实验 6：利用 Wireshark 进行协议分析

1、实验目的

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。

2、实验环境

- Windows操作系统
- 与因特网连接的计算机网络系统
- Wireshark

3、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：

a) 利用 Wireshark 分析 DNS 协议
b) 利用 Wireshark 分析 UDP 协议
c) 利用 Wireshark 分析 ARP 协议

4、实验方式

每位同学上机独立完成实验，并与指导教师讨论。

5、参考内容

要深入理解网络协议，需要仔细观察协议实体之间交换的报文序列。为探究协议操作细节，可使协议实体执行某些动作，观察这些动作及其影响。这些任务可以在仿真环境下或在如因特网这样的真实网络环境中完成。观察在正在运行协议实体间交换报文的基本工具被称为分组嗅探器（packet sniffer）。顾名思义，一个分组嗅探器俘获（嗅探）计算机发送和接收的报文。一般情况下，分组嗅探器将存储和显示出被俘获报文的各协议头部字段的内容。图 6-1 为一个分组嗅探器的结构。

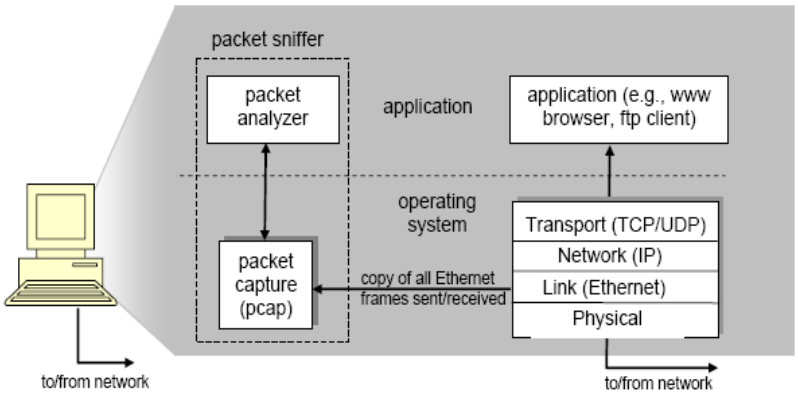

图 6-1 分组嗅探器的结构

图 6-1 右边是计算机上正常运行的协议（在这里是因特网协议）和应用程序（如：web 浏览器和 ftp 客户端）。分组嗅探器（虚线框中的部分）是附加计算机普通软件上的，主要有两部分组成。分组俘获库（packet capture library）接收计算机发送和接收的每一个链路层帧的拷贝。高层协议（如：HTTP、FTP、TCP、UDP、DNS、IP 等）交换的报文都被封装在链路层帧中，并沿着物理媒体（如以太网的电缆）传输。图 1 假设所使用的物理媒体是以太网，上层协议的报文最终封装在以太网帧中。

分组嗅探器的第二个组成部分是分析器。分析器用来显示协议报文所有字段的内容。为此，分析器必须能够理解协议所交换的所有报文的结构。例如：我们要显示图 6-1 中 HTTP 协议所交换的报文的各个字段。分组分析器理解以太网帧格式，能够识别包含在帧中的 IP 数据报。分组分析器也要理解 IP 数据报的格式，并能从 IP 数据报中提取出 TCP 报文段。然后，它需要理解 TCP 报文段，并能够从中提取出 HTTP 消息。最

后，它需要理解 HTTP 消息。

Wireshark 是一种可以运行在 Windows，UNIX，Linux 等操作系统上的分组分析器。Wireshark 是免费的，可以从 https://www.wireshark.org/download.html 得到,Wireshark 的 User's Guide 可以从 https://www.wireshark.org/docs/ 获得。运行 Wireshark 程序时，其图形用户界面如图 6-2 所示。最初，各窗口中并无数据显示。在用户选择接口，点击开始抓包按钮之后，Wireshark 的用户界面会变成如图 6-3 所示。

