# 实验2 真实网络环境协议分析

## 实验目的

了解和熟悉常见网络协议的内部通信过程

## 实验内容

安装网络包捕获软件，观察网络中的数据包

## 实验环境

PC机、Ethereal软件（wireshark），Windows8.1 操作系统

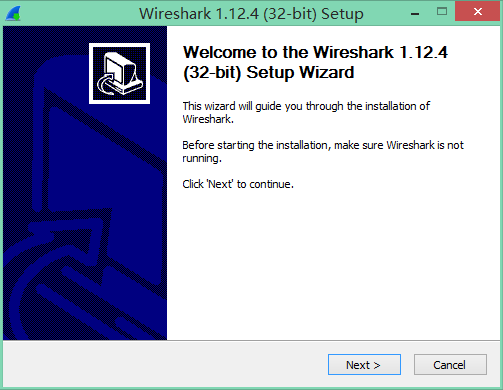
## 实验时间

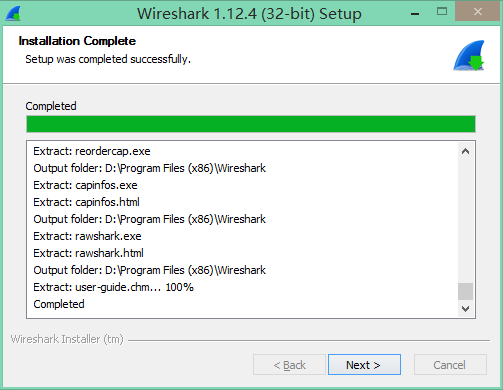
2机时。

## 实验步骤：

* **安装网络包捕获软件Ethereal**

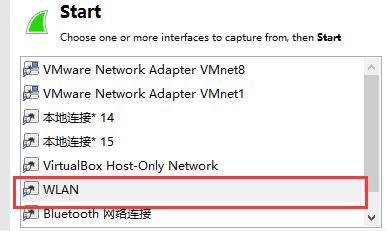
下载wireshark软件并安装。



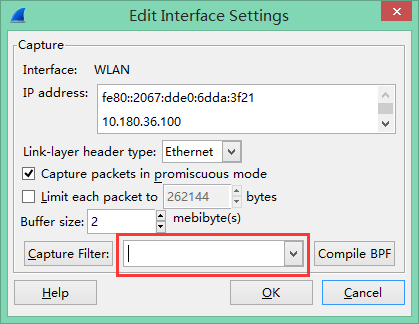


* **配置网络包捕获软件，捕获所有机器的数据包**

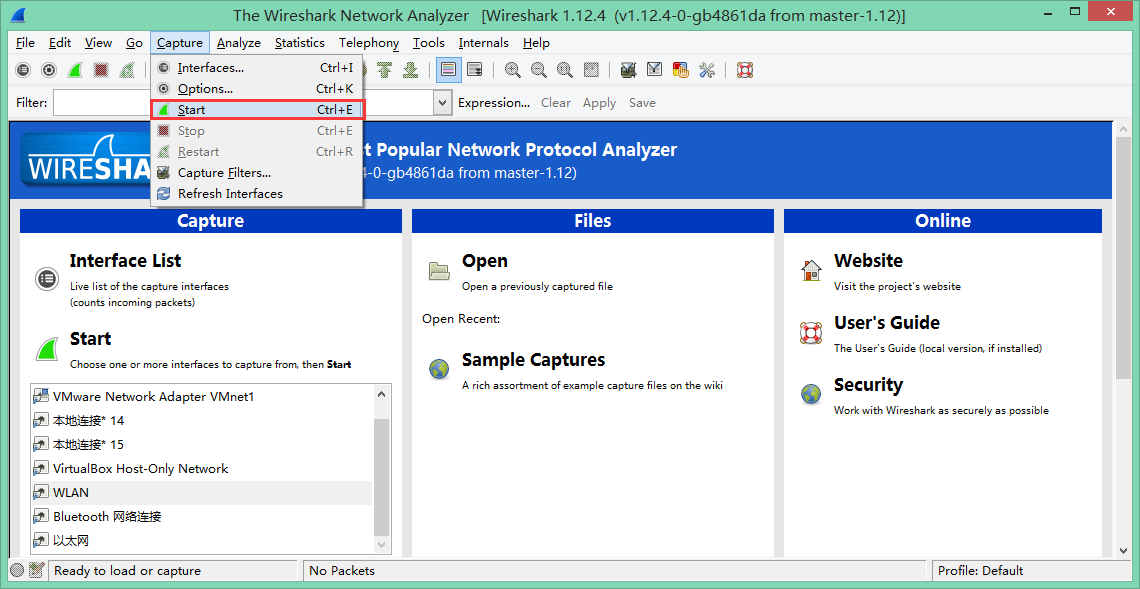
首先，需要在界面中选择当前的网络接口，由于本次实验是在无线网络的环境下完成的，因此这里我们选择WLAN。



