



哈爾濱工業大學 (深圳)
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

开课学期: 2022 春季
课程名称: 计算机网络
实验名称: 邮件客户端的设计与实现
学生班级: 1901105
学生学号: 190110509
学生姓名: 王铭
评阅教师:
报告成绩:

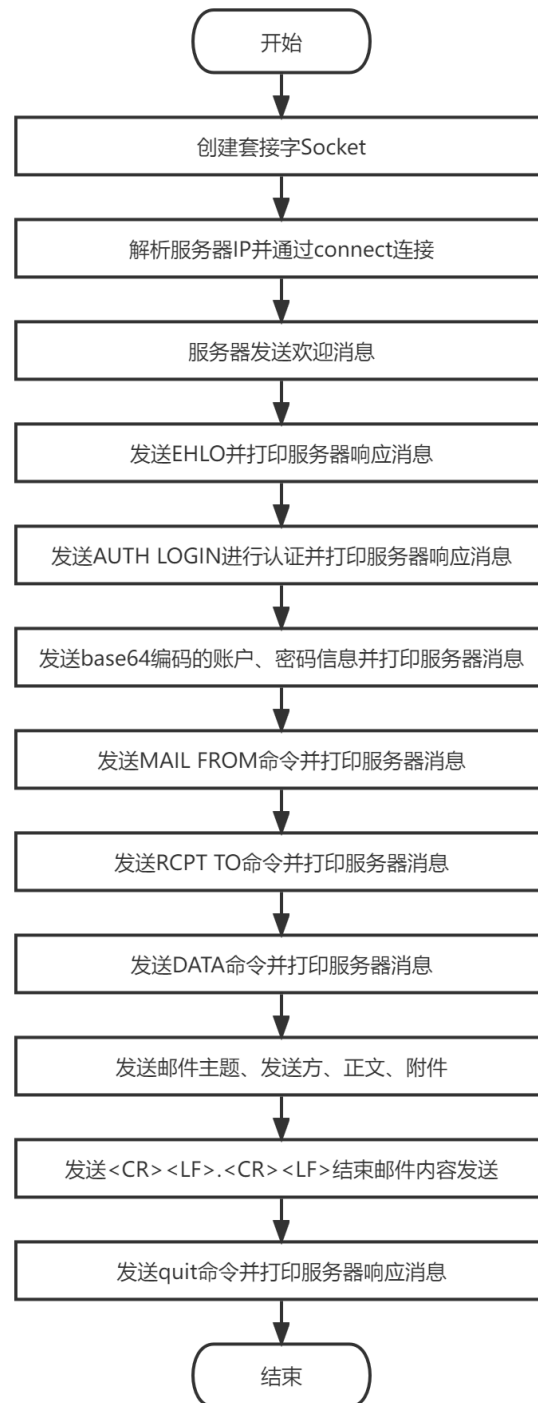
实验与创新实践教育中心制

2022 年 3 月

一、 实验详细设计

1. 邮件发送客户端详细设计

整体流程图如图所示：



定义打印从服务器端的返回消息函数 `printResponse`，通过 `socket` 文件描述符调用 `recv` 函数接收服务器端传回的消息。

```
void printResponse(int fd){
    int r_size;
    if ((r_size = recv(fd, buf, MAX_SIZE, 0)) == -1)
    {
        perror("recv");
        exit(EXIT_FAILURE);
    }
    buf[r_size] = '\0'; // Do not forget the null terminator
    printf("%s", buf);
}
```

图 1.1 打印服务器端返回的消息 printResponse 函数

首先，通过域名解析获取 QQ 邮箱服务器的 IP 地址，存于 dest_ip 中

```
// Get IP from domain name
if ((host = gethostbyname(host_name)) == NULL)
{
    perror("gethostbyname");
    exit(EXIT_FAILURE);
}

addr_list = (struct in_addr **) host->h_addr_list;
while (addr_list[i] != NULL)
    ++i;
strcpy(dest_ip, inet_ntoa(*addr_list[i-1]));
```