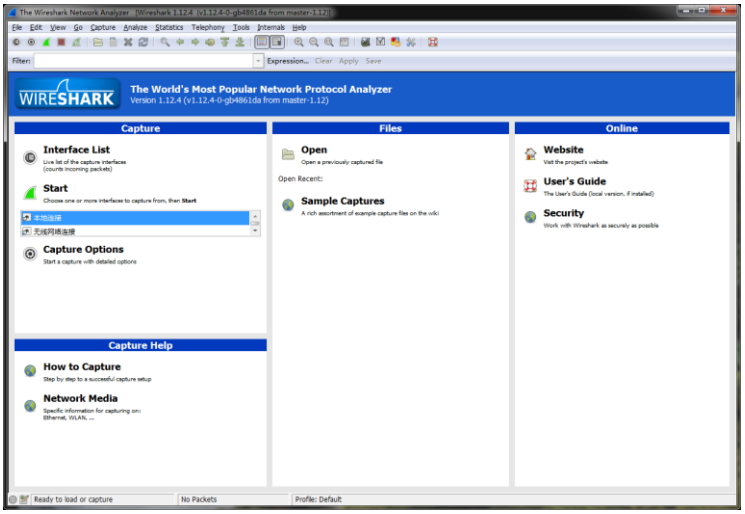

图 6-2 Wireshark 初始用户界面

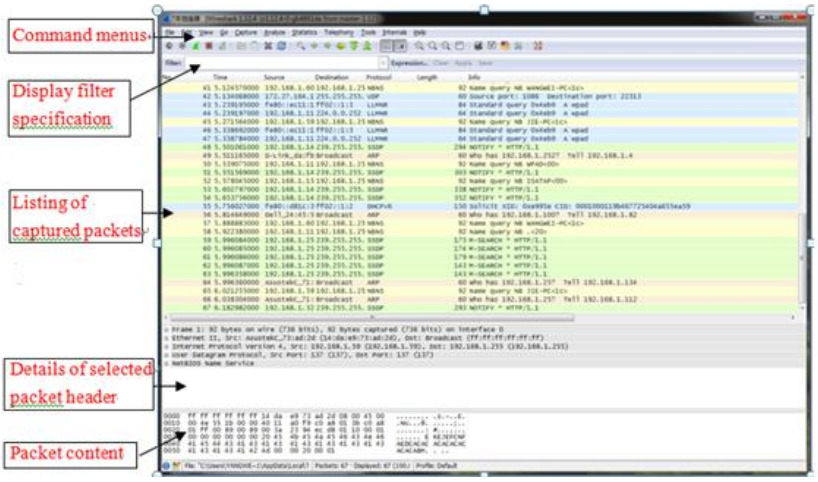

图 6-3 Wireshark 的用户界面

此时 Wireshark 的用户界面主要有 5 部分组成，如图 6-3 所示。

- **命令菜单**（command menus）：命令菜单位于窗口的最顶部，是标准的下拉式菜单。最常用菜单命令有两个：File、Capture。File 菜单允许你保存俘获的分组数据或打开一个已被保存的俘获分组数据文件或退出 Wireshark 程序。Capture 菜单允许你开始俘获分组。
- **俘获分组列表**（listing of captured packets）：按行显示已被俘获的分组内容，其中包括：Wireshark 赋予的分组序号、俘获时间、分组的源地址和目的地址、协议类型、分组中所包含的协议说明信息。单击某一列的列名，可以使分组按指定列进行排序。在该列表中，所显示的协议类型是发送或接收分组的**最高层协议**的类型。
- **分组头部明细**（details of selected packet header）：显示俘获分组列表窗口中被选中分组的头部详细信息。包括：与以太网帧有关的信息，与包含在该分组中的 IP 数据报有关的信息。单击以太网帧或 IP 数据报所在行左边的向右或向下的箭头可以展开或最小化相关信息。另外，如果利用 TCP 或 UDP 承载分组，Wireshark 也会显示 TCP 或 UDP 协议头部信息。最后，分组最高层协议的头部字段也会显示在此窗口中。
- **分组内容窗口**（packet content）：以 ASCII 码和十六进制两种格式显示被俘获帧的完整内容。
- 显示**筛选规则**（display filter specification）：在该字段中，可以填写协议的名称或其他信息，根据此内容可以对分组列表窗口中的分组进行过滤。

（一）Wireshark 的使用

- 启动主机上的 web 浏览器。
- 启动 Wireshark。你会看到如图 6-2 所示的窗口，只是窗口中没有任何分组列表。
- 开始分组俘获：选择“capture”下拉菜单中的“Capture Options”命令，会出现如图 6-3 所示的“Wireshark: Capture Options”窗口，可以设置分组俘获的选项。

- 在实验中，可以使用窗口中显示的默认值。在“Wireshark: Capture Options”窗口（如图 6-4 所示）的最上面有一个“Interface List”下拉菜单，其中显示计算机所具有的网络接口（即网卡）。当计算机具有多个活动网卡时，需要选择其中一个用来发送或接收分组的网络接口（如某个有线接口）。随后，单击“Start”开始进行分组俘获，所有由选定网卡发送和接收的分组都将被俘获。

