西安交通大學

XI'AN JIAOTONG UNIVERSITY

软件定义网络实验报告

The Process of Network

课程名称：软件定义网络

姓名：曾锦程

学院：计算机学院

专业：计算机科学与技术

学号：2203613040

指导老师：张鹏

2023 年 3 月 10 日

西安交通大学实验报告

专业： 计算机科学与技术
姓名： 曾锦程
学号： 2203613040
日期： 2023 年 3 月 10 日
地点： Personal Device

课程名称：	软件定义网络	指导老师：	张鹏	成绩：	
实验名称：	The Process of Network	实验类型：		同组学生姓名：	

一、 实验目的和要求

Use Wireshark to understand the process of network communication.

-Capture as more protocol packets as possible, including but not limited to DHCP, ARP, DNS, HTTP, TCP, UDP, ⋯

-A report that describes the detailed communication process, and what you newly understand after this Wireshark lab

二、 实验内容和步骤

-Start your device from a clean state, without an IP address

-Run Wireshark to capture packets

-Use your browser to visit a website (e.g., www.baidu.com)

-Read the captured packets in sequence

三、 实验环境

计算机，Wireshark, 个人服务器 43.143.132.10 以及域名 www.zjcblog.top, QQ。

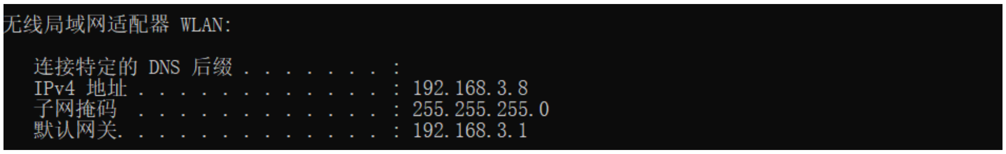
四、 实验过程

1. 将电脑设为 clean state

使用命令：

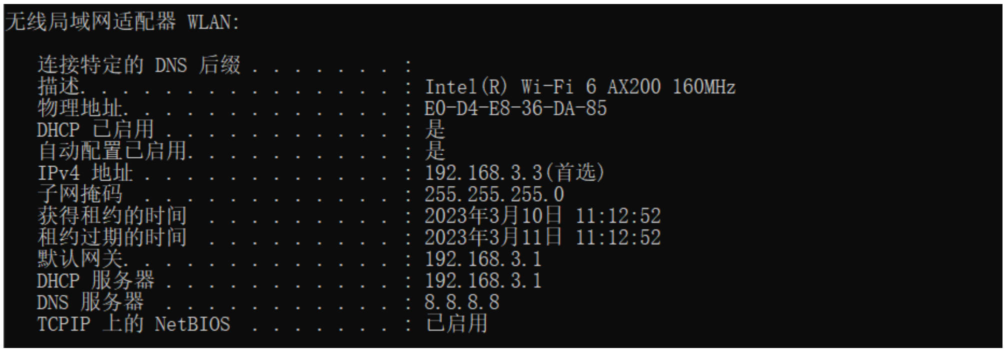
```
netsh interface ip set address name="WLAN" static 地址 + 子网掩码 + 网关
```

将原本动态设置 IP 地址设置为静态 (当重新设置成动态后，主机会向 DHCP 服务器发包，就能达到 clean state 的效果)。



```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   IPv4 地址 . . . . . . . . . . . . : 192.168.3.8
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.3.1
```

图 1: 手动设置 IP 地址后 WLAN



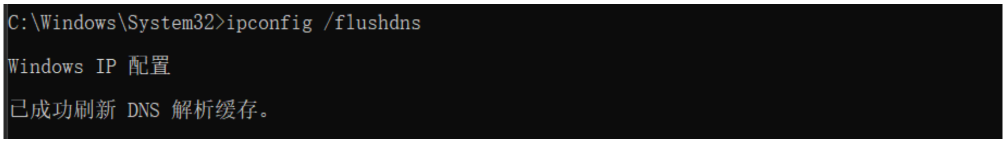
```
无线局域网适配器 WLAN:

   连接特定的 DNS 后缀 . . . . . . . :
   描述. . . . . . . . . . . . . . . : Intel(R) Wi-Fi 6 AX200 160MHz
   物理地址. . . . . . . . . . . . . : E0-D4-E8-36-DA-85
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   IPv4 地址 . . . . . . . . . . . . : 192.168.3.3(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间  . . . . . . . . . : 2023年3月10日 11:12:52
   租约过期的时间  . . . . . . . . . : 2023年3月11日 11:12:52
   默认网关. . . . . . . . . . . . . : 192.168.3.1
   DHCP 服务器 . . . . . . . . . . . : 192.168.3.1
   DNS 服务器  . . . . . . . . . . . : 8.8.8.8
   TCPIP 上的 NetBIOS  . . . . . . . : 已启用
```

图 2: 自动设置 IP 地址后 WLAN

使用命令:

```
ipconfig /flushdns
```

来刷新 dns 缓存，把 dns 服务器设置成 8.8.8.8(谷歌 dns 服务器)。

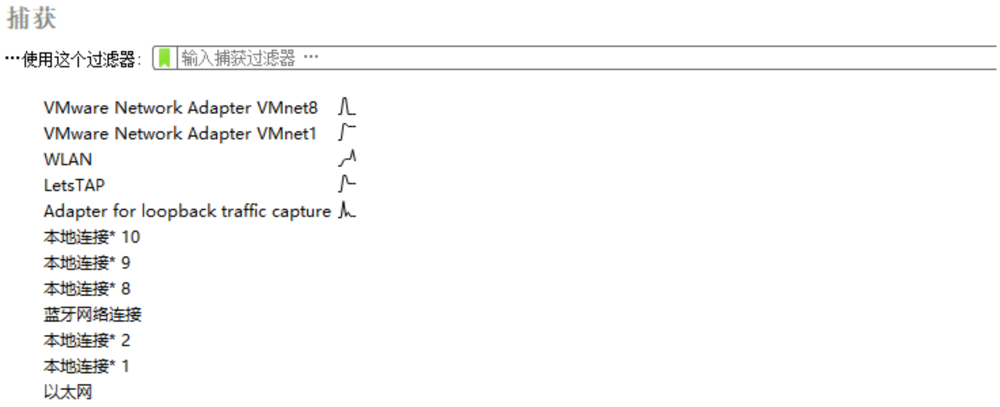


```
C:\Windows\System32>ipconfig /flushdns

Windows IP 配置

已成功刷新 DNS 解析缓存。
```

图 3: 刷新 DNS 缓存

2. Wireshark 抓包

保证在设置静态 IP 地址后开始使用 Wireshark 来抓包，接口选择 WLAN 口，之后开始捕获。



捕获

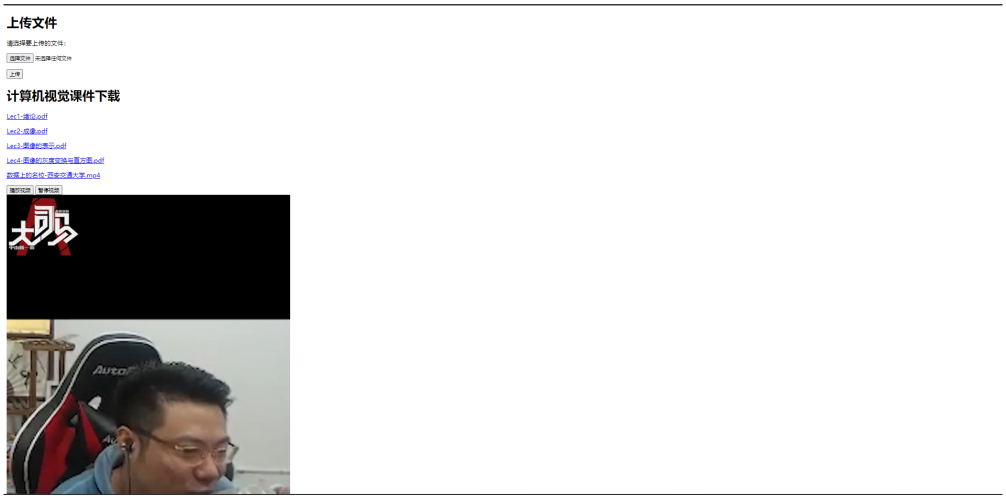
…使用这个过滤器：输入捕获过滤器 …

VMware Network Adapter VMnet8
VMware Network Adapter VMnet1
WLAN
LetsTAP
Adapter for loopback traffic capture
本地连接* 10
本地连接* 9
本地连接* 8
蓝牙网络连接
本地连接* 2
本地连接* 1
以太网

图 4: wireshark 选择接口

3. 打开网站和应用

在抓包过程中打开 QQ 和个人服务器网站 www.zjcblog.top 以及 www.bilibili.com, 进行实验观察，并进行协议分析。



上传文件

请选择要上传的文件:

未选择任何文件

计算机视觉课件下载

Lec1-绪论.pdf

Lec2-成像.pdf

Lec3-图像的表示.pdf

Lec4-图像的灰度变换与直方图.pdf

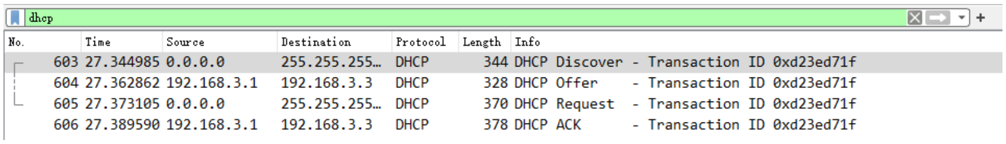
数据上的名校-西安交通大学.mp4

图 5: www.zjcblog.top 上提供的上传、下载、播放视频接口

五、 上网协议分析

1. DHCP

开启 filter:dhcp，发现抓到的包分为四种类型。

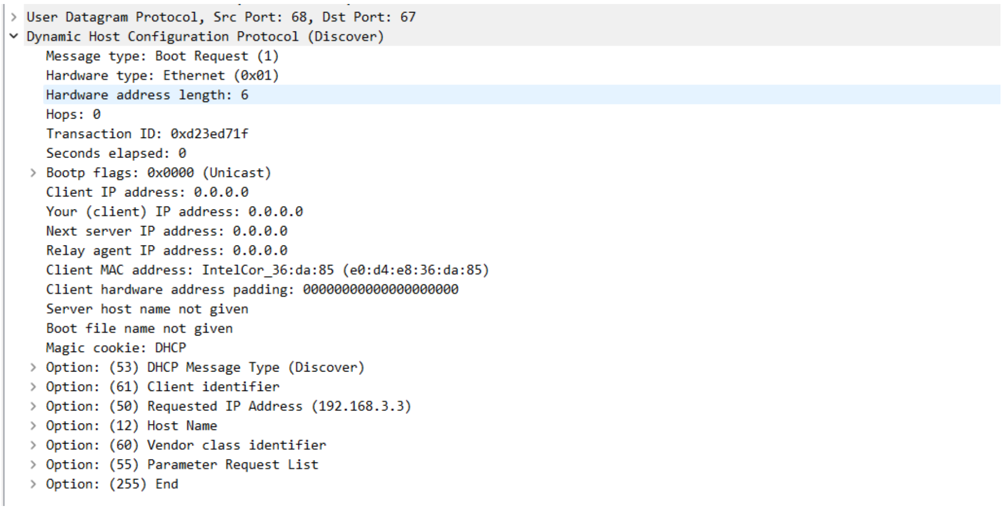


```
dhcp
No.   Time      Source       Destination   Protocol Length Info
  603 27.344985 0.0.0.0      255.255.255…  DHCP        344 DHCP Discover - Transaction ID 0xd23ed71f
  604 27.362862 192.168.3.1  192.168.3.3   DHCP        328 DHCP Offer    - Transaction ID 0xd23ed71f
  605 27.373105 0.0.0.0      255.255.255…  DHCP        370 DHCP Request  - Transaction ID 0xd23ed71f
  606 27.389590 192.168.3.1  192.168.3.3   DHCP        378 DHCP ACK      - Transaction ID 0xd23ed71f
```

图 6: 抓到的 DHCP 协议包

查阅并分析得知 DHCP 的基本工作过程如下:

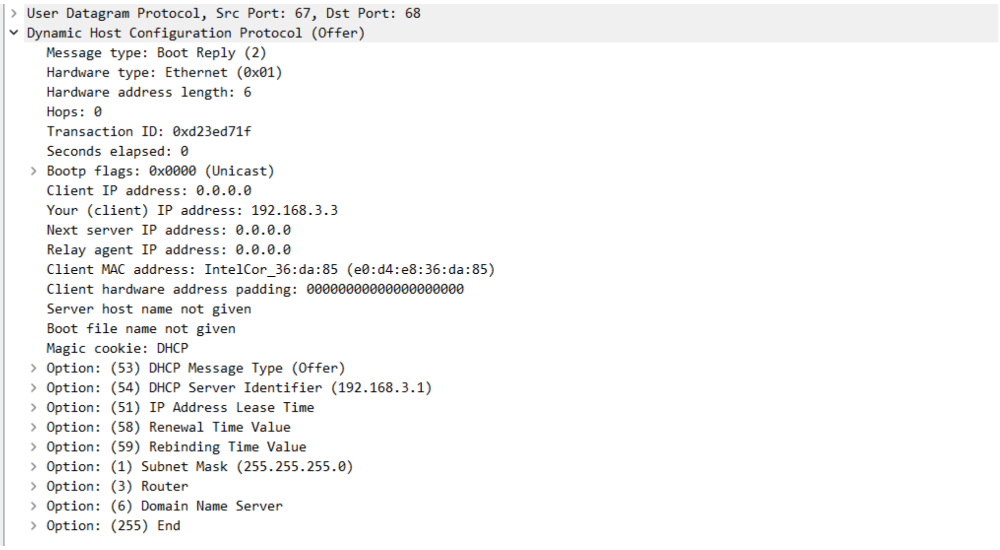
(1) 寻找 DHCP 服务器。DHCP 客户端（需要动态获得 IP 地址的主机）启动时，会广播发送一个 DHCP 发现 (DHCP Discover) 报文，由于客户端还不知道自己属于哪一个网络，所以 IP 分组的源地址为 0.0.0.0, 而目的地址为 255.255.255.255。



```
> User Datagram Protocol, Src Port: 68, Dst Port: 67
v Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xd23ed71f
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Discover)
  > Option: (61) Client identifier
  > Option: (50) Requested IP Address (192.168.3.3)
  > Option: (12) Host Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  > Option: (255) End
```

图 7: DHCP-Discover

(2) 提供 IP 租用地址。当 DHCP 服务器监听到客户端广播发送的 DHCP 发现报文后，它会从那些还没有租出的地址范围内选择最前面的空置 IP 地址，连同其他 TCP/IP 协议设定，应答给 DHCP 客户端一个 DHCP 提供 (DHCP Offer) 报文。由于客户端在开始的时候还没有 IP 地址，所以在其提供报文中会带有请求 DHCP 客户端的 MAC 地址信息。根据服务器端的设定，提供报文中还会包含一个租约期限的信息。



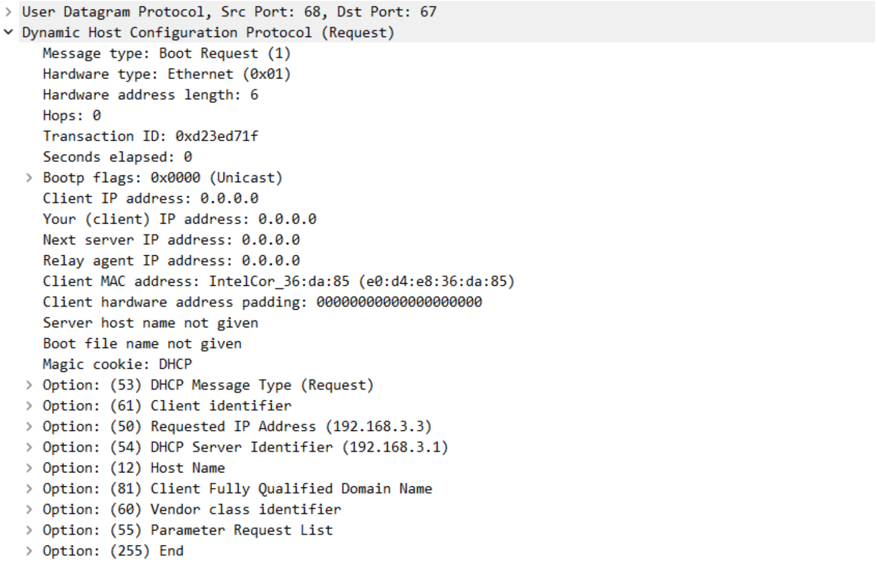
```
> User Datagram Protocol, Src Port: 67, Dst Port: 68
v Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xd23ed71f
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.3.3
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Offer)
  > Option: (54) DHCP Server Identifier (192.168.3.1)
  > Option: (51) IP Address Lease Time
  > Option: (58) Renewal Time Value
  > Option: (59) Rebinding Time Value
  > Option: (1) Subnet Mask (255.255.255.0)
  > Option: (3) Router
  > Option: (6) Domain Name Server
  > Option: (255) End
```

图 8: DHCP-Offer

(3) 接收 IP 租约。如果 DHCP 客户端受到网络多台 DHCP 服务器的提供报文，只会挑选并接受其中的一个提供报文 (通常是最先抵达的那个)，并且会广播发送一个 DHCP 请求报文，报文中包含选中的 DHCP 服务器的 IP 地址和需要的 IP 地址。

同时，客户端还会向网络发送一个 ARP 报文，查询网络上是否有其他机器使用该 IP 地址；如果发现该 IP 地址已经被占用，客户端则会发送一个 DHCP Decline 报文给 DHCP 服务器，拒绝接收其 DHCP 提供报文，并重新广播发送 DHCP 发现报文。(这里 DHCP 提供的 IP 地址并不发生冲突，所

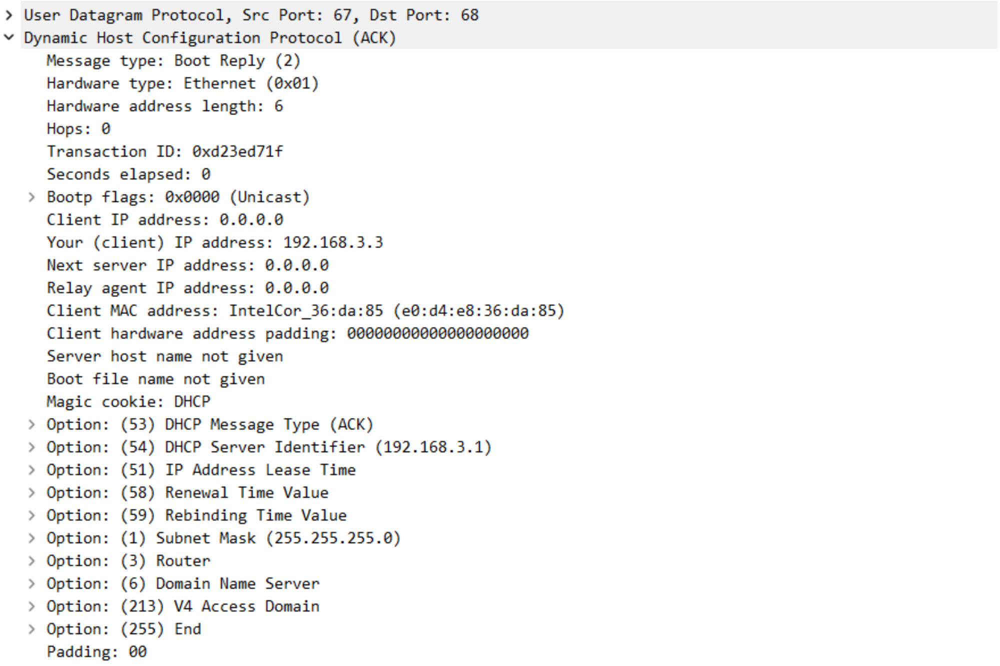
以没有出现该种包)



```
> User Datagram Protocol, Src Port: 68, Dst Port: 67
v Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xd23ed71f
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Request)
  > Option: (61) Client identifier
  > Option: (50) Requested IP Address (192.168.3.3)
  > Option: (54) DHCP Server Identifier (192.168.3.1)
  > Option: (12) Host Name
  > Option: (81) Client Fully Qualified Domain Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  > Option: (255) End
```

