

# 计算机网络原理实验报告

学院 计算机学院 专业 计算机科学与技术 班级 10012006

学号 2020303245 姓名 夏卓 实验时间: 2022/11/26

## 一、 实验名称:

网络协议分析与验证

## 二、 实验目的:

(1) 以用户访问一个 Web 网站首页为例, 深入理解 Web 服务系统的工作原理, 从计算机网络体系结构的角度出发, 分别从应用层、传输层、网络层以及数据链路层分析网络为 Web 服务提供的技术支持以及工作原理;

(2) 通过分析 Wireshark 抓包工具所抓取的数据报, 分析 DNS、Web 协议的工作原理, 进而类推到 FTP、SMTP、DHCP 等协议的分析;

(3) 分析 TCP 协议通过三次握手建立连接的时序关系, 以及通过四次挥手释放 TCP 连接的时序关系;

(4) 分析 ARP 协议工作原理

(5) 分析数据链路层工业以太网工作原理 (数据帧的语法);

## 三、 实验环境:

Win10, Intelx86, wireshark

## 四、 实验内容及步骤:

### 实验内容:

1. 学习 Wireshark 或 Ethereal 工具的使用方法。
2. 访问 [www.baidu.com](http://www.baidu.com) 网站首页, 并对通信完整过程抓包。
3. 通过对抓取的数据包进行分析, 深入理解网络协议工作原理。

### 实验步骤:

- (1) 安装 WINPCAP 组件;  
实验 2 中已经安装完成
- (2) 安装 wireshark 抓包工具;  
安装较为简单, 在此不再演示
- (3) 启动 wireshark 抓包工具, 并激活的网络接口上开始抓包;  
首先查看本机无线局域网 WLAN 配置信息:

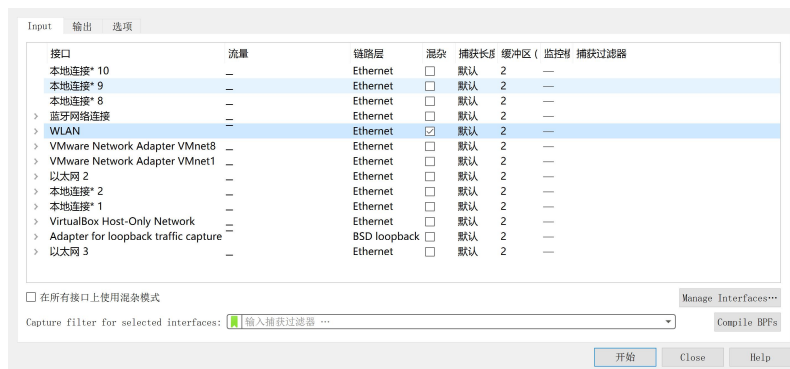
```
连接特定的 DNS 后缀 . . . . . :  
描述 . . . . . : Intel(R) Wi-Fi 6 AX200 160MHz  
物理地址. . . . . : 24-41-8C-F9-1E-74  
DHCP 已启用 . . . . . : 是  
自动配置已启用 . . . . . : 是  
本地链接 IPv6 地址. . . . . : fe80::105a:cd46:3b91:4c50%15(首选)  
IPv4 地址 . . . . . : 10.30.176.138(首选)
```

可以看到：

本机 MAC 地址为：24-41-8C-F9-1E-74

IP 地址为：10.30.176.138

启动 Wireshark，在捕获—选项中勾选无线局域网 WLAN，并启动混杂模式（该模式下会接收所有经过网卡的数据包，包括不是发给本机的包，即不验证 MAC 地址），点击 Start，启动抓包。



- (4) 用户在浏览器地址栏输入：[www.baidu.com](http://www.baidu.com) 回车，直到百度首先在浏览器上显示为止；

在管理员模式下，首先清空本机 ARP 缓存表，接着直接在命令行中输入命令 `curl www.baidu.com`，连接百度首页：

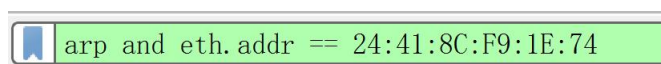
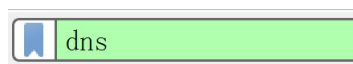
```
C:\WINDOWS\system32>arp -d *
C:\WINDOWS\system32>curl -I www.baidu.com
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: private, no-cache, no-store, proxy-revalidate, no-transform
Connection: keep-alive
Content-Length: 277
Content-Type: text/html
Date: Sat, 26 Nov 2022 06:53:02 GMT
Etag: "575e1f72-115"
Last-Modified: Mon, 13 Jun 2016 02:50:26 GMT
Pragma: no-cache
Server: bfe/1.0.8.18
```