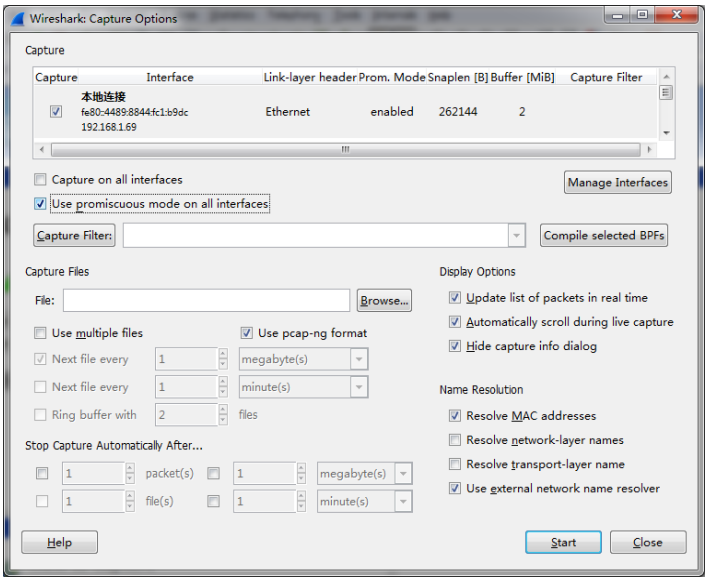

图 6-4 Wireshark 的 Capture Option

- 开始分组俘获后，会出现如图 6-5 所示的窗口。该窗口统计显示各类已俘获数据包。在该窗口的工具栏中有一个“stop”按钮，可以停止分组的俘获。但此时你最好不要停止俘获分组。
- 在运行分组俘获的同时，在浏览器地址栏中输入某网页的 URL，如：http://www.hit.edu.cn。为显示该网页，浏览器需要连接 www.hit.edu.cn 的服务器，并与之交换 HTTP 消息，以下载该网页。包含这些 HTTP 报文的以太网帧将被 Wireshark 俘获。
- 当完整的页面下载完成后，单击 Wireshark 菜单栏中的 stop 按钮，停止分组俘获。Wireshark 主窗口显示已俘获的你的计算机与其他网络实体交换的所有协议报文，其中一部分就是与 www.hit.edu.cn

服务器交换的 HTTP 报文。此时主窗口与图 6-3 相似。

- 在显示筛选规则中输入“http”，单击“回车”，分组列表窗口将只显示 HTTP 协议报文。
- 选择分组列表窗口中的第一条 http 报文。它应该是你的计算机发向 www.hit.edu.cn 服务器的 HTTP GET 报文。当你选择该报文后，以太网帧、IP 数据报、TCP 报文段、以及 HTTP 报文首部信息都将显示在分组首部子窗口中。单击分组首部详细信息子窗口中向右和向下箭头，可以最小化帧、以太网、IP、TCP 信息显示量，可以最大化 HTTP 协议相关信息的显示量。

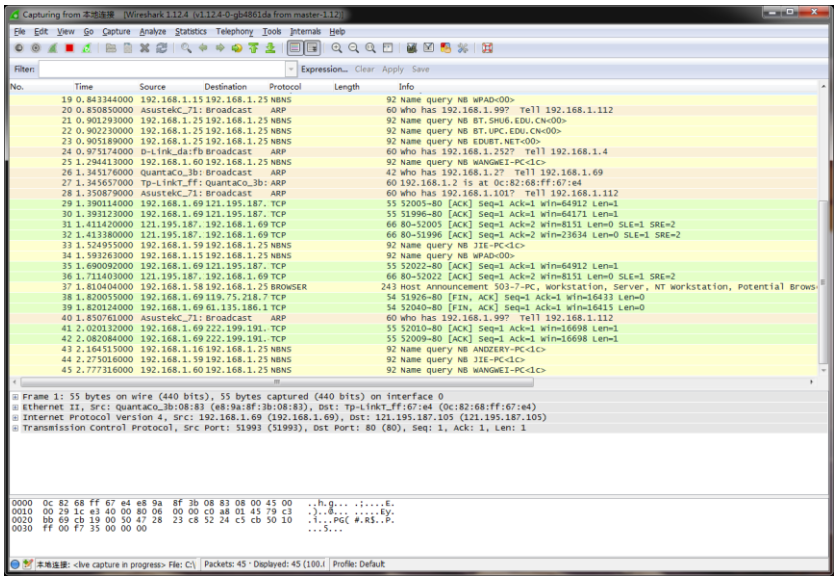

图 6-5 Wireshark 的抓包界面

（二）HTTP 分析

1）HTTP GET/response 交互

- ✧ 启动 Web browser，然后启动 Wireshark 分组嗅探器。在窗口的显示过滤说明处输入“http”，分组列表子窗口中将只显示所俘获到的 HTTP 报文。
- ✧ 开始 Wireshark 分组俘获。
- ✧ 在打开的 Web browser 窗口中输入一下地址：http://hitgs.hit.edu.cn/news