图 9: DHCP-Request

(4) 租约确认。当 DHCP 服务器接收到客户端的 DHCP 请求报文后，判断报文中 IP 地址是否与自己的地址相同。如果不相同，则 DHCP 服务器不做任何处理，只清除相应 IP 地址分配记录; 如果相同，DHCP 服务器就会向 DHCP 客户端响应一个 DHCP ACK 报文，并在选项字段中增加 IP 地址的使用租期信息，以确认 IP 的租约正式生效，也就结束了一个完整的 DHCP 工作过程。



```
> User Datagram Protocol, Src Port: 67, Dst Port: 68
v Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xd23ed71f
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.3.3
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (ACK)
  > Option: (54) DHCP Server Identifier (192.168.3.1)
  > Option: (51) IP Address Lease Time
  > Option: (58) Renewal Time Value
  > Option: (59) Rebinding Time Value
  > Option: (1) Subnet Mask (255.255.255.0)
  > Option: (3) Router
  > Option: (6) Domain Name Server
  > Option: (213) V4 Access Domain
  > Option: (255) End
    Padding: 00
```

图 10: DHCP-ACK

完整的工作图示如下图所示：

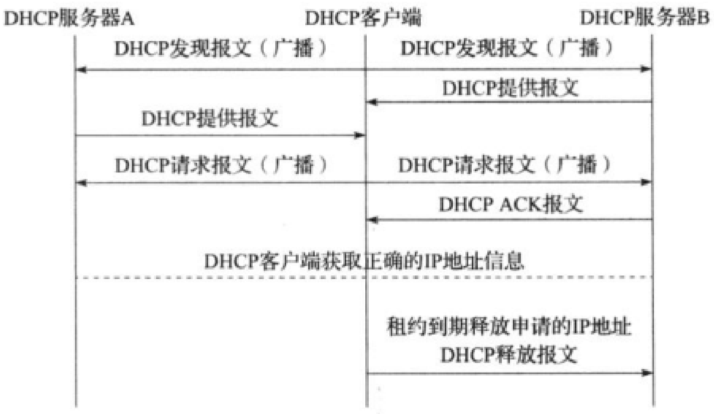
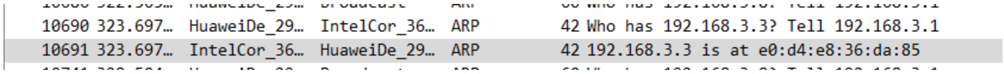


图 11: DHCP 的基本工作工程

2. ARP

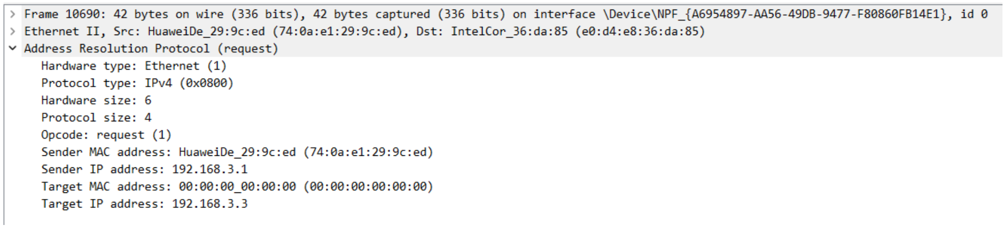
开启 filter:arp，发现抓到的包分为两种类型。



```
10690 323.697… HuaweiDe_29… IntelCor_36… ARP       42 Who has 192.168.3.3? Tell 192.168.3.1
10691 323.697… IntelCor_36… HuaweiDe_29… ARP       42 192.168.3.3 is at e0:d4:e8:36:da:85
```

图 12: 抓到的 ARP 包

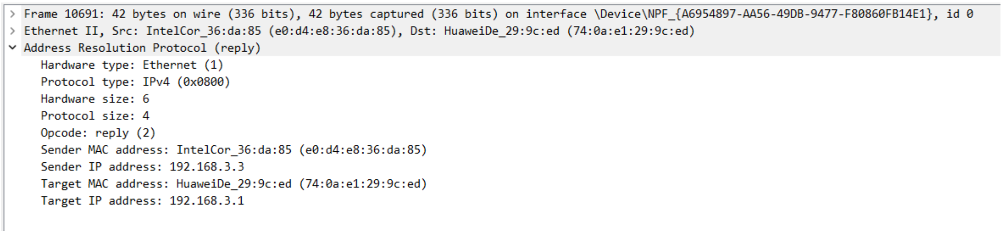
(1) 发送 ARP 请求广播：设备会发送一个广播的 ARP 请求，请求目标设备的 MAC 地址。这个请求包含了目标设备的 IP 地址。



```
> Frame 10690: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{A6954897-AA56-49DB-9477-F80860FB14E1}, id 0
> Ethernet II, Src: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed), Dst: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed)
    Sender IP address: 192.168.3.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.3.3
```

图 13: ARP-Request

(2)ARP 响应包被发送回到源设备：源设备接收到 ARP 响应包后，将目标设备的 MAC 地址存储到其 ARP 缓存中。

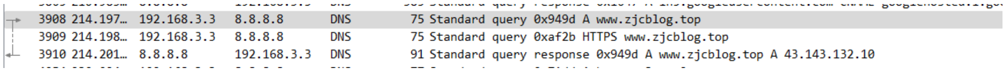


```
> Frame 10691: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{A6954897-AA56-49DB-9477-F80860FB14E1}, id 0
> Ethernet II, Src: IntelCor_36:da:85 (e0:d4:e8:36:da:85), Dst: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
    Sender IP address: 192.168.3.3
    Target MAC address: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed)
    Target IP address: 192.168.3.1
```

图 14: ARP-Reply

3. DNS

从浏览器输入 www.zjcblog.top, 并开启 filter:arp，发现抓到三个包。

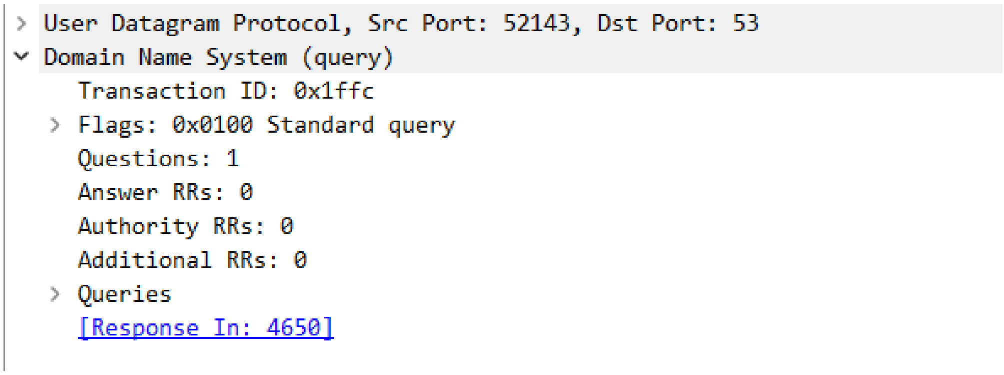


```
3908 214.197…  192.168.3.3  8.8.8.8      DNS   75 Standard query 0x949d A www.zjcblog.top
3909 214.198…  192.168.3.3  8.8.8.8      DNS   75 Standard query 0xaf2b HTTPS www.zjcblog.top
3910 214.201…  8.8.8.8      192.168.3.3  DNS   91 Standard query response 0x949d A www.zjcblog.top A 43.143.132.10
```

图 15: 抓到的 DNS 包

(1)DNS-Query-A

DNS 的 A 类型报文即建立一个域名到 IP 地址的映射,这里是在询问 DNS 服务器关于 www.zjcblog.top 的 A 映射。

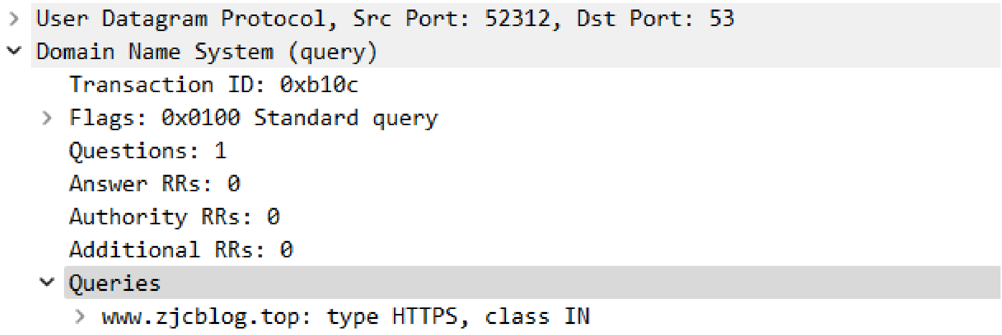


```
> User Datagram Protocol, Src Port: 52143, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x1ffc
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
    [Response In: 4650]
```

图 16: DNS-Query-A

(2)DNS-Query-HTTPS

DNS 的 HTTPS 报文，即在询问该网站是否有 HTTPS 服务，如果有则导向 HTTPS 服务器连接。(个人服务器并没有开启 HTTPS 服务)



