实验 5　IPv4 协议分析

一、IPv4 数据报首部

5．1．1　实验目的

掌握 IPv4 协议原理，理解 IPv4 分组首部结构及各字段的含义。

5．1．2　实验环境

1．连接外网的 Windows10 主机一台，并安装有科来网络分析系统。

2．通过科来网络分析系统捕获一段时间内的 IPv4 分组。

3．实验分组：一名同学一组。

5．1．3　实验内容

1．用科来网络分析系统捕获数据包。

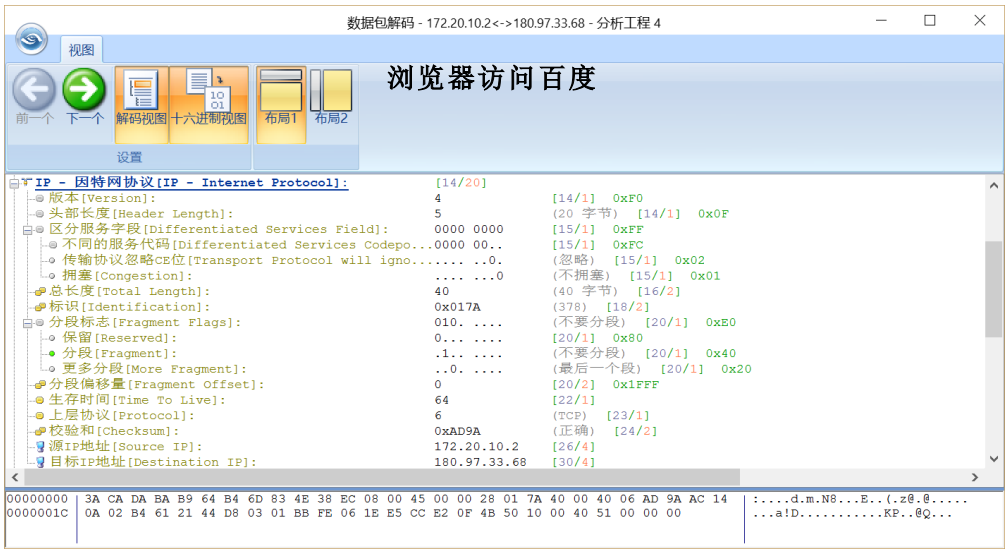
2．分析捕获到的 IP 数据包中首部各个字段的意义。

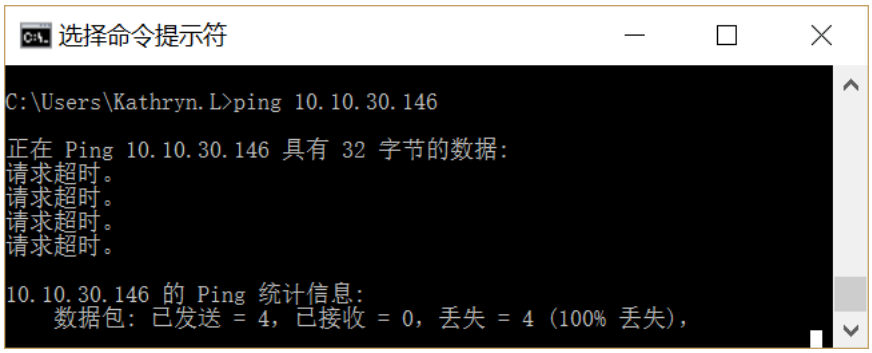
5．1．4　实验步骤

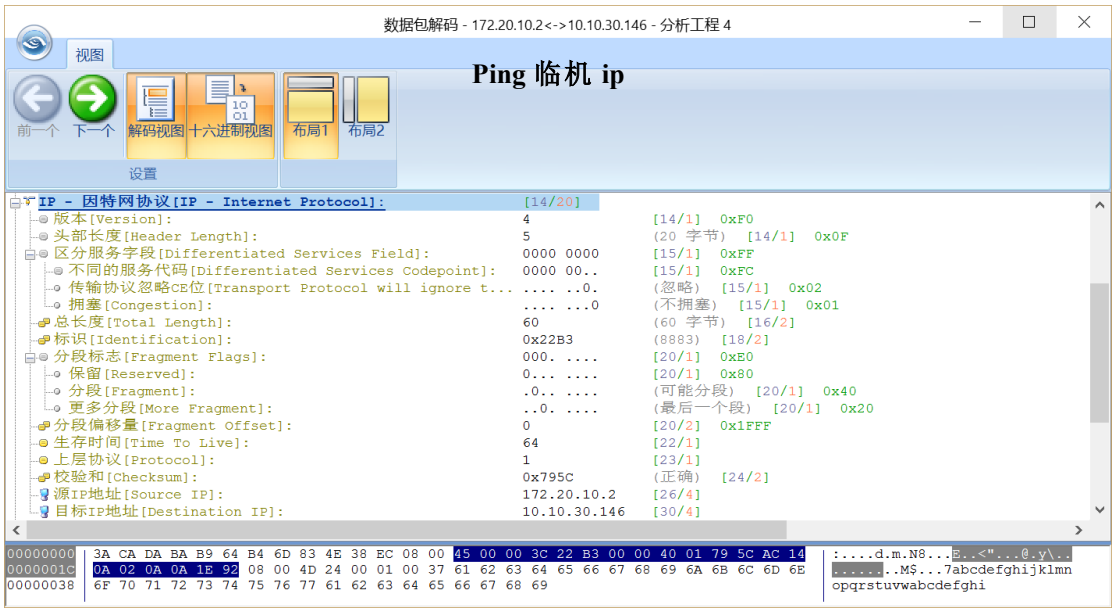
1．打开科来网络分析系统，开始捕获数据包。

2．用浏览器访问百度，用 ping 命令探测临机、网关和百度。

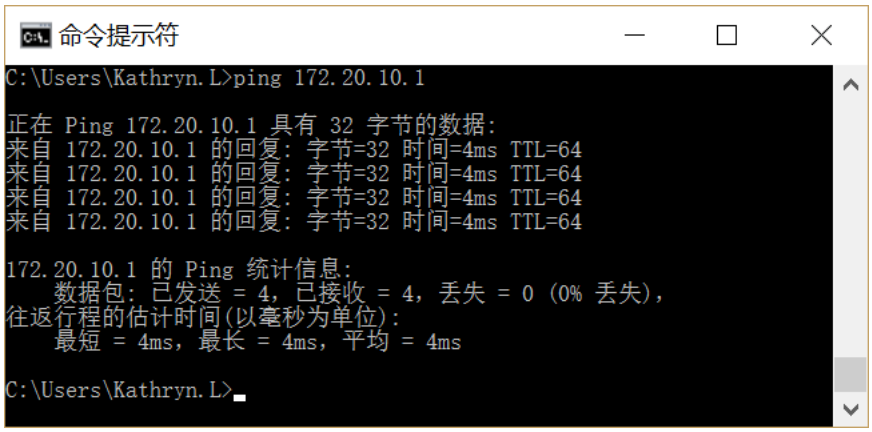
3．停止捕获，观察捕获到的数据包。

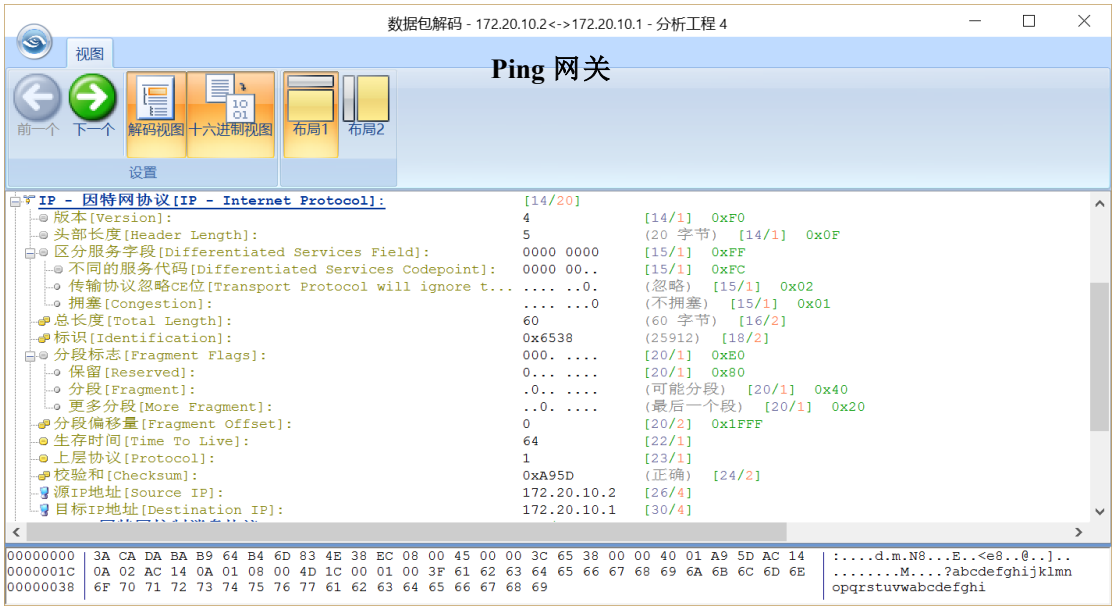


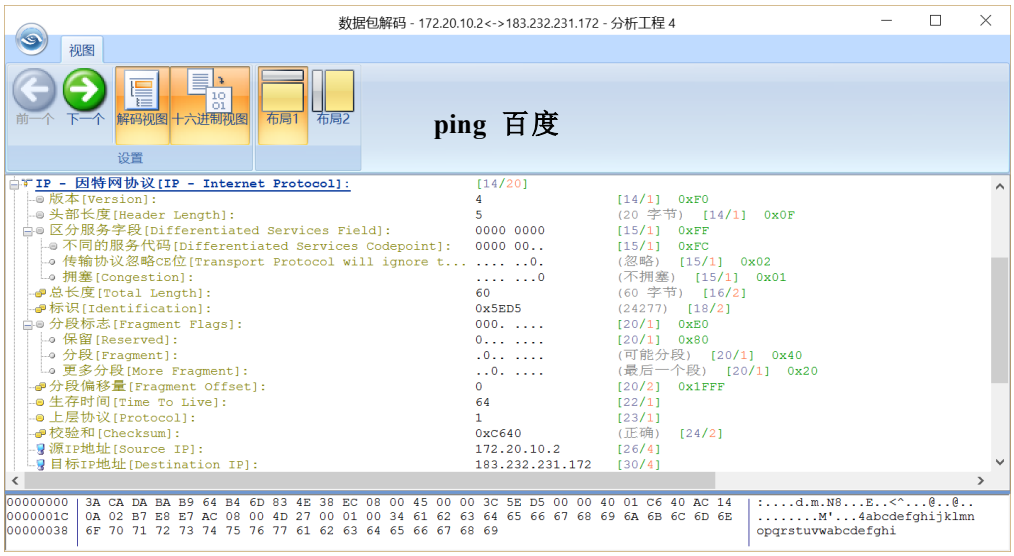

选择命令提示符
C:\Users\Kathryn.L>ping 10.10.30.146
正在 Ping 10.10.30.146 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
10.10.30.146 的 Ping 统计信息:
数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),