或者在清空 ARP 表后在浏览器中输入 [www.baidu.com](http://www.baidu.com) 回车，效果一样：



- (5) 抓包结束，开始对报文分析，对实验要求内容找到报文证据。

停止抓包，在 Wireshark 的显示过滤器中分别输入



就能过滤出所需要查看的 DNS 和 ARP 报文段，其中 ARP 的过滤条件中还添加了本机 MAC 地址的限制。

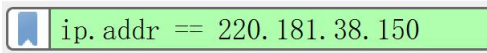
而为了查看 TCP 三次连接与四次挥手报文段，则需要输入 [www.baidu.com](http://www.baidu.com) 的 IP 地址进行过滤。为了得到它的 IP 地址，可在命令行中输入 `ping www.baidu.com` 命令进行连接：

```
C:\Users\xz276>ping www.baidu.com

正在 Ping www.a.shifen.com [220.181.38.150] 具有 32 字节的数据:
来自 220.181.38.150 的回复: 字节=32 时间=53ms TTL=49
来自 220.181.38.150 的回复: 字节=32 时间=25ms TTL=49
来自 220.181.38.150 的回复: 字节=32 时间=25ms TTL=49
来自 220.181.38.150 的回复: 字节=32 时间=24ms TTL=49

220.181.38.150 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 = 24ms, 最长 = 53ms, 平均 = 31ms
```

由此可以得到百度的 IP 地址为: 220.181.38.150  
于是在 wireshark 的显示过滤器中输入

 ip.addr == 220.181.38.150

即可过滤出源地址或者目标地址是百度 IP 地址的所有报文,由此即可筛选出 TCP 三次连接与四次挥手报文段

## 五、实验结果:

### DNS 请求报文和对应的应答报文:

No.	Time	Source	Destination	Protocol	Length	Info
2355	2022-11-25 23:13:20.262016	10.30.176.138	202.117.80.6	DNS	73	Standard query 0x83f7 A www.baidu.com
2356	2022-11-25 23:13:20.266666	202.117.80.6	10.30.176.138	DNS	135	Standard query response 0x83f7 A www.baidu.com

```
> Internet Protocol Version 4, Src: 202.117.80.6, Dst: 10.30.176.138
> User Datagram Protocol, Src Port: 53, Dst Port: 51151
< Domain Name System (response)
  Transaction ID: 0x83f7
  Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 3
  Authority RRs: 0
  Additional RRs: 0
  Queries
    < www.baidu.com: type A, class IN
      Name: www.baidu.com
      [Name Length: 13]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
    < Answers
      > www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
      > www.a.shifen.com: type A, class IN, addr 220.181.38.150
      > www.a.shifen.com: type A, class IN, addr 220.181.38.149
      [Request In: 2355]
      [Time: 0.004650000 seconds]
```

首先分析源地址和目的地址可知, 10.30.176.138 正是本机的 IP 地址, 而 202.117.80.6 正是 DNS 服务器的 IP 地址:

```
IPv4 地址 . . . . . : 10.30.176.138(首选)
子网掩码 . . . . . : 255.255.0.0
```

```
DNS 服务器 . . . . . : 202.117.80.6
                  202.117.80.7
```

因此这个过程即是客户端向 DNS 服务器发起 DNS 请求, 希望得到 [www.baidu.com](http://www.baidu.com) 的 IP 地址, 域名解析完成后, 服务器将 IP 地址返回给客户端, 注意最下方的 Answers:

```
< Answers
  > www.baidu.com: type CNAME, class IN, cname www.a.shifen.com
  > www.a.shifen.com: type A, class IN, addr 220.181.38.150
  > www.a.shifen.com: type A, class IN, addr 220.181.38.149
  [Request In: 2355]
  [Time: 0.004650000 seconds]
```

