



# 目录

<b>1 实验总体概述.....</b>	<b>3</b>
1.1 实验类别.....	3
1.2 实验内容 .....	3
1.3 实验目的 .....	3
1.4 实验环境 .....	3
<b>2 实验步骤与分析.....</b>	<b>3</b>
2.1 捕获在连接 Internet 过程中产生的网络层分组.....	3
2.2 ICMP 数据分组.....	9
2.3 发送指定长度的数据分组 .....	13
<b>3 实验总结 .....</b>	<b>16</b>

# 1 实验总体概述

## 1.1 实验类别

协议分析型。

## 1.2 实验内容

- 1) 捕获在连接 Internet 过程中产生的网络层分组：DHCP 分组，ARP 分组，IP 数据分组，ICMP 分组；
- 2) 分析各种分组的格式，说明各种分组在建立网络连接过程中的作用；
- 3) 分析 IP 数据分组分片的结构；
- 4) 分析 TCP 建立连接，拆除连接和数据通信的流程。

## 1.3 实验目的

通过本次实验了解计算机上网的工作过程，学习各种网络层分组的格式及其作用，理解长度大于 1500 字节 IP 数据组分片传输的结构。

## 1.4 实验环境

本次实验的环境为 1 台装有 Windows 10 操作系统的 PC 机，能够连接到 Internet，并安装了 WireShark 软件。

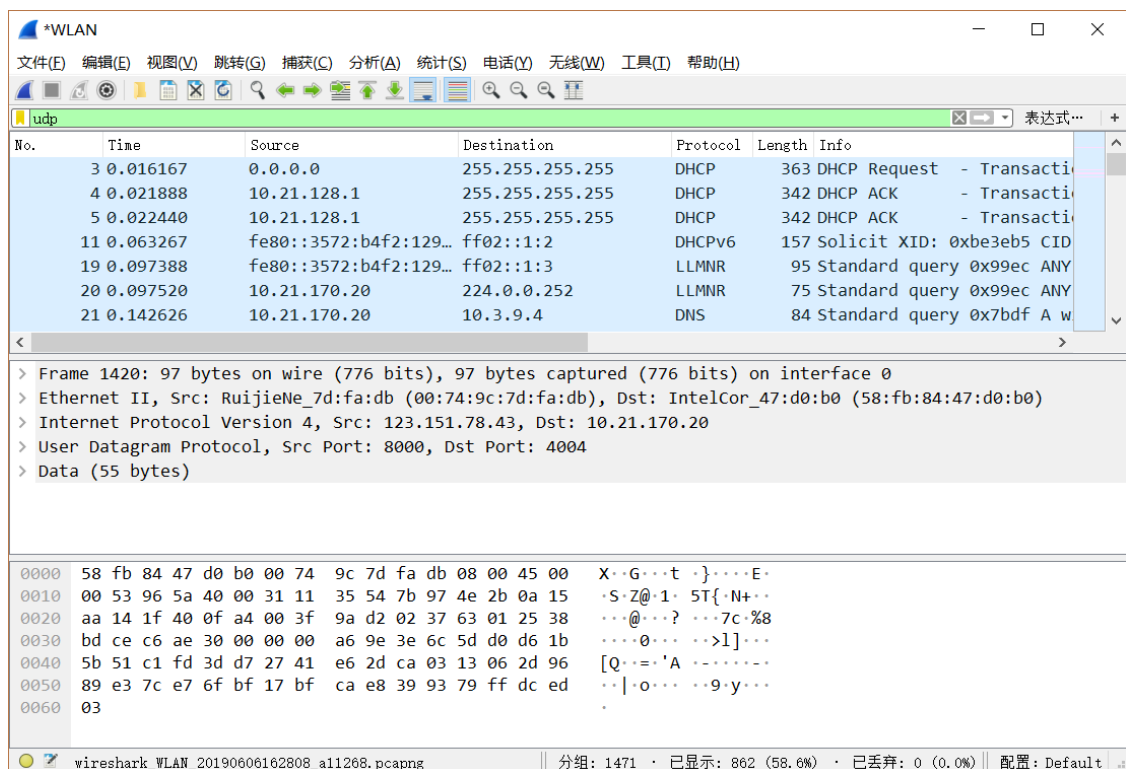
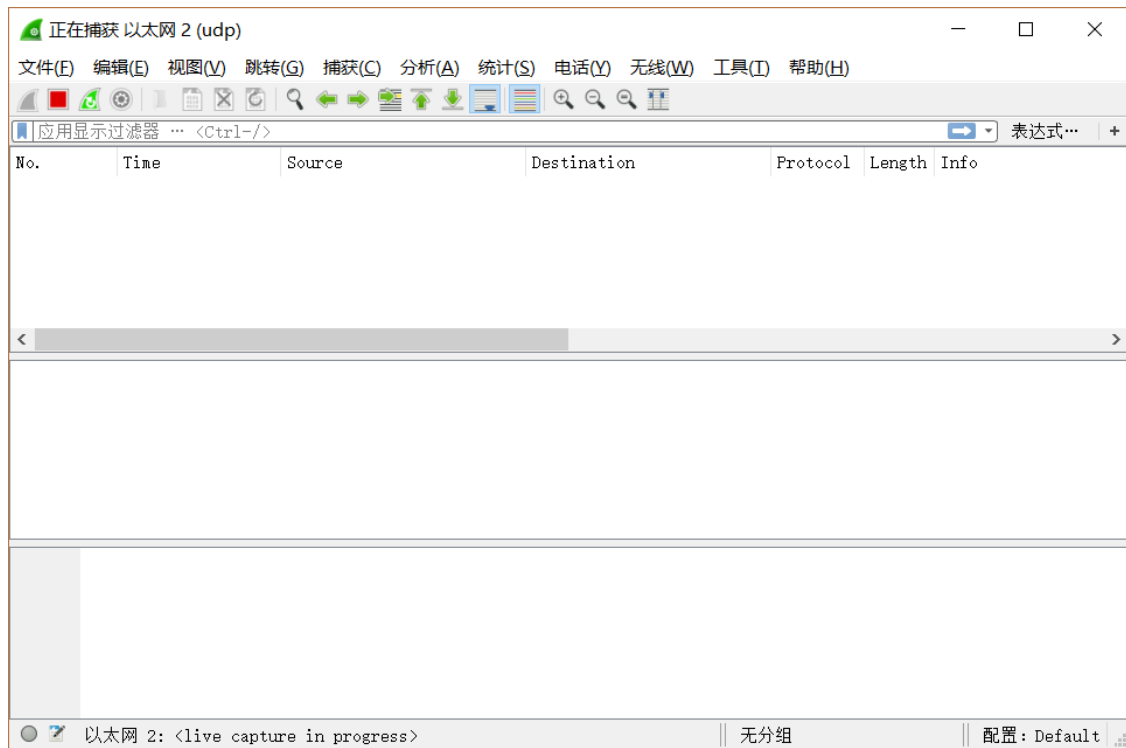
# 2 实验步骤与分析