图 1.2 解析域名获得 IP 字符串 dest_ip

创建流式 Socket，使用 IPV4 协议，获得创建的 socket 描述符。与服务器建立连接时，调用 inet_addr 函数，将目的 IP 转换为二进制网络字节，端口号要进行大小端的转换，调用 connect 函数与服务器建立连接。调用 printResponse 函数打印服务器端的返回消息。

```
s_fd = socket(AF_INET, SOCK_STREAM, 0);
struct sockaddr_in servaddr;
servaddr.sin_family = AF_INET;
servaddr.sin_port = swap16(port);
struct in_addr sin_addr;
sin_addr.s_addr = inet_addr(dest_ip);
servaddr.sin_addr = sin_addr;
bzero(&(servaddr.sin_zero), 8);
if (connect(s_fd, &servaddr, sizeof(servaddr)) == -1){
    perror("connect error");
    exit(1);
}
// Print welcome message
printResponse(s_fd);
```

图 1.3 创建 Socket 并与 QQ 的 SMTP 服务器连接

用 EHLO 命令进行交互并调用 printResponse 打印服务器响应消息

```
// Send EHLO command and print server response
const char* EHLO = "EHLO qq.com\r\n"; // TODO: Enter EHLO command here
send(s_fd, EHLO, strlen(EHLO), 0);
printResponse(s_fd);
```

图 1.4 发送 EHLO 命令

用 AUTH 命令进行交互,其中输入的用户名 user 和密码 pass 都需要调用 encode_str 函数,用 base64 进行编码再进行发送,发送结束后,调用 printResponse 打印服务器响应消息,用 free 函数释放使用堆内存编码的字符串。

```
// TODO: Authentication. Server response should be print
const char* AUT = "AUTH login\r\n";
send(s_fd,AUT,strlen(AUT),0);
printResponse(s_fd);

char* user64 = encode_str(user);
char* pass64 = encode_str(pass);
send(s_fd,user64,strlen(user64),0);
send(s_fd,ctrf,strlen(ctrf),0);
printResponse(s_fd);
send(s_fd,pass64,strlen(pass64),0);
send(s_fd,ctrf,strlen(ctrf),0);
printResponse(s_fd);
free(user64);
free(pass64);
```

图 1.5 进行身份验证并发送 base64 编码的用户名和密码

使用 MAIL FROM 指令表明发送方,调用 printResponse 打印服务器响应消息。

```
// TODO: Send MAIL FROM command and print server response
const char *FROM = "MAIL FROM:<1604350283@qq.com>\r\n";
send(s_fd,FROM,strlen(FROM),0);
printResponse(s_fd);
```

图 1.6 发送 MAIL from 命令

使用 RCPT TO 指令表明接收方,调用 printResponse 打印服务器响应消息

```
char TO[50] = "RCPT TO:<";
char temp[] = ">\r\n";
strcpy(TO+9,receiver);
strcpy(TO+9+strlen(receiver),temp);
send(s_fd,TO,strlen(TO),0);
printResponse(s_fd);
```

图 1.7 发送 RCPT TO 命令

发送 DATA 命令开始发送邮件内容

```
// TODO: Send DATA command and print server response
const char *DATA = "DATA\r\n";
send(s_fd,DATA,strlen(DATA),0);
printResponse(s_fd);
```

图 1.8 发送 DATA 命令

首先用 subject 命令发送邮件主题,用 from 命令发送邮件发送方邮箱地址,由于要实现发送附件功能,故采用 MIME 的 multipart/mixed 类型,用#BOUNDARY#区分邮件不同部分的内容。

```

// Send some message data
const char *SUBJECT = "subject:";
send(s_fd, SUBJECT, strlen(SUBJECT), 0);
send(s_fd, subject, strlen(subject), 0);
send(s_fd, ctrf, strlen(ctrf), 0);

const char* f = "from:";
send(s_fd, f, strlen(f), 0);
send(s_fd, from, strlen(from), 0);
send(s_fd, ctrf, strlen(ctrf), 0);
const char* header = "MIME-Version:1.0\r\nContent-Type:multipart/mixed;boundary=#BOUNDARY#\r\n\r\n";
send(s_fd, header, strlen(header), 0);