数据包解码 - 172.20.10.2<->10.10.30.146 - 分析工程 4
视图
前一个
下一个
解码视图
十六进制视图
布局1
布局2
设置
IP - 因特网协议[IP - Internet Protocol]: [14/20]
版本[Version]: 4 [14/1] 0xF0
头部长度[Header Length]: 5 (20 字节) [14/1] 0x0F
区分服务字段[Differentiated Services Field]: 0000 0000 [15/1] 0xFF
不同的服务代码[Differentiated Services Codepoint]: 0000 00.. [15/1] 0xFC
传输协议忽略CE位[Transport Protocol will ignore t...: .... ..0. (忽略) [15/1] 0x02
拥塞[Congestion]: ....0 (不拥塞) [15/1] 0x01
总长度[Total Length]: 60 (60 字节) [16/2]
标识[Identification]: 0x22B3 (8883) [18/2]
分段标志[Fragment Flags]: 000. .... [20/1] 0xE0
保留[Reserved]: 0... .... [20/1] 0x80
分段[Fragment]: .0.. .... (可能分段) [20/1] 0x40
更多分段[More Fragment]: ..0. .... (最后一个段) [20/1] 0x20
分段偏移量[Fragment Offset]: 0 [20/2] 0x1FFF
生存时间[Time To Live]: 64 [22/1]
上层协议[Protocol]: 1 [23/1]
校验和[Checksum]: 0x795C (正确) [24/2]
源IP地址[Source IP]: 172.20.10.2 [26/4]
目标IP地址[Destination IP]: 10.10.30.146 [30/4]
00000000 3A CA DA BA B9 64 B4 6D 83 4E 38 EC 08 00 45 00 00 3C 22 B3 00 00 40 01 79 5C AC 14
0000001C 0A 02 0A 0A 1E 92 08 00 4D 24 00 01 00 37 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E
00000038 6F 70 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69
:....d.m.N8...E..<"...@.y\..
......M$...7abcdefghijklmn
opqrstuvwabcdefghi

Ping 临机 ip


命令提示符
C:\Users\Kathryn.L>ping 172.20.10.1
正在 Ping 172.20.10.1 具有 32 字节的数据:
来自 172.20.10.1 的回复: 字节=32 时间=4ms TTL=64
来自 172.20.10.1 的回复: 字节=32 时间=4ms TTL=64
来自 172.20.10.1 的回复: 字节=32 时间=4ms TTL=64
来自 172.20.10.1 的回复: 字节=32 时间=4ms TTL=64
172.20.10.1 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 = 4ms, 最长 = 4ms, 平均 = 4ms
C:\Users\Kathryn.L>





