|  |  |
| --- | --- |
| 标题 | 实践作业3：SMTP协议分析与编程 |
| 发布时间 | 2016年10月26日 |
| 截止时间 | 2016年12月25日 |
| 评分方式: | 打分制:100.0分 |
| 作业内容 | 分析SMTP协议，编写一个简单的SMTP服务器。  1）利用Wireshark捕获数据包，分析一个电子邮件客户端（如outlook express）与SMTP服务器的交互过程。  2）编写一个简单的SMTP服务器。a）实现的服务器应能与outlook express等现有的客户端进行简单的交互；b）记录浏览器与服务的交互过程；c）在屏幕上直接显示接收到的邮件内容；d）将邮件附件中的图像（如bmp或jpg图像）显示在屏幕上；e）可以采用C/C++或Java语音进行实现。  3）提交报告，对以上内容进行总结和分析（包括程序的结构和说明）。 |

**信息安全 1410658 杨旭东**

**一、实验目的及任务**

1、通过对WireShark抓包实例进行分析，进一步加深对SMTP网络协议的理解。

2、编写一个简单的SMTP服务器，实现作业（2）中（a）（b）（c）（d）（e）项所要求的功能。

**二、实验环境**

1、系统环境：Windows 10 家庭中文版 1607

2、邮件客户端：

3、WireShark：Version 2.2.1

4、IDE：Visual Studio 2015 Update 3