接下来会弹出一个窗口，在这个窗口中，不对抓包过滤器添加任何内容，则默认抓取网络中的所有数据包。

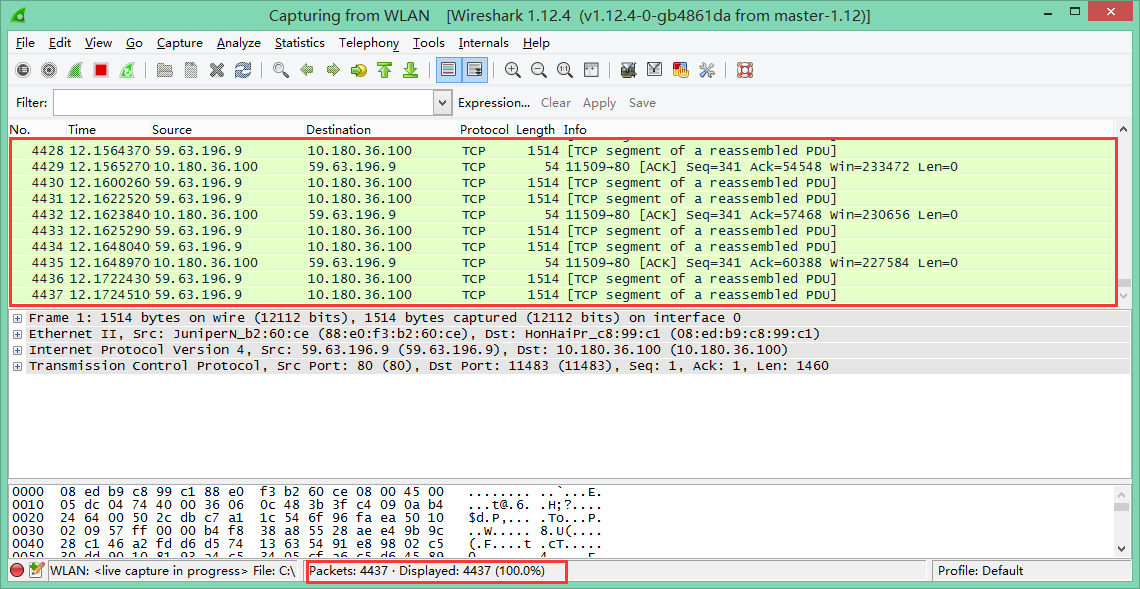


接下来在capture一栏中，选择Start开始抓包。



可以在下图中的红色方框中看到所有捕获的包，主要信息包括源地址，目的地址，协议名，长度以及包的主要信息内容。

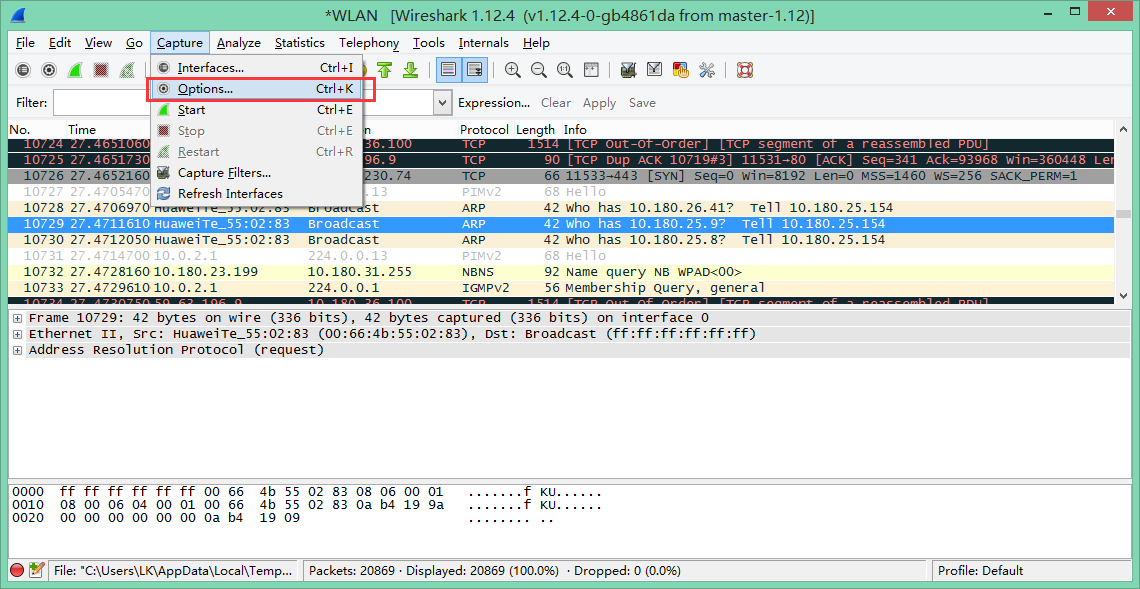
点击其中一个包，可以在下方看到该包的详细内容。其中紧挨着包列表的方框中，可以看到包的大小，捕获大小，包的源地址，目的地址，协议信息，源端口，目的端口等内容。以及包的大致内容等信息。



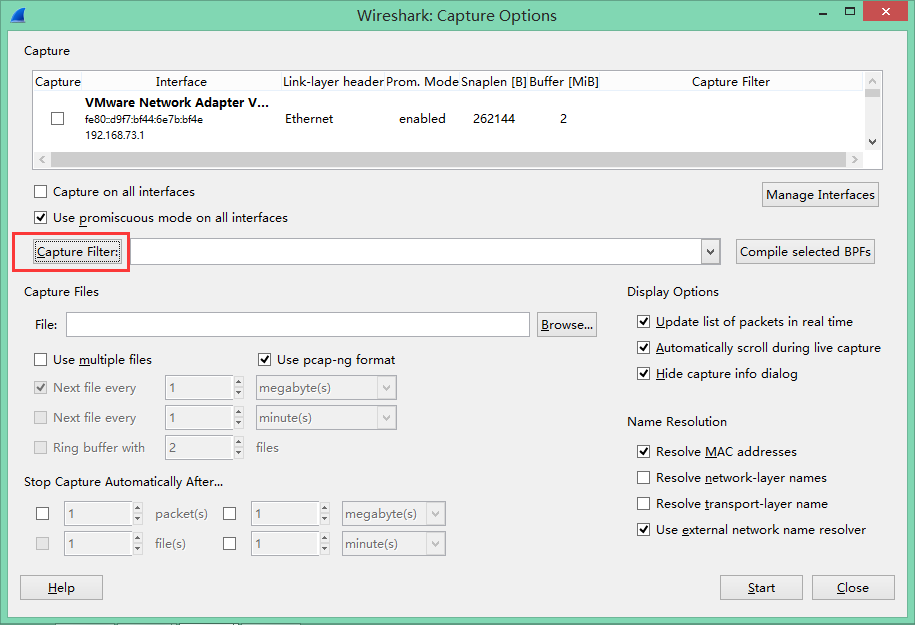
在最下方红框圈出的部分，显示了总共抓取到的包的数量以及显示的包的数量，其中抓取的包可以通过抓取过滤器进行过滤，而显示包数量可以通过显示过滤器进行过滤。

* **配置网络包捕获软件，只捕获特定IP或特定类型的包**

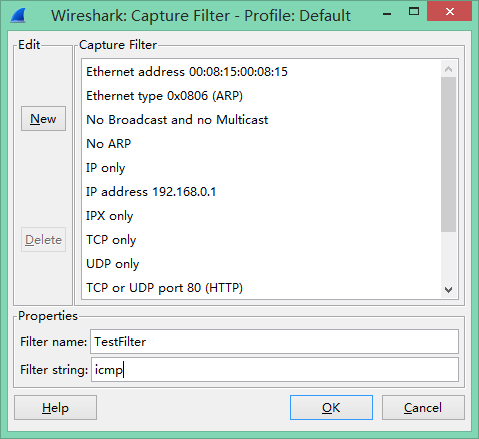
在capture捕获一栏中的option选项中



选择capture filter过滤器添加一个过滤器



输入过滤器的名字可以方便在下次直接使用，在Filter string中输入icmp表明该过滤器只捕获icmp协议的数据包. 当然，还可以通过and ,or 等符号添加更多的条件，扩大或缩小捕获数据包的范围，使捕获对象更加具体