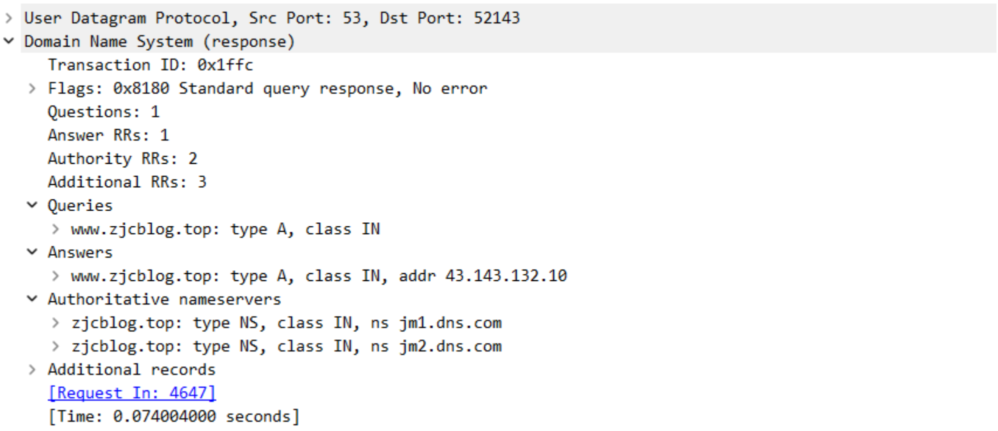
```
> User Datagram Protocol, Src Port: 52312, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0xb10c
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    > www.zjcblog.top: type HTTPS, class IN
```

图 17: DNS-Query-HTTPS

(2)DNS-Response-A

DNS 服务器这时候响应主机的请求，返回关于 www.zjcblog.top 的 A 映射，然而对于谷歌服务器来说则会同时返回服务器的认证服务器。

对于 DNS 服务器查询来说其实还分为递归查询和迭代查询，递归查询则不会返回目的 dns 服务器的地址，而迭代则会，同时让主机查询目的 DNS 服务器，这里则返回了目的 DNS 服务器。

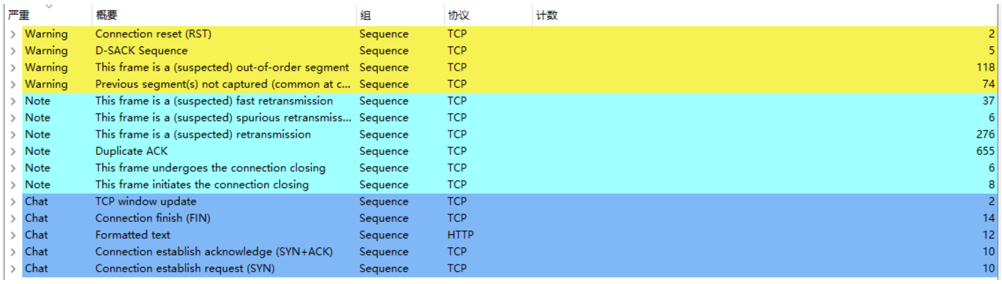


```
> User Datagram Protocol, Src Port: 53, Dst Port: 52143
v Domain Name System (response)
    Transaction ID: 0x1ffc
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 2
    Additional RRs: 3
  v Queries
    > www.zjcblog.top: type A, class IN
  v Answers
    > www.zjcblog.top: type A, class IN, addr 43.143.132.10
  v Authoritative nameservers
    > zjcblog.top: type NS, class IN, ns jm1.dns.com
    > zjcblog.top: type NS, class IN, ns jm2.dns.com
  > Additional records
    [Request In: 4647]
    [Time: 0.074004000 seconds]
```

图 18: DNS-Response-A

4. TCP

同时在 www.zjcblog.top 中开始使用接口 (上传、下载、播放视频) 并抓包，并开启 filter:ip.addr == 43.143.132.10 and tcp：

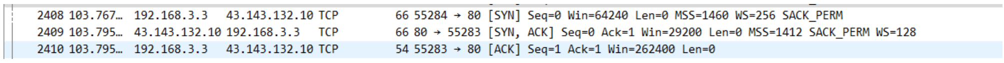


严重	概要	组	协议	计数
Warning	Connection reset (RST)	Sequence	TCP	2
Warning	D-SACK Sequence	Sequence	TCP	5
Warning	This frame is a (suspected) out-of-order segment	Sequence	TCP	118
Warning	Previous segment(s) not captured (common at c...	Sequence	TCP	74
Note	This frame is a (suspected) fast retransmission	Sequence	TCP	37
Note	This frame is a (suspected) spurious retransmiss...	Sequence	TCP	6
Note	This frame is a (suspected) retransmission	Sequence	TCP	276
Note	Duplicate ACK	Sequence	TCP	655
Note	This frame undergoes the connection closing	Sequence	TCP	6
Note	This frame initiates the connection closing	Sequence	TCP	8
Chat	TCP window update	Sequence	TCP	2
Chat	Connection finish (FIN)	Sequence	TCP	14
Chat	Formatted text	Sequence	HTTP	12
Chat	Connection establish acknowledge (SYN+ACK)	Sequence	TCP	10
Chat	Connection establish request (SYN)	Sequence	TCP	10

图 19: TCP 统计图

可以看出其中传送的 TCP 包非常多，这里只对部分包进行介绍。

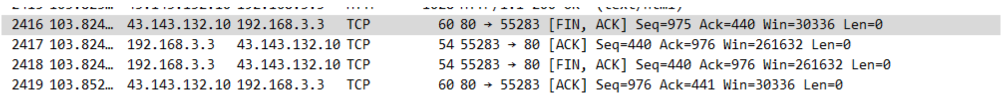
(1) 三次握手过程



```
2408 103.767… 192.168.3.3   43.143.132.10 TCP    66 55284 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2409 103.795… 43.143.132.10 192.168.3.3   TCP    66 80 → 55283 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM WS=128
2410 103.795… 192.168.3.3   43.143.132.10 TCP    54 55283 → 80 [ACK] Seq=1 Ack=1 Win=262400 Len=0
```

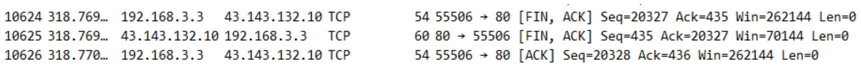
图 20: 三次握手过程

(2) 挥手过程



```
2416 103.824… 43.143.132.10 192.168.3.3   TCP    60 80 → 55283 [FIN, ACK] Seq=975 Ack=440 Win=30336 Len=0
2417 103.824… 192.168.3.3   43.143.132.10 TCP    54 55283 → 80 [ACK] Seq=440 Ack=976 Win=261632 Len=0
2418 103.824… 192.168.3.3   43.143.132.10 TCP    54 55283 → 80 [FIN, ACK] Seq=440 Ack=976 Win=261632 Len=0
2419 103.852… 43.143.132.10 192.168.3.3   TCP    60 80 → 55283 [ACK] Seq=976 Ack=441 Win=30336 Len=0
```

图 21: 四次挥手过程



```
10624 318.769…  192.168.3.3   43.143.132.10 TCP        54 55506 → 80 [FIN, ACK] Seq=20327 Ack=435 Win=262144 Len=0
10625 318.769…  43.143.132.10 192.168.3.3   TCP        60 80 → 55506 [FIN, ACK] Seq=435 Ack=20327 Win=70144 Len=0
10626 318.770…  192.168.3.3   43.143.132.10 TCP        54 55506 → 80 [ACK] Seq=20328 Ack=436 Win=262144 Len=0
```

图 22: 三次挥手过程

(3) 流量控制

在下载文件时，发现 TCP 传送过程中的窗口尺寸变化如下, 其中窗口尺寸不断变化，进行流量控制：

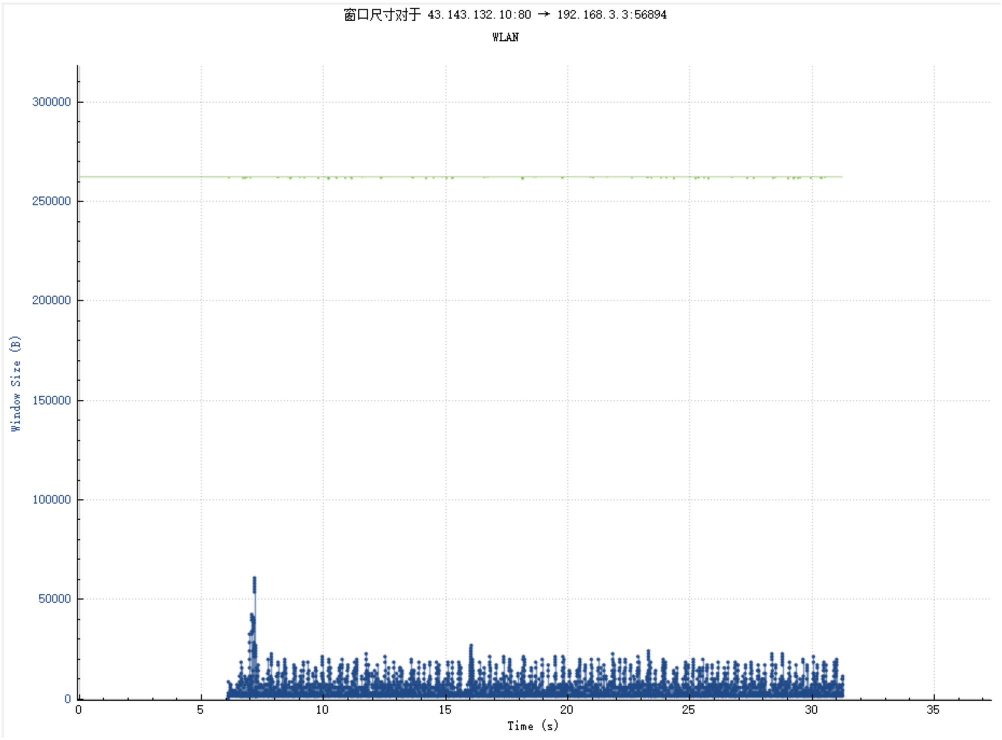
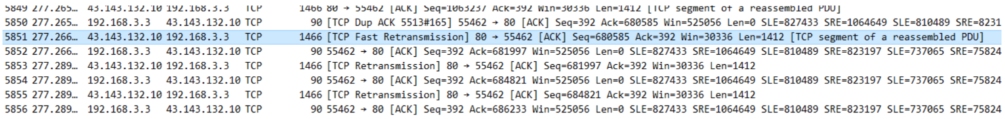