4. 将访问百度以及 ping 临机、网关和百度的 IP 数据包首部中各字段的值记录在下表中，需要记录 IP 数据报的版本号、首部长度、总长度、标识、标志、片偏移、生存时间、上层协议、源地址和目的地址。

	版本号	首部长度	总长度	标识	标志	片偏移	生存时间	上层协议	源地址	目的地址
访问百度	4	5	40	0X017A	010	0	64	6	172.20.10.2	180.97.33.68
Ping临机	4	5	60	0X22B3	000	0	64	1	172.20.10.2	10.10.30.146
Ping网关	4	5	60	0X6538	000	0	64	1	172.20.10.2	172.20.10.1
Ping百度	4	5	60	0X5ED5	000	0	64	1	172.20.10.2	183.232.231.172

5．比较所记录的各字段的值，理解首部字段的含义和作用。

5．1．5 实验要求

完成本次实验后，仔细观察所捕获的数据包，对网络层数据包首部的各字段进行整理，说明参数之间的关联性，进而加深理解网络层的工作过程。

参考如下

5．1．6 思考与讨论

1．在连续捕获到的数据包中，IPv4 首部哪些字段的值是不变的？说明原因。

2．在连续捕获到的数据包中，IPv4 首部哪些字段的值是变化的？说明原因。

二、IPv4 数据报分段

5．2．1 实验目的

理解并掌握 IP 协议首部与分段有关的字段的含义和作用。

5．2．2 实验环境

1．软硬件环境：安装科来网络分析系统的连网的 Windows XP 主机一台。
2．实验分组：每名同学一组。

5．2．3 实验内容

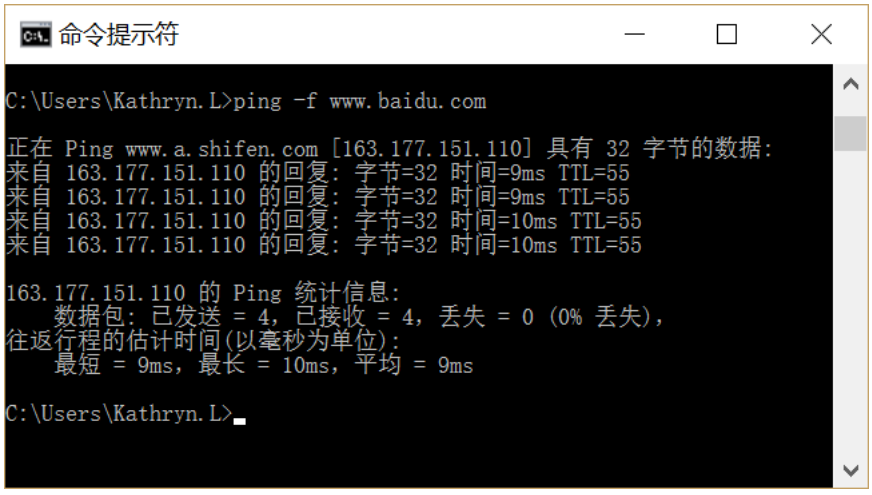
用 PING 命令发送设置不分段的数据包，发送可分段数据包，捕获数据包，分析首部相关字段的内容，理解这些字段的含义和作用。