5、调试环境：MFC Release x86

6、编程语言：C++

**三、实验原理**

1、利用Wireshark软件抓取本地PC的数据包，观察其主要使用的网络协议。

2、根据所获数据包的内容分析相关协议，从而加深对SMTP网络协议的理解。

3、利用Winsock实现本地SMTP服务器，并实现与邮件客户端（如outlook）的交互。

4、SMTP工作原理：

（1）客户端使用TCP协议连接SMTP服务器的25端口；

（2）客户端发送HELO报文将自己的域地址告诉给SMTP服务器；

（3）SMTP服务器接受连接请求，向客户端发送请求账号密码的报文（AUTH LOGIN）；

（4）客户端向SMTP服务器传送账号和密码，如果验证成功，向客户端发送一个OK命令，表示可以开始报文传输；

（5）客户端使用MAIL命令将邮件发送者的名称发送给SMTP服务器；

（6）SMTP服务器发送OK命令做出响应；

（7）客户端使用RCPT命令发送邮件接收者地址，如果SMTP服务器能识别这个地址，就向客户端发送OK命令，否则拒绝这个请求；

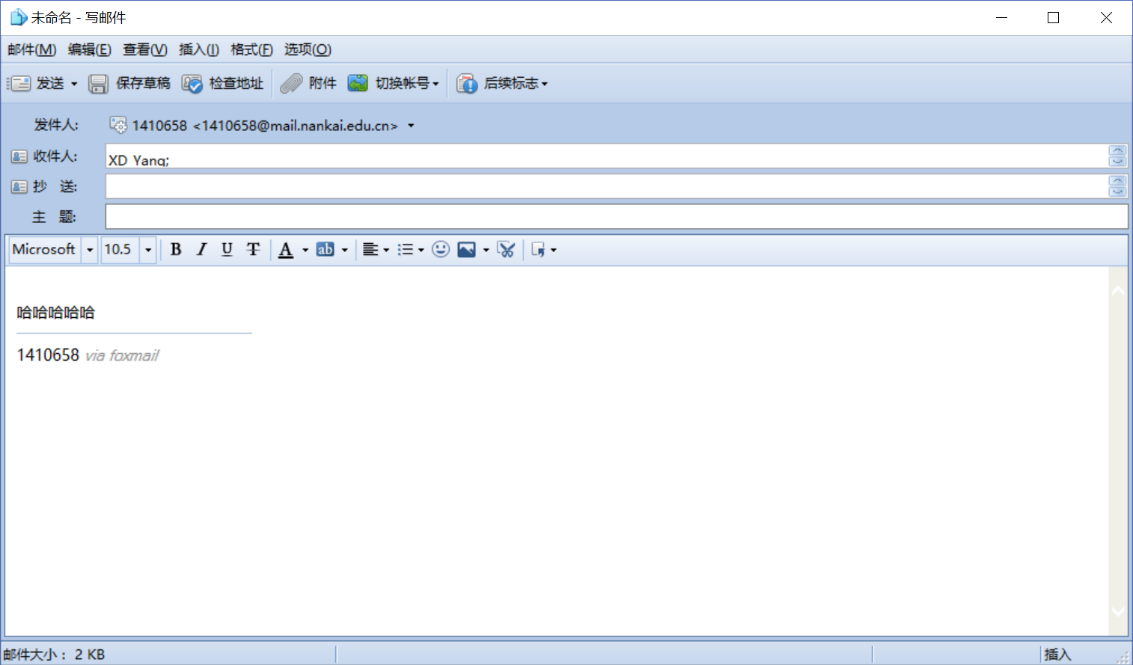
（8）收到SMTP服务器的OK命令后，客户端使用DATA命令发送邮件的数据。

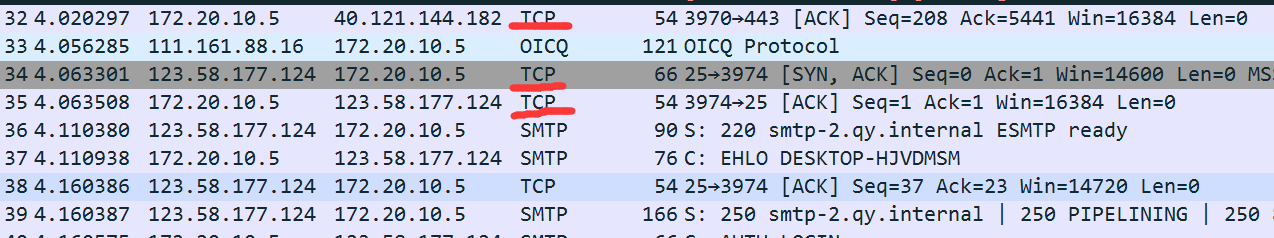
（9）客户端发送QUIT命令终止连接。