图 23: 窗口尺寸变化图

(4) 快速重传

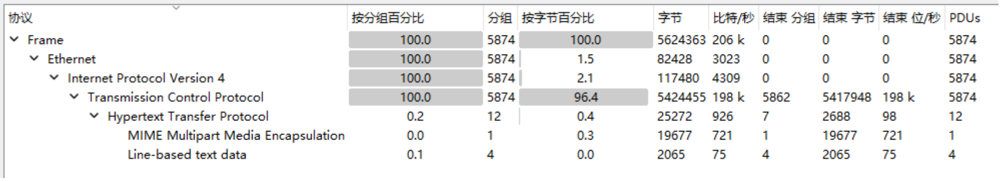
使用快速重传进行拥塞控制。



```
5849 277.265…  43.143.132.10 192.168.3.3   TCP      1466 80 → 55462 [ACK] Seq=1063237 Ack=392 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
5850 277.265…  192.168.3.3   43.143.132.10 TCP        90 [TCP Dup ACK 5513#165] 55462 → 80 [ACK] Seq=392 Ack=680585 Win=525056 Len=0 SLE=827433 SRE=1064649 SLE=810489 SRE=8231
5851 277.266…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Fast Retransmission] 80 → 55462 [ACK] Seq=680585 Ack=392 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
5852 277.266…  192.168.3.3   43.143.132.10 TCP        90 55462 → 80 [ACK] Seq=392 Ack=681997 Win=525056 Len=0 SLE=827433 SRE=1064649 SLE=810489 SRE=823197 SLE=737065 SRE=75824
5853 277.289…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Retransmission] 80 → 55462 [ACK] Seq=681997 Ack=392 Win=30336 Len=1412
5854 277.289…  192.168.3.3   43.143.132.10 TCP        90 55462 → 80 [ACK] Seq=392 Ack=684821 Win=525056 Len=0 SLE=827433 SRE=1064649 SLE=810489 SRE=823197 SLE=737065 SRE=75824
5855 277.289…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Retransmission] 80 → 55462 [ACK] Seq=684821 Ack=392 Win=30336 Len=1412
5856 277.289…  192.168.3.3   43.143.132.10 TCP        90 55462 → 80 [ACK] Seq=392 Ack=686233 Win=525056 Len=0 SLE=827433 SRE=1064649 SLE=810489 SRE=823197 SLE=737065 SRE=75824
```

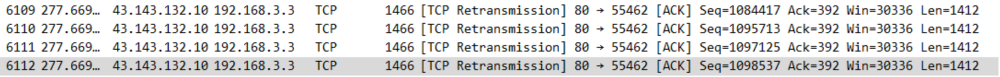
图 24: 快速重传

(5) 重传



协议	按分组百分比	分组	按字节百分比	字节	比特/秒	结束 分组	结束 字节	结束 位/秒	PDUs
⌄ Frame	100.0	5874	100.0	5624363	206 k	0	0	0	5874
⌄ Ethernet	100.0	5874	1.5	82428	3023	0	0	0	5874
⌄ Internet Protocol Version 4	100.0	5874	2.1	117480	4309	0	0	0	5874
⌄ Transmission Control Protocol	100.0	5874	96.4	5424455	198 k	5862	5417948	198 k	5874
⌄ Hypertext Transfer Protocol	0.2	12	0.4	25272	926	7	2688	98	12
MIME Multipart Media Encapsulation	0.0	1	0.3	19677	721	1	19677	721	1
Line-based text data	0.1	4	0.0	2065	75	4	2065	75	4

图 25: TCP 总数



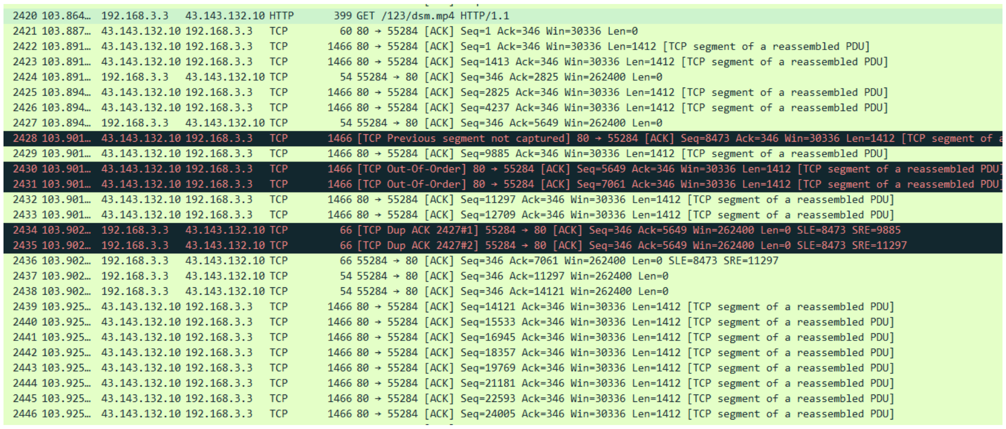
```
6109 277.669…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Retransmission] 80 → 55462 [ACK] Seq=1084417 Ack=392 Win=30336 Len=1412
6110 277.669…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Retransmission] 80 → 55462 [ACK] Seq=1095713 Ack=392 Win=30336 Len=1412
6111 277.669…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Retransmission] 80 → 55462 [ACK] Seq=1097125 Ack=392 Win=30336 Len=1412
6112 277.669…  43.143.132.10 192.168.3.3   TCP      1466 [TCP Retransmission] 80 → 55462 [ACK] Seq=1098537 Ack=392 Win=30336 Len=1412
```

图 26: 重传

可以计算出重传率为 4.7%

(6) 播放视频

在捕获 udp 包时播放了网站的 mp4 视频，发现没有捕捉到 udp，而是仍使用的 HTTP 即 TCP，说明 TCP 也可以偶尔传输视频信息等等。这里也猜测是网页代码编写问题，导致播放视频本质是请求文件导致不是流媒体播放，则不是 UDP。

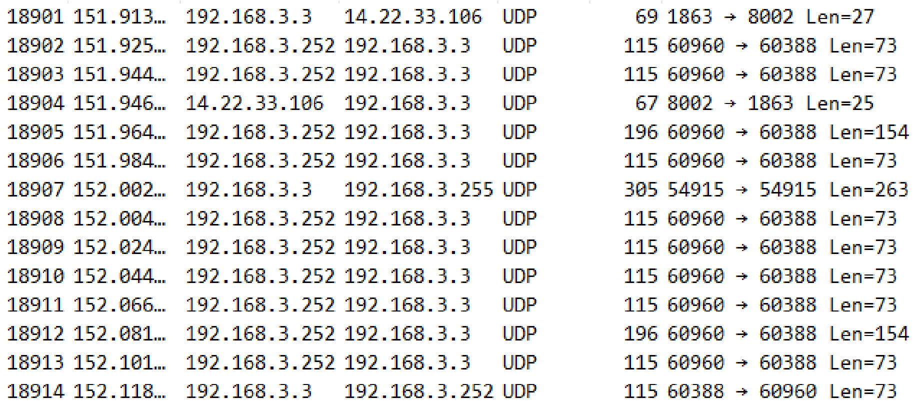


```
2420 103.864…  192.168.3.3   43.143.132.10 HTTP   399 GET /123/dsm.mp4 HTTP/1.1
2421 103.887…  43.143.132.10 192.168.3.3   TCP     60 80 → 55284 [ACK] Seq=1 Ack=346 Win=30336 Len=0
2422 103.891…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=1 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2423 103.891…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=1413 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2424 103.891…  192.168.3.3   43.143.132.10 TCP     54 55284 → 80 [ACK] Seq=346 Ack=2825 Win=262400 Len=0
2425 103.894…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=2825 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2426 103.894…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=4237 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2427 103.894…  192.168.3.3   43.143.132.10 TCP     54 55284 → 80 [ACK] Seq=346 Ack=5649 Win=262400 Len=0
2428 103.901…  43.143.132.10 192.168.3.3   TCP   1466 [TCP Previous segment not captured] 80 → 55284 [ACK] Seq=8473 Ack=346 Win=30336 Len=1412 [TCP segment of a
2429 103.901…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=9885 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2430 103.901…  43.143.132.10 192.168.3.3   TCP   1466 [TCP Out-Of-Order] 80 → 55284 [ACK] Seq=5649 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2431 103.901…  43.143.132.10 192.168.3.3   TCP   1466 [TCP Out-Of-Order] 80 → 55284 [ACK] Seq=7061 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2432 103.901…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=11297 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2433 103.901…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=12709 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2434 103.902…  192.168.3.3   43.143.132.10 TCP     66 [TCP Dup ACK 2427#1] 55284 → 80 [ACK] Seq=346 Ack=5649 Win=262400 Len=0 SLE=8473 SRE=9885
2435 103.902…  192.168.3.3   43.143.132.10 TCP     66 [TCP Dup ACK 2427#2] 55284 → 80 [ACK] Seq=346 Ack=5649 Win=262400 Len=0 SLE=8473 SRE=11297
2436 103.902…  192.168.3.3   43.143.132.10 TCP     66 55284 → 80 [ACK] Seq=346 Ack=7061 Win=262400 Len=0 SLE=8473 SRE=11297
2437 103.902…  192.168.3.3   43.143.132.10 TCP     54 55284 → 80 [ACK] Seq=346 Ack=11297 Win=262400 Len=0
2438 103.902…  192.168.3.3   43.143.132.10 TCP     54 55284 → 80 [ACK] Seq=346 Ack=14121 Win=262400 Len=0
2439 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=14121 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2440 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=15533 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2441 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=16945 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2442 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=18357 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2443 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=19769 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2444 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=21181 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2445 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=22593 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
2446 103.925…  43.143.132.10 192.168.3.3   TCP   1466 80 → 55284 [ACK] Seq=24005 Ack=346 Win=30336 Len=1412 [TCP segment of a reassembled PDU]
```

