哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

实验名称	利用 Wireshark 进行协议分析				
姓名	王艺丹		院系	计算机	
班级	1903601		学号	1190201303	
任课教师	李全龙		指导教师	李全龙	
实验地点	G207		实验时间	2021.11.20	
实验课表现	出勤、表现得分(10)		实验报告得分(40)	实验总分	
	操作结果得分(50)				
教师评语					

计算机科学与技术学院 SINCE 1956...
School of Computer Science and Technology

实验目的：

1. 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。

实验内容：

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：

1. 利用 Wireshark 分析 DNS 协议
2. 利用 Wireshark 分析 UDP 协议
3. 利用 Wireshark 分析 ARP 协议

实验过程：

一. 学习Wireshark的使用

在初始用户界面，由于本实验在 WLAN 的环境下进行，因此选择嗅探 WLAN。

二. 设计并实现一个支持 Cache 功能的 HTTP 代理服务器

主要思想为：

为 cache分配缓存，通过MakeCache函数将数据存入缓存；

当访问某网站时，首先通过url寻找本地对应缓存文件，若无对应文件，即为第一次访问某网站，则代理服务器将该请求返回的响应数据写入缓存即相应文件中；

若匹配到本地缓存文件，利用MakeNewHTTP函数改造请求报文，即为报文增加if-modified-since头部行，再向服务器端发送请求，解析Date字段获得最网站最新更新时间；通过服务器返回的数据码判断是否为最新的数据，若返回304，则内容并未再次更新，直接将缓存中的内容发给服务器即可；若返回200，则令服务器同时返回响应报文，将此响应报文直接发给客户端，同时更新本地缓存。

三. 扩展 HTTP 代理服务器

a) 网站过滤：允许/不允许访问某些网站；

设置字符串数组存储屏蔽网站地址，对请求的 HTTP 报文头部进行解析，提取其中的访问地址 url ，并与屏蔽网站地址进行匹配，若匹配成功，则代码跳转至error 部分，打印相关提示信息，立即关闭套接字，断开连接。

b) 用户过滤：支持/不支持某些用户访问外部网站；

更改套接字绑定的主机地址， ProxyServerAddr.sin_addr.S_un.S_addr = inet_addr("127.0.0.1"); //仅本机用户可访问服务器；也可以通过数组存储禁止访问的人或者资源，实现不同的用户屏蔽。

c) 网站引导：将用户对某个网站的访问引导至一个模拟网站（钓鱼）

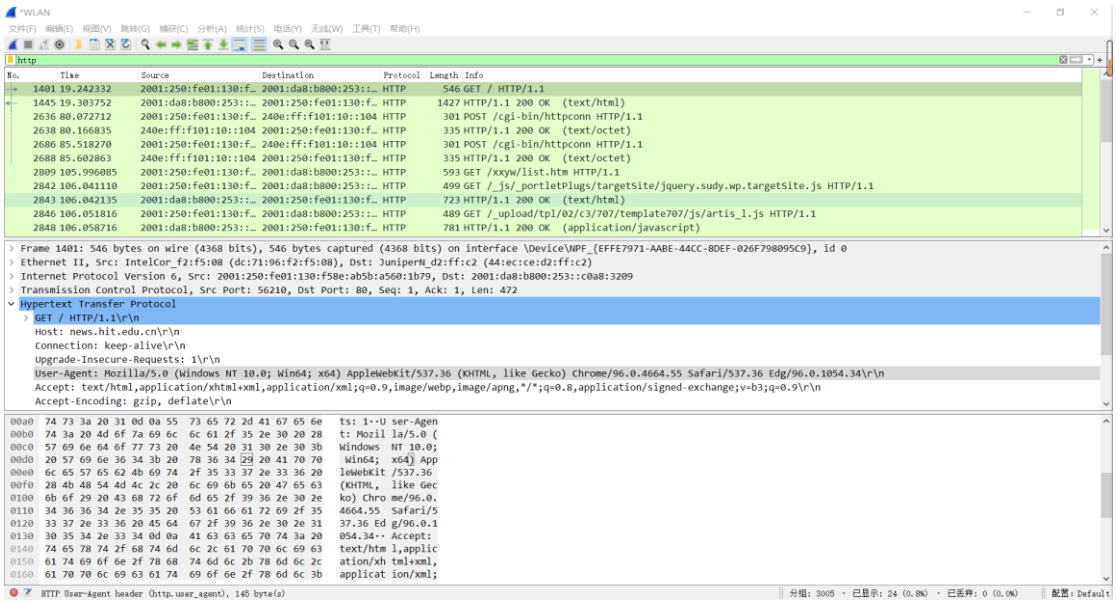
设置字符串数组存储钓鱼网站地址，设置引导目的网站地址，同样解析匹配url，若匹配成功，则更改HTTP头部字段的url(访问网址)与host主机地址，实现网页跳转，同时打印输出相关提示信息。

