**《计算机网络》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验名称** | IP 和 TCP 数据分组的捕获和解析 | | **学 院** | 计算机学院（国家示范性软件学院） | **指导教师** | **程莉** |
| **班 级** | **班内序号** | **学 号** | | **学生姓名** | **成绩** | |
| **2022211304** |  | **2022211119** | | **赵宇鹏** |  | |
| **实**  **验**  **内**  **容** | 1）捕获在使用网络过程中产生的分组（packet）： IP 数据包、ICMP 报文、DHCP 报文、TCP 报文段。  2）分析各种分组的格式，说明各种分组在建立网络连接和通信过程中的作用。  3）分析 IP 数据报分片（片段）的结构：理解长度大于 1500 字节 IP 数据报分片传输的结构  4）分析 TCP 建立连接、拆除连接和数据通信的过程。 | | | | | |
| **学生**  **实验**  **报告**  （附页） |  | | | | | |
| **实**  **验**  **成**  **绩**  **评**  **定** | **评语**:  **成绩**:  助教签名：  年 月 日 | | | | | |

注：评语要体现每个学生的工作情况，可以加页。



**计算机网络实验报告**



**题目：IP 和 TCP 数据分组的捕获和解析**

**班 级： 2022211304 \_\_\_**

**学 号： 2022211119\_\_\_\_\_**

**姓 名： 赵宇鹏\_\_\_\_\_\_\_\_\_**

**学 院：计算机学院（国家示范性软件学院）**

**2024年 5 月 29 日**

**一、本次实验内容：**

1）捕获在使用网络过程中产生的分组（packet）： IP 数据包、ICMP 报文、DHCP 报文、TCP 报文

段。

2）分析各种分组的格式，说明各种分组在建立网络连接和通信过程中的作用。

3）分析 IP 数据报分片（片段）的结构：理解长度大于 1500 字节 IP 数据报分片传输的结构

4）分析 TCP 建立连接、拆除连接和数据通信的过程。

**二、实验设备环境：**

1 台装有 Windows 操作系统的 PC 机，要求能够连接到 Internet，并安装 Wireshark 软件。

**三、实验步骤 ：**

**准备工作** ：

1）启动计算机，连接网络确保能够上网（为便于查找数据包和分析，不运行其他网络应用程序）。

2）运行 wireshark，选择活跃的网卡（网络接口）。

1、捕获 DHCP 报文并分析：

1）在 Wireshark 中设置捕获过滤器：udp port 67，开始监控。

2）运行 cmd，进入命令行窗口，使用命令：C:>ipconfig /release，释放主机的 IP 地址，断网；

此时 wireshark 的主窗口将显示一条 DHCP 消息。

3）使用命令：C:>ipconfig /renew，重新分配 IP 地址，此时 wireshark 的主窗口将显示 4 条 DHCP 消息。

4）停止捕获，保存捕获数据。

5）分析捕获到的 DHCP 报文的格式，理解 DHCP 的功能和分配 IP 地址的过程。

DHCP（动态主机配置协议）是一个应用层协议。当我们将客户主机ip地址设置为动态获取方式时，DHCP服务器就会根据DHCP协议给客户端分配IP，使得客户主机能够利用这个IP上网。DHCP通常被用于局域网环境，主要作用是集中的管理、分配IP地址，使客户端动态地获得IP地址、Gateway地址、DNS服务器地址等信息，并能够提升地址的使用率。

DHCP的实现分为4步，分别是：

第一步：客户端在局域网内发起一个DHCP　Discover包，目的是想发现能够给它提供IP的DHCP 服务端。

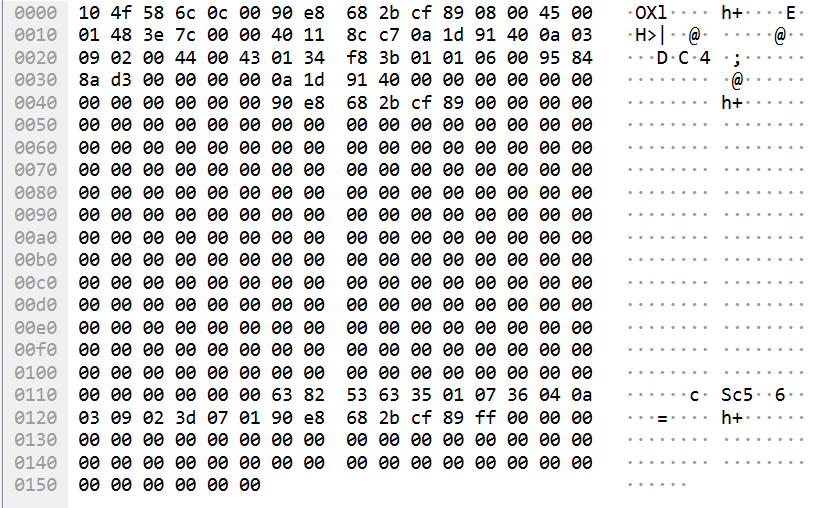
第二步：可用的DHCP 服务端接收到Discover包之后，通过发送DHCP Offer包给予客户端应答，意在告诉客户端它可以提供IP地址。