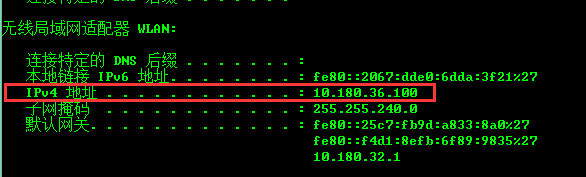


接下来进行测试，我们使用基于ICMP协议的ping 命令来进行测试。

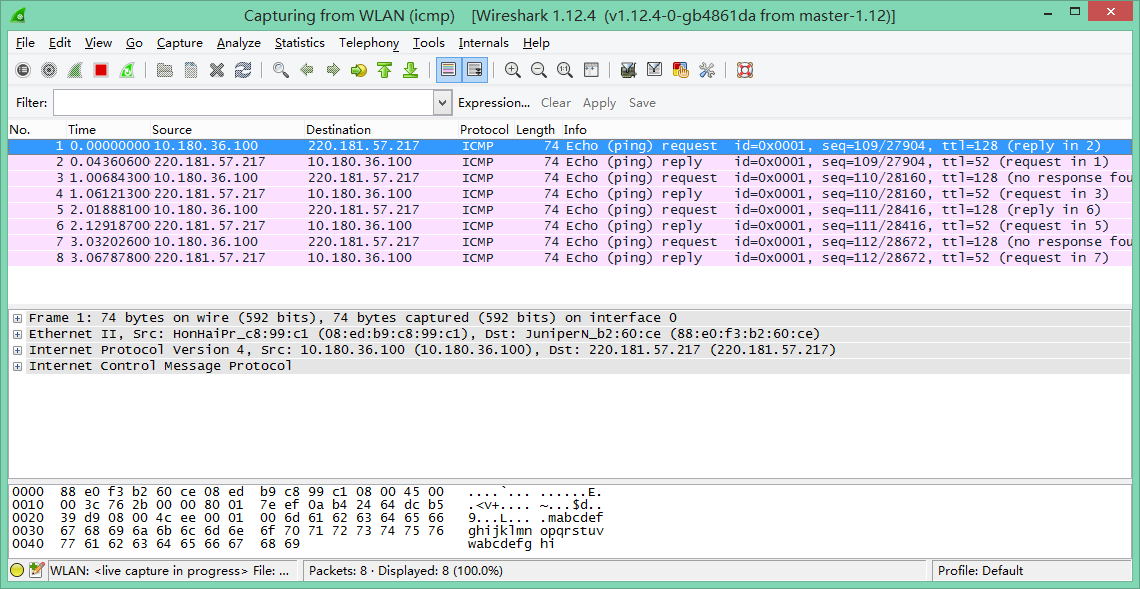
这个命令一共发送了4个数据包包到baidu.com的服务器



我们使用ipconfig命令查看本机的IP地址，为10.180.36.100

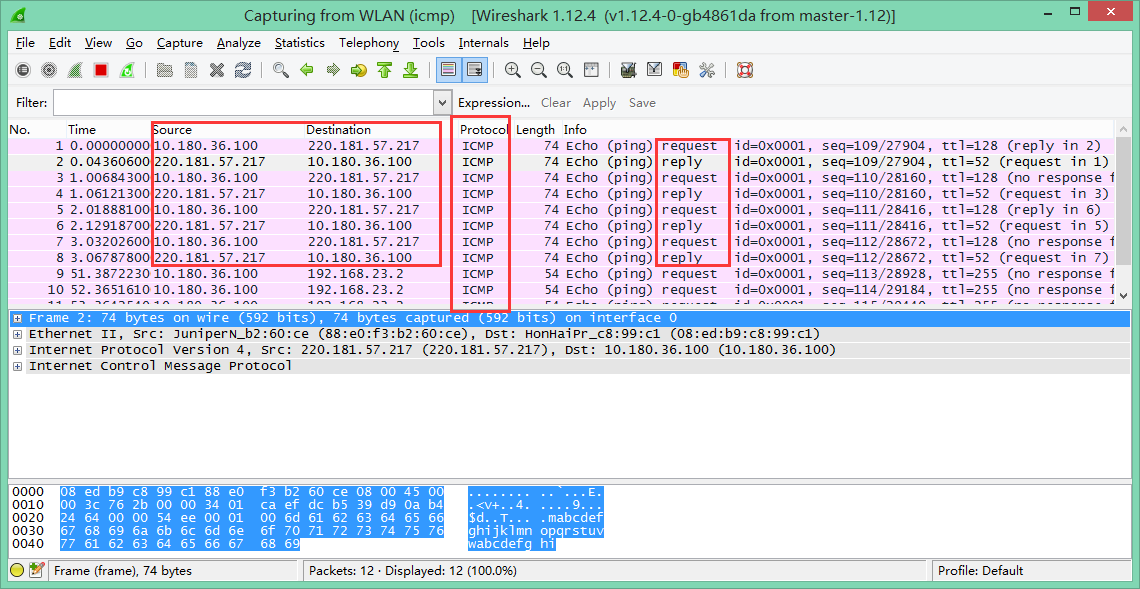


接下来返回主界面，发现已经捕捉到了一些数据包，由于这次我们限定了捕捉数据包的类型，而ICMP协议数据包数量很少，事实上这里除了我们自己发送的数据包意外，并没有接收到任何其他基于ICMP协议的数据包，因此总数据包的量很少。

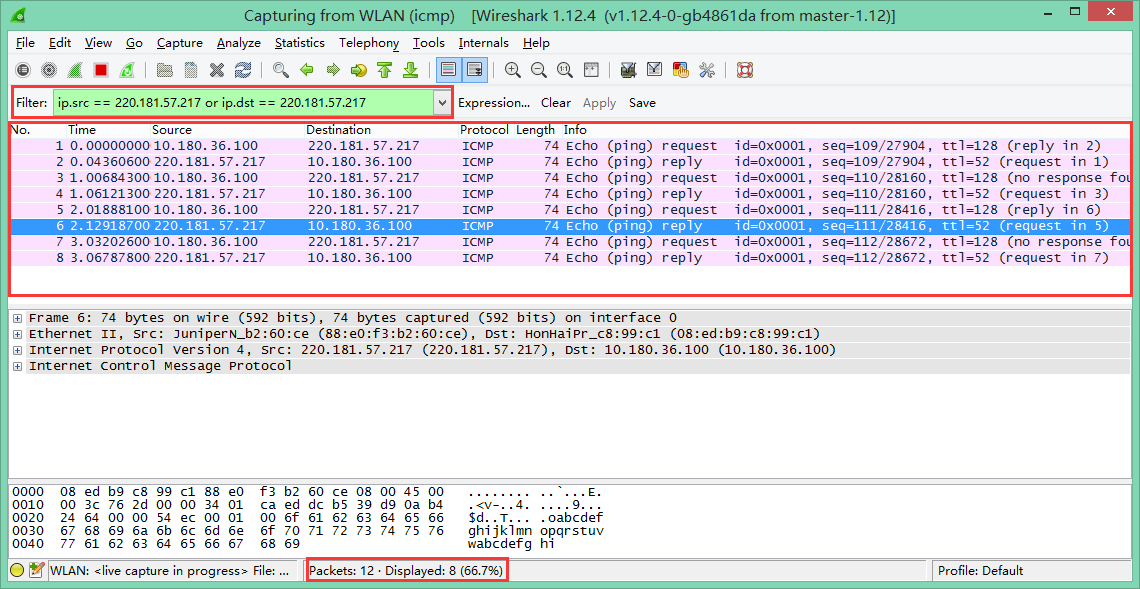


在图中，我们可以发现数据包的一些特点和内容，如下图所示，首先，所有数据包的协议都是ICMP的，这是我们设置了过滤器后筛选的结果，另外，我们发现前几个数据包的Source源地址和Destination目的地址都是本机地址（10.180.36.100）和百度的IP地址（220.181.57.217），总共8个包，分别对应了4次的本机请求包和服务器回应的数据包，也就是ping命令发出得4个包和接收4个包。

在每个数据包的最后，对应的数据包的大致内容，包括延迟时间和数据包的生存最长时间等。



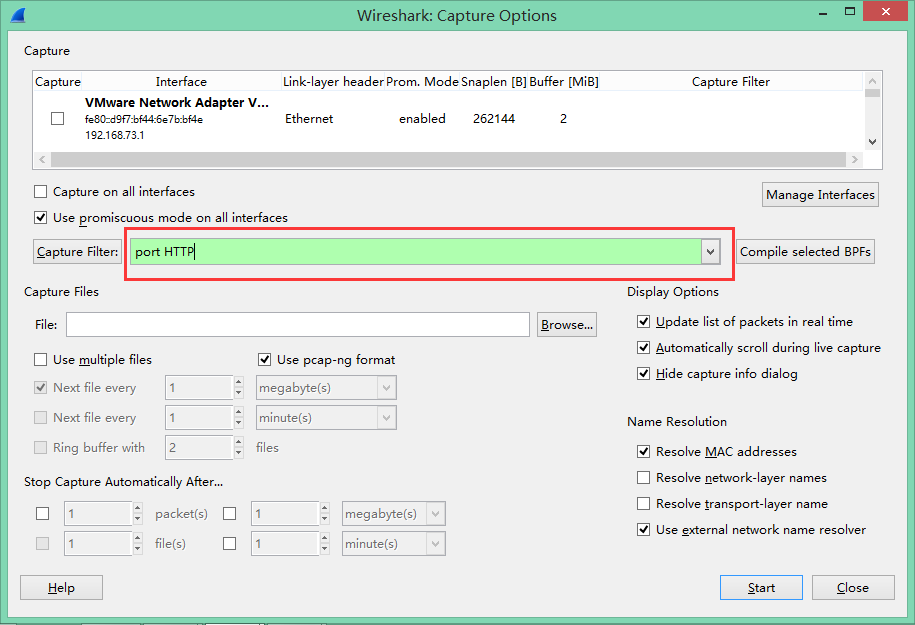
如果我们希望仅仅查看收发的数据包，还可以在display filter中输入相应的指令进行筛选。不同于capture filter, display filter只是在捕获数据包后对要显示的数据包进行筛选，但事实上那些没有显示的数据包依然被捕获。如下图所示：