**四、实验步骤**

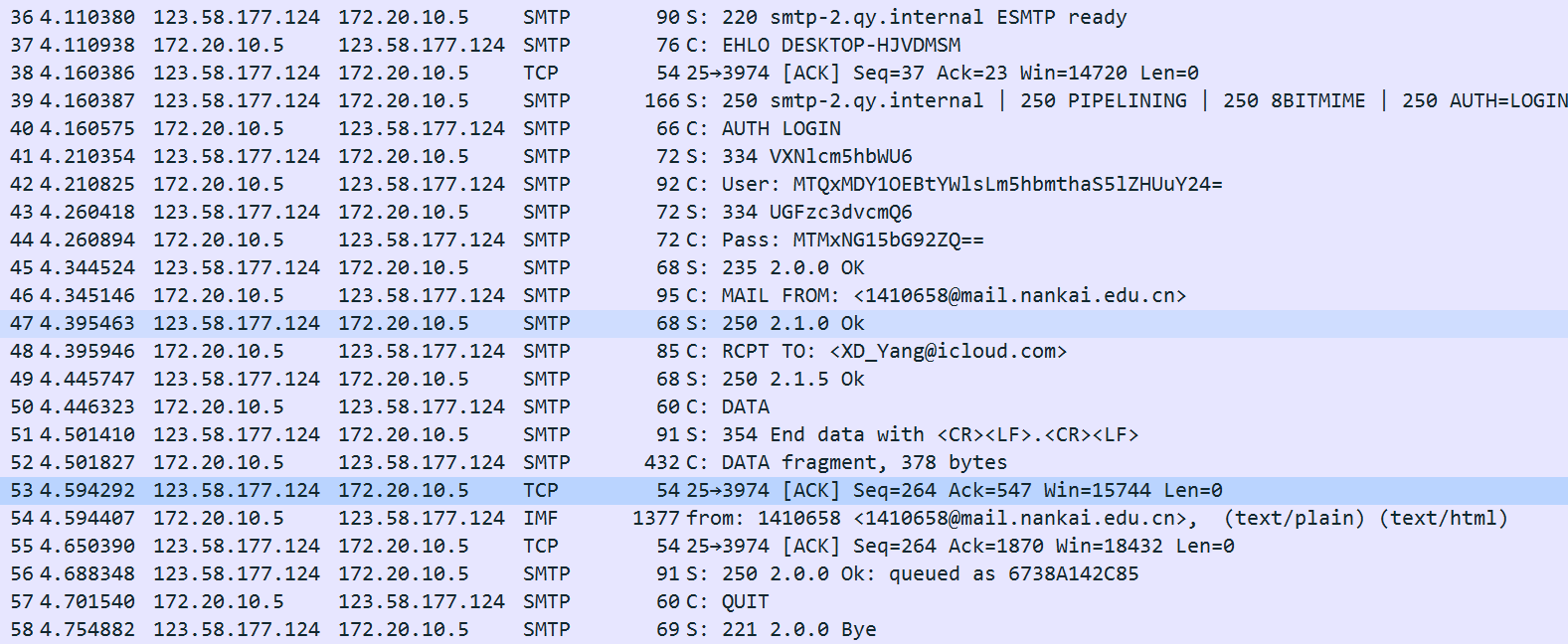
**1.SMTP协议抓包分析**

 开始我使用foxmail邮箱登陆结果总也抓不到数据包，并不知道是何原因，之后在foxmail客户端登陆了南开邮箱，抓包成功。配置如下：

打开Wireshark，开始捕获，然后使用foxmail客户端发送一封邮件：

停止捕获，查看Wireshark中的数据。

因为SMTP协议是基于TCP的，所以先进行三次握手，客户端与服务器建立TCP连接，服务器返回连接信息，表示是否连接成功（S: 220 smtp-2.qy.internal ESMTP ready服务器发送220表示已经准备就绪）。

客户端向服务器发送命令HELO，并加上本机的主机名DESKTOP-HJVDMSM，服务器响应并回复250表示服务器可用。

客户端向服务器发送用户登录命令AUTH LOGIN，服务器回复的两个334分别表示用户名和密码，之后客户端分别向服务器发送编码后的用户名和密码（SMTP要求用户名和密码都通过64位编码后再发送，不接受明文），服务器分别回复235表示身份验证成功。

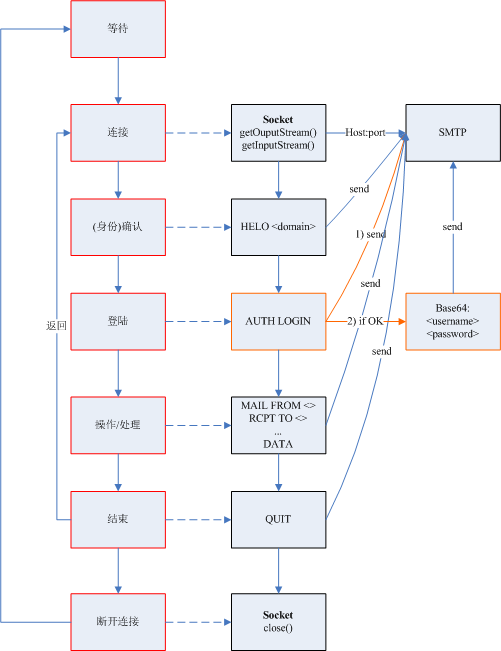
客户端分别先后向服务器发送MAIL FROM和RCPT TO命令，后面分别加上发件人的邮箱地址和收件人的邮箱地址，服务器分别回应250表示成功接受。