实验结果：

一. Wireshark 分析 HTTP 协议

1) HTTP GET/response 交互

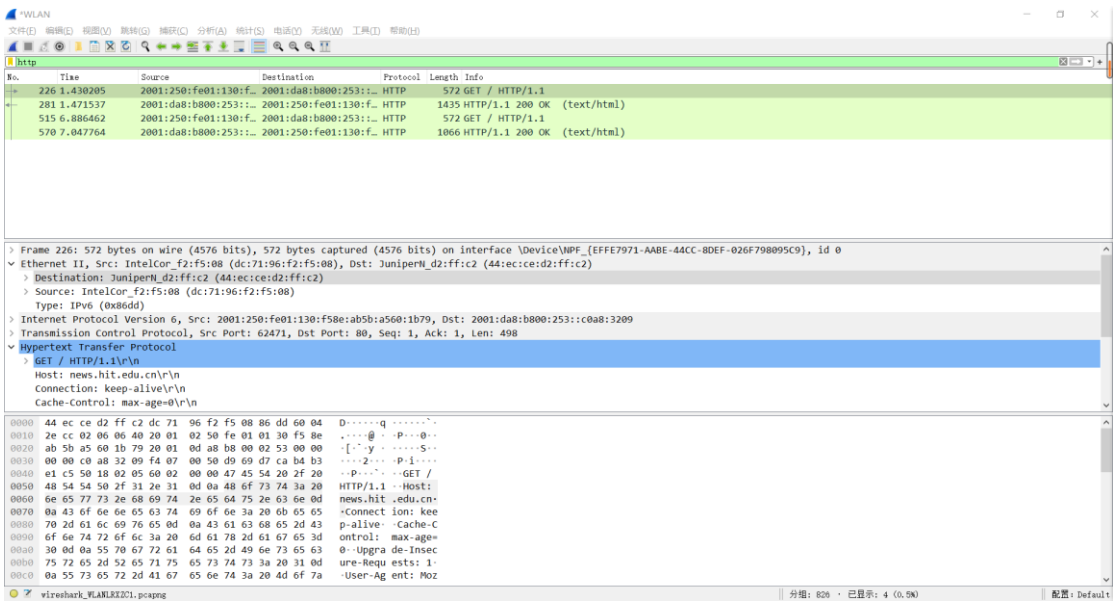
按照实验过程中给出的步骤进行试验，得到俘获的分组如下所示



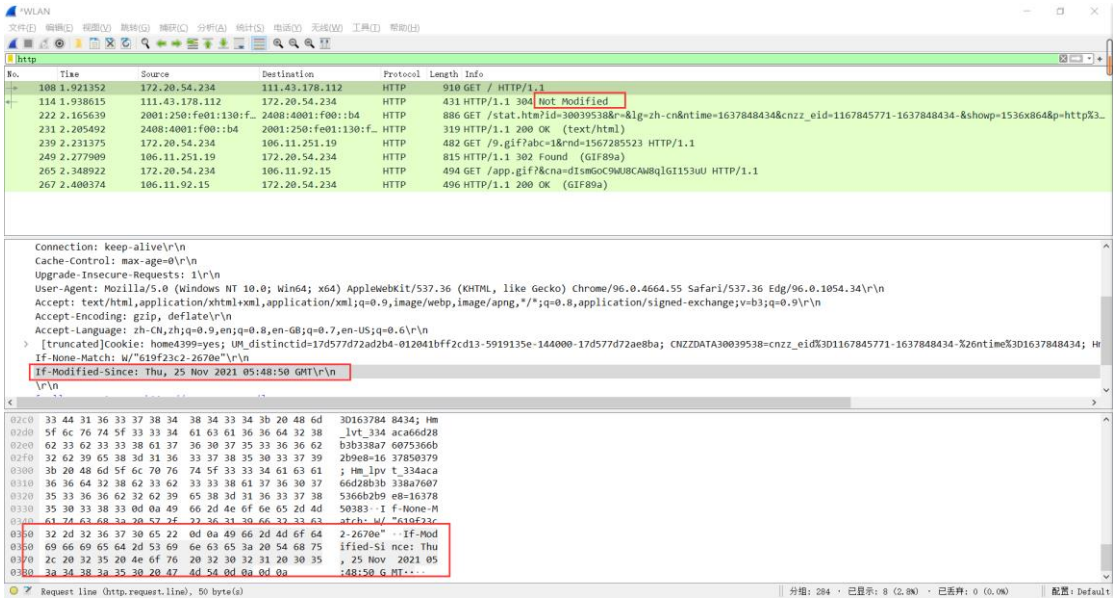
- **你的浏览器运行的是 HTTP1.0，还是 HTTP1.1？**
  HTTP1.1
- **你所访问的服务器所运行 HTTP 协议的版本号是多少？**
  HTTP1.1
- **你所访问的服务器所运行 HTTP 协议的版本号是多少？**
  zh-CN。也就是简体中文
- **你的计算机的 IP 地址是多少？**
  2001:250:fe01:130:f58e:ab5b:a560:1b79
- **服务器的 IP 地址是多少**
  2001:da8:b800:253::c0a8:3209
- **从服务器向你的浏览器返回的状态代码是多少**
  200

2) HTTP 条件 GET/response交互

捕获数据包如下：



由于访问 http://news.hit.edu.cn 无 if-modified-since 行，更改访问 www.4399.com，捕获内容如下：



- **分析你的浏览器向服务器发出的第一个 HTTP GET 请求的内容，在该请求报文中，是否有一行是： IF-MODIFIED-SINCE？**

  没有

- **分析服务器响应报文的内容，服务器是否明确返回了文件的内容？如何获知？**

  服务器明确返回了文件的所有内容。

  服务器返回的状态码为 200，表示明确返回了文件。

- **分析你的浏览器向服务器发出的较晚的“HTTP GET”请求，在该请求报文中是否有一行是：IF-MODIFIED-SINCE？如果有，在该首部行后面跟着的信息是什么？**