5．2．4 实验步骤

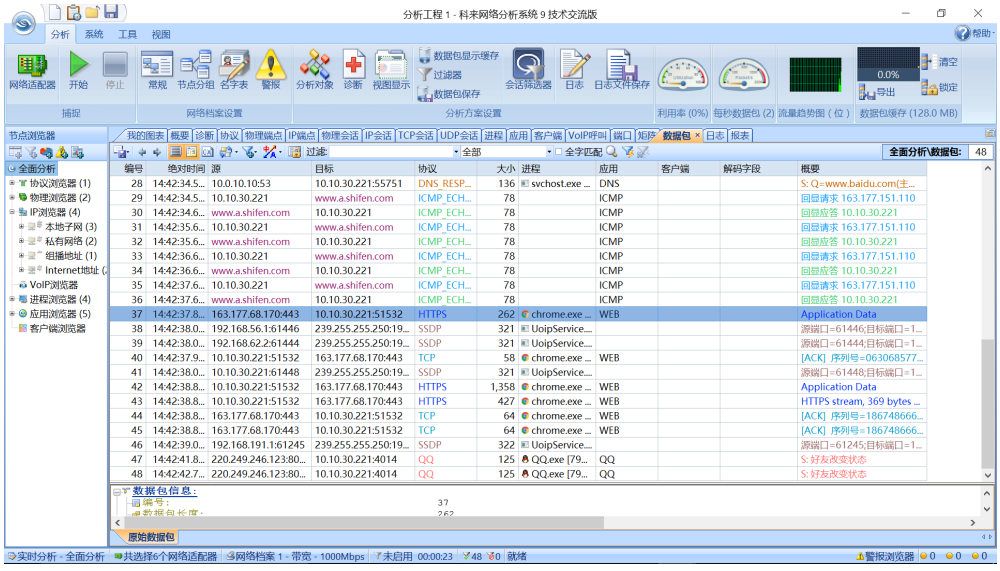
（一）、确认分段标志位

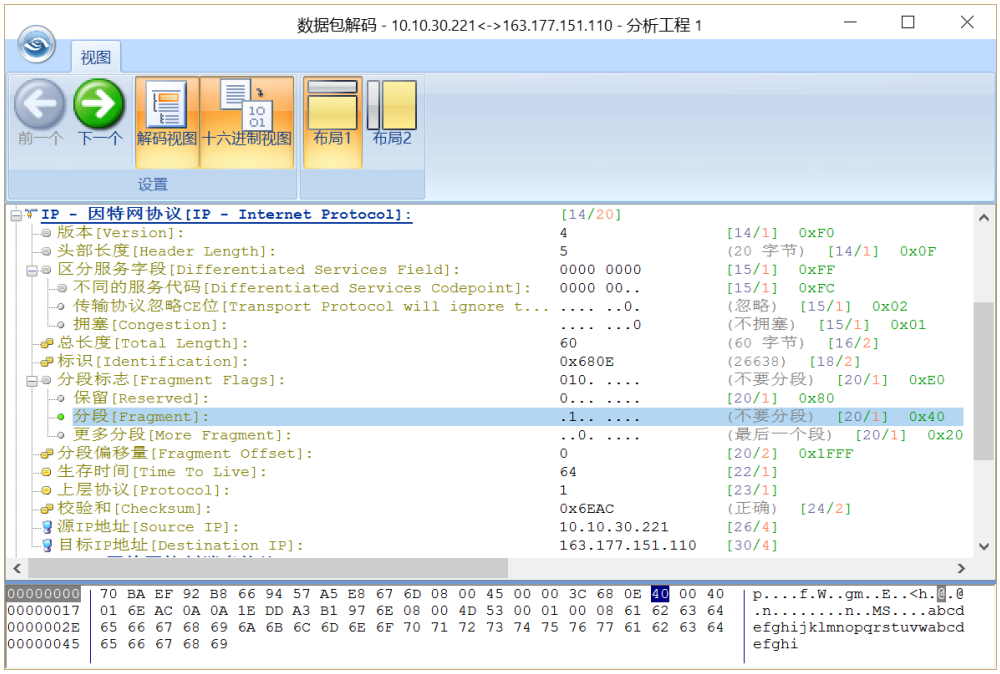
1．打开科来网络分析系统，开始捕获数据包。

2．在命令窗口执行 ping –f　www.baidu.com 命令，设置 DF 值为 1，即不分段。



3．停止捕获，观察捕获的数据包中 DF 字段的值。





（二）、IPv4 分段与重组

1．打开科来网络分析系统，开始捕获数据包。

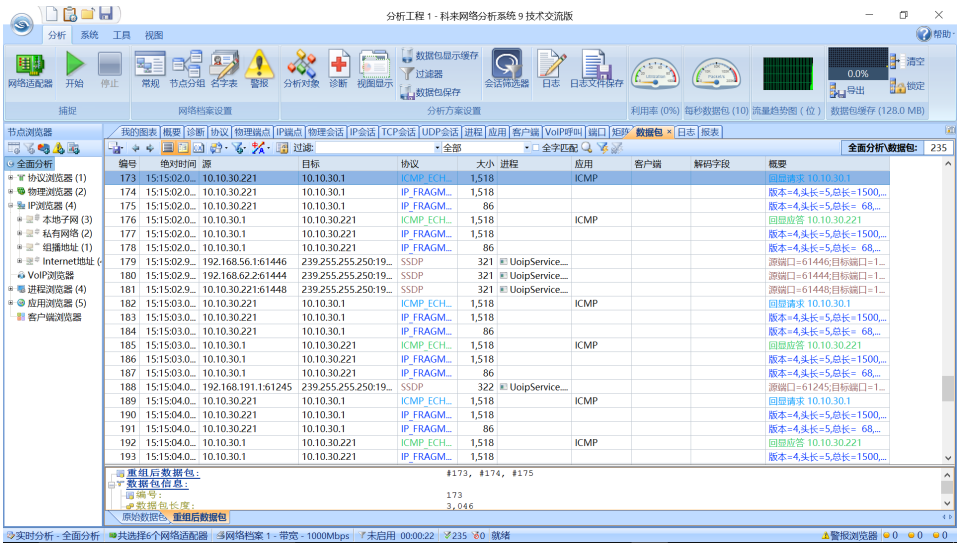
2．在本机运行命令：“ping –l 3000 网关 IP”，向局域网网关发送较大的 ping 分组。