✧停止分组俘获。

根据俘获窗口内容，思考以下问题：

✧你的浏览器运行的是 HTTP1.0，还是 HTTP1.1？你所访问的服务器所运行 HTTP 协议的版本号是多少？

✧你的浏览器向服务器指出它能接收何种语言版本的对象？

✧你的计算机的 IP 地址是多少？服务器 http://hitgs.hit.edu.cn/news 的 IP 地址是多少？

✧从服务器向你的浏览器返回的状态代码是多少？

2）HTTP 条件 GET/response 交互

✧启动浏览器，清空浏览器的缓存（在浏览器中，选择“工具”菜单中的“Internet 选项”命令，在出现的对话框中，选择“删除文件”）。

✧启动 Wireshark 分组俘获器。开始 Wireshark 分组俘获。

✧在浏览器的地址栏中输入以下 URL: http://hitgs.hit.edu.cn/news,在你的浏览器中重新输入相同的 URL 或单击浏览器中的“刷新”按钮。

✧停止 Wireshark 分组俘获，在显示过滤筛选说明处输入“http”,分组列表子窗口中将只显示所俘获到的 HTTP 报文。

根据俘获窗口内容，思考以下问题：

✧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？

✧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？

✧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？

✧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。

（三）TCP 分析

注：访问以下网址需要设置代理服务器。如无法访问可与实验 TA

联系，下载 tcp-Wireshark-trace 文件，利用该文件进行 TCP 协议分析。

A. 俘获大量的由本地主机到远程服务器的 TCP 分组

（1）启动浏览器，打开http://gaia.cs.umass.edu/Wireshark-labs/alice.txt网页，得到ALICE'S ADVENTURES IN WONDERLAND文本，将该文件保存到你的主机上。

（2）打开http://gaia.cs.umass.edu/Wireshark-labs/TCP-Wireshark-file1.html，如图6-6所示，窗口如下图所示。在Browse按钮旁的文本框中输入保存在你的主机上的文件ALICE'S ADVENTURES IN WONDERLAND的全名（含路径），此时不要按“Upload alice.txt file”按钮。

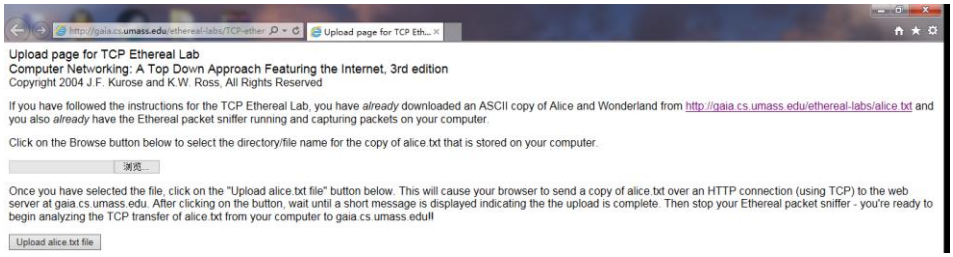

图6-6 Wireshark-labs网页截图

（3）启动Wireshark，开始分组俘获。

（4）在浏览器中，单击“Upload alice.txt file”按钮，将文件上传到gaia.cs.umass.edu服务器，一旦文件上传完毕，一个简短的贺词信息将显示在你的浏览器窗口中。

（5）停止俘获。

B. 浏览追踪信息

在显示筛选规则中输入“tcp”,可以看到在本地主机和服务器之间传输的一系列 tcp 和 http 报文,你应该能看到包含 SYN 报文的三次握手。也可以看到有主机向服务器发送的一个 HTTP POST 报文和一系列的“http continuation”报文。

根据操作思考以下问题：

- 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？

- 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？

C. TCP 基础

根据操作思考以下问题：

- 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？
- 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？
- **你能从捕获的数据包中分析出 tcp 三次握手过程吗？**
- 包含 HTTP POST 命令的 TCP 报文段的序号是多少？
- 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？
- 前六个 TCP 报文段的长度各是多少？
- 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？
- 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？
- TCP 连接的 throughput (bytes transferred per unit time)是多少？请写出你的计算过程。

（四）IP 分析

通过分析执行 traceroute 程序发送和接收到的 IP 数据包，我们将研究 IP 数据包的各个字段，并详细研究 IP 分片。

A. 通过执行 traceroute 执行捕获数据包

为了产生一系列 IP 数据报，我们利用 traceroute 程序发送具有不同大小的数据包给目的主机 X。回顾之前 ICMP 实验中使用的 traceroute 程序，源主机发送的第一个数据包的 TTL 设位 1，第二个为 2，第三个为 3，

等等。每当路由器收到一个包，都会将其 TTL 值减 1。这样，当第 n 个数据包到达了第 n 个路由器时，第 n 个路由器发现该数据包的 TTL 已经过期了。根据 IP 协议的规则，路由器将该数据包丢弃并将一个 ICMP 警告消息送回源主机。