## 2.1 捕获在连接 Internet 过程中产生的网络层分组

- (1) 准备工作：将PC机的网卡禁用，断开网络；



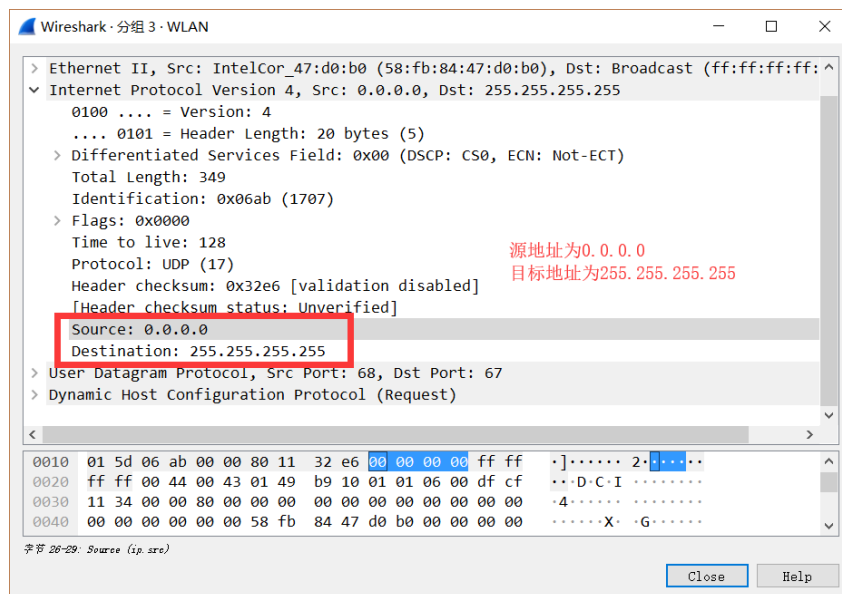
(2) 开启WirkShark监控，设置捕获过滤器，仅捕获UDP报文。开启网卡，连接网络，一段时间后发现WirkShark捕获了如下所示的数据包：



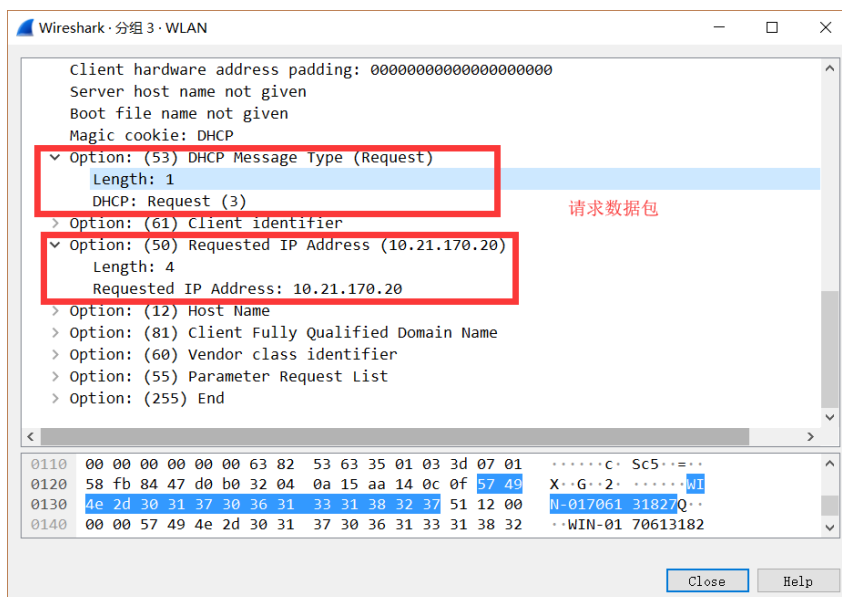
## 1. DHCP分组

DHCP动态主机设置协议（Dynamic Host Configuration Protocol）是一个局域网的网络协议，使用UDP协议工作，主要有两个用途：给内部网络或网络服务提供商自动分配IP地址、给内部网络管理员作为对所有计算机作中央管理的手段。

由捕获的数据包可见，其IP头部的目标地址为ff ff ff ff，即表明该包为一个广播包，同时可以看到其源地址为00 00 00 00。



根据DHCP的数据包部分的译码输出，我们可以得到DHCP Message Type域为1（表明是申请IP地址），以及硬件地址类型和硬件地址长度等信息，并且最终申请的IP地址为10.21.170.20。



可以看到网关在之后发来了DHCP ACK数据包，用于确认IP地址的分配。

Source	Destination	Protocol	Length	Info
0.0.0.0	255.255.255.255	DHCP	363	DHCP Request - Transaction ID 0xd9cf1134
10.21.128.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xd9cf1134
10.21.128.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0xd9cf1134
fe80::3572:b4f2:129... ff02::1:2		DHCPv6	157	Solicit XID: 0xbe3eb5 CID: 0001000120d17875c85

不仅如此，我们还可以发现，DHCP服务使用了UDP的67与68端口。其中客户端向68端口广播请求配置，而服务器向67端口广播回应请求。

```

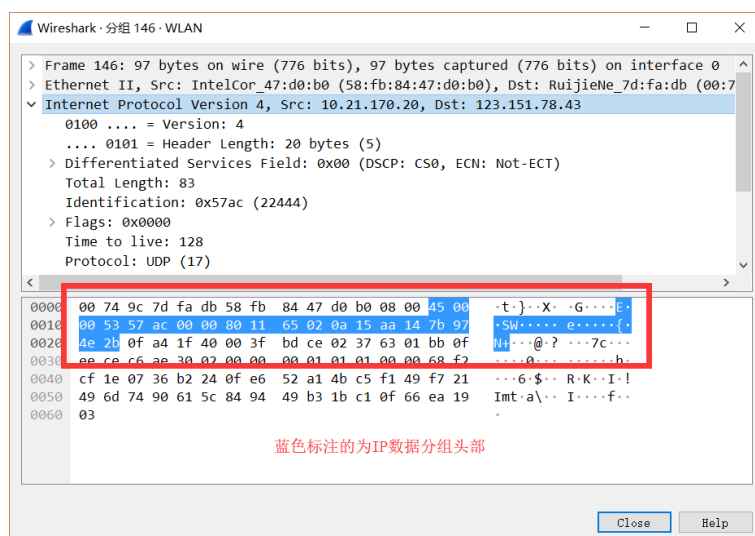
User Datagram Protocol, Src Port: 68, Dst Port: 67
  Source Port: 68
  Destination Port: 67
  Length: 329
  Checksum: 0xb910 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  > [Timestamps]
  