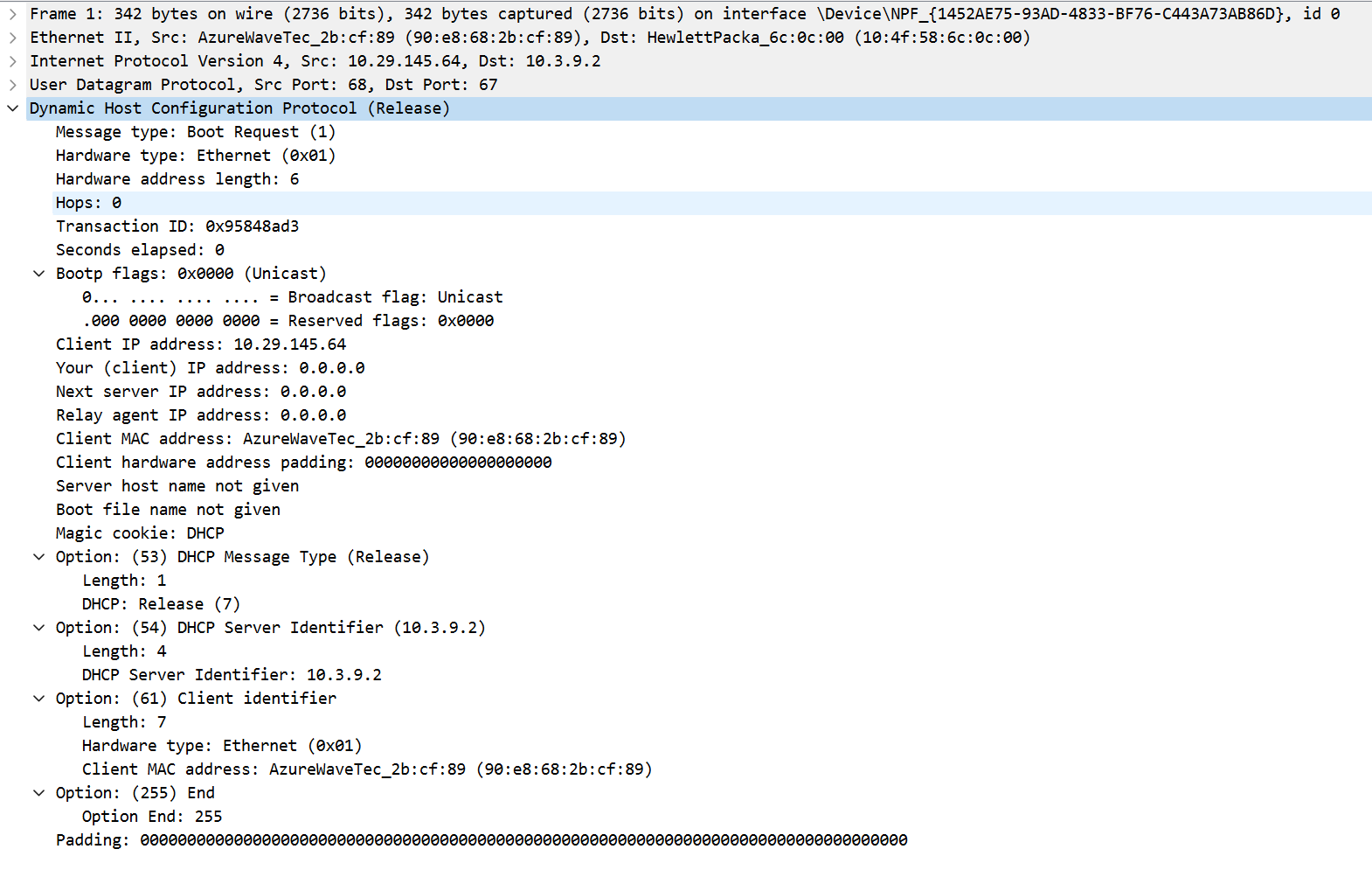
第三步：客户端接收到Offer包之后，发送DHCP Request包请求分配IP。

第四步：DHCP 服务端发送ACK数据包，确认信息，客户端在接收到这个报文之后才会确认分配给它的IP和其他信息可以被允许使用。

现在开始对捕获到的报文进行分析：

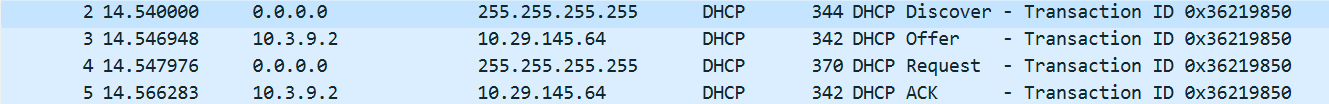
DHCP Release:



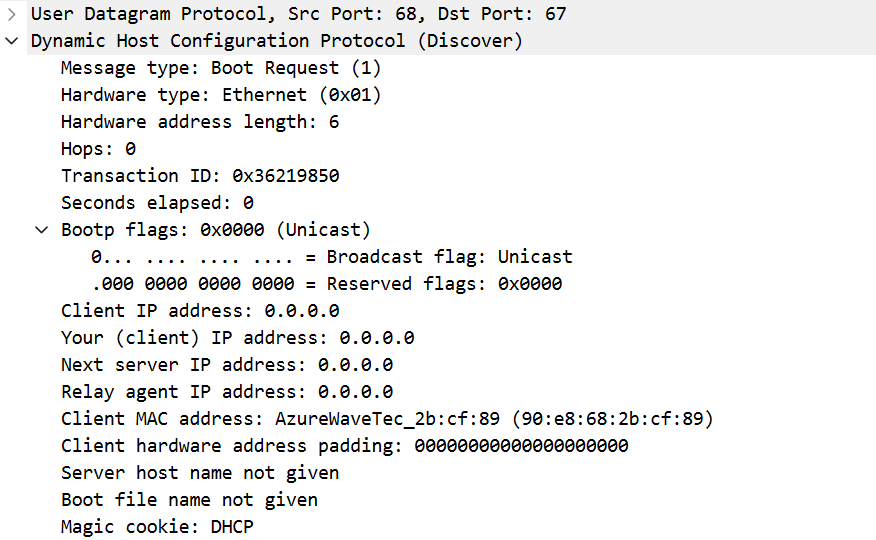


进行释放主机地址操作

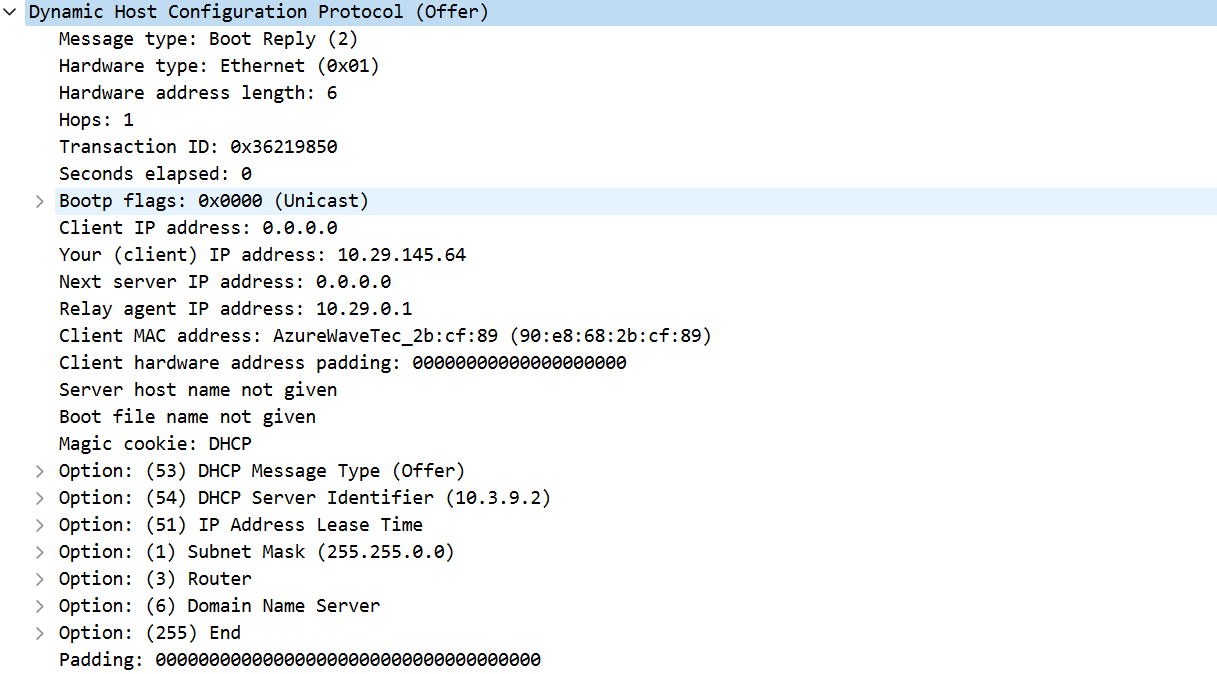
继续在命令提示符窗口输入ipconfig /renew 请求网络连接，得到

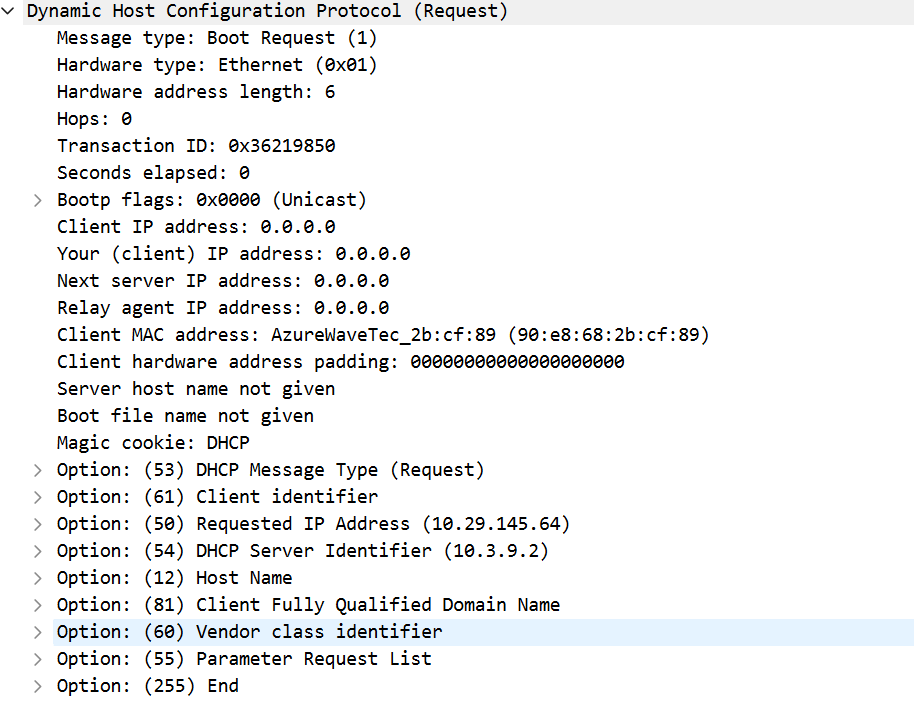


客户端使用IP地址0.0.0.0发送了一个广播包，此时的目的IP地址为255.255.255.255，客户端通过这个广播包来找到可以为它提供服务的DHCP服务器

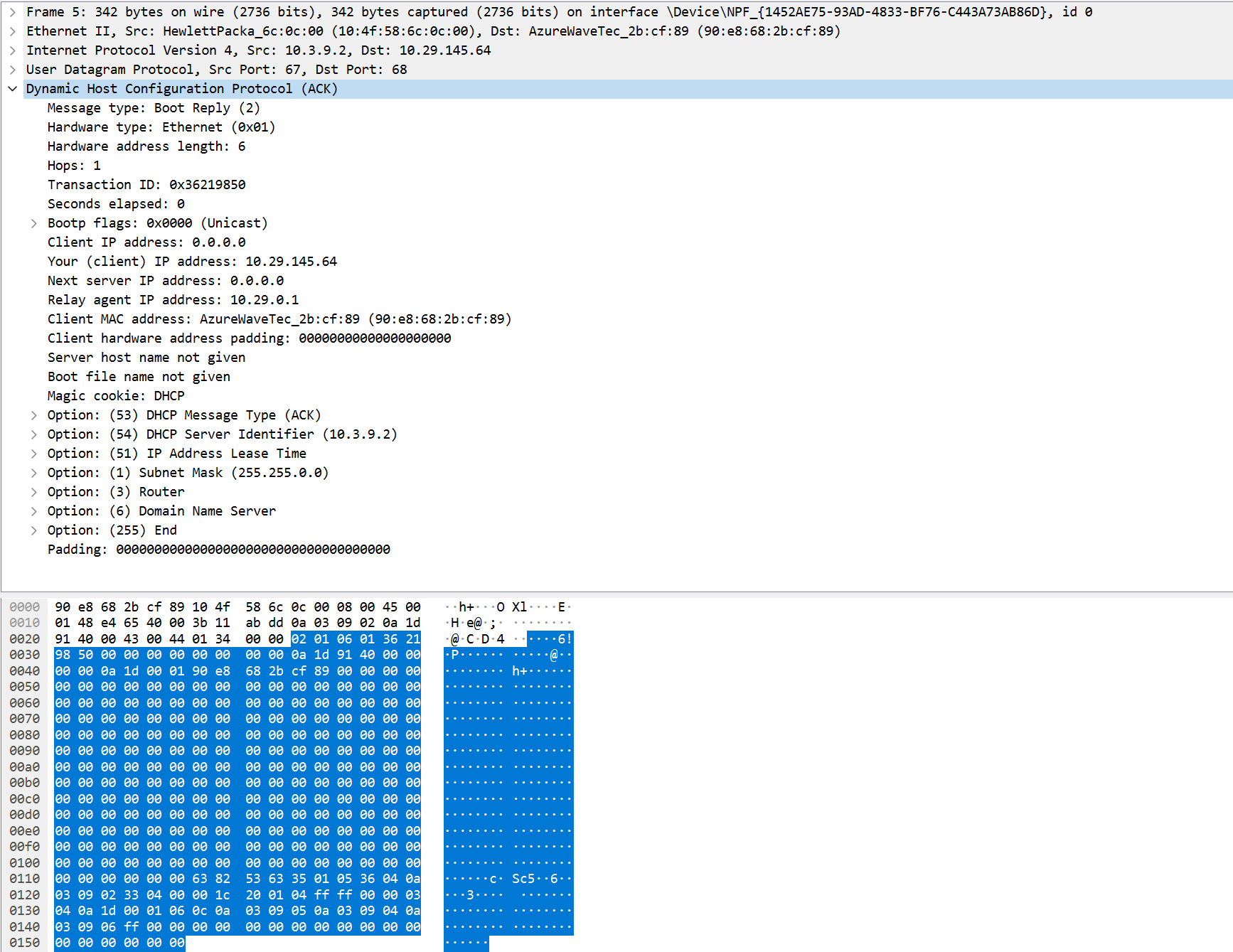


在这个广播包中有客户的MAC地址，并且可以看到目的端口是67，当DHCP服务器收到一条DHCP Discover数据包时，会为客户端分配 IP，并向之前的 MAC 地址发送 