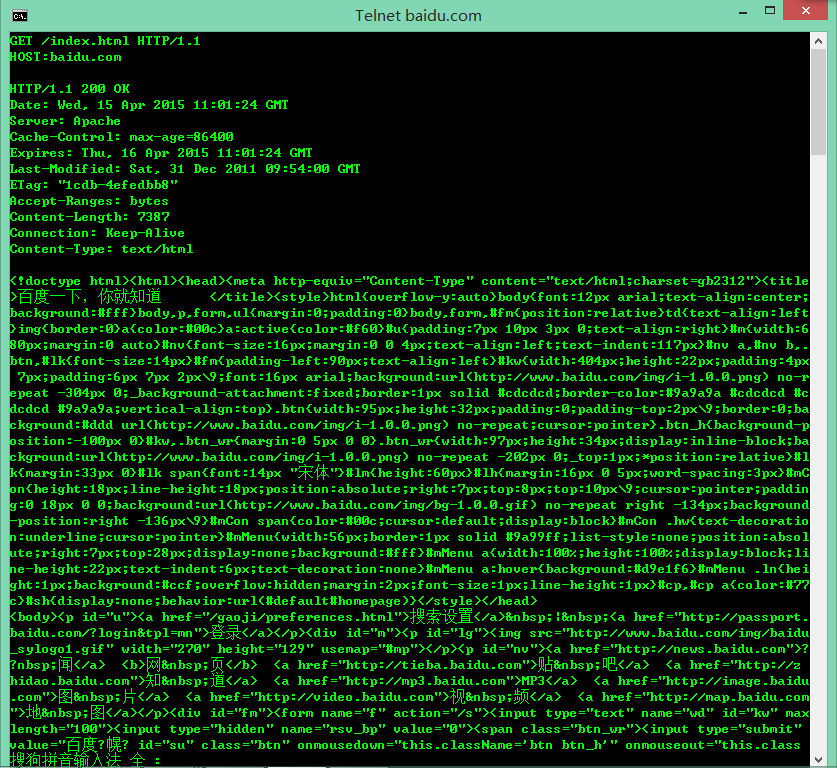
如图，捕获了12个数据包，但只显示了8个。因为我们将显示的数据包限定为和百度有关的数据包，即源地址和目的地址中必须有一个是百度服务器的地址。通过or 和ip.src, ip.dst来设置源IP地址和目的IP地址。

* **跟踪一次HTTP会话数据包**

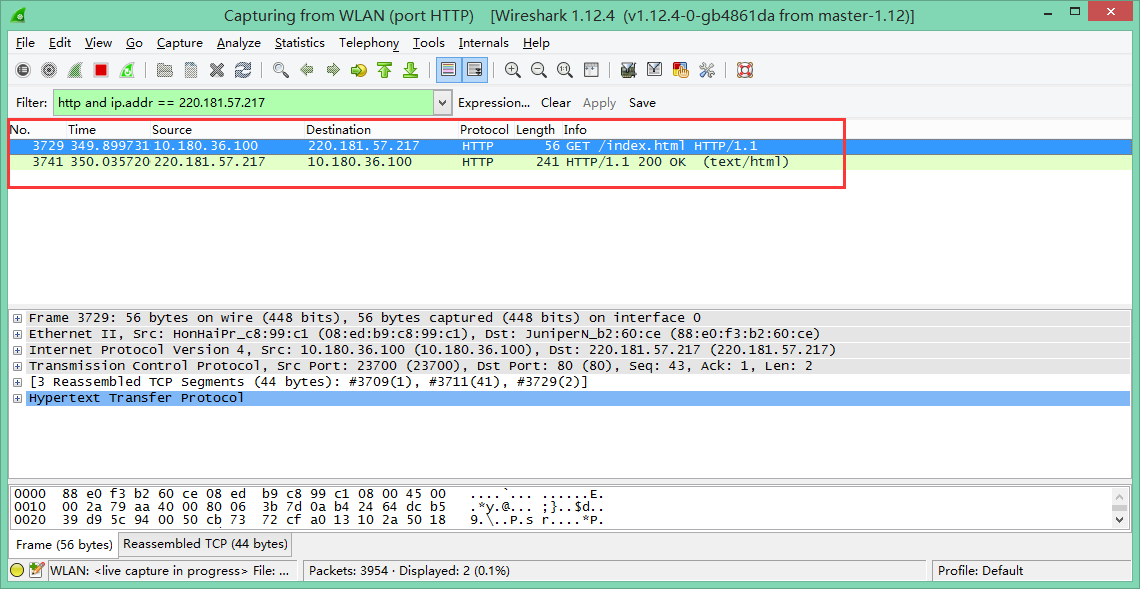
将capture filter中的筛选条件改为port HTTP, 使过滤器过滤掉那些和HTTP协议无关的数据包。



由于现在绝大多数情况下使用的都是HTTPS协议，因此本次实验使用telnet的方法连接服务器，并使用GET命令，以HTTP1.1的协议传送文本，来捕获数据包。

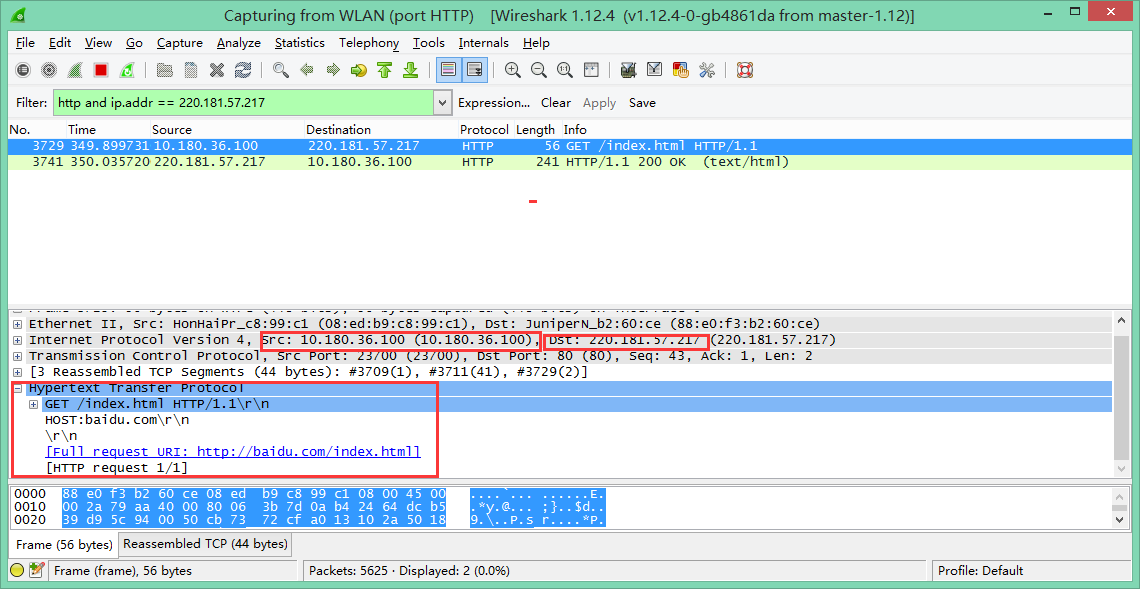


使用telnet连接服务器并请求HTTP页面数据的方法已经在上次实验中描述过了，这里不多做赘述。请求得到响应并成功收到HTML数据后，停止数据包的捕获，回到wireshark主界面，可以看到：