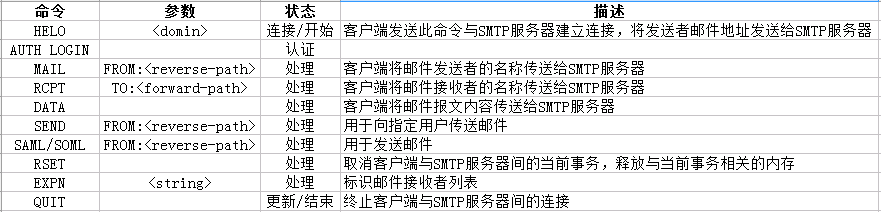
接下来客户端向服务器发送命令DATA，表示将要向服务器发送邮件正文，服务器回应354 End data with <CR><LF>.<CR><LF>表示开始邮件输入，以"."结束。

然后客户端将邮件内容发送给服务器（大小为378 bytes），服务器回应250表示接收成功。

邮件已成功发送到服务器，客户端向服务器发送命令QUIT，释放服务器连接，服务器返回221表示同意。双方断开连接，通信过程结束。

**2.SMTP服务器的实现**

（1）程序流程图：

（2）程序响应命令表：

（3）命令可能的服务器返回值：

500 格式错误，命令不可识别（此错误也包括命令行过长）

501 参数格式错误

502 命令不可实现

503 错误的命令序列

504 命令参数不可实现

211 系统状态或系统帮助响应

214 帮助信息

220 <domain> 服务就绪

221 <domain> 服务关闭传输信道

421 <domain> 服务未就绪，关闭传输信道（当必须关闭时，此应答可以作为对任何命令的响应）

250 要求的邮件操作完成

251 用户非本地，将转发向<forward-path>

450 要求的邮件操作未完成，邮箱不可用（例如，邮箱忙）

550 要求的邮件操作未完成，邮箱不可用（例如，邮箱未找到，或不可访问）

451 放弃要求的操作；处理过程中出错

551 用户非本地，请尝试<forward-path>

452 系统存储不足，要求的操作未执行

552 过量的存储分配，要求的操作未执行

553 邮箱名不可用，要求的操作未执行（例如邮箱格式错误）