```

通过多次对DHCP数据包的查看与分析，我们可以看到DHCP服务动态申请IP地址的4个步骤：

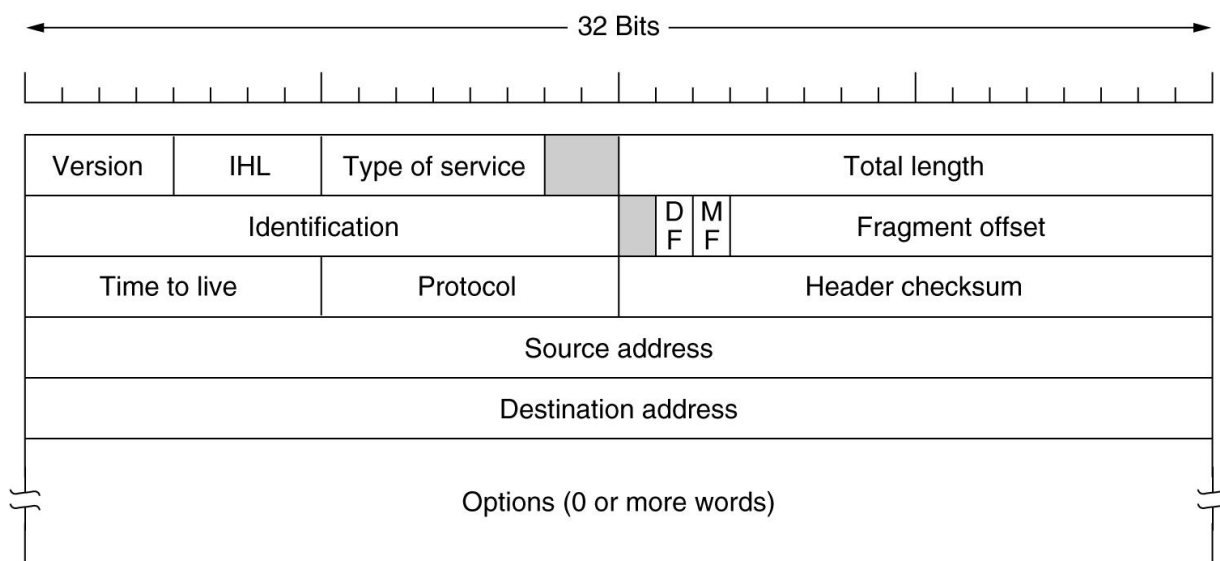
- ① 客户机请求IP（DHCP DISCOVER广播包）；
- ② Server响应（DHCP OFFER广播包）；
- ③ 客户机选择IP（DHCP REQUEST广播包）；
- ④ Server确定租约（DHCP ACK/DHCP NAK广播包）。

## 2. IP数据分组

本次实验捕获的一个IP数据分组如下所示：



IPv4 协议下分组的首部各字段如下所示：



Internet Protocol Version 4, Src: 10.21.170.20, Dst: 123.151.78.43

```

0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 83
  Identification: 0x57ac (22444)
> Flags: 0x0000
  Time to live: 128
  Protocol: UDP (17)
  Header checksum: 0x6502 [validation disabled]
  [Header checksum status: Unverified]
  Source: 10.21.170.20
  Destination: 123.151.78.43
  
```

对捕获的 IP 数据分组的首部进行分析，可以得到如下所示结果：

字段	报文（16 进制）	内容
首部长度	5	报头长度为 20 字节
服务类型	00	正常时延、正常吞吐量、正常可靠性
总长度	00 53	分组长度 83 字节
标识	57 ac	标识为 22444
标志	00 00	DF=MF=0，允许分片，且后面没有其他分片
偏移值	00	偏移量为 0
生存周期	80	每跳生存周期为 128 秒
协议	11	十进制为 17，表示携带的数据来自 UDP 协议
头部校验和	65 02	头部校验和为 25858
源地址	0a 15 aa 14	源地址为 10.21.170.20
目的地址	7b 97 4e 2b	目的地址为 153.121.78.43

### 3. ARP数据分组

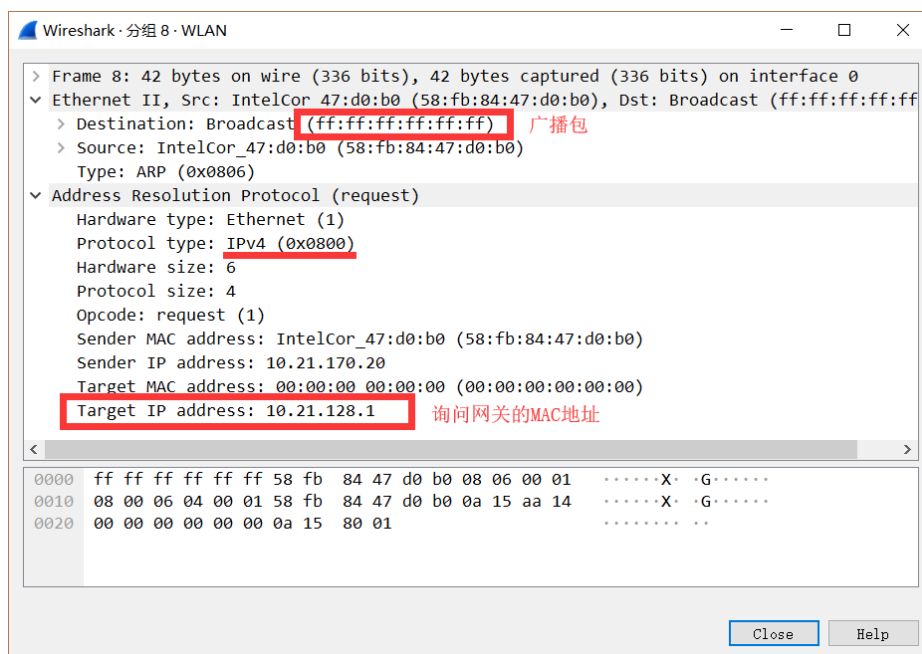
ARP，是一种地址解析协议，实现通过主机的IP地址得知其物理地址。在TCP/IP网络环境下，每个主机都各自拥有一个32位的IP地址，这种互联网地址是在网际范围标识主机的一种逻辑地址。为了让报文在物理网路上传送，必须知道对方目的主机的物理地址。这样就存在着把IP地址变换成物理地址的地址转换问题。