  有

  本次实验中，IF-MODIFIED-SINCE 字段的内容是： If-Modified-Since: Thu, 25 Nov 2021 05:48:50 GMT 表示浏览器上一次缓存该网页内容的时间。因此在该首部行后面跟着的信息是缓存最后更新的时间。

- **服务器对较晚的 HTTP GET 请求的响应中的 HTTP 状态代码是多少？服务器是否明确返回了文件的内容？请解释。**

  从图中可以看到，服务器返回的状态码是 304，表示 Not Modified。即不需要重新加载网页，服务器没有明确返回文件的内容。

- **总结**

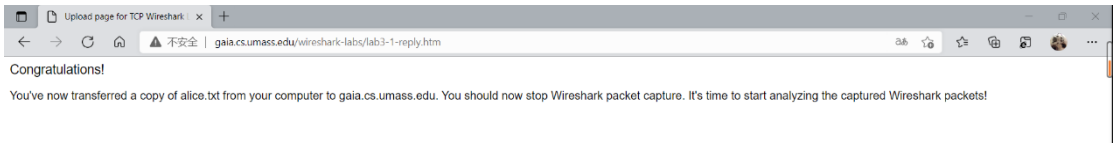
✧ Web 浏览器第一次访问一个网页中实际上不会添加 IF-MODIFIED-SINCE 字段，因为此时该网页在本地没有缓存，浏览器明确地在服务器端请求网页，该字段也就失去了作用

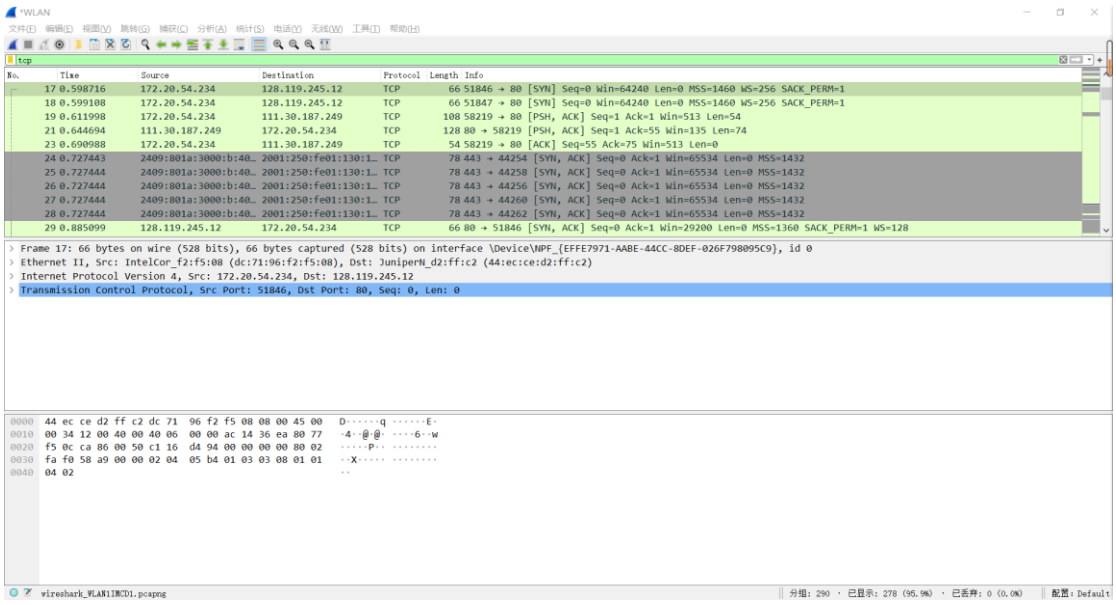
✧ 如果服务器返回的状态码为 200，则表明返回了文件。

✧ 如果服务器返回的状态码为 304，则表示网页的内容没有更新，可以使用本地缓存的内容。

二. 利用 Wireshark 分析 TCP 协议

1) 俘获大量的由本地主机到远程服务器的 TCP 分组





2) 浏览追踪信息

- **向 gaia.cs.umass.edu 服务器传送文件的客户端主机的 IP 地址和 TCP 端口号是多少？**

  主机 IP 地址：172.20.54.234

  TCP 端口号：51846

- **Gaia.cs.umass.edu 服务器的 IP 地址是多少？对这一连接，它用来发送和接收 TCP 报文的端口号是多少？**

  服务器 IP 地址：128.119.245.12

  用来发送和接收 TCP 报文的端口号：80

3) TCP 基础

- **客户服务器之间用于初始化 TCP 连接的 TCP SYN 报文段的序号（sequence number）是多少？在该报文段中，是用什么来标示该报文段是 SYN 报文段的？**

```
Acknowledgment number (raw): 0
1000 .... = Header Length: 32 bytes (8)
Flags: 0x002 (SYN)
   000. .... .... = Reserved: Not set
   ...0 .... .... = Nonce: Not set
   .... 0... .... = Congestion Window Reduced (CWR): Not set
   .... .0.. .... = ECN-Echo: Not set
   .... ..0. .... = Urgent: Not set
   .... ...0 .... = Acknowledgment: Not set
   .... .... 0... = Push: Not set
   .... .... .0.. = Reset: Not set
   .... .... ..1. = Syn: Set
```