354 开始邮件输入，以<CRLF>.<CRLF>结束

554 操作失败

（4）编程思路：

通过继承CAsyncSocket，重载其中的OnAccept，OnClose，OnReceive函数：

OnAccept：用来接收客户端请求建立连接；

OnReceive：实现对客户端的监听并对请求做出响应；

OnClose：关闭连接。

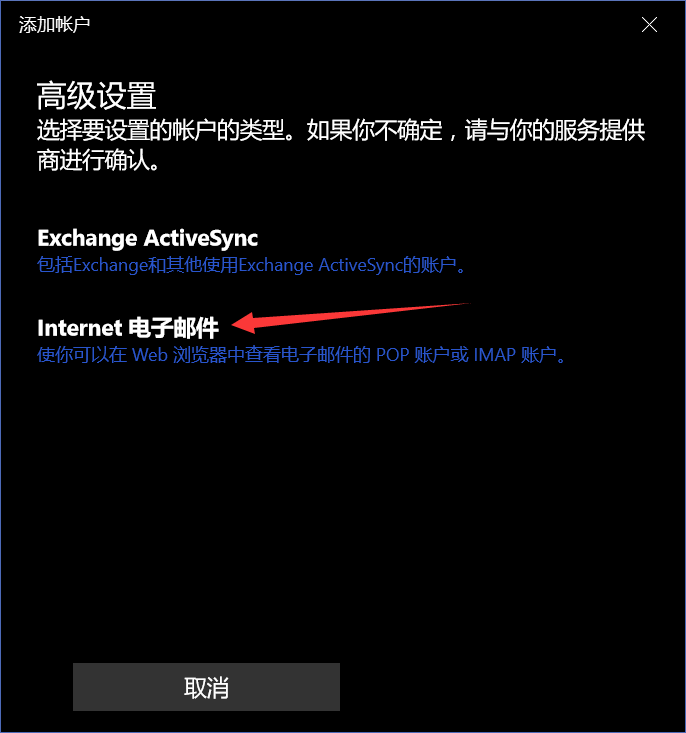
函数的实现在，SMTPsocket.cpp文件中。

其中，用户名密码需要用base64编码发送，邮件的内容以及附件也是base64编码，我的项目中的base.h文件负责处理这些编码问题。

在窗口初始化时就创建了一个socket用来连接客户端，所以我的程序界面上并没有任何按钮，窗口打开时socket就建立好了。

具体实现请参照代码。

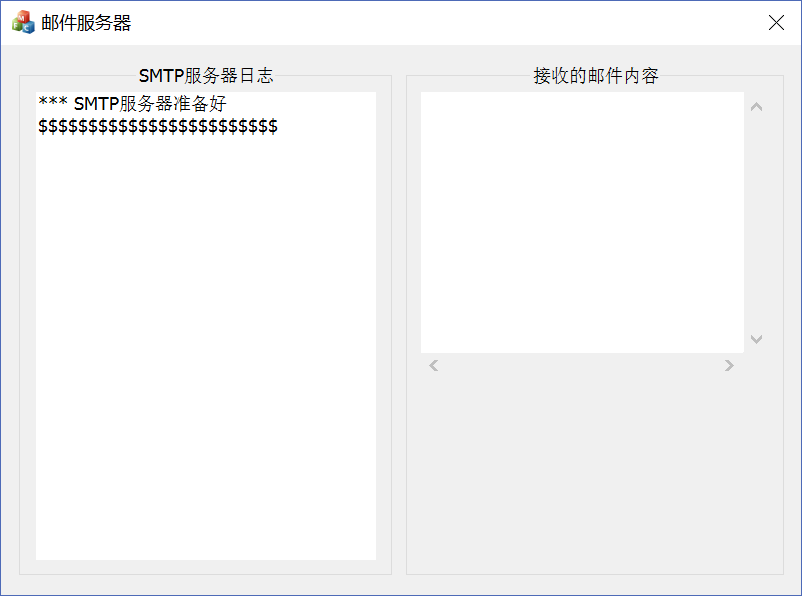
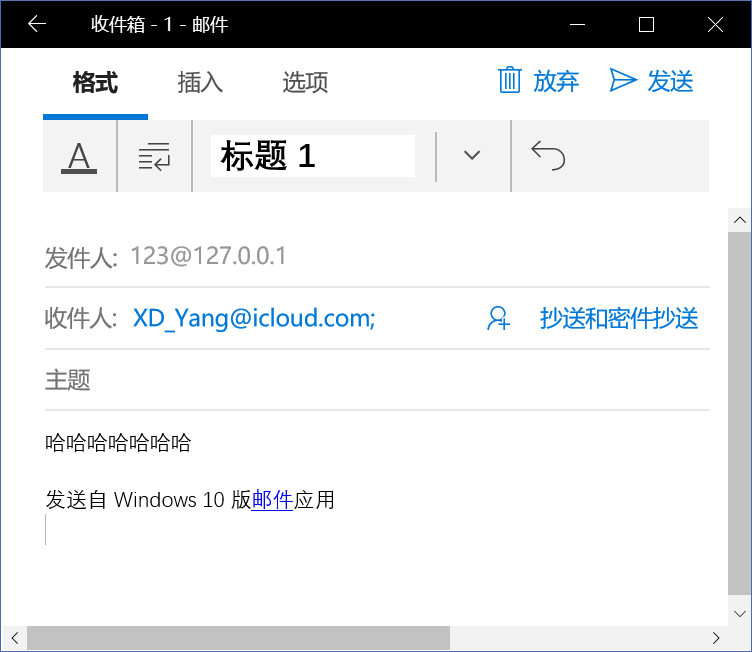
**五、实验效果**

 首先配置本地的SMTP的邮箱。在这里使用Outlook UWP客户端。

选择类型为Internet电子邮件，之后主要将电子邮件地址，传入电子邮件服务器，传出电子邮件服务器设为本地地址127.0.0.1，账户类型选择POP3，下面的身份验证以及SSL均取消。其他选项随意填。完成后登陆。



打开SMTP服务器。

用之前配置好的邮箱发送一封邮件：

SMTP服务器收到并记录交互过程：

再发一封带有图片附件的：

服务器接收，并显示图片：

左边日志上方保留了上次的交互过程。

至此，实验完成。