在 Windows 自带的 tracert 命令不允许用户改变由 tracert 命令发送的 ICMP echo 请求消息（ping 消息）的大小。一个更优秀的 traceroute 程序是 pingplotter，下载并安装 pingplotter。ICMP echo 请求消息的大小可以通过下面方法在 pingplotter 中进行设置。Edit->Options->Packet，然后填写 Packet Size(in bytes，default=56)域。实验步骤：

（1） 启动 Wireshark 并开始数据包捕获

（2） 启动 pingplotter 并“Address to Trace Window”域中输入目的地址。

在“# of times to Trace”域中输入“3”，这样就不过采集过多的数据。Edit->Options->Packet，将 Packet Size(in bytes,default=56)域设为 56，这样将发送一系列大小为 56 字节的包。然后按下“Trace”按钮。得到的 pingplotter 窗口如图 6-7 所示。

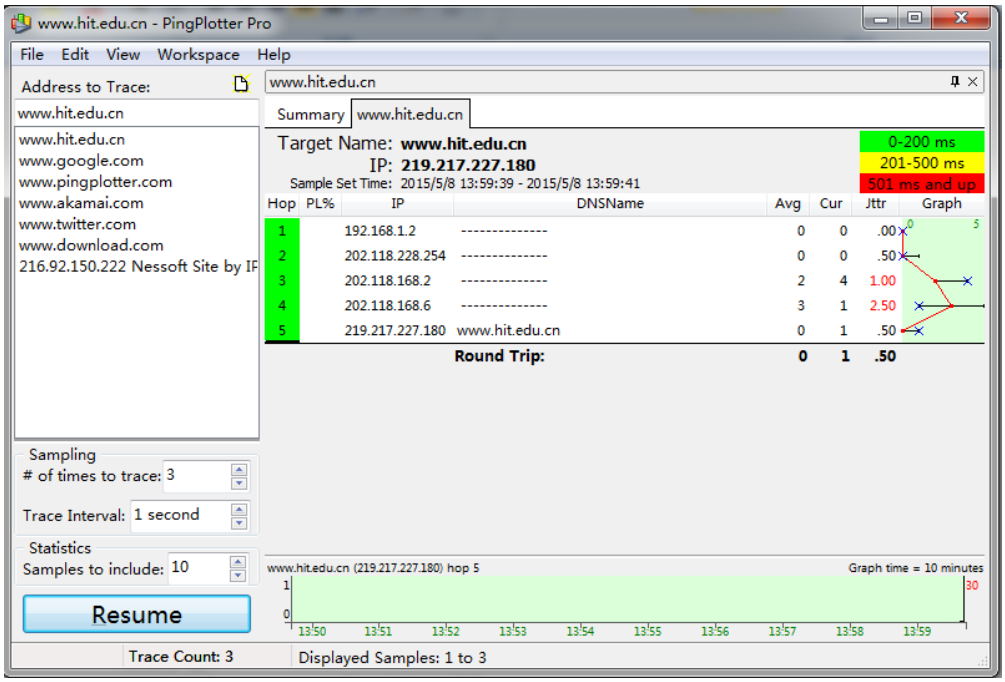

图 6-7 pingplotter 窗口

（1） Edit->Options->Packet，然后将 Packet Size(in bytes,default=56)域改为 2000，这样将发送一系列大小为 2000 字节的包。然后按下“Resume”按钮。

（2） 最后，将 Packet Size(in bytes,default=56)域改为 3500，发送一系列大小为 3500 字节的包。然后按下“Resume”按钮。

（3） 停止 Wireshark 的分组捕获。

注：如无法访问可与实验 TA 联系，下载已有的 ip-Wireshark-trace 文件，利用该文件进行 IP 协议分析

B. 对捕获的数据包进行分析

（1）在你的捕获窗口中，应该能看到由你的主机发出的一系列ICMP Echo Request包和中间路由器返回的一系列ICMP TTL-exceeded消息。选择第一个你的主机发出的ICMP Echo Request消息，在packet details窗口展开数据包的Internet Protocol部分，如图6-8所示。