```

图 1.9 发送邮件主题、发件人并构建 MIME 头

构造 MIME 头，用--#BOUNDARY#表示开始，正文部分采用 text/plain 类型，Content-Description 描述为邮件主体 body，若输入的 msg 不是文件，则直接发送从命令行输入的 msg 作为正文内容，若输入的 msg 是文件，则读取文件内容作为正文内容并发送。

```

if(msg){
    const char* cheader1 = "--#BOUNDARY#\r\nContent-Type:text/plain\r\nContent-Description:body\r\n\r\n";
    send(s_fd, cheader1, strlen(cheader1), 0);
    FILE *t = fopen(msg, "r+");
    if(t){
        send(s_fd, msg, strlen(msg), 0);
    }
    else{
        int nCount;
        while( (nCount = fread(buf, 1, MAX_SIZE, t)) > 0 ){
            send(s_fd, buf, nCount, 0);
        }
        fclose(t);
    }
}

```

图 1.10 发送邮件正文

若需要发送附件，则构造 MIME 头，用--#BOUDARY#表示开始，类型选择 application/octet/stream，name 采用传入的路径/文件名，编码说明用 base64 编码，内容描述为附件 attachment。

调用 encode_file 对附件进行 base64 编码，编码结束后用 send 进行发送

```

if(att_path){
    FILE *rawfile = fopen(att_path, "r+");
    if(!rawfile){
        perror("file error!\n");
        exit(1);
    }
    FILE *file = fopen("temp", "wt");
    encode_file(rawfile, file);
    const char* cheader2 = "\r\n--#BOUNDARY#\r\nContent-Type:application/octet-stream;name=";
    send(s_fd, cheader2, strlen(cheader2), 0);
    send(s_fd, att_path, strlen(att_path), 0);
    const char* cheader3 = "\r\nContent-Transfer-Encoding: base64\r\nContent-Description:attachment\r\n\r\n";
    send(s_fd, cheader3, strlen(cheader3), 0);
    fclose(file);
    fclose(rawfile);
    int nCount;
    file = fopen("temp", "r+");
    while( (nCount = fread(buf, 1, MAX_SIZE, file)) > 0 ){
        send(s_fd, buf, nCount, 0);
    }
    fclose(file);
}

```

图 1.11 发送附件

邮件内容以<CR><LF>.<CR><LF>结尾，调用 printResponse 打印服务器响应消息。

```

// TODO: Message ends with a single period
send(s_fd, end_msg, strlen(end_msg), 0);
printResponse(s_fd);

```

图 1.12 发送邮件内容结束标记

发送 quit 命令，邮件发送结束。

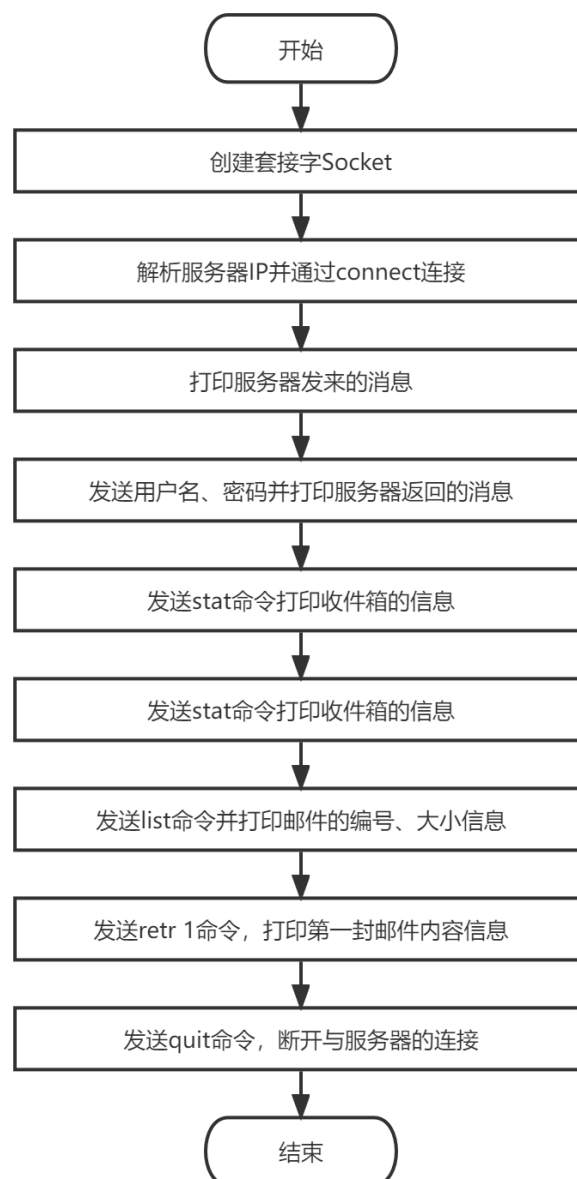
```
// TODO: Send QUIT command and print server response
const char *QUIT = "quit\r\n";
send(s_fd, QUIT, strlen(QUIT), 0);
printResponse(s_fd);

close(s_fd);
```