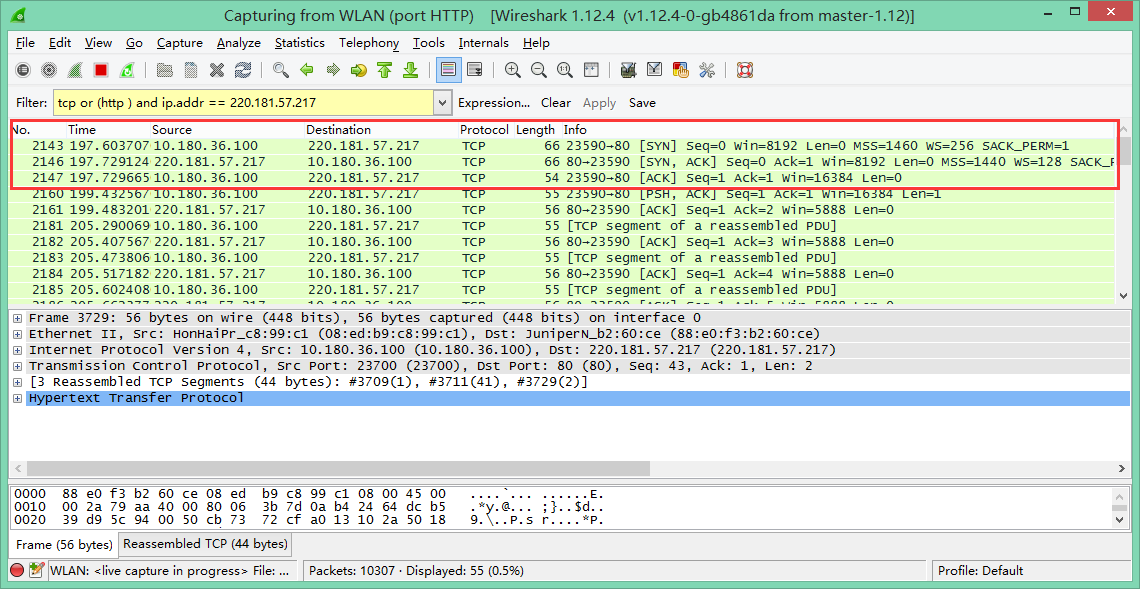
两个包中，第一个包是主机向服务器发送请求的包，可以在最右侧看到内容包括GET /index.html HTTP/1.1的内容，第二个包是服务器响应，200代表成功。

下面查看第一个包的内容。



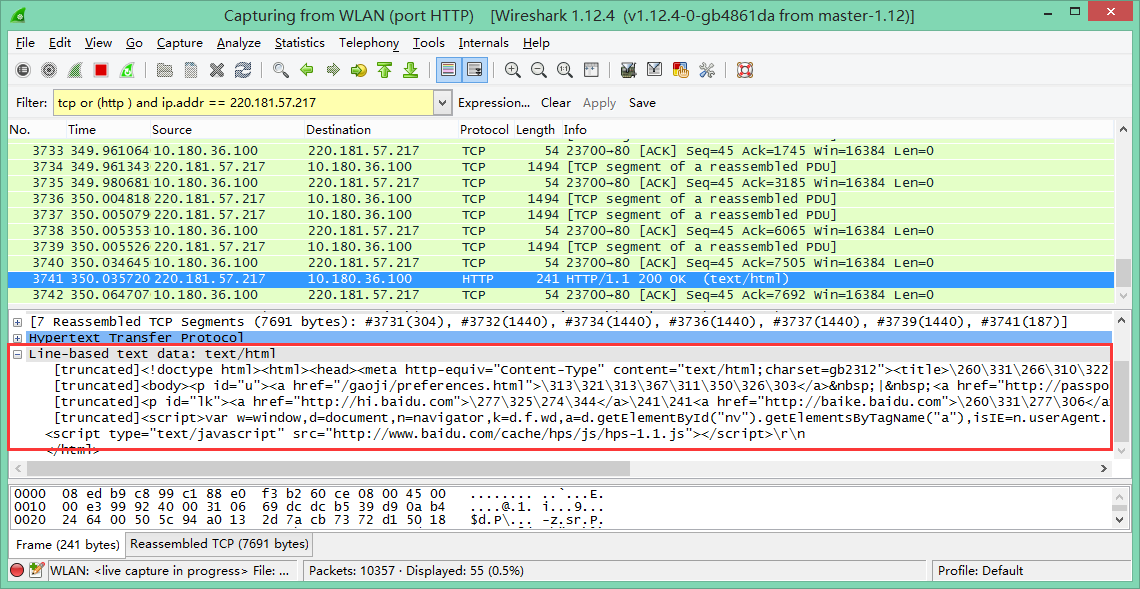
可以在包的详细内容中看到数据包源地址，目的地址，并在HTTP中找到了请求的详细内容。

我们将筛选范围扩大，可以看到TCP协议的包：



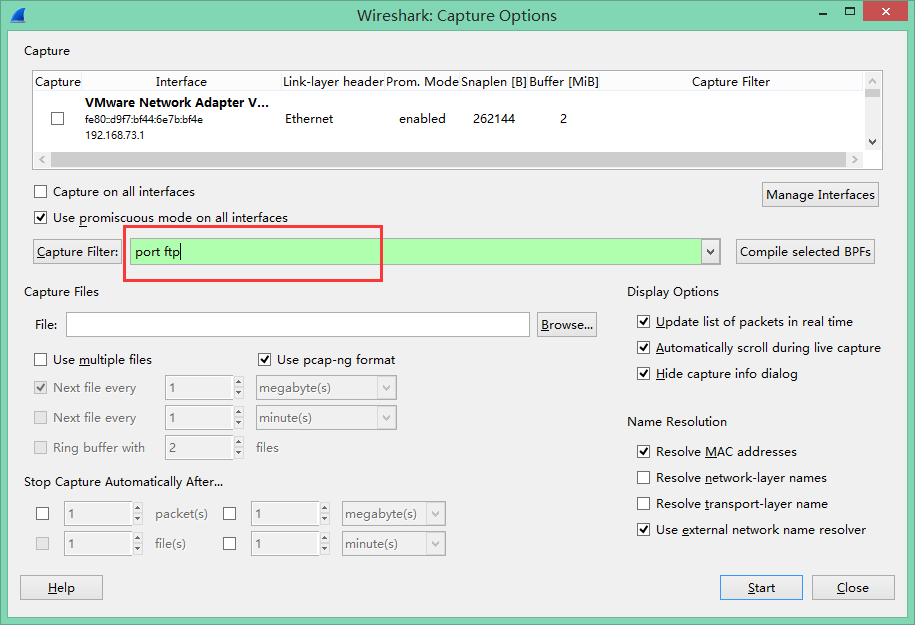
由于HTTP协议是基于TCP协议建立连接的，所以会有三个 TCP的数据包，分别代表了三次握手协议，用于建立连接。