可以看到, 服务器返回了百度其中两个 IP 地址, 分别为: 220.181.38.150 和 220.181.38.149, 且与 ping 所得结果一致, 除此之外, 我们还可以看到百度别名是 [www.a.shifen.com](http://www.a.shifen.com)。

TCP 三次握手和对应的四次挥手报文段：

No.	Time	Source	Destination	Protocol	Length	Info
2358	2022-11-25 23:13:20.270086	10.30.176.138	220.181.38.150	TCP	74	10637 → 80 [SVN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM TSval=501
3165	2022-11-25 23:13:21.217948	220.181.38.150	10.30.176.138	TCP	74	80 → 10637 [SVN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1380 WS=32 SACK_PERM
3166	2022-11-25 23:13:21.218027	10.30.176.138	220.181.38.150	TCP	54	10637 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
3167	2022-11-25 23:13:21.229261	10.30.176.138	220.181.38.150	HTTP	132	HEAD / HTTP/1.1
3168	2022-11-25 23:13:21.258207	220.181.38.150	10.30.176.138	TCP	60	80 → 10637 [ACK] Seq=1 Ack=79 Win=78464 Len=0
3169	2022-11-25 23:13:21.258955	220.181.38.150	10.30.176.138	HTTP	386	HTTP/1.1 200 OK
3170	2022-11-25 23:13:21.269663	10.30.176.138	220.181.38.150	TCP	54	10637 → 80 [FIN, ACK] Seq=79 Ack=333 Win=65792 Len=0
3253	2022-11-25 23:13:21.323897	220.181.38.150	10.30.176.138	TCP	60	80 → 10637 [ACK] Seq=333 Ack=80 Win=78464 Len=0
3254	2022-11-25 23:13:21.324591	220.181.38.150	10.30.176.138	TCP	60	80 → 10637 [FIN, ACK] Seq=333 Ack=80 Win=78464 Len=0
3256	2022-11-25 23:13:21.324645	10.30.176.138	220.181.38.150	TCP	54	10637 → 80 [ACK] Seq=80 Ack=334 Win=65792 Len=0

> Frame 3167: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface \Device\NPF\_{A5DC845C-48EB-4B6F-99C9-761246A1197F}, id 0

> Ethernet II, Src: IntelCor\_f9:1e:74 (24:41:8c:f9:1e:74), Dst: HuaweiTe\_8a:36:95 (04:f3:52:8a:36:95)

> Internet Protocol Version 4, Src: 10.30.176.138, Dst: 220.181.38.150

> Transmission Control Protocol, Src Port: 10637, Dst Port: 80, Seq: 1, Ack: 1, Len: 78

> Hypertext Transfer Protocol

> HEAD / HTTP/1.1\r\n

Host: www.baidu.com\r\n

User-Agent: curl/7.71.1\r\n

Accept: \*/\*\r\n

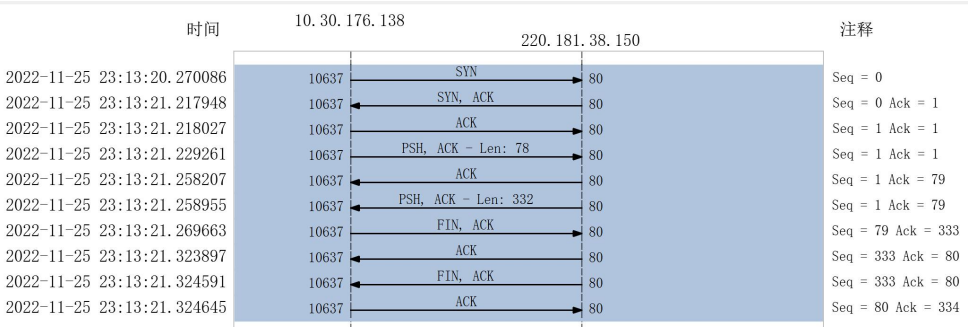
\r\n

[Full request URI: http://www.baidu.com/]

[HTTP request 1/1]

[Response in frame: 3169]