DHCP Offer 分组，分组内容中有分配的 IP 地址，分组的 OPTION 字段中附带服务器标识符、IP 租期、子网掩码、路由器以及域名服务器等信息。



此后，如果有多个DHCP服务器，那么客户端就会收到多个DHCP Offer组，因此客户端仍然会发送广播，发送一个DHCP Request分组，指明请求的IP地址和服务器标识符

最后，服务器向客户端发送DHCP ACK进行确认，完成IP地址的分配



2、 发送 ICMP 报文，捕获并分析格式。

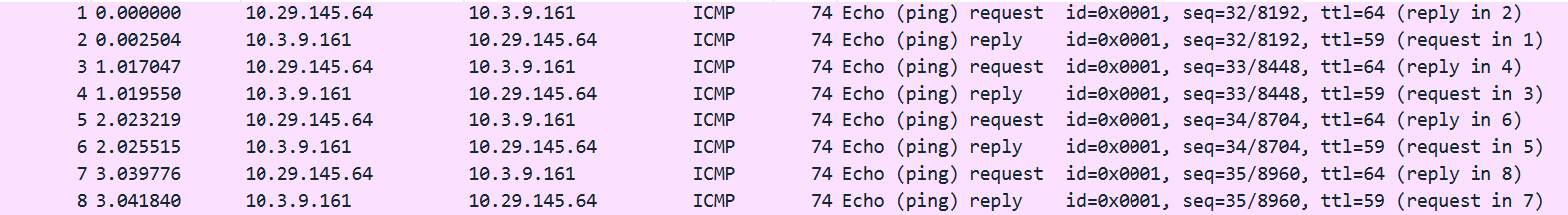
1） 开启 wireshark，在过滤器输入icmp使其只显示ICMP报文，开始监控，在命令行窗口使用 ping [www.bupt.edu.cn](http://www.bupt.edu.cn)命令, 捕获 ICMP 报文。