另外，可以在收到的HTTP数据包中找到HTML的详细内容。如下图所示：

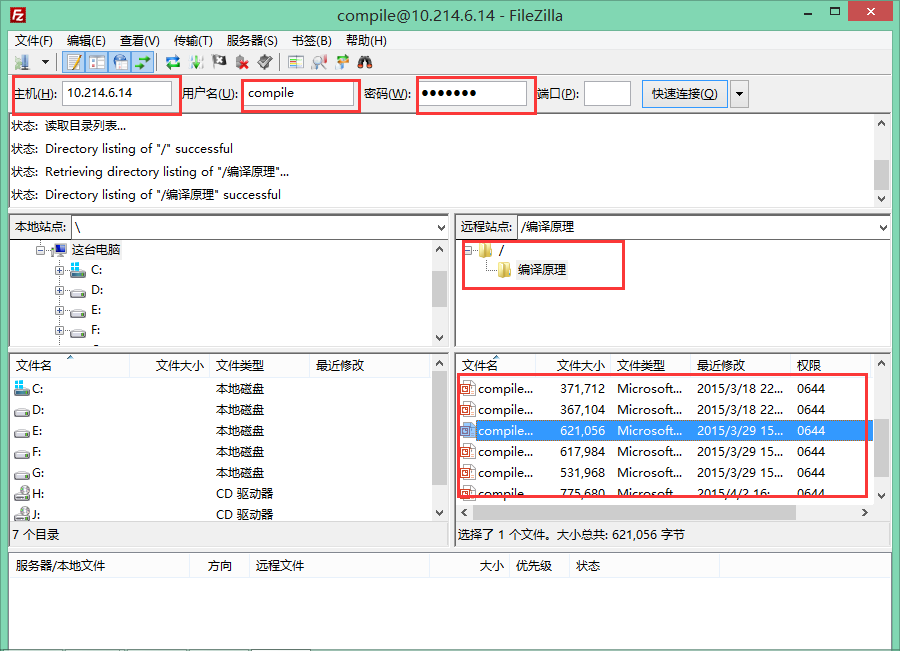


* **跟踪一次FTP会话的数据包**

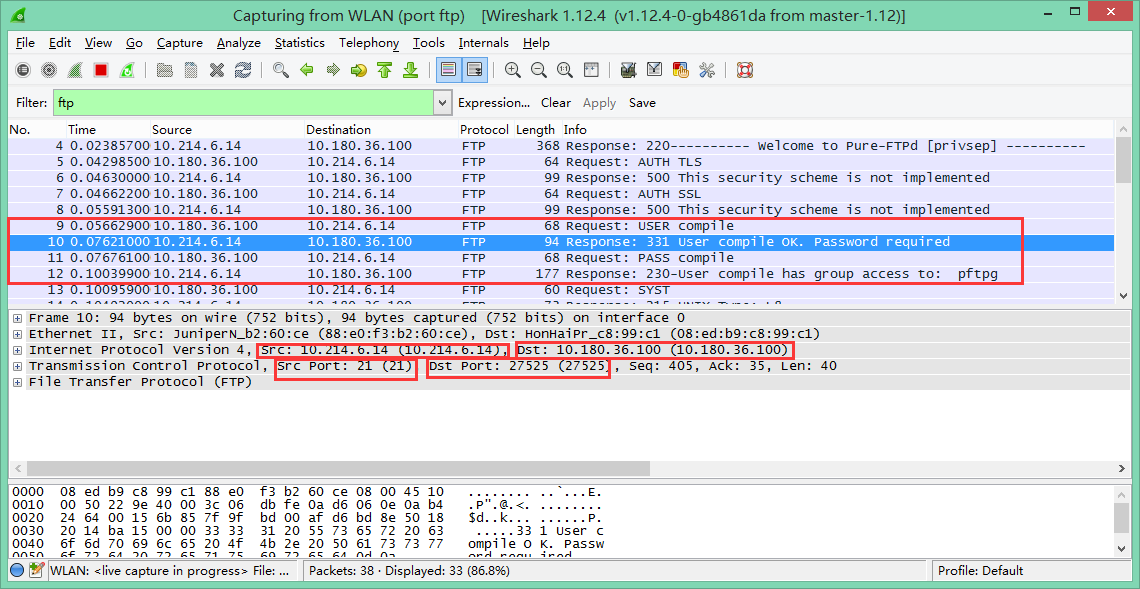
将抓取范围限定到FTP协议的数据包



用FTP客户端登陆FTP：

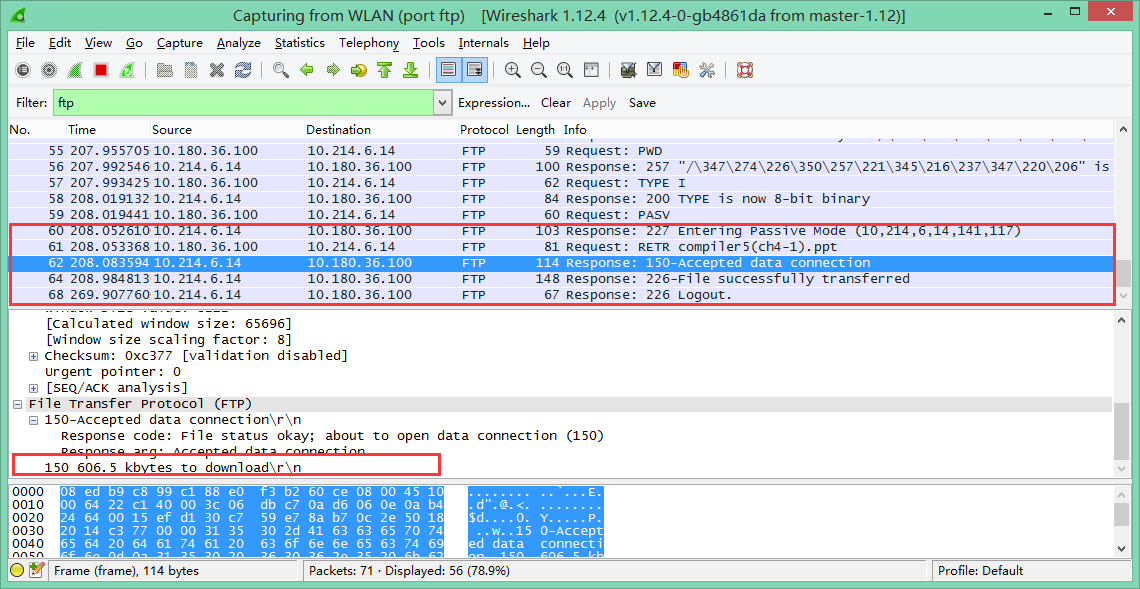


选中其中一个文件下载到本机，并结束数据包的抓取，返回主界面：



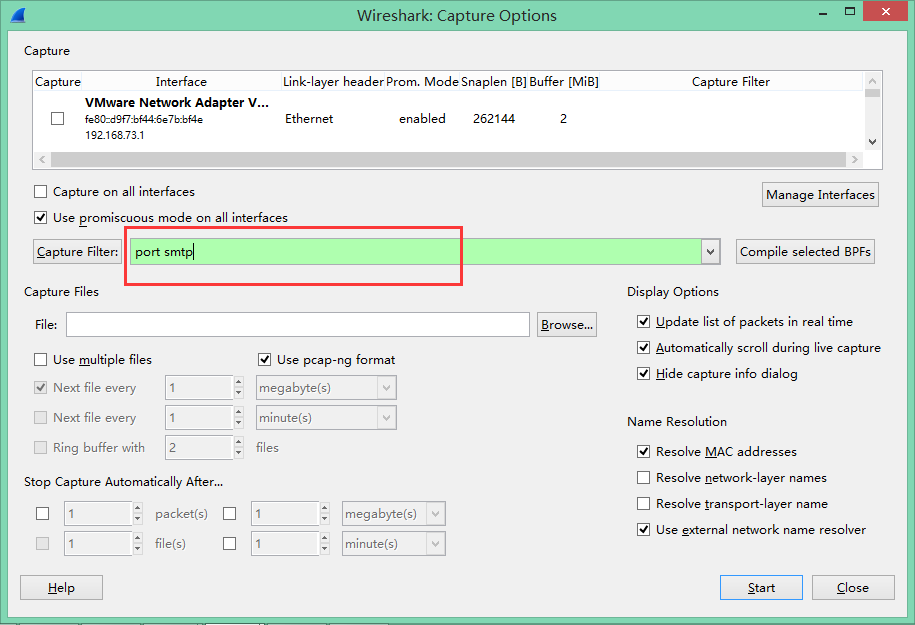
数据包列表中的红框部分的数据包主要用于身份的验证。点击其中一个数据包可以得到本机和服务器的IP地址，以及两者会话的端口。比如本机使用27525端口，服务器使用21端口