以以太网环境为例，为了正确地向目的主机传送报文，必须把目的主机的32位IP地址转换成为48位以太网的地址。这就需要在互连层有一组服务将IP地址转换为相应物理地址，这组协议就是ARP协议。

在捕获的所有数据包中，我们还可以观察到ARP协议的分组，如下图所示：

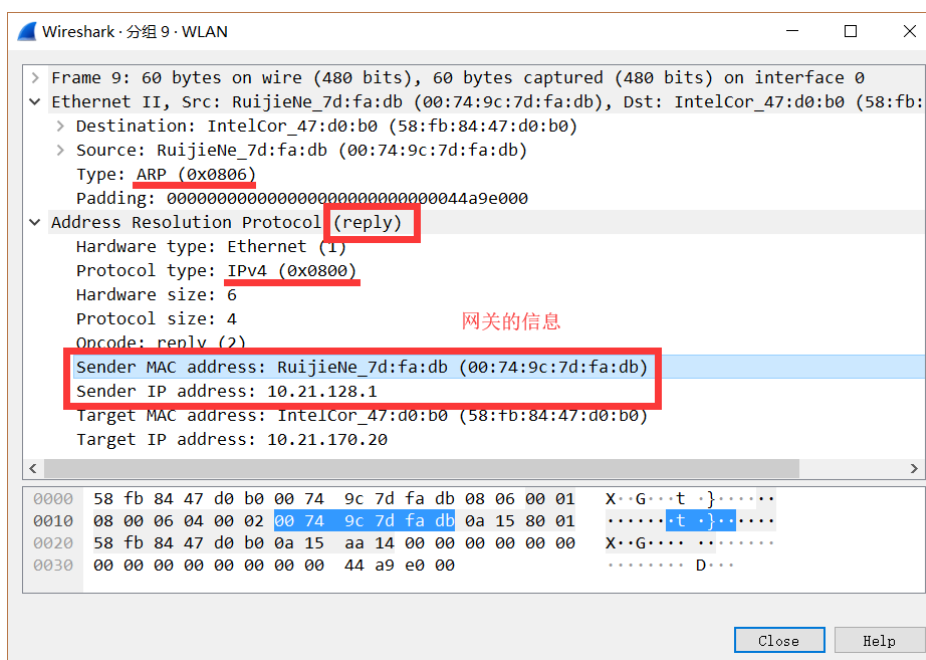
Source	Destination	Protocol	Length	Info
IntelCor_47:d0:b0	Broadcast	ARP	42	Who has 10.21.128.1? Tell 10.21.170.20
RuijieNe_7d:fa:db	IntelCor_47:d0:b0	ARP	60	10.21.128.1 is at 00:74:9c:7d:fa:db
IntelCor_47:d0:b0	Broadcast	ARP	42	Who has 10.21.128.1? Tell 10.21.170.20
RuijieNe_7d:fa:db	IntelCor_47:d0:b0	ARP	60	10.21.128.1 is at 00:74:9c:7d:fa:db

对第一个数据包进行分析，可以发现该数据包是一个广播包，在网络连接初期用于请求网关10.21.128.1的MAC地址。



网关在之后很短的时间内便给出了回应，向主机发送了自己的MAC地址，如图所示：





由于主机和网关在同一网段，所以它们之间使用ARP协议进行信息交换的过程比较简单，也十分迅速。若主机请求另一个网段上某一主机的MAC地址，则信息交换过程将比较复杂，需要经过中间路由器的转发。

#### 4. 总结主机连接至网络的工作过程

通过实验现象与结果可以看出，计算机在连接网络时会向本网段广播ARP请求，以询问DHCP服务器的MAC地址。在获得其MAC地址后，主机和服务器直接方可进行正常的通信。

在连接网络时，计算机还会以广播方式（将IP数据分组中的源地址设置为 0.0.0.0、目的地址设置为255.255.255.255）发送一个DHCP Request 的请求信息，该信息中包含向它所选定的DHCP服务器，请求IP地址10.21.170.20；DHCP服务器接受到主机发出的请求之后，返回DHCP Reply的响应信息，为该主机分配10.21.170.20的IP地址。整个过程共经历了四部分来完成IP地址的动态分配。

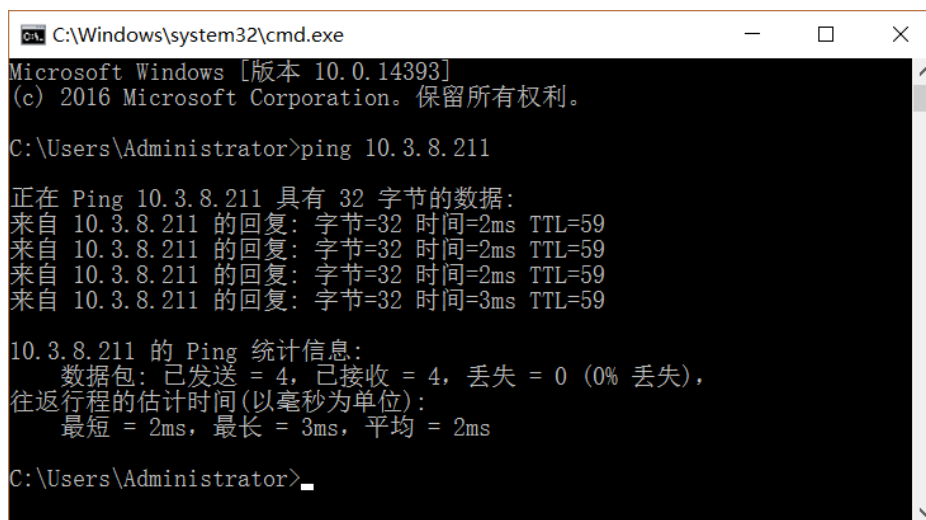
## 2.2 ICMP 数据分组

ICMP（Internet Control Message Protocol）是Internet控制报文协议。它是TCP/IP协议族的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、