2） 停止监控，分析其格式。

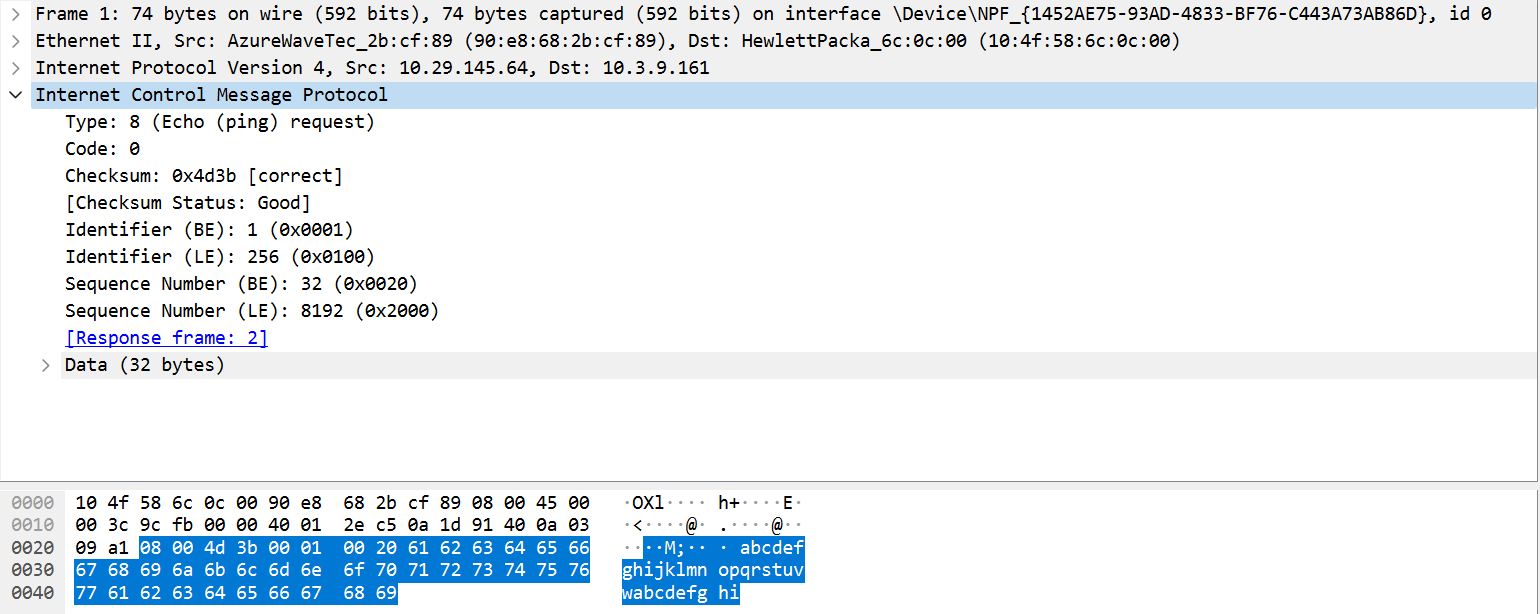
注：可以不设置捕获过滤器，而在主窗口设置显示过滤器为：icmp，从而只显示 icmp 报文，不显示