初始化 TCP 连接的 SYN 报文段的序号是：0

通过 Flag 标志位，表示该报文段是 SYN 报文段（将其中的 SYN 位置设为 1）

- **服务器向客户端发送的 SYNACK 报文段序号是多少？该报文段 中，Acknowledgement 字段的值是多少？Gaia.cs.umass.edu 服务器 是如何决定此值的？在该报文段中，是用什么来标示该报文段是 SYNACK 报文段的？**

  对三次握手过程进行分析可知，服务器发送的 acknowledgement number 字段由上一次客户端发送给服务器的 seq number + 1 得到的

  如果 Flags 标志位的 SYN 和 ACK 位均为 1，则该报文是一个 SYN ACK 报文段

- **你能从捕获的数据包中分析出 tcp 三次握手过程吗？**

  - 建立连接时,客户端发送 syn 包(syn=j)到服务器,并进入 SYN_SEND 状态,等待服务器确认；

```
17 0.598716    172.20.54.234    128.119.245.12    TCP    66 51846 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
```

  - 服务器收到 syn 包,必须确认客户的 SYN（ack=j+1）,同时自己也发送一个 SYN 包（syn=k）,即 SYN+ACK 包,此时服务器进入 SYN_RECV 状态；

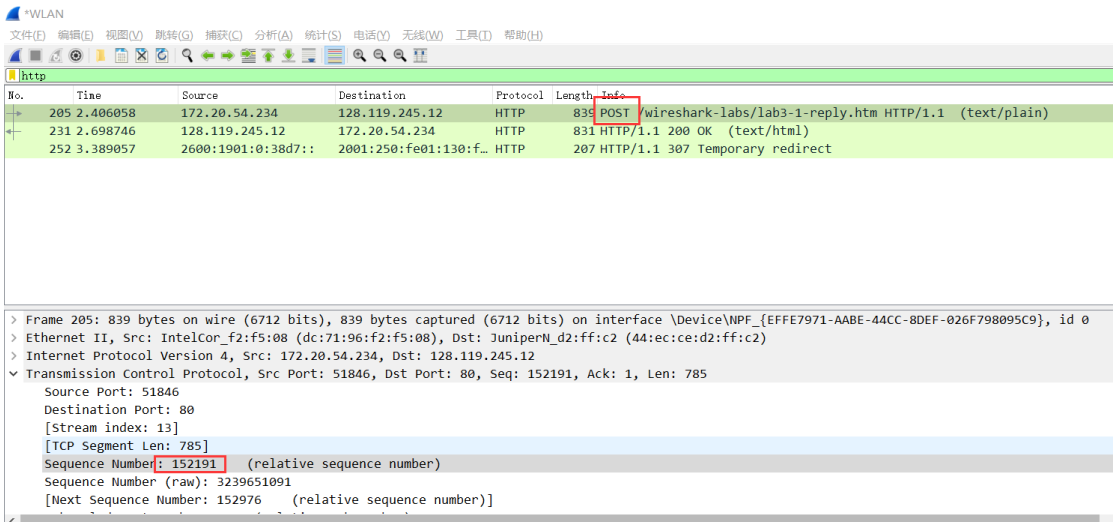
```
18 0.599108    172.20.54.234    128.119.245.12    TCP    66 51847 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
```

  - 第三次握手：客户端收到服务器的 SYN＋ACK 包,向服务器发送确认包 ACK(ack=k+1),此包发送完毕,客户端和服务器进入 ESTABLISHED 状态,完成三次握手.

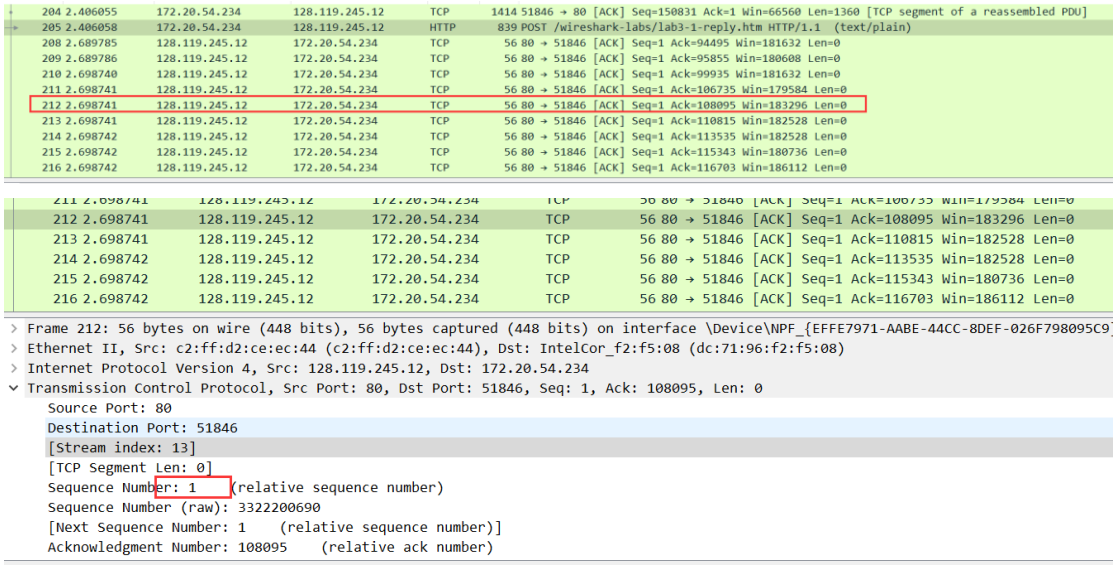
```
29 0.885099    128.119.245.12    172.20.54.234    TCP    66 80 → 51846 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1360 SACK_PERM=1 WS=128
```