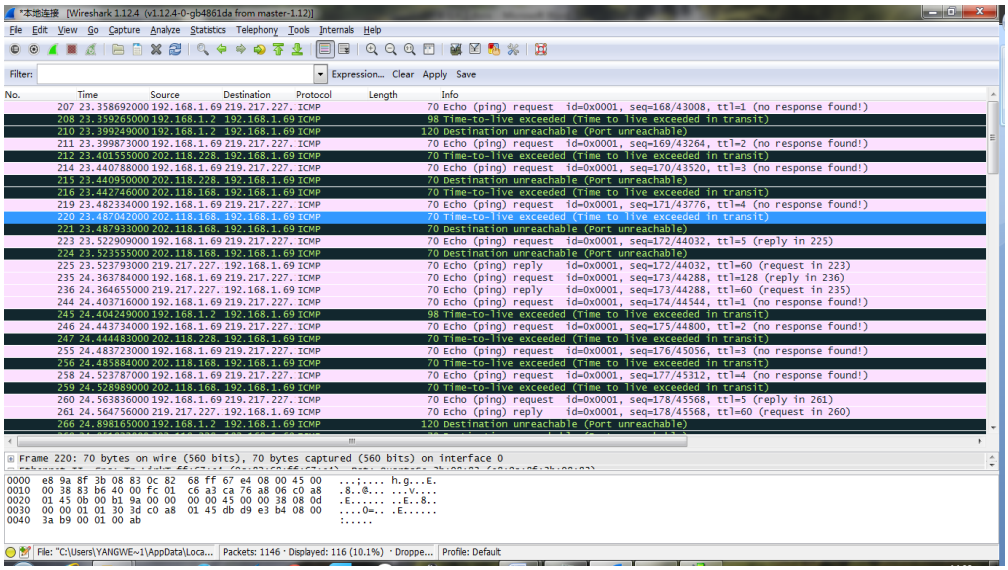

图6-8 Wrieshark窗口

思考下列问题：

- 你主机的IP地址是什么？
- 在IP数据包头中，上层协议（upper layer）字段的值是什么？
- 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？
- 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

（2）单击Source列按钮，这样将对捕获的数据包按源IP地址排序。选择第一个你的主机发出的ICMP Echo Request消息，在packet details窗口展开数据包的Internet Protocol部分。在“listing of captured packets”窗口，你会看到许多后续的ICMP消息（或许还有你主机上运行的其他协议的数据包）

思考下列问题：

- 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？
- 哪些字段必须保持常量？哪些字段必须改变？为什么？
- 描述你看到的IP数据包Identification字段值的形式。

（3）找到由最近的路由器（第一跳）返回给你主机的ICMP Time-to-live exceeded消息。

思考下列问题：

- Identification字段和TTL字段的值是什么？
- 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？

（4）单击Time列按钮，这样将对捕获的数据包按时间排序。找到在将包大小改为2000字节后你的主机发送的第一个ICMP Echo Request消息。

思考下列问题：

- 该消息是否被分解成不止一个IP数据报？
- 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

C. 找到在将包大小改为3500字节后你的主机发送的第一个ICMP Echo Request消息。

思考下列问题：

➢ 原始数据包被分成了多少片？

➢ 这些分片中IP数据报头部哪些字段发生了变化？

（五）抓取 ARP 数据包

（1）利用 MS-DOS 命令：arp 或 c:\windows\system32\arp 查看主机上 ARP 缓存的内容。

（2）在命令行模式下输入：ping 192.168.1.82（或其他 IP 地址）

（3）启动 Wireshark，开始分组俘获。抓取的数据包大致如下图 6-9 所示。

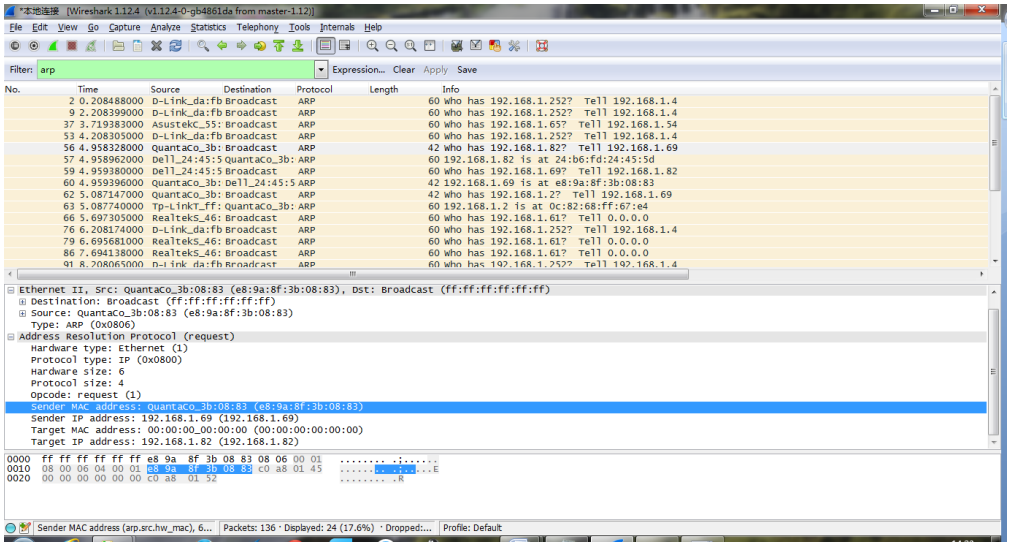

图 6-9 ARP 广播包

从 Wireshark 的第一栏中，我们看到这是个 ARP 解析的广播包，如上图。由于这个版本的 Wireshark 使用的是 Ethernet II 来解码的，我们先看看 Ethernet II 的封装格式。如图 6-10 所示。