主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。

## 1. 使用ping命令产生ICMP分组

将 WireShark 的监控打开后，使用计算机中的命令提示符对校园网内的主机“10.3.8.211”发出ping命令。



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

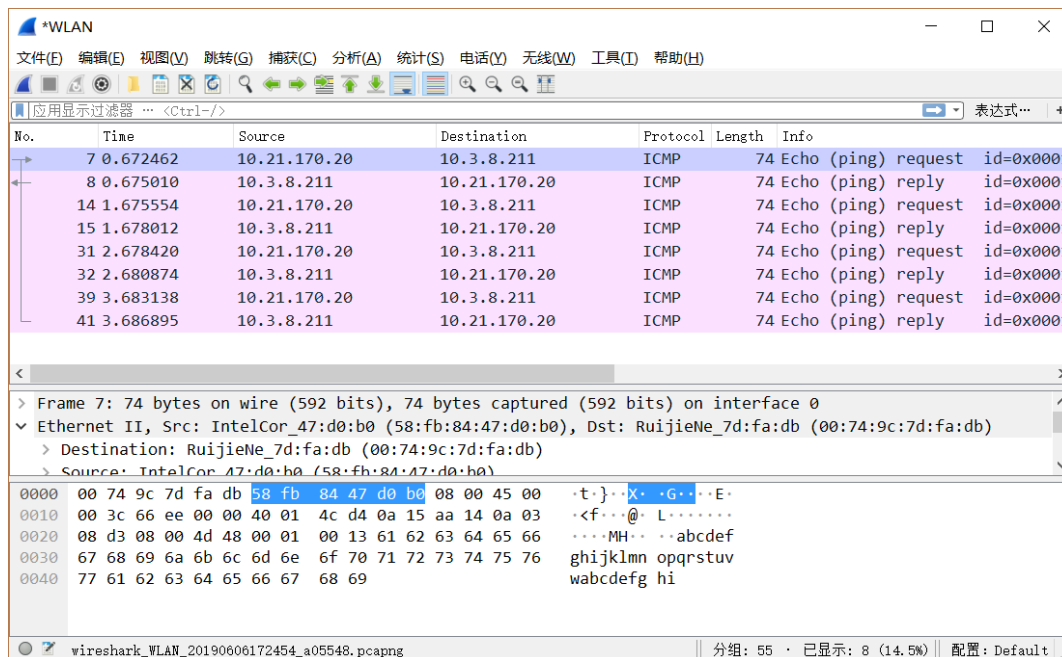
C:\Users\Administrator>ping 10.3.8.211

正在 Ping 10.3.8.211 具有 32 字节的数据:
来自 10.3.8.211 的回复: 字节=32 时间=2ms TTL=59
来自 10.3.8.211 的回复: 字节=32 时间=2ms TTL=59
来自 10.3.8.211 的回复: 字节=32 时间=2ms TTL=59
来自 10.3.8.211 的回复: 字节=32 时间=3ms TTL=59

10.3.8.211 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 = 2ms, 最长 = 3ms, 平均 = 2ms