- **包含 HTTP POST 命令的 TCP 报文段的序号是多少？**

  如图所示，为 152191



- **如果将包含 HTTP POST 命令的 TCP 报文段看作是 TCP 连接上的 第一个报文段，那么该 TCP 连接上的第六个报文段的序号是多少？是何时发送的？该报文段所对应的 ACK 是何时接收的？**

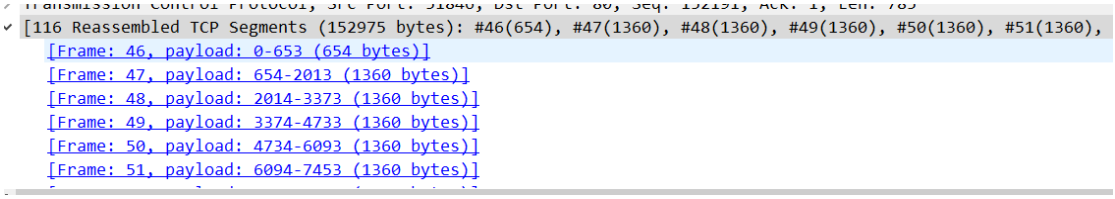


第六个报文段的序号是：1

在 http post 发送之前，tcp 连接建立之后发送

对应的 ack 是服务器返回的第六个 ack

- **前六个 TCP 报文段的长度各是多少？**



654，1360，1360，1360，1360，1360

- **在整个跟踪过程中，接收端公示的最小的可用缓存空间是多少？ 限制发送端的传输以后，接收端的缓存是否仍然不够用？**

最小可用缓存空间是 65536。

首先，由于本次实验规模较小，没有出现接收端缓存不够用的情况；其次，限制发送端传输后，接收端的缓存仍然可能出现不够用的现象。

- **在跟踪文件中是否有重传的报文段？进行判断的依据是什么？**

  没有。没有出现重复的序列号

- **TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。**

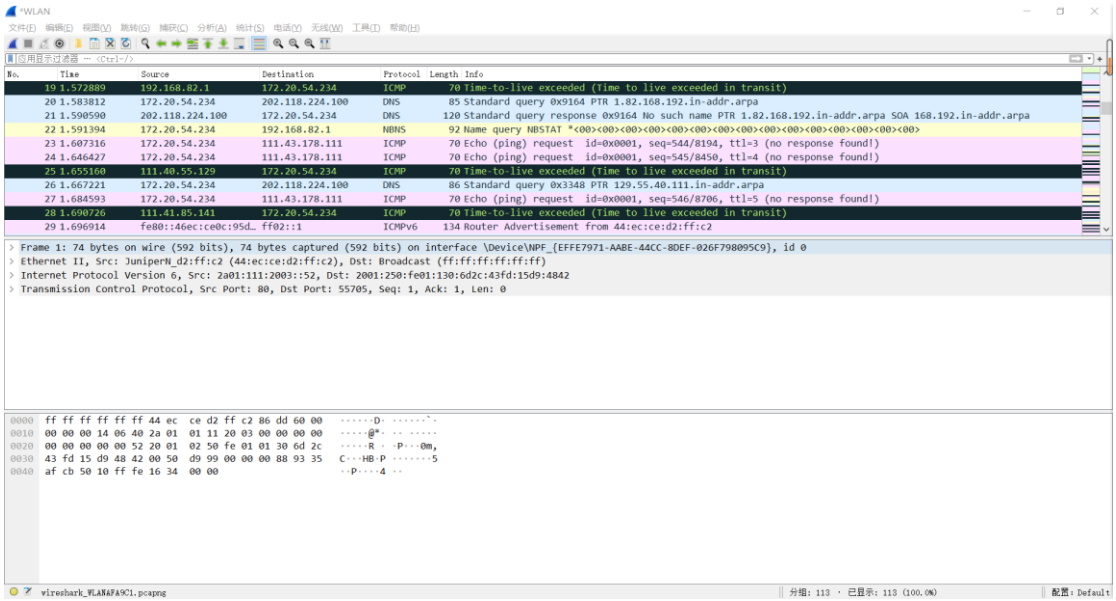
  数据总长度：$152685 + 108 * 54 = 164422B$

  时间间隔：$1.74586s$

  吞吐量：$\frac{164422}{1.74} = 9.6e5\ bps$

三. 利用 Wireshark 分析 IP 协议

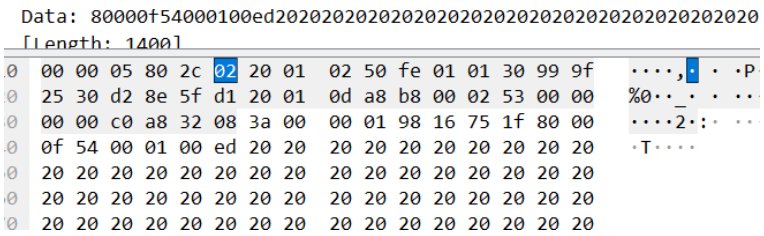
1) 执行 traceout 捕获数据包



- **你主机的 IP 地址是什么？**

  172.20.54.214

- **在 IP 数据包头中，上层协议（upper layer）字段的值是什么？**



上层协议字段的值为：02

- **IP 头有多少字节？该 IP 数据包的净载为多少字节？并解释你是怎样确定该 IP 数据包的净载大小的？**

  IP 头有 20 字节

  IP 包的净载为总长度减去头部长度，即为 56B-20B=36B

- **该 IP 数据包分片了吗？解释你是如何确定该 P 数据包是否进行了分片**

  没有分片，若有分片，则必然在某些数据包中存在偏移量，但是所有的数据包偏移量均为 0

- **你主机发出的一系列 ICMP 消息中 IP 数据报中哪些字段总是发生改变？**