其他协议数据。

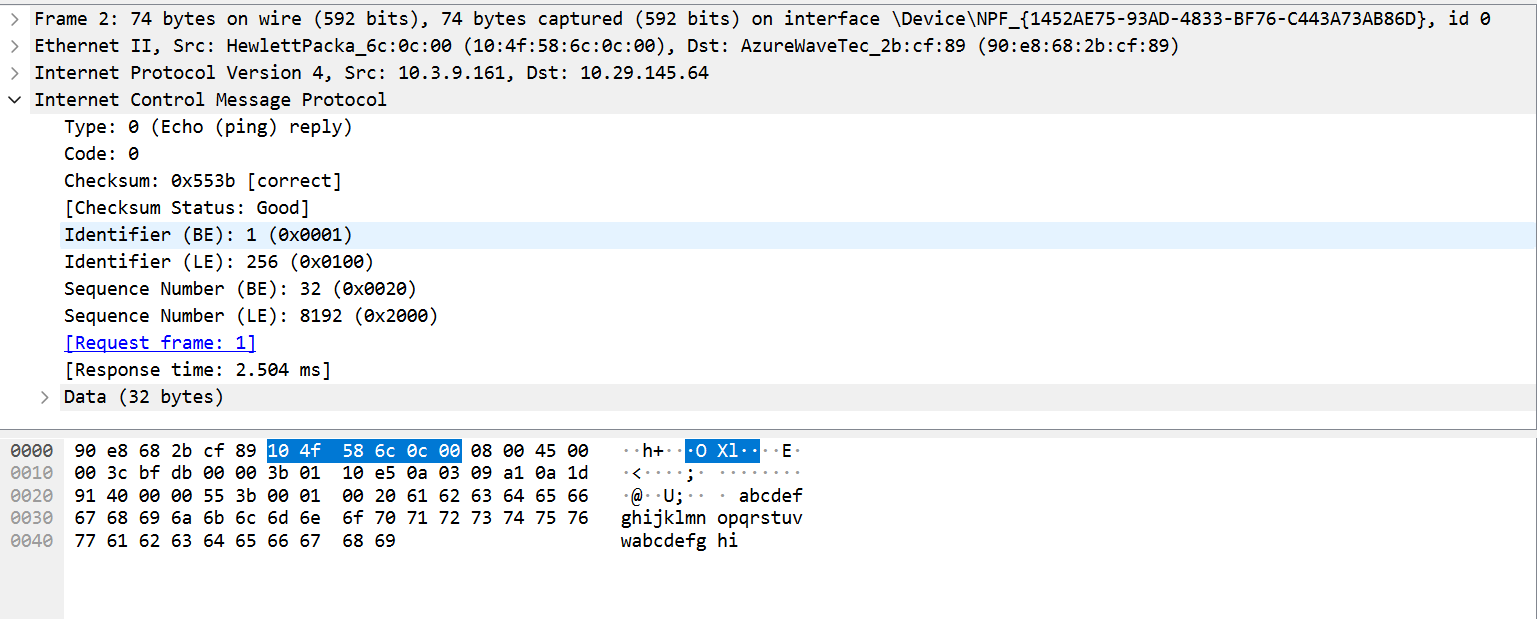
共获得请求和回复ICMP报文



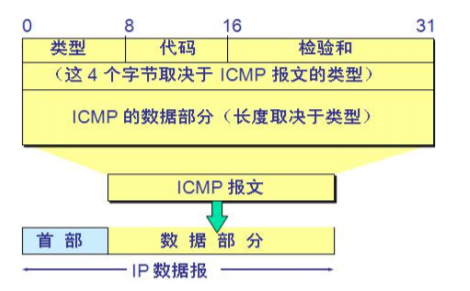
第一组Request的ICMP



第一组Reply的ICMP



ICMP的分组格式如下：



由以上格式我们可以得出ICMP报头格式如下：

|  |  |  |
| --- | --- | --- |
| 字段（字节数） | 内容 | 解释 |
| Type(1) | 08H | 指示 ICMP 消息的类型为回显请求（Echo Request） |
| Code(1) | 00H | 代码，与 Type 字段一起使用，提供特定类型的附加信息。 |
| Checksum(2) | 4d 3bH | 校验和，用于错误检测，确保消息的完整性。 |
| Identifier(2) | 00 01H | 标识符，用于区分不同进程 ping 消息 |
| Sequence Number(2) | 00 20H | ping 请求序号 |

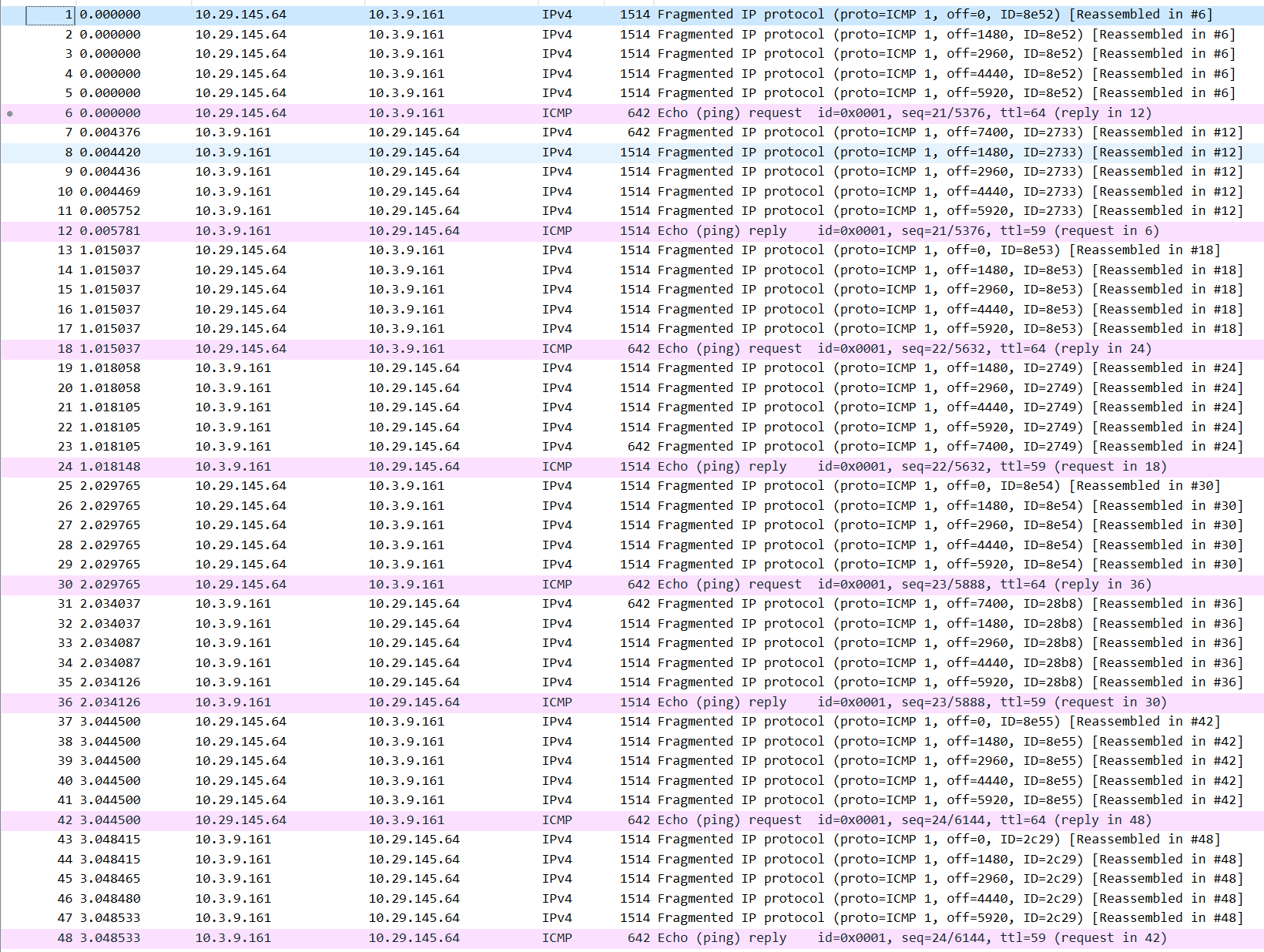
而Request相应的Reply是对Request的回复，分组头除了Type变成0表示回显应答以及校验和发生相应变化外与Request的报头没有其他区别。

3、 分析 IP 数据报的分片传输过程

1） 通过ping [www.bupt.edu.cn获得其IP](http://www.bupt.edu.cn获得其IP)地址为10.3.9.161，打开Wireshark，开启抓包，在过滤器一栏输入 ip.src eq 10.3.9.161