找到传送文件的数据包，可以获得数据包的大小等信息，如图：

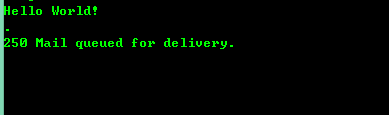


* **跟踪一次SMTP会话的数据包**

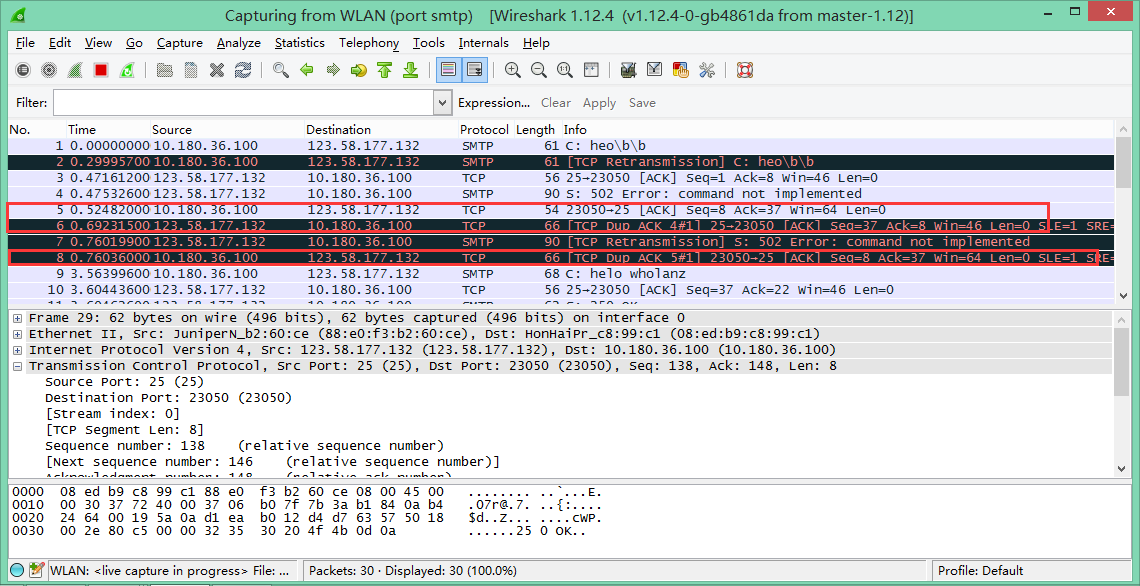
将抓包范围限定在SMTP协议的数据包：



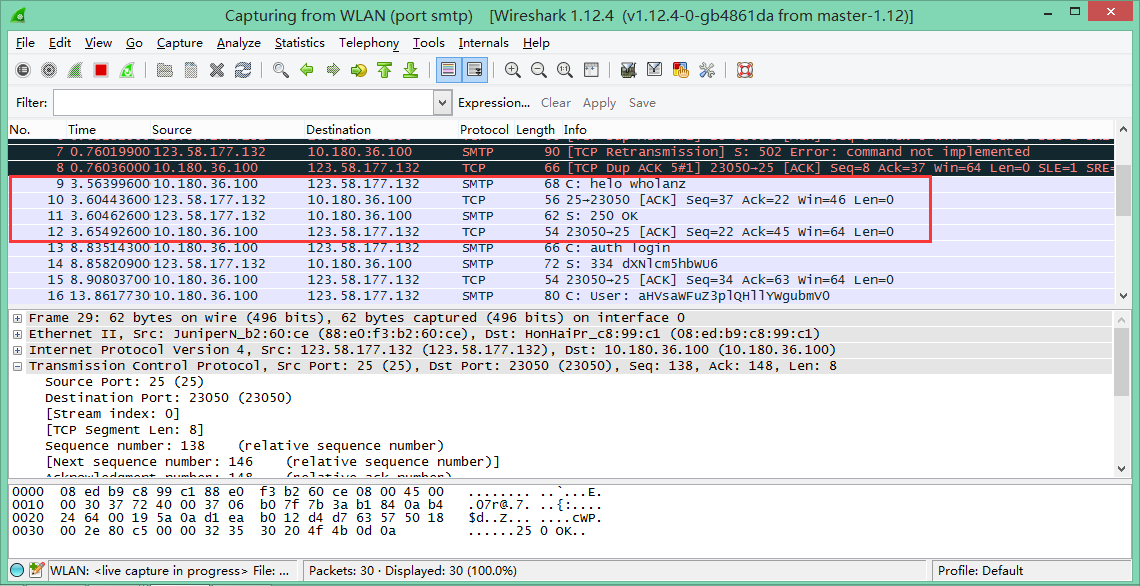
由于现在大多数邮件并不是通过SMTP协议发送的，所以本次实验通过telnet连接到服务器并发送一封邮件，过程已经在上次的实验中详细描述过，这里不多做赘述