  TTL，头部校验和，ID 鉴别码

- **哪些字段必须保持常量？哪些字段必须改变？为什么？**
  常量：IP 版本，在数据传输过程中，通常只用一个 IP 版本
  必须改变：TTL，头部校验和，ID 鉴别码
- **描述你看到的 IP 数据包 Identification 字段值的形式。**
  加一递增的 16 位
- **Identification 字段和 TTL 字段的值是什么？**
  Identification：29668
  TTL：49
- **最近的路由器（第一跳）返回给你主机的 ICMP Time-to-live exceeded 消息中这些值是否保持不变？为什么？**
  不变。因为最近一个路由器发送回来的时候，该值是维持不变的。
- **该消息是否被分解成不止一个 IP 数据报？**

  > [2 IPv4 Fragments (1980 bytes): #2390(1480), #2391(500)]

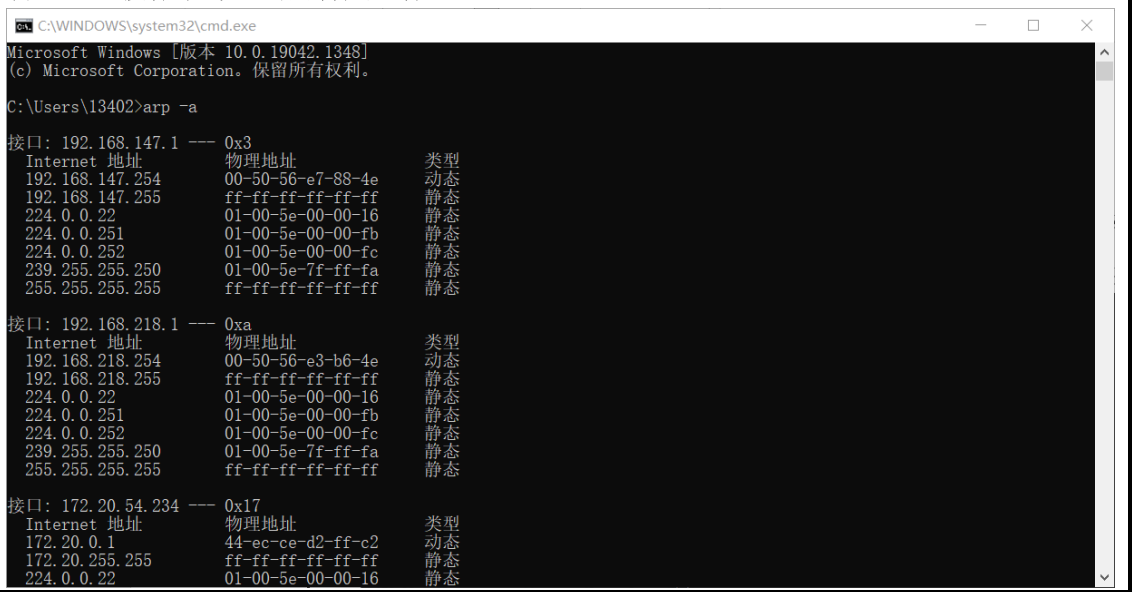
  被分成了两个
- **观察第一个 IP 分片，IP 头部的哪些信息表明数据包被进行了分片？IP 头部的哪些信息表明数据包是第一个而不是最后一个分片？该分片的长度是多少**
  表明了被分片：偏移量和 More Fragment 段
  数据包是第一个而不是最后一个分片：偏移量为 0，More Fragment 段为 1
  该分片的长度是多少：1480
- **原始数据包被分成了多少片？**

  [3 IPv4 Fragments (3480 bytes): #2953(1480), #2955(1480), #2954(520)]

  可以看到被分为了 3 片
- **这些分片中 IP 数据报头部哪些字段发生了变化？**
  More Fragment 段的值：前两片为 1，最后一片 0
  偏移量：第二片的分片为 1480，最后一片为 2960。

四. 利用 Wireshark 分析 ARP 协议

- **利用 MS-DOS 命令：arp 或 c:\windows\system32\arp 查看主机 上 ARP 缓存的内容。说明 ARP 缓存中每一列的含义是什么？**



Internet 地址，物理地址，类型（动态/静态）

- **ARP 数据包的格式是怎样的？由几部分构成，各个部分所占的字节数是多少？**

图1.　ARP 数据包的格式

部分	字节数
硬件类型	2
协议类型	2
硬件地址长度	1
协议地址长度	1
OP	2
发送端 MAC 地址	6
发送端 IP 地址	4
目的 MAC 地址	6
目的 IP 地址	4