2．停止捕获后，分析捕获到的数据包，观察分组的分段与重组，记录如下内容：

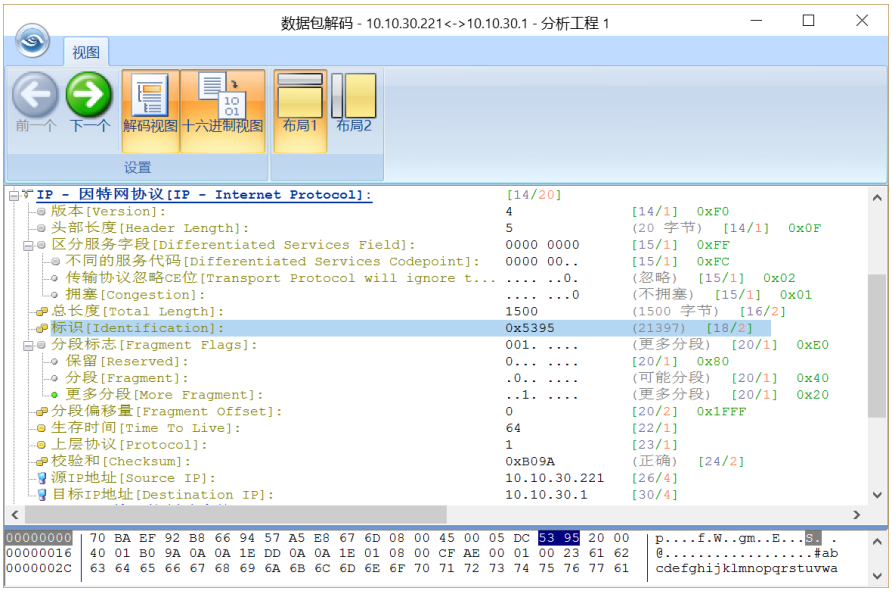
1）应答分组的返回时间比起通常的 ping 应答要长一些，并且有可能刚开始几个 ping 请求会得不到应答，为什么？

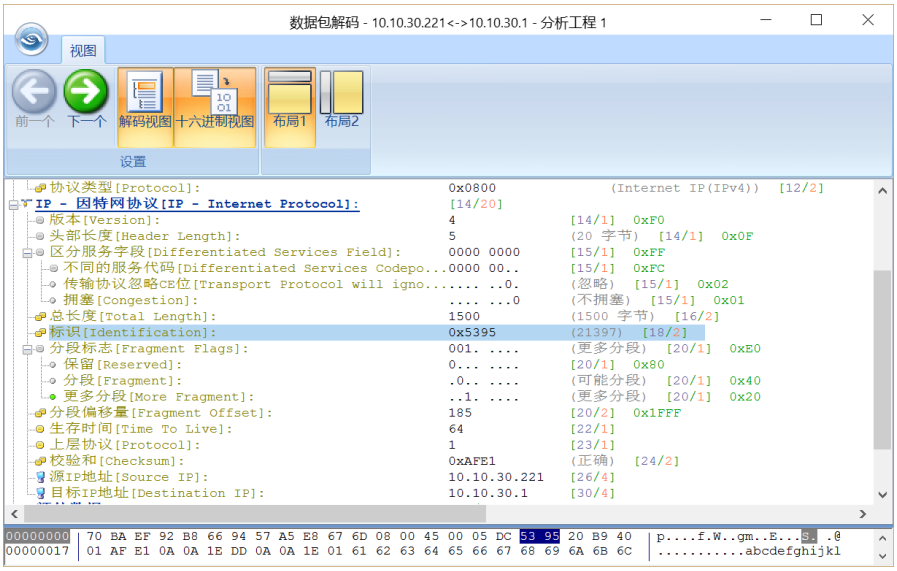
答：理论上 IP 数据报经过的路由器越多，所花费的时间也会越多。Ping 网关需要遍历所有路由器所以比起通常的 ping 应答要长一些。而网速变化也对时长有很大影响。