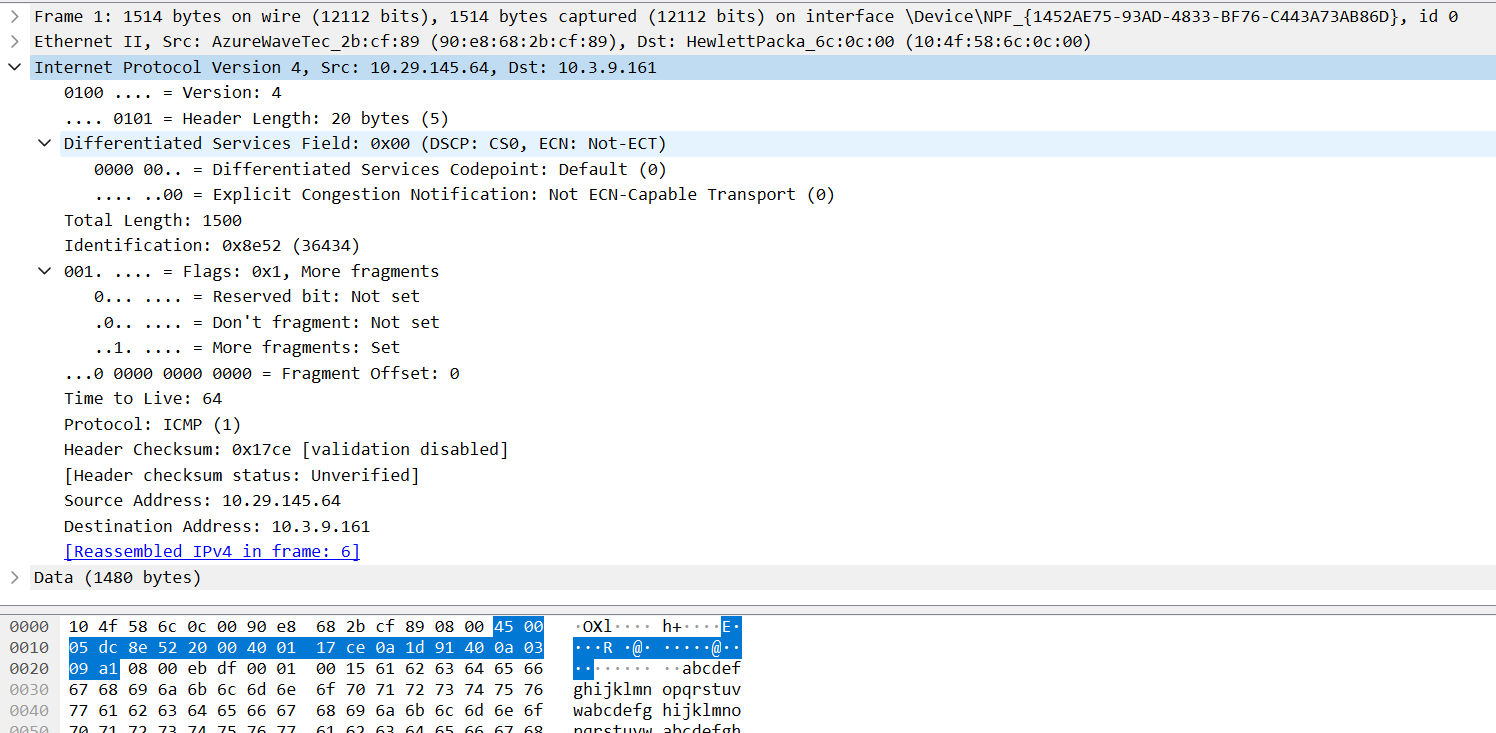
2）运行 cmd，进入命令行窗口，使用命令：C:>ping -l 8000 www.bupt.edu.cn，制作 8000 字节的 IP 数据报并发送，捕获后分析其分片传输的包结构。

在过滤器中输入ip.src eq 10.3.9.161过滤出48个分组，共计4对请求与回复



由于前面分析请求与回复的报文差别不大，因此此处只拿第一对请求与回复中的请求报文进行分析

第一个分片



|  |  |  |
| --- | --- | --- |
| 字段（位数） | 内容(默认二进制) | 解释 |
| Version(4) | 0100 | 版本：4 |
| Header Length(4) | 0101 | 头部长度20字节 |
| DSCP(6) | 0000 00 | 区分服务信息：默认 |
| ECN(2) | 00 | 显式拥塞通知：否 |
| Total Length(16) | 05 dcH | 总长度：1500字节 |
| Identification(16) | 8e 52H | 分组标识：0x8e52 |
| Flags(3) | 001 | DF：0允许分片  MF：1后面还有其他分片 |
| Fragment Offset(13) | 0 0000 0000 0000 | 分片偏移量：0 |
| TTL(8) | 40H | 生存期：64跳 |
| Protocol(8) | 01H | 协议：ICMP |
| Header CheckSum(16) | 17 ceH | 头部校验和 |
| Source Address(32) | 0a 1d 91 40H | 源地址：10.29.145.64 |
| Destination Address(32) | 0a 03 09 a1H | 目的地址：10.3.9.161 |

前六个分片的IP包头大部分一致，不一致处见下表：

|  |  |  |
| --- | --- | --- |
| 序号 | MF | 偏移量 |
| 1 | 1 | 0 |
| 2 | 1 | 0 0000 1011 1001 = 185->185\*8=1480 |
| 3 | 1 | 370 \* 8 = 2960 |
| 4 | 1 | 555 \* 8 = 4440 |
| 5 | 1 | 740 \* 8 = 5920 |
| 6 | 0 | 925 \* 8 = 7400 |

IP报头长度为20字节，而以太网数据链路层的MTU为1500字节，除去报头后剩余1480字节，正好为8的倍数，可以分配，所以8000字节的数据包被拆分成6个分片，前五个长度为1480字节，第六个长度为600字节，分片正确，通过Identification来判断是不是同一数据包，通过MF标志位为0来确定数据的结尾。