- **如何判断一个 ARP 数据是请求包还是应答包？**

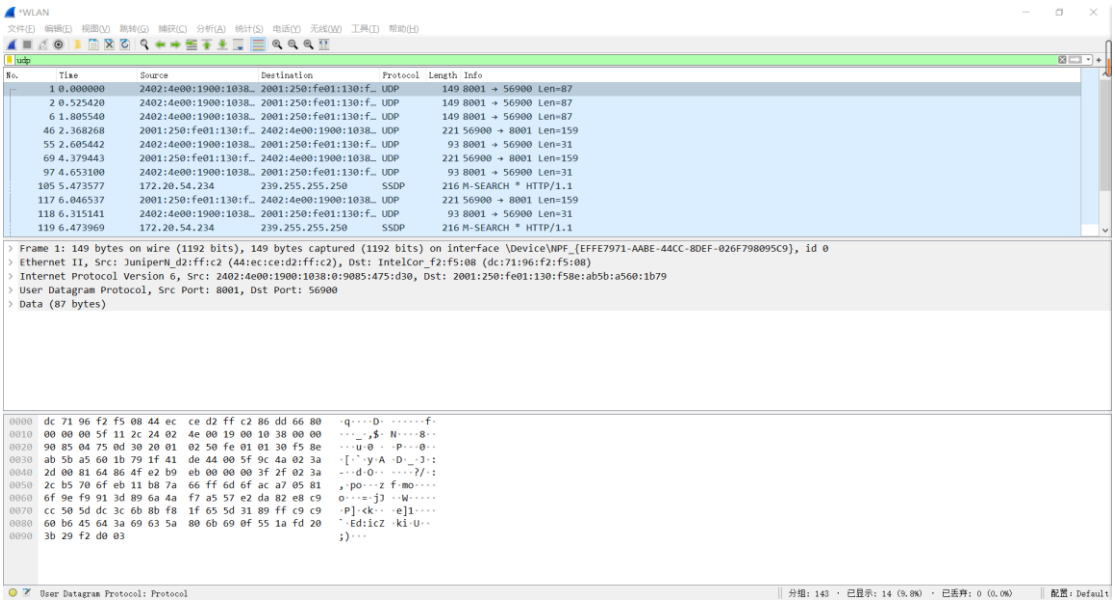
  检测 OP 字段，OP=0x0001 为请求包，OP=0x0002 为应答包

- **为什么 ARP 查询要在广播帧中传送，而 ARP 响应要在一个有着明确目的局域网地址的帧中传送？**

  ARP 不知道目的地址所对应的 MAC 地址

  目的地址可以通过查询报文获取查询主机的 MAC 地址，因此只向其发送即可。

五. 利用 Wireshark 分析 UDP 协议



- **消息是基于 UDP 的还是 TCP 的？**

UDP

- **你的主机 ip 地址是什么？目的主机 ip 地址是什么？**

  我的主机 IP 地址：2402:4e00:1900:1038:0:9085:475:d30

  目的主机 IP 地址：2001:250:fe01:130:f58e:ab5b:a560:1b79

- **你的主机发送 QQ 消息的端口号和 QQ 服务器的端口号分别是多少？**

```
Source Port: 8001
Destination Port: 56900
```

  发送：8001

  服务器：56900

- **数据报的格式是什么样的？都包含哪些字段，分别占多少字节？**

  UDP 数据报格式包括：

  首部和数据，其中数据存放具体内容。首部有源端口号、目的端口号、数据报长度、校验和。

  分别占多少字节：

  源端口号占 2 个字节、目的端口号占 2 个字节、数据报长度占 2 个字节、校验和占 2 个字节。

部分	字节数
源端口号	2
目的端口号	2
UDP 长度	2
UDP 校验和	2

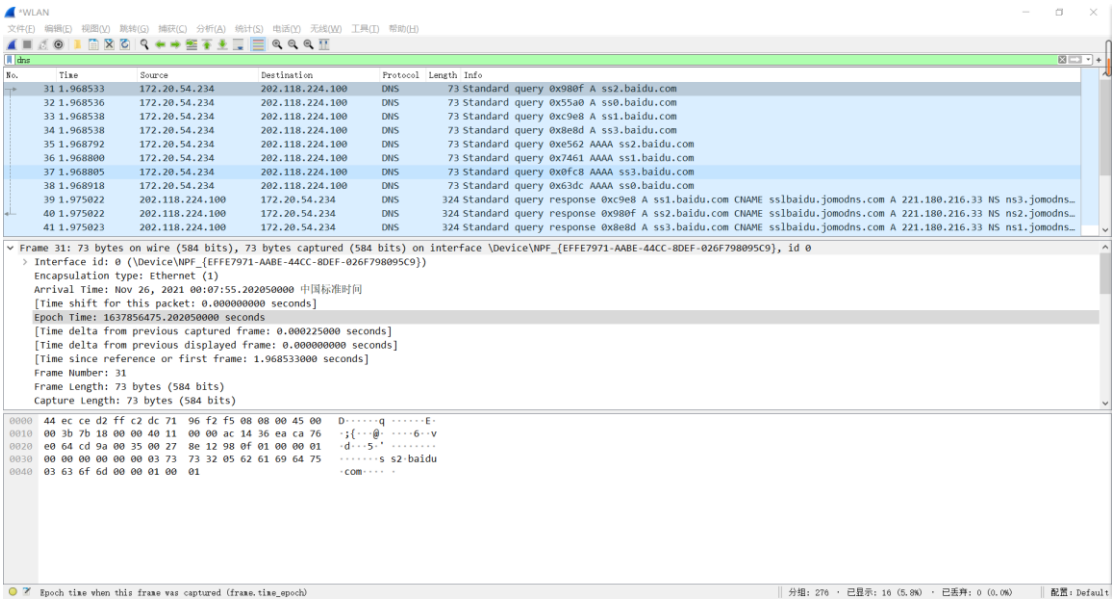
- **为什么你发送一个 ICQ 数据包后，服务器又返回给你的主机一个 ICQ 数据包？这 UDP 的不可靠数据传输有什么联系？对比前面的 TCP 协议分析，你能看出 UDP 是无连接的吗？**