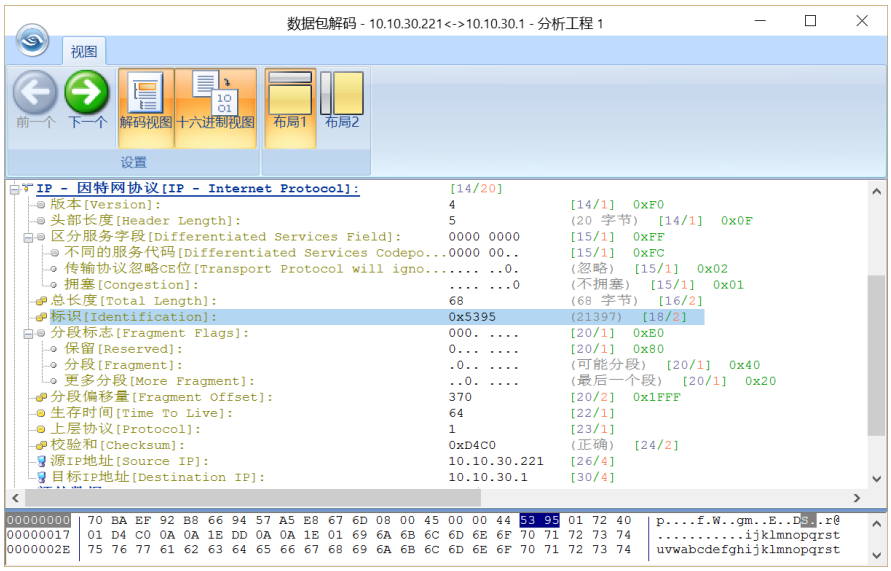
2）在科来网络分析系统中可以看到每个 ping 请求都被分成了 3 个 IPv4 分段，将截图放到实验记录中，并回答下列问题。



（1） 三个段的标识字段的值是多少？说明什么？







答：如图，三个字段的标识字段都为（53 95）

说明三个数据分段属于一个数据报：相同的标识字段的值使分片后的各数据报片最后能正确的重装成原来的数据报。

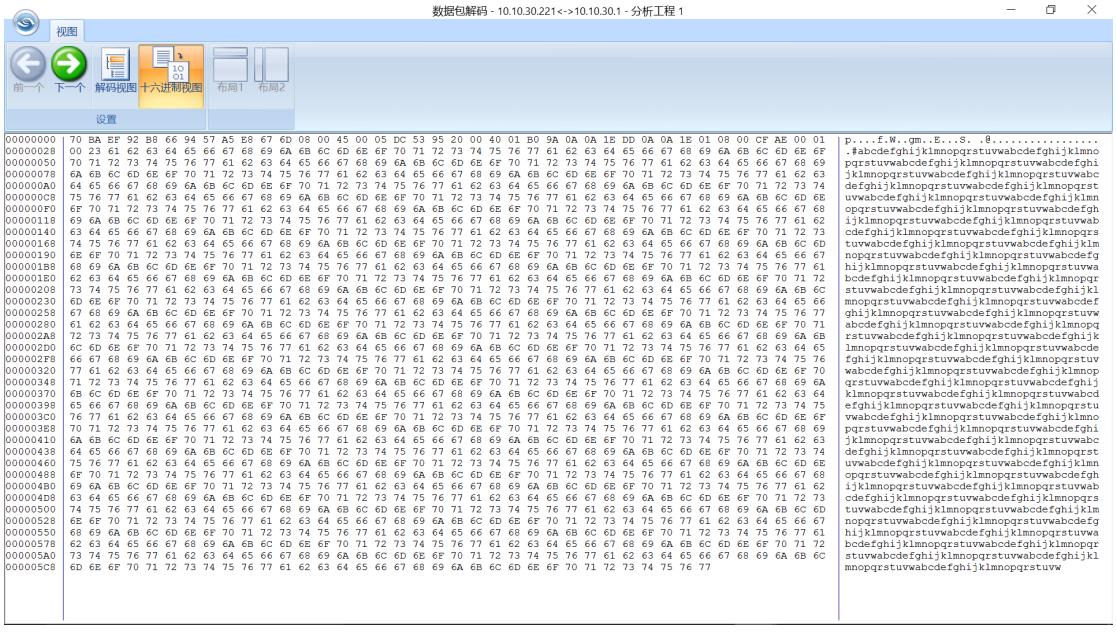
（2)片偏移字段的值依次是多少？是否可以确保在乱序到达时也能正确重组出原来的分组？