4、 捕获建立连接和释放连接过程的 TCP 报文段并分析

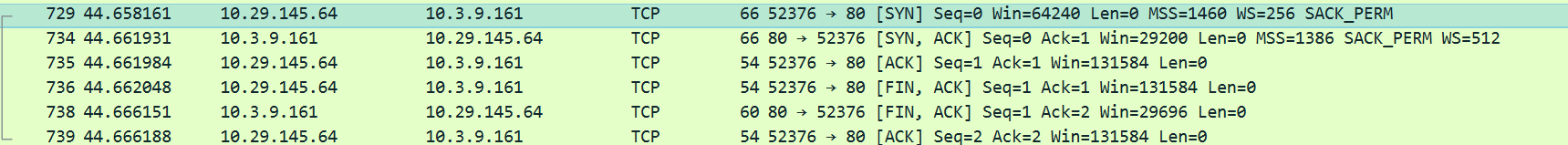
开启 wireshark 监控，打开一个网页，停止监控。

分析捕获到的 TCP 报文段，画出建立连接和释放连接过程的消息序列图。

捕获方法及过程：

开启wireshark监控，打开[www.bupt.edu.cn](http://www.bupt.edu.cn)网页，显示过滤器输入tcp && ip.addr == 10.3.9.161显示相应TCP信息，之后关闭网页，通知监控。

三次握手四次挥手如下：



三次握手流程如下：

第一次握手：客户端给服务端发送一个SYN报文，表示希望建立连接，并在SYN包中放入初始序列号（seq = x，此处为0），在报文中SYN = 1，seq = x。

第二次握手：服务端收到客户端发送的SYN报文后，会回复一个SYN-ACK包，表示同意建立连接，并在SYN-ACK包中放入自己的初始序列号（seq=y，此处为0）和对客户端序列号的确认号（ack=x+1），表示自己已经收到了客户端的SYN，在报文中,SYN = 1, ACK = 1, ack = x + 1, seq = y.

第三次握手：客户端收到服务端发送的SYN-ACK报文后，会发送一个ACK报文，确认服务器的序列号（ack=y+1），表示连接建立成功。。

三次握手中第一次握手：客户端发送网络包，服务端收到了，这样服务端就能得出结论：客户端的发送能力、服务端的接收能力是正常的。第二次握手：服务端发包，客户端收到了，这样客户端就能得出结论：服务端的接收、发送能力，客户端的接收、发送能力是正常的。不过此时服务器并不能确认客户端的接收能力是否正常。第三次握手：客户端发包，服务端收到了，这样服务端就能得出结论：客户端的接收、发送能力正常，服务器自己的发送、接收能力也正常。

ACK = 1, seq = x + 1, ack = y + 1

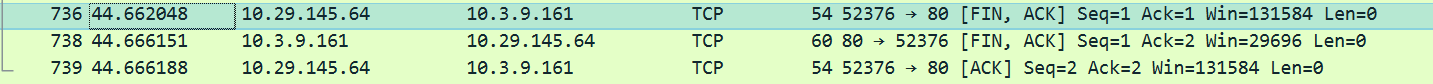
SYN = 1, ACK = 1, seq = y, ack = x + 1

SYN = 1, seq = x

服务端

客户端

四次挥手：



在这里服务端把第二次挥手和第三次挥手合并在一起。

第一次挥手：客户端发送一个FIN报文（此处发送的是FIN-ACK报文），报文中会指定一个序列号（seq =u），表示不再发送数据了，但仍可以接收数据，等待服务端确认。

第二次挥手：服务端收到 FIN 之后，会发送 ACK 报文，且把客户端的序列号值 +1 作为 ACK 报文的序列号值，表明已经收到客户端的报文，服务端收到连接释放报文段后即发出确认报文（ACK=1，ack=u+1，seq=v），客户端到服务端的连接释放。

第三次挥手：服务端主动断开连接，和客户端的第一次挥手一样，发送 FIN 报文，且指定一个序列（seq = w）在这里FIN = 1,ACK = 1, seq = w, ack = u + 1。

第四次挥手：客户端收到 FIN 之后，一样发送一个 ACK 报文作为应答，且把服务端的序列号值 +1 作为自己 ACK 报文的序列号值（seq = u + 1, ack = w + 1），连接完全关闭。

FIN = 1, ACK = 1, seq = w, ack = u + 1

ACK = 1, seq = v, ack = u + 1

FIN = 1, seq = u

ACK = 1, seq = u + 1, ack = w + 1

通过上面的分析我们可以看到第二次挥手和第三次挥手可以合并在一起以提高信道利用率。

为了保证客户端发送的最后一个ACK报文段能够到达服务器发挥作用，我们设定等待计时器2MSL，当客户端发送完ACK后等待2MSL再完全关闭连接。因为这个ACK有可能丢失，从而导致服务器收不到对FIN-ACK的确认报文。服务器会超时重传FIN-ACK，接着客户端再重传一次确认，重新启动时间等待计时器。最后客户端和服务器都能正常的关闭。假设客户端不等待2MSL，而是在发送完ACK之后直接释放关闭，一但这个ACK丢失的话，服务器就无法正常的进入关闭连接状态。

**实验结论和实验心得**

完成本次实验大概花费了3小时，其中主要时间花费在查询报文格式以及了解报文内容上，由于做本实验时课程内容还没讲到TCP的连接与释放，所以也花费了较多时间查询学习了三次握手四次挥手的相关内容。

我学习并且灵活运用了 Wireshark 的筛选功能，从诸多报文中筛选需要的一部分，节省了许多时间。在本次实验中，我通过将Wireshark所抓包与课程中的理论知识相结合，对DHCP，ICMP，IP以及TCP协议报文的内容和部分功能有了进一步的认识，对其计算机网络的交互原理有了更深刻的理解。