  服务器返回接受的结果给客户端

  服务器只发送一次 ACK，无法保证可靠传输

  可以看出：UDP 数据报没有序列号，过程中也没有三次握手发送数据，每次只发送一个数据报，然后等待响应。

六. 利用 Wireshark 分析 DNS 协议



在控制台查询后，可以看到，通过 DNS 协议来获取域名的地址。

七. 协议总结

(一)HTTP 协议

1. HTTP 协议支持客户/服务器模式；

2. 简单快速：客户向服务器请求服务时，只需传送请求方法和路径；请求 方法常用 GET、HEAD、POST 等，每种方法规定了客户与服务器联系 的不同类型；HTTP 协议简单，服务器程序规模小，通信速度较快；

3. 灵活性：HTTP 允许传输任意类型数据对象；正在传输数据类型由 Content-Type 标记；

4. 无连接：无连接是指每次连接只处理一个请求；服务器处理完客户请求， 并收到客户应答后，即断开连接，节省传输时间；

5. 无状态：指协议对于事务处理无记忆能力；应答快，但传输数据量较大。

(二)TCP 协议

1. 提供面向连接的、可靠的、基于流的数据传输服务，传输单位是报文段；

2. 使用超时重发、数据确认等方式确保数据被正确发送至目的地。

(三)IP 协议

1. 任务: 负责对数据包进行路由选择和存储转发；

2. 负责为分组交换网上的不同主机提供通信服务。在发送数据时，网络层 把运输层产生的报文段和用户数据报封装成分组（IP 数据报）或包进行传送；

3. IP 协议:逐跳发送模式；根据数据包的目的地 IP 地址决定数据如何发送； 如果数据包不能直接发送至目的地，IP 协议负责寻找一个合适的下一跳 路由器，并将数据包交付给该路由器转发；

4. ICMP 协议：因特网控制报文协议，用于检测网络连接；

(四)Internet 协议

1. 两个相邻节点之间传送数据时，数据链路层将网络层交下来的 IP 数据报组装成帧，在两个相邻的链路上传送帧（frame)。每一帧包括数据和必要的控制信息；

2. 网卡接口的网络驱动程序，处理数据在物理媒介上的传输；不同的物理 网络具有电气特性，网络驱动程序隐藏实现细节，为上层协议提供一致接口

(五)ARP 协议

1. 地址解析协议 ARP（Address Resolution Protocol），负责完成逻辑地址向物 理地址的动态映射，将 32 位逻辑地址（IP 地址）转换为 48 位的物理地址（MAC 地址）。

(六)UDP 协议

1. 为了在给定的主机上能识别多个目的地址，同时允许多个应用程序在同一台主机上工作并能独立地进行数据包的发送和接收，设计用户数据报协议 UDP。

2. UDP 使用底层的互联网协议来传送报文，同 IP 一样提供不可靠的无连 接数据包传输服务。它不提供报文到达确认、排序、及流量控制等功能。

(七)DNS 协议

1. DNS 是一种可以将域名和 IP 地址相互映射的以层次结构分布的数据库系统。

2. DNS 系统采用递归查询请求的方式来响应用户的查询，为互联网的运行提供关键性的基础服务。

3. 目前绝大多数的防火墙和网络都会开放 DNS 服务，DNS 数据包不会被拦截，因此可以基于 DNS 协议建立隐蔽信道，从而顺利穿过防火墙，在客户端和服务器之间传输数据。

问题讨论：

- 在实验过程中，发现了某些网站，在第二次访问的报文中并没有IF-MODIFIED-SINCE字段，（例如http://news.hit.edu.cn）。更改访问www.4399.com后获得了IF-MODIFIED-SINCE字段；经过查阅资料得知，某些网站对缓存的判断已经抛弃了IF-MODIFIED-SINCE的通信方式。
- 由于Wireshark十分灵敏，可能会出现同时抓到很多包的情况，此时可以通过过滤器进行筛选。
- 针对服务器缓存的问题，在本次实验中并没有出现过缓存溢出的情况，在后续的改进中，增大了数据的规模，会发现窗口的大小会一直不停地增加。

心得体会：

结合实验过程和结果给出实验的体会和收获：

- 通过本次实验，从实践中对协议进行了分析和总结，对各种协议有了更深刻全面的理解
- 掌握了抓包工具的使用