SMTP协议同样是基于TCP协议建立连接的，因此我们同样可以看到代表三次握手协议的三个TCP数据包：

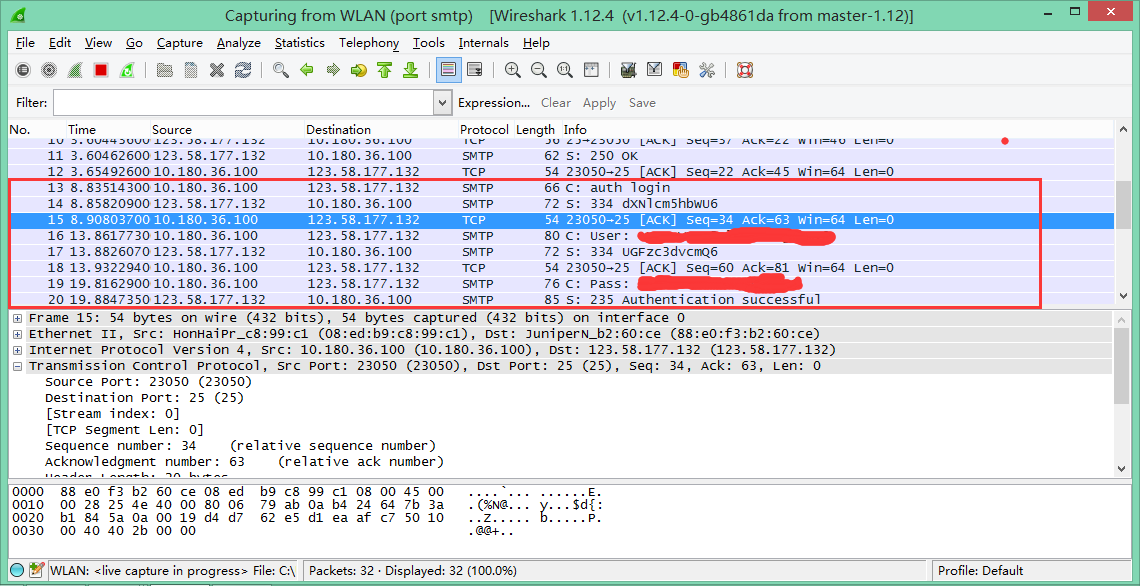


连接到服务器后，还需要进行身份确认：



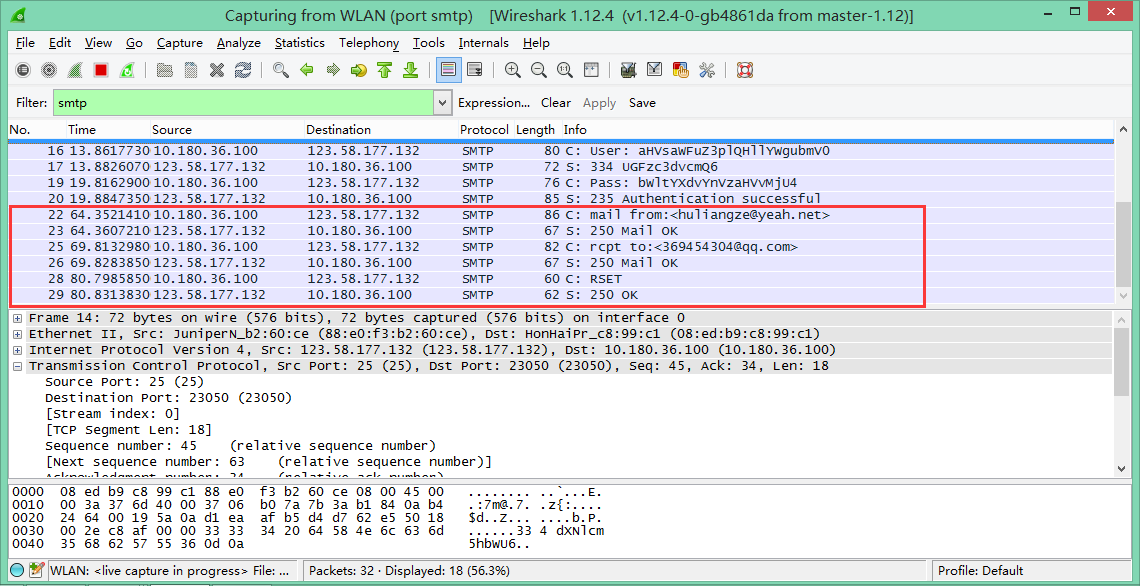
图中的两个SMTP包内容分别是本机向服务器主动表明身份以及服务器确认。

接下来需要用base64加密过后的邮箱账号密码登陆服务器：



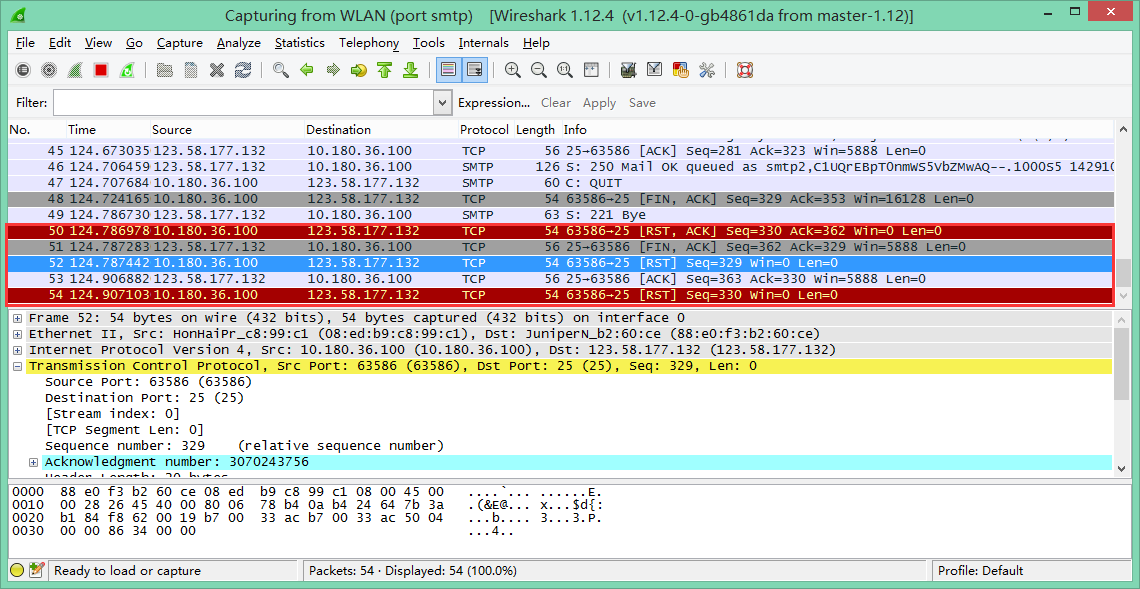
图中的几个SMTP包内容分别是服务器要求输入账号密码以及本机发送账号密码，具体过程和原理在上次实验中已经讲述过，这里不多做赘述

接下来确认邮件发送者以及邮件接收者：

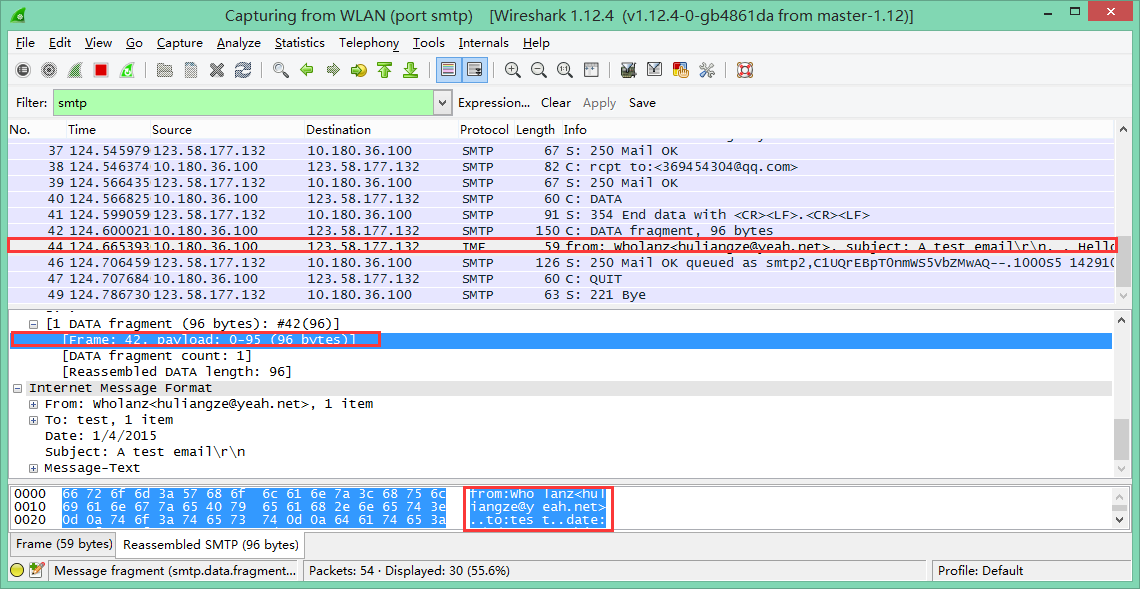


几个SMTP包的内容分别包括了邮件发送者地址，邮件接收者地址，已经服务器确认的信息。

最后和服务器断开连接。



另外，还可以在下图的数据包中找到相应的邮件属性信息：



比如，在红框的数据包中，我们发现邮件大小有96K，并且在数据包中找到了邮件的详细内容（下方红框中标注），在16进制内容中翻译后得到结果。

## 实验心得

本次实验让我学会了如何巧妙的使用wireshark软件抓取网络中的数据包，并分析数据包的内容，最大的收获包括如下几点：

1. 抓取数据包和分析数据包的过程让我对HTTP协议、ICMP协议、FTP协议以及SMTP协议有了更加深入的了解，并熟悉了协议的行为过程。
2. 学会了如何利用筛选器筛选数据包
3. 对网络协议产生了更加浓厚的兴趣，期望在以后的实验中学到更多