图 6-10 Ethernet II 的封装格式

从 Ethernet II 知道了是 ARP 解析以后，我们来看看 Wireshark 是如何判断是 ARP 请求呢还是应答的。

以太网的 ARP 请求和应答的分组格式，如图 6-11 所示。

图 6-11 ARP 请求和应答的分组格式

从上图中我们了解到判断一个 ARP 分组是 ARP 请求还是应答的字段是“OP”，当其值为 0×0001 时是请求，为 0×0002 时是应答。如下两图：

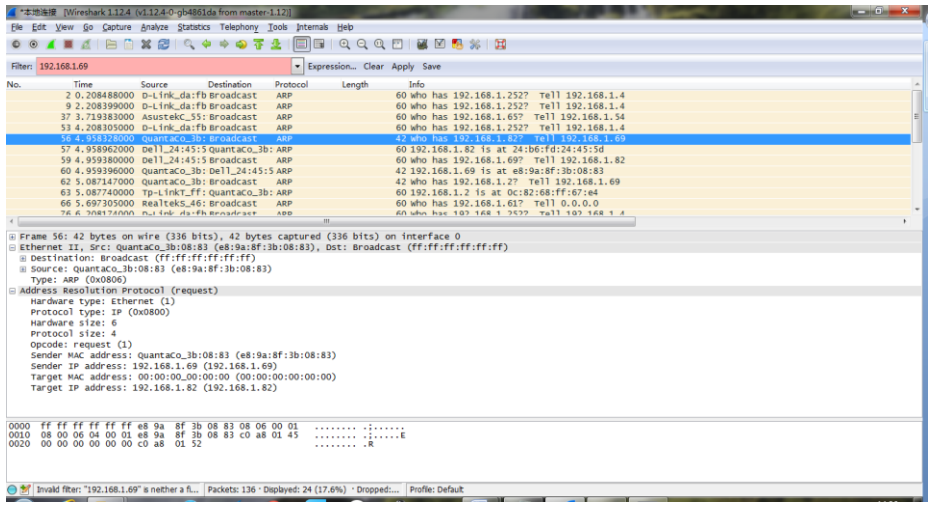

图 6-12 ARP 请求包格式

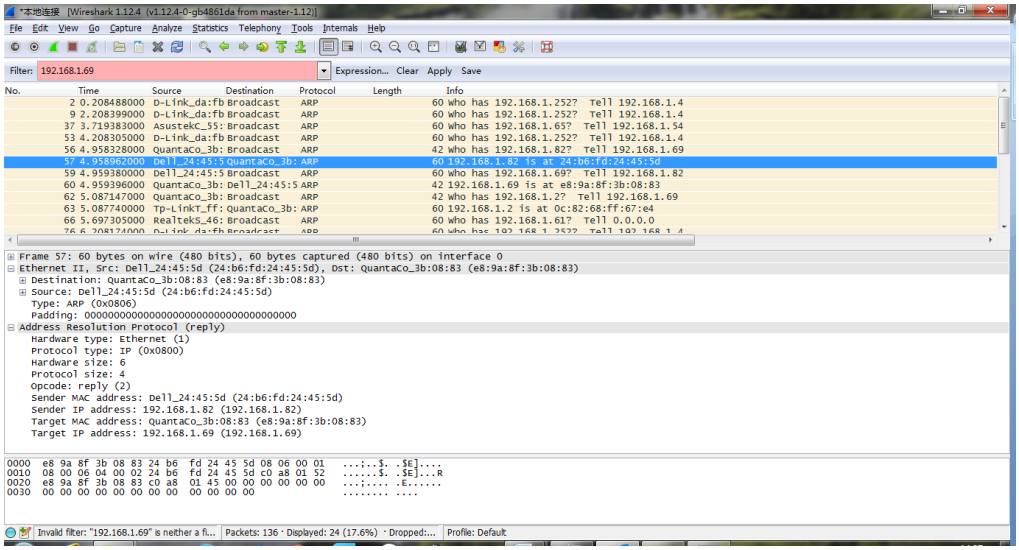

图 6-13 ARP 应答包格式

思考下面问题：

（1）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?