答：如图，分别是 0 、185 、370

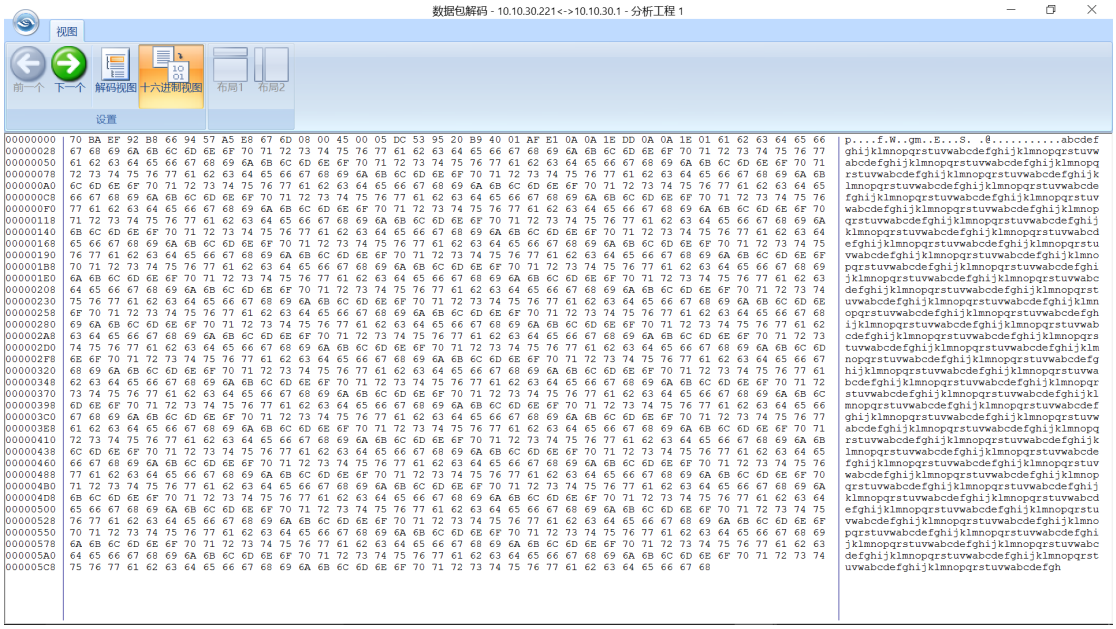
可以确保正确重组。

（3) 第一个分段的最后一个字母、第二个分段的第一个字母和最后一个字母、第三个分段中的第一个字母分别是什么？

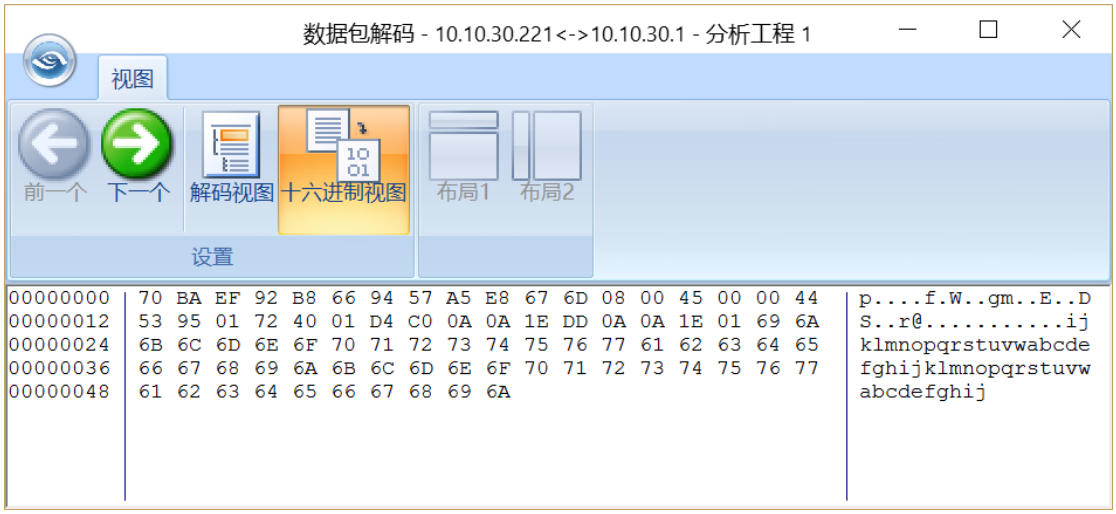
答：如图：



第一分段的最后一个字母（w）



第二分段第一个字母（a）最后一个字母（h）



第三分段第一个字母（i）

5．2．5 实验要求

完成本次实验后，对捕获的数据包进行详细分析，重点分析网络层数据包首部中的标识、标志和片偏移字段的含义和作用。

5．2．6 思考与讨论

1. IP 分组被分段后，其对应的标识字段、标志字段和片偏移字段的值是什么？说明其含义。

2．什么情况下 IPv4 分组需要分段？分段和重组分别在哪一端进行？

3．三个分段的总的数据长度为 1500+1500+68-3*20=3008，比 ping 命令后面的参数 3000 多了 8，为什么？