C:\Users\Administrator>
```

停止监控后，可以看到WireShark软件捕获的ICMP分组如下所示：



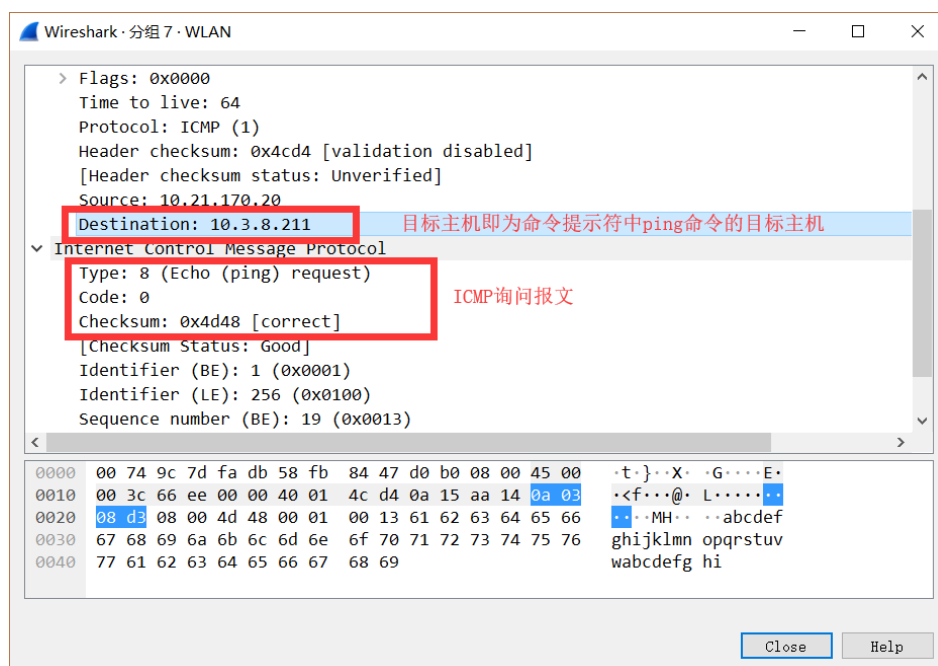
No.	Time	Source	Destination	Protocol	Length	Info
7	0.672462	10.21.170.20	10.3.8.211	ICMP	74	Echo (ping) request id=0x0001
8	0.675010	10.3.8.211	10.21.170.20	ICMP	74	Echo (ping) reply id=0x0001
14	1.675554	10.21.170.20	10.3.8.211	ICMP	74	Echo (ping) request id=0x0001
15	1.678012	10.3.8.211	10.21.170.20	ICMP	74	Echo (ping) reply id=0x0001
31	2.678420	10.21.170.20	10.3.8.211	ICMP	74	Echo (ping) request id=0x0001
32	2.680874	10.3.8.211	10.21.170.20	ICMP	74	Echo (ping) reply id=0x0001
39	3.683138	10.21.170.20	10.3.8.211	ICMP	74	Echo (ping) request id=0x0001
41	3.686895	10.3.8.211	10.21.170.20	ICMP	74	Echo (ping) reply id=0x0001

> Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
v Ethernet II, Src: IntelCor\_47:d0:b0 (58:fb:84:47:d0:b0), Dst: RuijieNe\_7d:fa:db (00:74:9c:7d:fa:db)  
v Destination: RuijieNe\_7d:fa:db (00:74:9c:7d:fa:db)  
v Source: IntelCor\_47:d0:b0 (58:fb:84:47:d0:b0)

Offset	Hex	ASCII
0000	00 74 9c 7d fa db 58 fb 84 47 d0 b0 08 00 45 00	.t.}.X.G..E.
0010	00 3c 66 ee 00 00 40 01 4c d4 0a 15 aa 14 0a 03	<f...@.L.....
0020	08 d3 08 00 4d 48 00 01 00 13 61 62 63 64 65 66	...MH...abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmn opqrstuv
0040	77 61 62 63 64 65 66 67 68 69	wabcdegh i

可以看到，计算机一共发送了四个ICMP询问报文，所有报文均得到了应答。这与命

令提示符中所显示的结果是一致的。

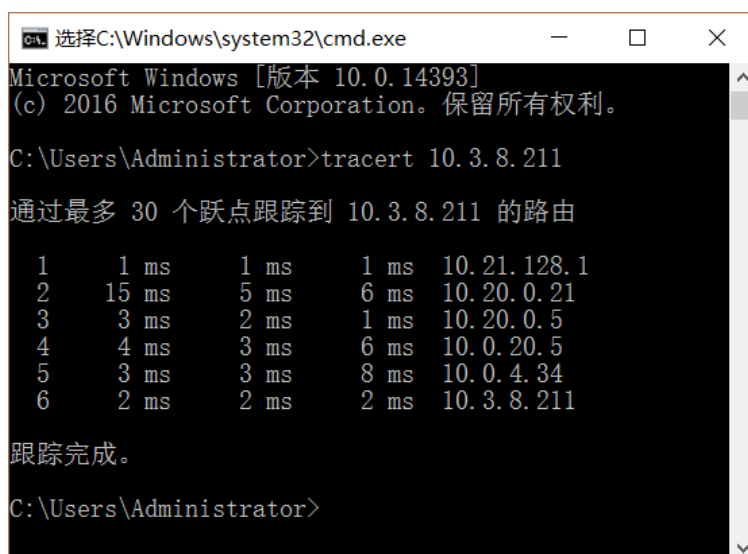


对报文进一步分析，我们可以得到如下表所示结果：

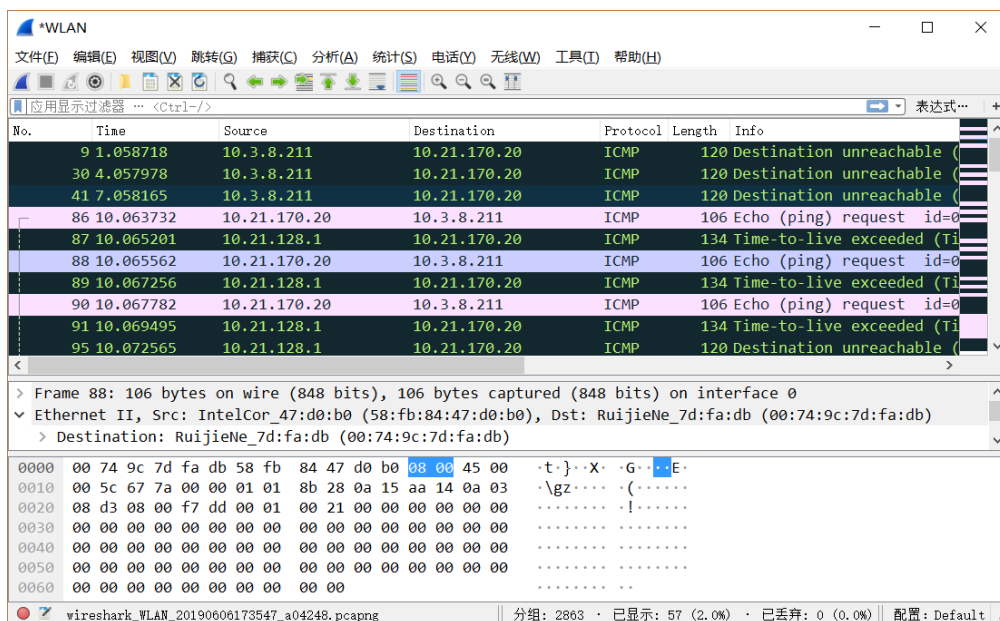
字段	报文（16 进制）	内容
类型	08	询问一台主机是否处于活动状态
代码	00	
校验和	4d 48	头部校验和为 0x4d48

## 2. 使用ping命令捕获ICMP分组

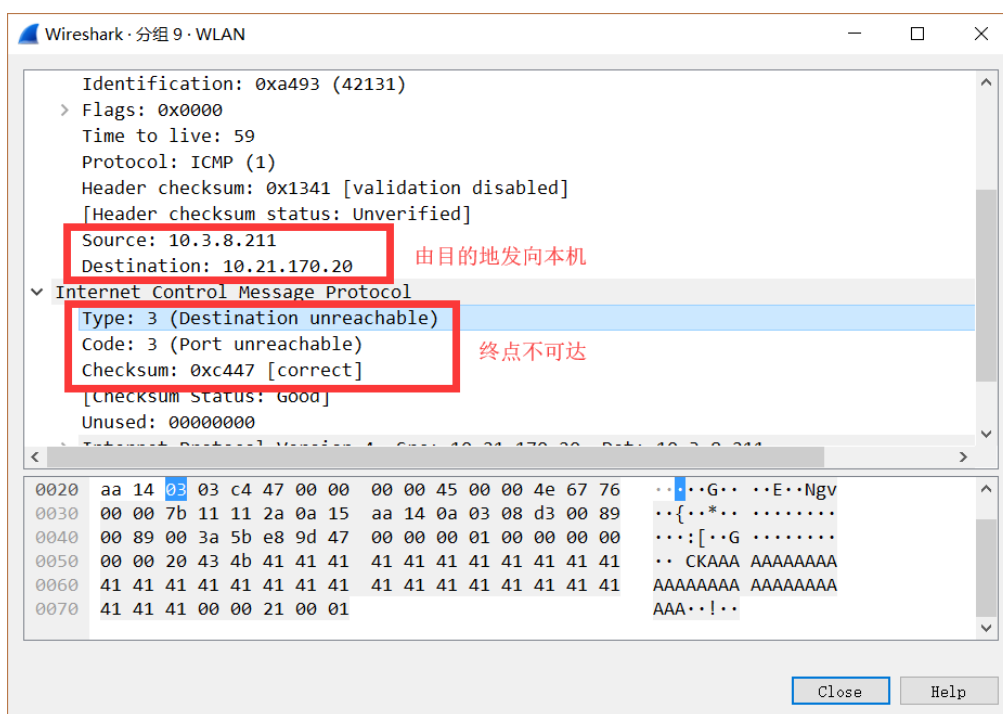
将WireShark的监控打开后，使用计算机中的命令提示符对校园网内的主机“10.3.8.211”发出tracert命令，如下图所示：



可以看到，数据包经过了6个跃点跟踪到了10.3.8.211的路由。停止Wireshark的监控后，我们可以看到，软件在本次监控过程中捕获了大量的ICMP数据分组。



对第一个数据包进行分析，我们可以看出，该数据包为一个ICMP差错报告报文。



对报文进一步分析，我们可以得到如下表所示结果：

字段	报文（16进制）	内容
类型	03	终点不可达
代码	03	端口不可达
校验和	c4 47	头部校验和为 0xc447

## 2.3 发送指定长度的数据分组

将Wireshark的监控打开后，使用命令提示符中的ping命令向校园网内一台主机“10.3.8.211”发送大小为8000字节的数据，得到的结果如下所示：