为了清晰的看到三次握手和四次挥手的过程，可以使用流量图显示：



可以看到：

第一次握手：客户端发送一个 TCP，标志位为 Seq = 0，代表客户端请求建立连接；

第二次握手：服务器发回确认包，标志位为 Seq=0，ACK = 1；

第三次握手：客户端再次发送确认包 Seq=1，ACK = 1。

中间是双方进行通信的数据信息：

首先客户端发送一个 HTTP 的 HEAD 请求，服务器收到请求之后返回了一个 ACK 进行确认，之后将 HTTP 的头部信息返回给客户端。

第一次挥手：客户端向服务器发送一个 [FIN+ACK] 数据包，主动断开连接；

第二次挥手：服务器收到 FIN 数据包之后，向客户端发送 ACK 进行确认；

第三次挥手：服务器向客户端发送[FIN+ACK]，表示自己也没有数据要发送了；

第四次挥手：客户端收到 FIN 数据包之后，一样发送一个 ACK 报文作为应答。

ARP 请求报文和对应的应答报文：

No.	Time	Source	Destination	Protocol	Length	Info
932	2022-11-26 16:12:54.169923	IntelCor_f9:1e:74	Broadcast	ARP	42	Who has 10.30.0.1? Tell 10.30.176.138
933	2022-11-26 16:12:54.228092	HuaweiTe_8a:36:95	IntelCor_f9:1e:74	ARP	56	10.30.0.1 is at 04:f3:52:8a:36:95
2889	2022-11-26 16:12:58.041150	HuaweiTe_8a:36:95	IntelCor_f9:1e:74	ARP	56	Who has 10.30.176.138? Tell 10.30.0.1
2890	2022-11-26 16:12:58.041163	IntelCor_f9:1e:74	HuaweiTe_8a:36:95	ARP	42	10.30.176.138 is at 24:41:8c:f9:1e:74

> Frame 933: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF\_{A5DC845C-48EB-4B6F-99C9-761246A1197F}, id 0

> Ethernet II, Src: HuaweiTe\_8a:36:95 (04:f3:52:8a:36:95), Dst: IntelCor\_f9:1e:74 (24:41:8c:f9:1e:74)

> Address Resolution Protocol (reply)

Hardware type: Ethernet (1)

Protocol type: IPv4 (0x0800)

Hardware size: 6

Protocol size: 4

Opcode: reply (2)

Sender MAC address: HuaweiTe\_8a:36:95 (04:f3:52:8a:36:95)

Sender IP address: 10.30.0.1

Target MAC address: IntelCor\_f9:1e:74 (24:41:8c:f9:1e:74)

Target IP address: 10.30.176.138



由于清空了本机 ARP 表，因此可以看到客户端首先发送广播报文以获取网关 MAC 地址：

```
Opcode: request (1)
Sender MAC address: IntelCor_f9:1e:74 (24:41:8c:f9:1e:74)
Sender IP address: 10.30.176.138
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
Target IP address: 10.30.0.1
```

得到网关 IP 为 10.30.0.1、MAC 地址为 04:f3:52:8a:36:95

```
Sender MAC address: HuaweiTe_8a:36:95 (04:f3:52:8a:36:95)
Sender IP address: 10.30.0.1
Target MAC address: IntelCor_f9:1e:74 (24:41:8c:f9:1e:74)
Target IP address: 10.30.176.138
```

可以看到其与命令行中得到的信息是一致的：

```
获得租约的时间 . . . . . : 2022年11月26日 15:15:28
租约过期的时间 . . . . . : 2022年11月29日 15:38:13
默认网关. . . . . : 10.30.0.1
DHCP 服务器 . . . . . : 10.30.0.1
```

## 六、实验总结

遇到的问题：

在本次实验过程，对抓取的报文进行分析，发现 ARP 协议没有工作。这是因为当主机 PC1 想发送数据给主机 PC2 时，首先会在自己的本地 ARP 缓存表中检查主机 PC2 匹配的 MAC 地址，没有找到相应的条目，才将 ARP 请求帧广播到本地网络上的所有主机，因此需要使用 `arp -d *` 命令清空 ARP 缓存表。DNS 同理，需要使用 `flushdns` 命令清除 DNS 解析程序缓存。

