOA ns1.jomodns.com
18	2.852294	202.118.224.101	172.20.15.139	DNS	120	Standard query response 0xd95e A ss3.baidu.com CNAME sslbaidu.jomodns.com A 222.199.191.33
21	2.855409	202.118.224.101	172.20.15.139	DNS	161	Standard query response 0x8867 AAAA ss3.baidu.com CNAME sslbaidu.jomodns.com SOA ns1.jomodns.com
274	3.211850	172.20.15.139	202.118.224.101	DNS	75	Standard query 0x9e72 A s1.bdstatic.com
275	3.212346	172.20.15.139	202.118.224.101	DNS	75	Standard query 0x27e4 AAAA s1.bdstatic.com
276	3.215619	172.20.15.139	202.118.224.101	DNS	72	Standard query 0x33be A t1.baidu.com
277	3.216572	172.20.15.139	202.118.224.101	DNS	72	Standard query 0x7fec AAAA t1.baidu.com
278	3.220127	172.20.15.139	202.118.224.101	DNS	72	Standard query 0x3d7d A t3.baidu.com
279	3.220153	172.20.15.139	202.118.224.101	DNS	72	Standard query 0xdaae A t2.baidu.com
280	3.220602	172.20.15.139	202.118.224.101	DNS	72	Standard query 0x0359 AAAA t2.baidu.com
281	3.220602	172.20.15.139	202.118.224.101	DNS	72	Standard query 0xcc44 AAAA t3.baidu.com
282	3.226550	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xde42 A t10.baidu.com
283	3.227530	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xdef5 AAAA t10.baidu.com
284	3.227942	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xa2e8 A t11.baidu.com
285	3.228110	172.20.15.139	202.118.224.101	DNS	73	Standard query 0x6089 A t12.baidu.com
286	3.228172	172.20.15.139	202.118.224.101	DNS	73	Standard query 0xf499 AAAA t11.baidu.com
287	3.228441	172.20.15.139	202.118.224.101	DNS	73	Standard query 0x28eb AAAA t12.baidu.com
289	3.249668	202.118.224.101	172.20.15.139	DNS	122	Standard query response 0x9e72 A s1.bdstatic.com CNAME wwwbaidu.jomodns.com A 222.199.191.40
290	3.249669	202.118.224.101	172.20.15.139	DNS	169	Standard query response 0x27e4 AAAA s1.bdstatic.com CNAME wwwbaidu.jomodns.com SOA ns1.jomodns.com