图 27: TCP 播放视频

5. UDP

这里通过 QQ 与宿舍好友开启语音通话，开启 UDP 传输：



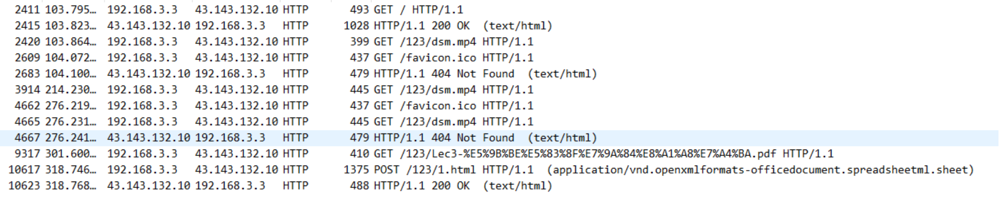
```
18901 151.913… 192.168.3.3   14.22.33.106  UDP         69 1863 → 8002 Len=27
18902 151.925… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18903 151.944… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18904 151.946… 14.22.33.106  192.168.3.3   UDP         67 8002 → 1863 Len=25
18905 151.964… 192.168.3.252 192.168.3.3   UDP        196 60960 → 60388 Len=154
18906 151.984… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18907 152.002… 192.168.3.3   192.168.3.255 UDP        305 54915 → 54915 Len=263
18908 152.004… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18909 152.024… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18910 152.044… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18911 152.066… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18912 152.081… 192.168.3.252 192.168.3.3   UDP        196 60960 → 60388 Len=154
18913 152.101… 192.168.3.252 192.168.3.3   UDP        115 60960 → 60388 Len=73
18914 152.118… 192.168.3.3   192.168.3.252 UDP        115 60388 → 60960 Len=73
```

图 28: 语音通话中的 UDP

同时之前的 DNS、DHCP 其实都是 UDP 包，主要传输控制信息，且数据长度低，适合无连接的 UDP 传输。

6. HTTP

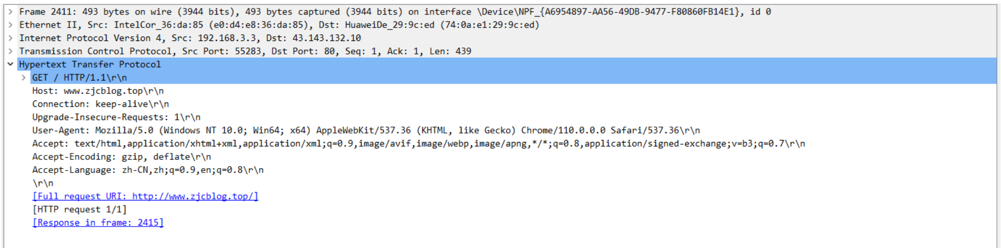
同时在 www.zjcblog.top 中开始使用接口 (上传、下载、播放视频) 并抓包，并开启 filter:ip.addr == 43.143.132.10 and http：



```
 2411 103.795… 192.168.3.3   43.143.132.10 HTTP      493 GET / HTTP/1.1
 2415 103.823… 43.143.132.10 192.168.3.3   HTTP     1028 HTTP/1.1 200 OK  (text/html)
 2420 103.864… 192.168.3.3   43.143.132.10 HTTP      399 GET /123/dsm.mp4 HTTP/1.1
 2609 104.072… 192.168.3.3   43.143.132.10 HTTP      437 GET /favicon.ico HTTP/1.1
 2683 104.100… 43.143.132.10 192.168.3.3   HTTP      479 HTTP/1.1 404 Not Found  (text/html)
 3914 214.230… 192.168.3.3   43.143.132.10 HTTP      445 GET /123/dsm.mp4 HTTP/1.1
 4662 276.219… 192.168.3.3   43.143.132.10 HTTP      437 GET /favicon.ico HTTP/1.1
 4665 276.231… 192.168.3.3   43.143.132.10 HTTP      445 GET /123/dsm.mp4 HTTP/1.1
 4667 276.241… 43.143.132.10 192.168.3.3   HTTP      479 HTTP/1.1 404 Not Found  (text/html)
 9317 301.600… 192.168.3.3   43.143.132.10 HTTP      410 GET /123/Lec3-%E5%9B%BE%E5%83%8F%E7%9A%84%E8%A1%A8%E7%A4%BA.pdf HTTP/1.1
10617 318.746… 192.168.3.3   43.143.132.10 HTTP     1375 POST /123/1.html HTTP/1.1  (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet)
10623 318.768… 43.143.132.10 192.168.3.3   HTTP      488 HTTP/1.1 200 OK  (text/html)
```

图 29: 抓到的 HTTP 包

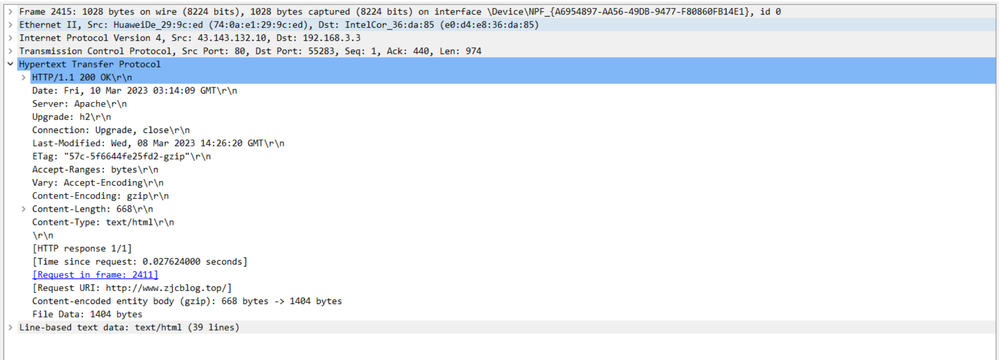
(1) 主机发出 GET，期望与网页进行访问连接。



```
> Frame 2411: 493 bytes on wire (3944 bits), 493 bytes captured (3944 bits) on interface \Device\NPF_{A6954897-AA56-49DB-9477-F80860FB14E1}, id 0
> Ethernet II, Src: IntelCor_36:da:85 (e0:d4:e8:36:da:85), Dst: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed)
> Internet Protocol Version 4, Src: 192.168.3.3, Dst: 43.143.132.10
> Transmission Control Protocol, Src Port: 55283, Dst Port: 80, Seq: 1, Ack: 1, Len: 439
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.zjcblog.top\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
    \r\n
    [Full request URI: http://www.zjcblog.top/]
    [HTTP request 1/1]
    [Response in frame: 2415]
```

图 30: HTTP-GET

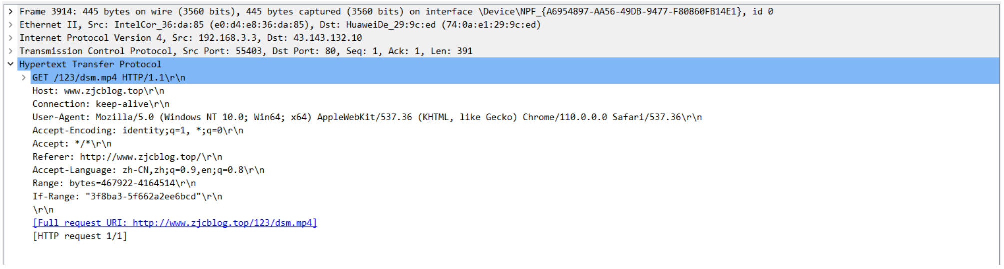
(2) 服务器回应，状态码 200 代表成功访问连接。



```
> Frame 2415: 1028 bytes on wire (8224 bits), 1028 bytes captured (8224 bits) on interface \Device\NPF_{A6954897-AA56-49DB-9477-F80860FB14E1}, id 0
> Ethernet II, Src: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed), Dst: IntelCor_36:da:85 (e0:d4:e8:36:da:85)
> Internet Protocol Version 4, Src: 43.143.132.10, Dst: 192.168.3.3
> Transmission Control Protocol, Src Port: 80, Dst Port: 55283, Seq: 1, Ack: 440, Len: 974
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Fri, 10 Mar 2023 03:14:09 GMT\r\n
    Server: Apache\r\n
    Upgrade: h2\r\n
    Connection: Upgrade, close\r\n
    Last-Modified: Wed, 08 Mar 2023 14:26:20 GMT\r\n
    ETag: "57c-5f6644fe25fd2-gzip"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
  > Content-Length: 668\r\n
    Content-Type: text/html\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.027624000 seconds]
    [Request in frame: 2411]
    [Request URI: http://www.zjcblog.top/]
    Content-encoded entity body (gzip): 668 bytes -> 1404 bytes
    File Data: 1404 bytes
> Line-based text data: text/html (39 lines)
```

图 31: HTTP-200

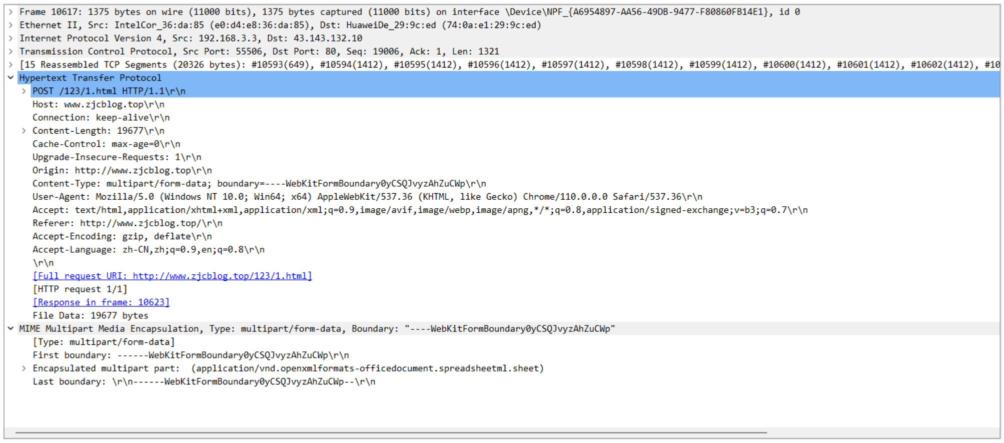
(3) 主机向服务器请求网页文件，需要使用 GET，比如如下图的视频文件。



```
> Frame 3914: 445 bytes on wire (3560 bits), 445 bytes captured (3560 bits) on interface \Device\NPF_{A6954897-AA56-49DB-9477-F80860FB14E1}, id 0
> Ethernet II, Src: IntelCor_36:da:85 (e0:d4:e8:36:da:85), Dst: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed)
> Internet Protocol Version 4, Src: 192.168.3.3, Dst: 43.143.132.10
> Transmission Control Protocol, Src Port: 55403, Dst Port: 80, Seq: 1, Ack: 1, Len: 391
v Hypertext Transfer Protocol
  > GET /123/dsm.mp4 HTTP/1.1\r\n
    Host: www.zjcblog.top\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36\r\n
    Accept-Encoding: identity;q=1, *;q=0\r\n
    Accept: */*\r\n
    Referer: http://www.zjcblog.top/\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
    Range: bytes=467922-4164514\r\n
    If-Range: "3f8ba3-5f662a2ee6bcd"\r\n
    \r\n
    [Full request URI: http://www.zjcblog.top/123/dsm.mp4]
    [HTTP request 1/1]
```