ARP 数据包解析：

```
Type: ARP (0x0806)
v Address Resolution Protocol (request) request: 表示这是一个请求数据包
  Hardware type: Ethernet (1) Hardware: 硬件类型，标识链路层协议
  Protocol type: IPv4 (0x0800) Protocol: 协议类型，标识网络层协议
  Hardware size: 6 硬件地址长度：也就是 MAC 地址长度，表示6个字节，即48
  Protocol size: 4 协议地址长度：也就是 IP 地址长度，表示4字节，32位
  Opcode: request (1) 操作码：标识这个数据包的类型，1表示请求，2表示响应
  Sender MAC address: ce:28:82:fe:fe:12 (ce:28:82:fe:fe:12) 源MAC地址
  Sender IP address: 192.168.2.4 源IP地址
  Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff) 目标MAC地址
  Target IP address: 192.168.2.1 目标IP地址
```

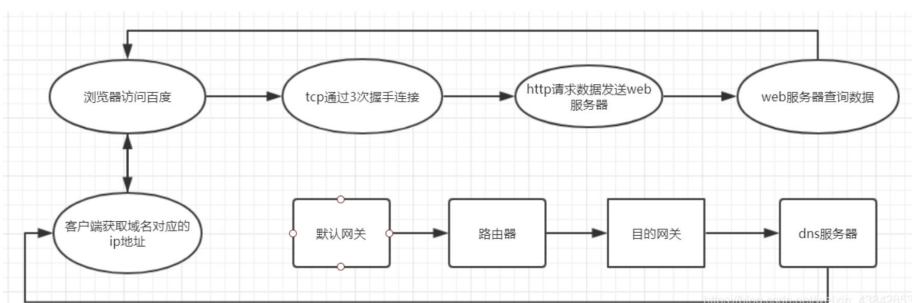
TCP 数据包解析：

```
v Transmission Control Protocol, Src Port: 64373, Dst Port: 22, Seq: 1, Ack: 1, Len: 0
  Source Port: 64373
  Destination Port: 22
  [Stream index: 14]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 1 (relative sequence number) 第三次握手：物理机发送ACK确认。(client你好！64373可以)
  Sequence Number (raw): 1094817303
  [Next Sequence Number: 1 (relative sequence number)] 第一次握手：物理机发送SYN到client。(client你好！22端口请求建立连接)
  Acknowledgment Number: 1 (relative ack number) 第二次握手：client发送SYN+ACK到物理机。(物理机你好！我允许你的22，你允许我的64373吗？)
  Acknowledgment number (raw): 3572071675
  0101 .... = Header Length: 20 bytes (5)
  v Flags: 0x010 (ACK)
```

## DNS 数据包解析：



## 客户端访问百度首页的整个过程：



其中关键是要先解析出百度域名 `www.baidu.com` 所对应的 `ip` 地址，其又包括以下几个步骤：

1. 先要知道默认网关的 `mac` 地址（使用 `arp` 获取默认网关的 `mac` 地址）
2. 组织数据发送给默认网关(`ip` 还是 `dns` 服务器的 `ip`,但是 `mac` 地址是默认网关的 `mac` 地址)
3. 默认网关拥有转发数据的能力,把数据转发给路由器
4. 路由器根据自己的路由协议,来选择一个合适的较快的路径转发数据给目的网关
5. 目的网关(`dns` 服务器所在的网关),把数据转发给 `dns` 服务器
6. `dns` 服务器查询解析出 `baidu.com` 对应的 `ip` 地址,并把它原路返回给请求这个域名的客户端

## 教师评语：

成绩：\_\_\_\_\_ 教师签名：\_\_\_\_\_ 批阅日期：\_\_\_\_\_

# 计算机网络原理实验报告

学院 计算机学院 专业 计算机科学与技术 班级 10012006

学号 2020303245 姓名 夏卓 实验时间: 2022/11/26

## 一、实验名称:

网络广播报文发送

## 二、实验目的:

设计并实现基于广播方式的通信程序;理解受限广播报文在局域网中传输的特点;初步认识对等通信模式;理解广播风暴产生的原因和解决该问题的方法。

## 三、实验环境:

Win10, Intelx86, wireshark, Visual Studio 2019

## 四、实验内容及步骤:

### 实验内容:

1. 从抓取的报文中过滤出源 IP = 发送方 IP 地址的某一个报文(可以手动或者采用过滤器);
2. 分析广播报文传输层采用 UDP/TCP, 理解广播或者组播为什么不是 TCP?
3. 抓取发送的广播报文, 找出通信的五元组信息和数据帧首部信息, 分析目的 IP 地址、源 IP 地址、协议类型、目的 MAC 地址、源 MAC 地址等与单播 UDP 用户数据报的不同。