（2）清除主机上 ARP 缓存的内容,抓取 ping 命令时的数据包。分析数据包,回答下面的问题：

- 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？
- 如何判断一个ARP数据是请求包还是应答包？
- 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？

（六）抓取 UDP 数据包

（1）启动 Wireshark，开始分组捕获；

（2）发送 QQ 消息给你的好友；

（3）停止 Wireshark 组捕获；

（4）在显示筛选规则中输入“udp”并展开数据包的细节，如图 6-14 所示。

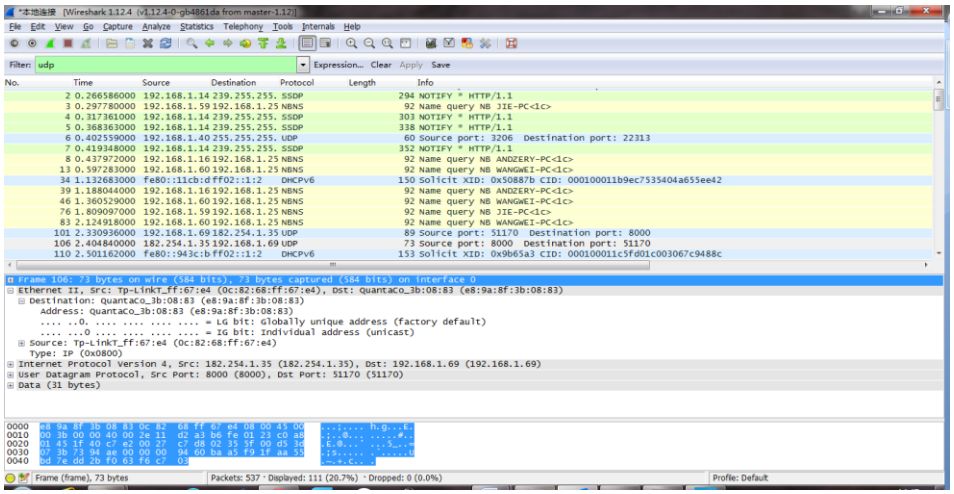

图 6-14 UDP 抓包图

分析 QQ 通讯中捕获到的 UDP 数据包。根据操作思考以下问题：

- 消息是基于UDP的还是TCP的？
- 你的主机ip地址是什么？目的主机ip地址是什么？
- 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？
- 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？
- 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？

（七）利用 WireShark 进行 DNS 协议分析

（1）打开浏览器键入:www.baidu.com

（2）打开 Wireshark,启动抓包.

（3）在控制台回车执行完毕后停止抓包.Wireshark 捕获的 DNS 报文如图 6-15 所示。

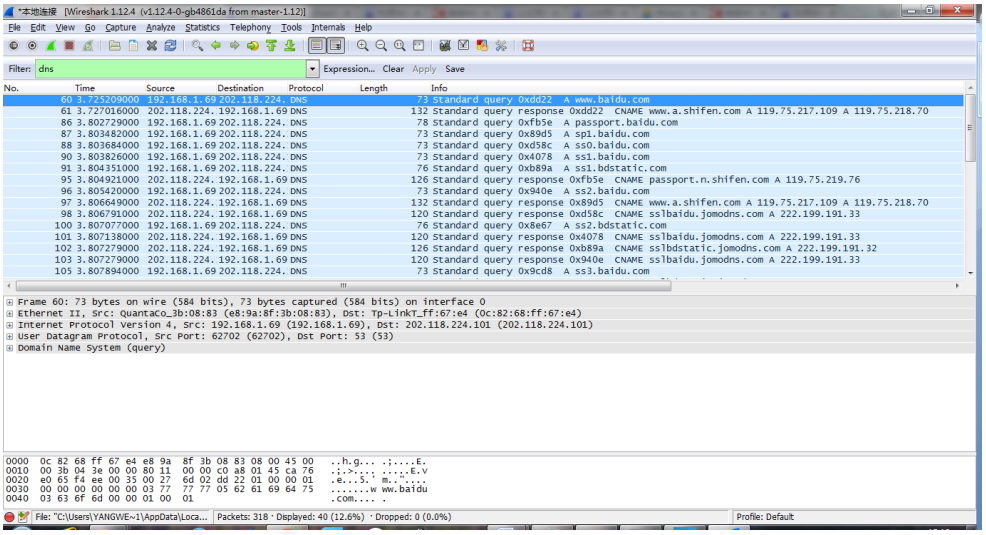

图 6-15 DNS 报文

6. 实验报告

要求撰写实验报告对利用 Wireshark 分析 HTTP、TCP、IP、以太网帧、ARP、DNS 等的抓包分析实验过程、发现的问题、得到的结果、对协议的认识等内容进行总结（可结合每个实验后面的思考题进行分析、总结）。