```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping -l 8000 10.3.8.211

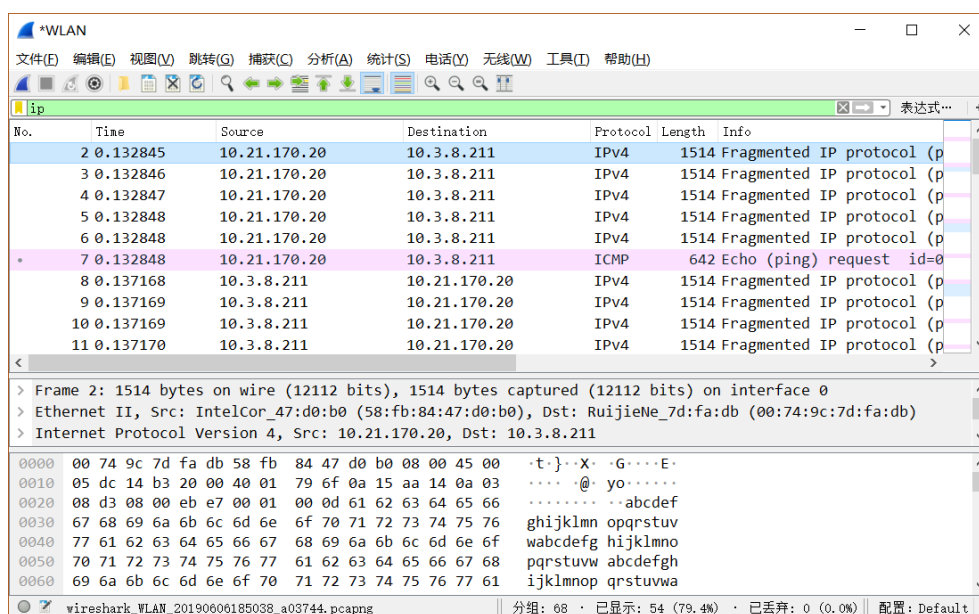
正在 Ping 10.3.8.211 具有 8000 字节的数据:
来自 10.3.8.211 的回复: 字节=8000 时间=6ms TTL=59
来自 10.3.8.211 的回复: 字节=8000 时间=3ms TTL=59
来自 10.3.8.211 的回复: 字节=8000 时间=5ms TTL=59
来自 10.3.8.211 的回复: 字节=8000 时间=3ms TTL=59

10.3.8.211 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 = 3ms, 最长 = 6ms, 平均 = 4ms