图 32: HTTP-GET

(4) 主机上传文件时，需要使用 POST，这里上传的是一张图片，于是会再封装一层 MIME 协议，用于 HTTP 的多媒体传输服务：



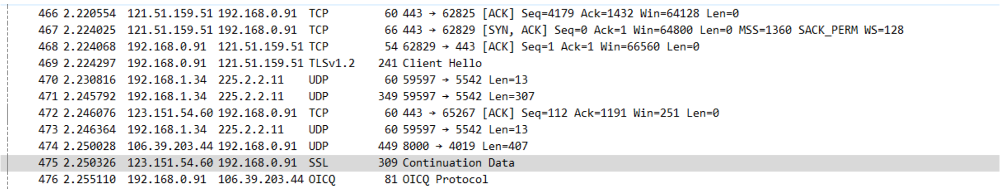
```
> Frame 10617: 1375 bytes on wire (11000 bits), 1375 bytes captured (11000 bits) on interface \Device\NPF_{A6954897-AA56-49DB-9477-F80860FB14E1}, id 0
> Ethernet II, Src: IntelCor_36:da:85 (e0:d4:e8:36:da:85), Dst: HuaweiDe_29:9c:ed (74:0a:e1:29:9c:ed)
> Internet Protocol Version 4, Src: 192.168.3.3, Dst: 43.143.132.10
> Transmission Control Protocol, Src Port: 55506, Dst Port: 80, Seq: 19006, Ack: 1, Len: 1321
> [15 Reassembled TCP Segments (20326 bytes): #10593(649), #10594(1412), #10595(1412), #10596(1412), #10597(1412), #10598(1412), #10599(1412), #10600(1412), #10601(1412), #10602(1412), #10
v Hypertext Transfer Protocol
  > POST /123/1.html HTTP/1.1\r\n
    Host: www.zjcblog.top\r\n
    Connection: keep-alive\r\n
  > Content-Length: 19677\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Origin: http://www.zjcblog.top\r\n
    Content-Type: multipart/form-data; boundary=----WebKitFormBoundary0yCSQJvyzAhZuCWp\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Referer: http://www.zjcblog.top/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8\r\n
    \r\n
    [Full request URI: http://www.zjcblog.top/123/1.html]
    [HTTP request 1/1]
    [Response in frame: 10623]
    File Data: 19677 bytes
v MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "----WebKitFormBoundary0yCSQJvyzAhZuCWp"
    [Type: multipart/form-data]
    First boundary: ------WebKitFormBoundary0yCSQJvyzAhZuCWp\r\n
  > Encapsulated multipart part:  (application/vnd.openxmlformats-officedocument.spreadsheetml.sheet)
    Last boundary: \r\n------WebKitFormBoundary0yCSQJvyzAhZuCWp--\r\n
```

图 33: HTTP-POST

7. OICQ

在登录 QQ 的时候，发现了 OICQ 协议，现对其进行抓包分析:

以下是刚打开 QQ 登录时抓到的包，TLSv1.2 与 SSL 都是保证传输层安全的协议，其他数据包则主要用于建立起客户端到 QQ 服务器的连接，三次握手等等。

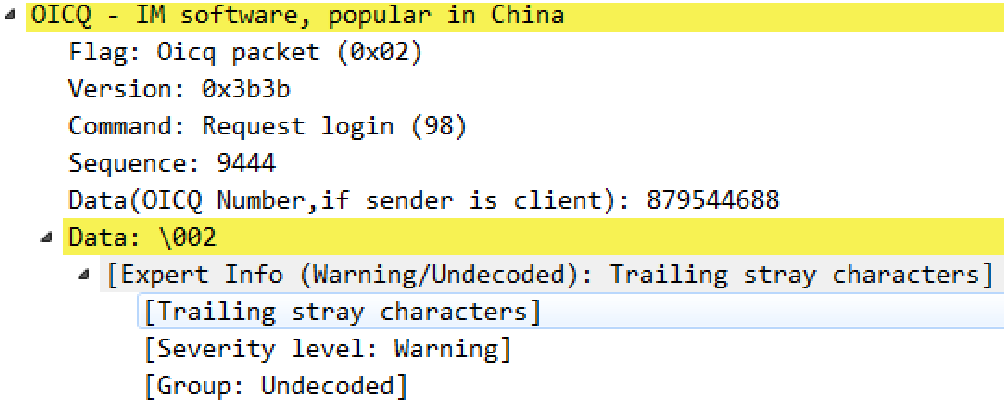


```
466 2.220554  121.51.159.51 192.168.0.91  TCP       60 443 → 62825 [ACK] Seq=4179 Ack=1432 Win=64128 Len=0
467 2.224025  121.51.159.51 192.168.0.91  TCP       66 443 → 62829 [SYN, ACK] Seq=0 Ack=1 Win=64800 Len=0 MSS=1360 SACK_PERM WS=128
468 2.224068  192.168.0.91  121.51.159.51 TCP       54 62829 → 443 [ACK] Seq=1 Ack=1 Win=66560 Len=0
469 2.224297  192.168.0.91  121.51.159.51 TLSv1.2  241 Client Hello
470 2.230816  192.168.1.34  225.2.2.11    UDP       60 59597 → 5542 Len=13
471 2.245792  192.168.1.34  225.2.2.11    UDP      349 59597 → 5542 Len=307
472 2.246076  123.151.54.60 192.168.0.91  TCP       60 443 → 65267 [ACK] Seq=112 Ack=1191 Win=251 Len=0
473 2.246364  192.168.1.34  225.2.2.11    UDP       60 59597 → 5542 Len=13
474 2.250028  106.39.203.44 192.168.0.91  UDP      449 8000 → 4019 Len=407
475 2.250326  123.151.54.60 192.168.0.91  SSL      309 Continuation Data
476 2.255110  192.168.0.91  106.39.203.44 OICQ      81 OICQ Protocol
```

图 34: OICQ-Before

(1) 登录

然而值得注意的是，这实际上是关闭 QQ 即账号下线抓到的，语义相反。

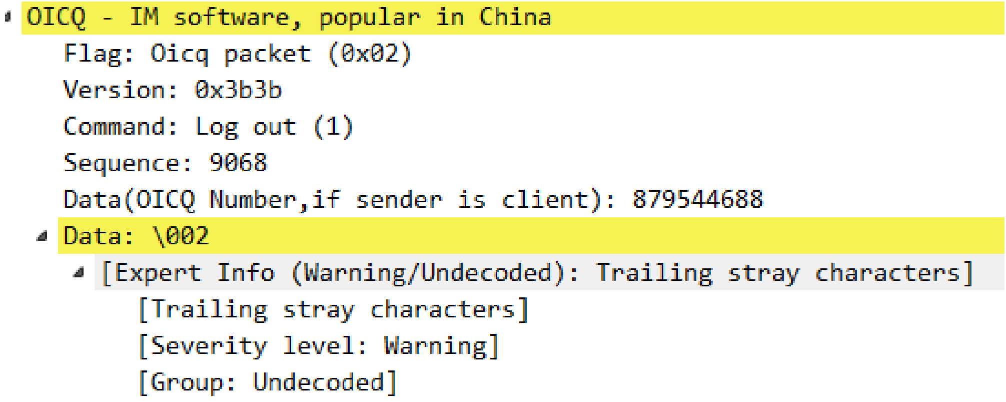


```
◢ OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Request login (98)
    Sequence: 9444
    Data(OICQ Number,if sender is client): 879544688
  ◢ Data: \002
    ◢ [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 35: OICQ-Login

(2) 登出

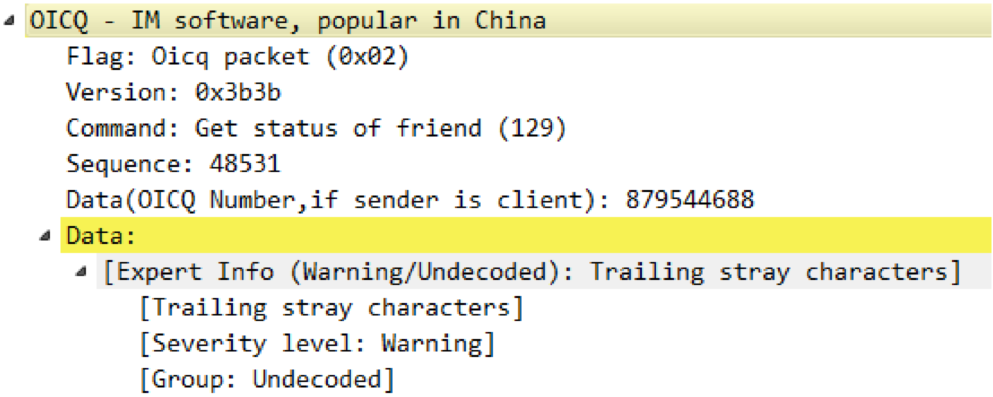
然而值得注意的是，这实际上是登录 QQ 时候抓到的，语义相反。