### 实验步骤:

1. 编写程序, 发送三层广播报文;

首先在客户端的命令行中输入 ipconfig 获取客户端 ip 地址:

```
无线局域网适配器 WLAN:
    连接特定的 DNS 后缀 . . . . . : 
    本地链接 IPv6 地址. . . . . : fe80::3227:41ba:605e:b305%21
    IPv4 地址 . . . . . : 192.168.26.31
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 192.168.26.28
```

可以看到客户端的 IP 地址为 192.168.26.31。

接下来, 我们需要根据客户端实际 IP 地址对程序进行修改。需要注意的是, 在创建广播套接字时, Winsock 支持的套接字类型中, 只有数据报套接字 (SOCK\_DGRAM) 才支持广播通信, 然后将数据报套接字绑定在指定的地址和端口:

```
// 创建套接字
if ((sock = socket(AF_INET, SOCK_DGRAM, 0)) == INVALID_SOCKET)
{
    printf("create socket failed!\n");
    WSACleanup();
    return FALSE;
}
sockAddrFrom.sin_family = AF_INET;
sockAddrFrom.sin_addr.s_addr = inet_addr("192.168.26.31");
sockAddrFrom.sin_port = htons(SEND_PORT); // 套接字上绑定IP地址和端口号
if (bind(sock, (LPSOCKADDR)&sockAddrFrom, sizeof(sockAddrFrom)))
{
    closesocket(sock);
    WSACleanup();
    return FALSE;
}
```

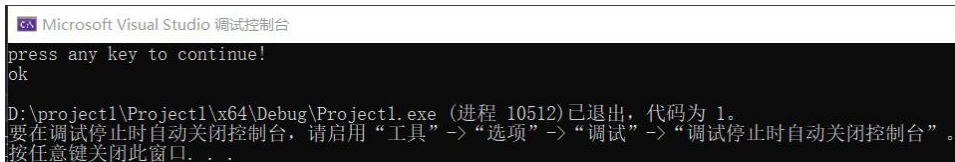
接着我们需要设置套接字相应选项，启用广播模式：

```
// 套接字选项设置
if (setsockopt(sock, SOL_SOCKET, SO_BROADCAST, (char *)&optReturn,
    sizeof(optReturn)) == SOCKET_ERROR)
{
    closesocket(sock);
    WSACleanup();
    return FALSE;
}
return TRUE;
```

最后通过 sendto() 函数发送广播报文，发送地址为 INADDR\_BROADCAST (广播地址)：

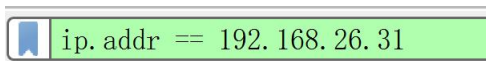
```
int lengthSend = 0;
sockAddrTo.sin_family = AF_INET;
sockAddrTo.sin_addr.s_addr = INADDR_BROADCAST;
sockAddrTo.sin_port = htons(RECV_PORT);
lengthSend = sendto(sock, LpBuffer, strlen(LpBuffer), MSG_DONTROUTE,
    (struct sockaddr *)&sockAddrTo, sizeof(sockAddr));
if (lengthSend == SOCKET_ERROR)
{
    // 发送失败
    closesocket(sock);
    WSACleanup();
    return FALSE;
}
return TRUE;
```

2. 在另一台利用抓包工具，抓取广播报文，并对报文首部进行分析；  
首先启动 wireshark 开始抓包，然后客户端启动程序进行广播报文的发送：



Microsoft Visual Studio 调试控制台  
press any key to continue!  
ok  
D:\project1\Project1\x64\Debug\Project1.exe (进程 10512) 已退出，代码为 1。  
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。  
按任意键关闭此窗口。...