C:\Users\Administrator>

```

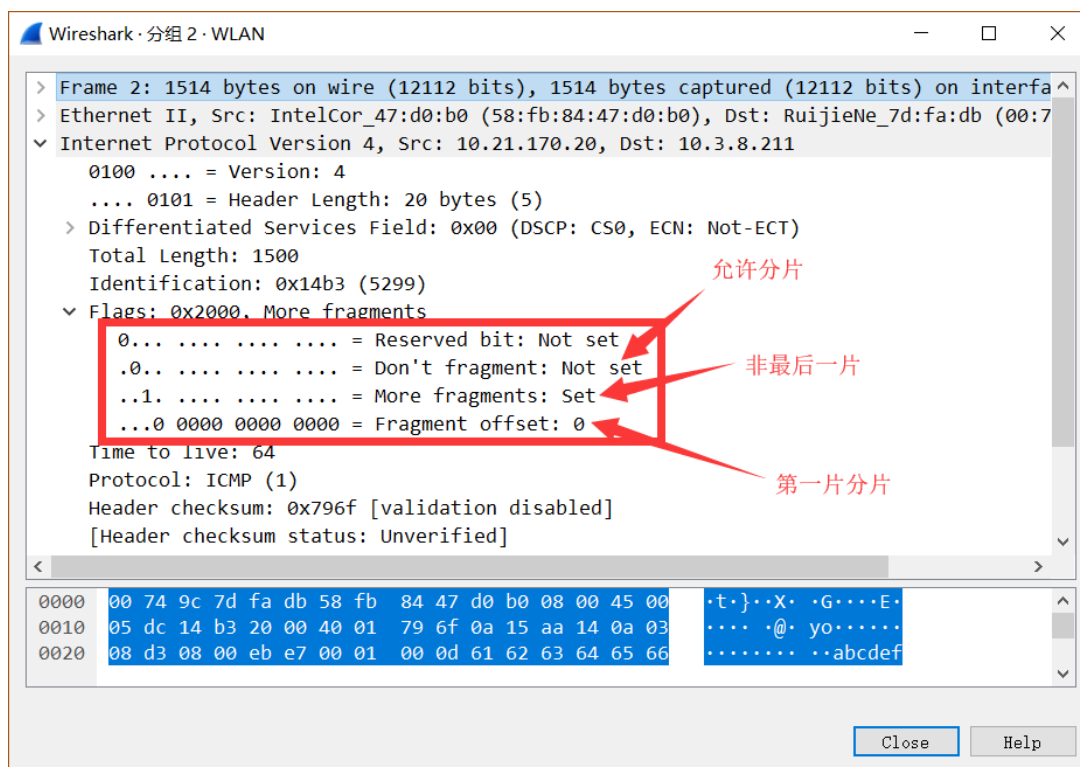
停止监控后，可以看到Wireshark软件捕获的IP分组如下所示：



我们抽取其中的一组数据包进行分析，经过筛选与分析，本报告以ping命令发出的一次请求为例：

No.	Time	Source	Destination	Protocol	Length	Info
2	0.132845	10.21.170.20	10.3.8.211	IPv4	1514	Fragmented IP protocol
3	0.132846	10.21.170.20	10.3.8.211	IPv4	1514	Fragmented IP protocol
4	0.132847	10.21.170.20	10.3.8.211	IPv4	1514	Fragmented IP protocol
5	0.132848	10.21.170.20	10.3.8.211	IPv4	1514	Fragmented IP protocol
6	0.132848	10.21.170.20	10.3.8.211	IPv4	1514	Fragmented IP protocol
7	0.132848	10.21.170.20	10.3.8.211	ICMP	642	Echo (ping) request id

如下所示为该组数据包中第一个数据包的详细内容：



对该数据包进行分析后，可将其所包含的重要信息如下所示：

字段	报文（16 进制）	内容
首部长度	5	报头长度为 20 字节
服务类型	00	正常时延、正常吞吐量、正常可靠性
总长度	05 dc	分组长度 1500 字节
标识	14 b3	标识为 5299
标志	20 00	DF=0, MF=1，允许分片，且非最后一片
偏移值	00	偏移量为 0
生存周期	40	每跳生存周期为 64 秒
协议	1	十进制为 1，表示携带的数据使用 ICMP 协议
头部校验和	79 6f	头部校验和为 0x796f
源地址	0a 15 aa 14	源地址为 10.21.170.20
目的地址	0a 03 08 d3	目的地址为 10.3.8.211

对其之后的5个数据包也进行同样的分析后，我们可以将这6个数据包的信息归纳为如下所示的表格中：

字段	报文（16 进制）					
	数据包 1	数据包 2	数据包 3	数据包 4	数据包 5	数据包 6
首部长度	5					
服务类型	0					
总长度	05 dc					02 74
标识	14b3					
标志	20 00	20 b9	21 72	22 2b	22 e4	03 9d
偏移值	00 00	00 b9	01 72	02 2b	02 e4	03 9d
生存周期	40					
协议	1					
头部校验和	79 6f	78 b6	77 fd	77 44	76 8b	99 3a
源地址	0a 15 aa 14					
目的地址	0a 03 08 d3					

可以看到，这一组数据包的首部长度、服务类型、标识、生存周期、协议、源地址、目标地址是一致的，其中标识一致表明了它们属于同一组的报文分片，而再下一个数据包的标识字段则为0xc6d1，表明其并不属于这一组。

下面对该组数据包的标志（Flag）字段进行分析：

① 6个数据包中的DF位均为0，这说明本组数据包是允许分片的，且也只有当DF=0时，后面的分析才是有意义的；

② 前5个数据包中的MF位均为1，而第6个数据包的MF位为0，这代表前5个数据包的后面均还存在同一组的数据包，而第六个数据包则为最后一个数据包；

③ 这6个数据包的偏移值（Offset）中，每两个相邻的数据包的差值均为0xb9，为十进制的185，该部分还需乘以8字节才能够得到真值，即1480字节。

我们知道以太网的最大数据部分长度为1500字节，因此发送总长度为8000字节的数据报是必须要经过分片的。上述的1480字节加上对应IP数据包长达20字节的首部恰好为1500字节，符合预期。

不仅如此，除了前5个数据包的总长度均为0x05dc（十进制的1500）外，第6个数据包的总长度为0x0274（十进制的628），将这六个数据包数据部分的总长度相加，并将之与预期的结果进行对比，即：

$$\text{总长度} = (1500 - 20) \times 5 + (628 - 20) = 8008$$

$$ping \text{ 命令指定的数据部分长度} + ICMP \text{ 首部长度} = 8000 + 8 = 8008$$



由计算结果可知，实验与预期数据相吻合，说明本部分实验是成功的。

### 3 实验总结

在计算机网络“网络层数据分组的捕获和解析”实验中，我完成了对网络数据包的抓取与分析。本次实验的进展十分顺利，不过也许是个人比较谨慎、仔细，我从实验开始至实验报告大致完成共花费了5个小时，但我从本次实验中得到的收获是巨大的；除了实验中的内容之外，我还借助此次机会，将TCP、IGMP等其他协议的数据包进行了捕获与分析，对TCP的三次握手及可靠传输机制，还有IGMP中使用到的组播技术有了更加深刻的理解。

本次实验虽然难度不是特别大，但其中有许多前期工作是需要注意的。比如同学们需要提前熟悉WireShark这款软件，以及需要利用课上、课下的时间将网络数据包的格式以及各字段的作用学得透彻。只有这样，我们才能够在实验过程中变得更加得心应手，从实验中得到的收获也更大。

在开学初期，王老师便向我们推荐了WireShark这款软件，在当时对于没有学习过《计算机网络》课程的我来说，在课下使用这个软件进行抓包与分析是一个很大的挑战。随着这学期我们在计算机网络理论课程上对于计算机网络不断深入的了解，我在课堂上学到了许多关于网络中通信交互的基本知识，发现网络中的信息传递在大多数场合下是以数据包的形式。我还借助互联网以及图书馆中的文献资料熟悉了因特网中许多其他类型数据包的格式、机制及原理。

通过本次实验的动手实践，我对理论课程上学习到的知识有了更为深刻的理解与把握。在实验之前，我通过抓包分析了一定数量的数据报文，对计算机网络体系中各层次的数据包发送和接收工作原理有了比较清晰的认识。

通过本学期课程的学习以及最后一次实验的完成，我对计算机网络中使用数据包进行通信的机制有了一个较为基础、全面的认识。但要想深入理解，还需在今后的学习生涯中深入实践。王老师在这学期的计算机网络课程中让我首次接触到了系统化的计算机网络知识，为我接下来深入理解计算机网络及现代网络技术奠定了基础。我将会在今后的学习生涯中继续开拓、不断实践。在此感谢王老师本学期带给我们的精彩的课程，感谢王老师对同学们的耐心讲解与悉心教导。