```
◢ OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Log out (1)
    Sequence: 9068
    Data(OICQ Number,if sender is client): 879544688
  ◢ Data: \002
    ◢ [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 36: OICQ-Log out

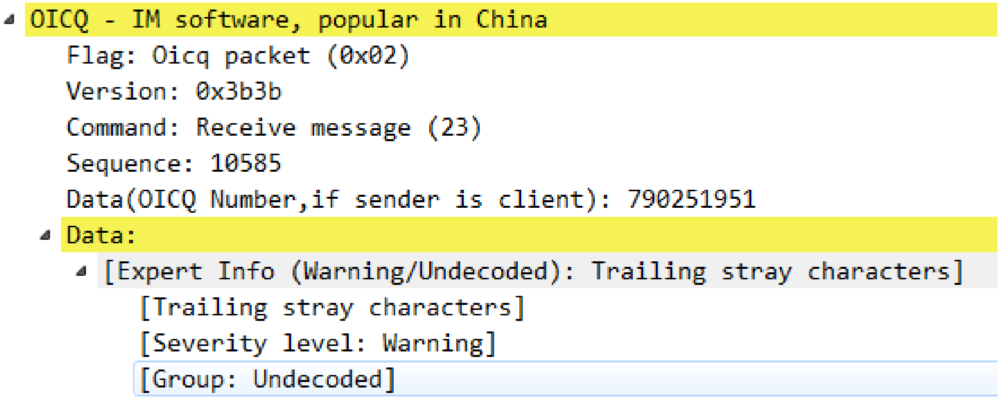
(3) 获得好友状态



```
OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Get status of friend (129)
    Sequence: 48531
    Data(OICQ Number,if sender is client): 879544688
  Data:
    [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 37: OICQ-Get status of friend

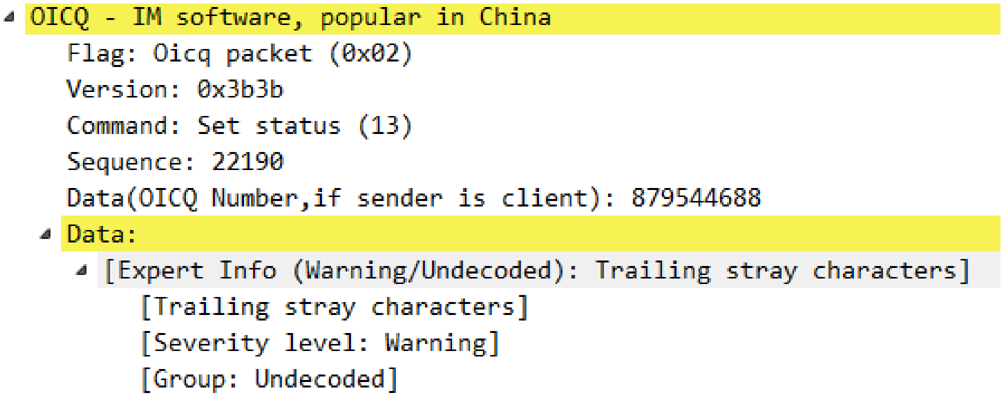
(4) 接收消息



```
OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Receive message (23)
    Sequence: 10585
    Data(OICQ Number,if sender is client): 790251951
  Data:
    [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 38: OICQ-Receive message

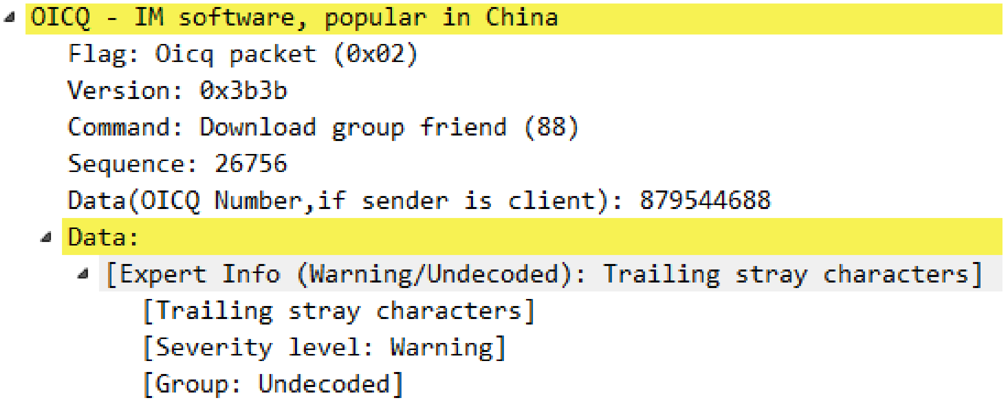
(5) 设置账号状态



```
OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Set status (13)
    Sequence: 22190
    Data(OICQ Number,if sender is client): 879544688
  Data:
    [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 39: OICQ-Set status

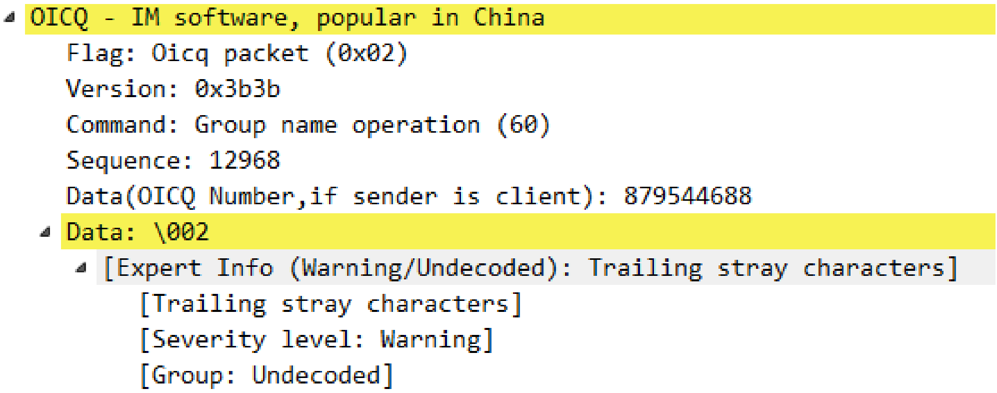
(6) 下载朋友列表



```
OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Download group friend (88)
    Sequence: 26756
    Data(OICQ Number,if sender is client): 879544688
  Data:
    [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 40: OICQ-Download group friend

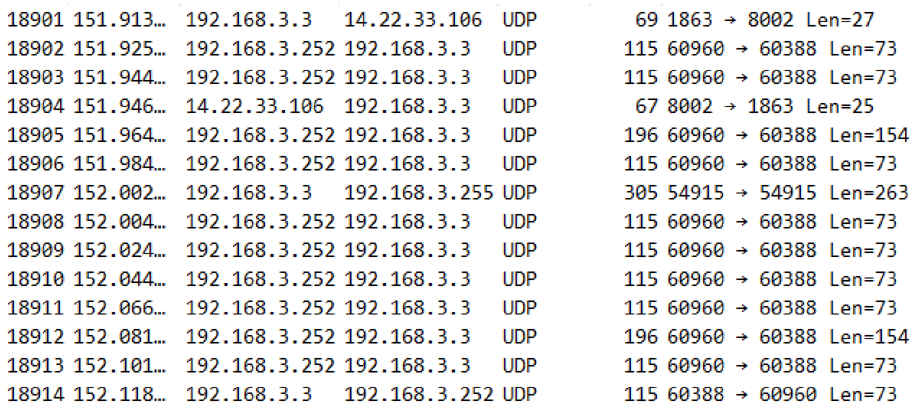
(7) 组名操作



```
OICQ - IM software, popular in China
    Flag: Oicq packet (0x02)
    Version: 0x3b3b
    Command: Group name operation (60)
    Sequence: 12968
    Data(OICQ Number,if sender is client): 879544688
  Data: \002
    [Expert Info (Warning/Undecoded): Trailing stray characters]
        [Trailing stray characters]
        [Severity level: Warning]
        [Group: Undecoded]
```

图 41: OICQ-Group name operation

下图是与舍友 QQ 语音通话的报文截获图，14.22.33.106 为腾讯服务器的 IP 地址，而在其中发现语音 UDP 传输从主机到服务器再到主机的模式，更改成了同一局域网中的 UDP 语音传输，这样大大提高了语音通话质量，是一种很智能的协议模式。



```
18901 151.913… 192.168.3.3   14.22.33.106  UDP    69 1863 → 8002 Len=27
18902 151.925… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18903 151.944… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18904 151.946… 14.22.33.106  192.168.3.3   UDP    67 8002 → 1863 Len=25
18905 151.964… 192.168.3.252 192.168.3.3   UDP   196 60960 → 60388 Len=154
18906 151.984… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18907 152.002… 192.168.3.3   192.168.3.255 UDP   305 54915 → 54915 Len=263
18908 152.004… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18909 152.024… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18910 152.044… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18911 152.066… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18912 152.081… 192.168.3.252 192.168.3.3   UDP   196 60960 → 60388 Len=154
18913 152.101… 192.168.3.252 192.168.3.3   UDP   115 60960 → 60388 Len=73
18914 152.118… 192.168.3.3   192.168.3.252 UDP   115 60388 → 60960 Len=73
```

图 42: 语音传输链路转移的过程

六、　实验感悟

(1) 基本了解了主机上网过程，由上述这些协议共同服务实现，其实除此之外还有 NAT 协议等等，让我对计算机网络通信又有了更深的理解。

(2) 发现在网页上播放视频，并不一定就是 UDP 传输，也有可能是 TCP 传输，一切都要从实际出发，不能光依赖于理论。

(3) 学习到了 Wireshark 抓包软件的各种用法，包括过滤器、专家信息、TCP 流图、分级协议统计等等。

(4) 简单分析了 QQ 的 OICQ 协议，发现其报文 OICQ - IM software, popular in China 很有意思，并且了解了平时对 QQ 操作的各种协议包。还发现了 OICQ 的优化链路功能，检测到两台主机为统一局域网下 (同一公网 IP)，从主机到服务器再到主机的 UDP 语音传输链路，转换成了局域网内主机到主机的 UDP 语音传输。