停止抓包，输入客户端 ip 地址，进行数据包的过滤：



ip.addr == 192.168.26.31

在分组详情界面，可以看到各层数据信息：



Frame 81: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF\_{A5DC84...} Ethernet II, Src: LiteonTe\_e7:64:db (80:30:49:e7:64:db), Dst: Broadcast (ff:ff:ff:ff:ff:ff) Internet Protocol Version 4, Src: 192.168.26.31, Dst: 255.255.255.255 User Datagram Protocol, Src Port: 2000, Dst Port: 1000 Data (6 bytes)

五、实验结果：

实验结果如下所示：

No.	Time	Source	Destination	Protocol	Length	Info
80	2022-11-26 10:31:35.105087	192.168.26.31	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
81	2022-11-26 10:31:35.117970	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
83	2022-11-26 10:31:35.177900	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
84	2022-11-26 10:31:35.231415	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
85	2022-11-26 10:31:35.300966	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
86	2022-11-26 10:31:35.371575	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
87	2022-11-26 10:31:35.424564	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
88	2022-11-26 10:31:35.486957	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
89	2022-11-26 10:31:35.548737	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
90	2022-11-26 10:31:35.612857	192.168.26.31	255.255.255.255	UDP	48	2000 → 1000 Len=6
92	2022-11-26 10:31:35.742739	192.168.26.31	255.255.255.255	UDP	49	2000 → 1000 Len=7
96	2022-11-26 10:31:35.799889	192.168.26.31	255.255.255.255	UDP	49	2000 → 1000 Len=7

> Frame 81: 48 bytes on wire (384 bits), 48 bytes captured (384 bits) on interface \Device\NPF\_{A5DC84...}

> Ethernet II, Src: LiteonTe\_e7:64:db (80:30:49:e7:64:db), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

> Internet Protocol Version 4, Src: 192.168.26.31, Dst: 255.255.255.255

> User Datagram Protocol, Src Port: 2000, Dst Port: 1000

Source Port: 2000Destination Port: 1000Length: 14Checksum: 0x2060 [unverified][Checksum Status: Unverified][Stream index: 3]> [Timestamps]UDP payload (6 bytes)> Data (6 bytes)

其中通信五元组信息包含在如下所示位置：

Source Address: 192.168.26.31  
Destination Address: 255.255.255.255

> User Datagram Protocol, Src Port: 2000, Dst Port: 1000

Source Port: 2000Destination Port: 1000Length: 14Checksum: 0x2060 [unverified][Checksum Status: Unverified][Stream index: 3]

由此可以看出通信五元组为：  
源 IP 地址：192.168.26.31，即客户端 IP 地址  
目的 IP 地址：255.255.255.255，即广播地址  
协议号：UDP  
源端口号：2000  
目的端口号：1000

UDP 数据帧首部格式为：

用户数据包协议		
偏移位	0~15	16~31
0	源端口	目标端口
32	数据包长度	校验和
64+	数据（如果有）	

其中数据包长度为：14  
校验和为：0x2060

另外，由下图可以得出目的 MAC 地址以及源 MAC 地址：

```
Ethernet II, Src: LiteonTe_e7:64:db (80:30:49:e7:64:db), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: LiteonTe_e7:64:db (80:30:49:e7:64:db)
  Type: IPv4 (0x0800)
```

即源 MAC 地址为 80:30:49:e7:64:db，目的 MAC 地址为 ff:ff:ff:ff:ff:ff  
其中源 MAC 地址为客户机 MAC 地址，目的 MAC 地址为广播地址

## 六、 实验总结

由实验结果可以看出，广播 UDP 用户数据报与单播用户数据报的不同之处在于广播报文中，目的 IP 地址为 255.255.255.255，目的 MAC 地址为 ff:ff:ff:ff:ff:ff，而单播报文中，目的 IP 地址为服务器 IP 地址，目的 MAC 地址为下一跳主机或路由器的 MAC 地址。但在源 IP 地址和源 MAC 地址以及协议类型上，广播报文与单播报文是相同的。

IP 多播首先要知道的是只有 UDP 有多播，而没有 TCP 多播。这是因为多播的重点是高效的把同一个包尽可能多的发送到不同的，甚至可能是未知的设备。但是 TCP 连接可能要求丢包重发或者延时或重组顺序，这些操作可能非常消耗资源，不适用于许多使用多播的应用场景。（同时多播不知道发出的包是不是已经到达，这个也导致不能使用 TCP）。

**教师评语：**

**成绩：** \_\_\_\_\_ **教师签名：** \_\_\_\_\_ **批阅日期：** \_\_\_\_\_