图 1.13 发送 quit 命令，断开连接

2. 邮件接收客户端详细设计

整体流程图如图所示：



与发送端类似，通过 `gethostbyname` 函数获取 pop 服务器的 IP 地址，存于 `dest_ip` 字符串中。

```
// Get IP from domain name
if ((host = gethostbyname(host_name)) == NULL)
{
    perror("gethostbyname");
    exit(EXIT_FAILURE);
}

addr_list = (struct in_addr **) host->h_addr_list;
while (addr_list[i] != NULL)
    ++i;
strcpy(dest_ip, inet_ntoa(*addr_list[i-1]));
```

图 2.1 解析域名获得 IP 字符串 `dest_ip`

与发送端类似，创建流式 Socket，使用 IPV4 协议，获得创建的 socket 描述符。与服务器建立连接时，调用 `inet_addr` 函数，将目的 IP 转换为二进制网络字节，端口号要进行大小端的转换，调用 `connect` 函数与服务器建立连接。调用 `printResponse` 函数打印服务器端的返回消息。

```
s_fd = socket(AF_INET, SOCK_STREAM, 0);
struct sockaddr_in servaddr;
servaddr.sin_family = AF_INET;
servaddr.sin_port = swap16(port);
struct in_addr sin_addr;
sin_addr.s_addr = inet_addr(dest_ip);
servaddr.sin_addr = sin_addr;
bzero(&(servaddr.sin_zero), 8);
if (connect(s_fd, &servaddr, sizeof(servaddr)) == -1) {
    perror("connect error");
    exit(1);
}
// Print welcome message
if ((r_size = recv(s_fd, buf, MAX_SIZE, 0)) == -1)
{
    perror("recv");
    exit(EXIT_FAILURE);
}
buf[r_size] = '\0'; // Do not forget the null terminator
printf("%s", buf);
```

图 2.2 创建 Socket 并与 QQ 的 POP 服务器连接

发送用户名、密码，不用进行 base64 编码。

```
// TODO: Send user and password and print server response
send(s_fd, user, strlen(user), 0);
printResponse(s_fd);
send(s_fd, pass, strlen(pass), 0);
printResponse(s_fd);
```

图 2.3 发送账户、密码进行登录

发送 list 命令查看邮件列表和大小信息，调用 printResponse 打印服务器端响应消息。

```
// TODO: Send LIST command and print server response
const char *LIST = "list\r\n";
send(s_fd, LIST, strlen(LIST), 0);
printResponse(s_fd);
```

图 2.4 发送 list 命令获取邮件个数、大小

发送 retr 1 命令，调用 printResponse 获取第一封邮件的内容

```
// TODO: Retrieve the first mail and print its content
const char *RETR = "retr 1\r\n";
send(s_fd, RETR, strlen(RETR), 0);
printResponse(s_fd);
```

图 2.5 发送 retr 1 命令获取第一封邮件内容

发送 quit 命令，断开连接。

```
// TODO: Send QUIT command and print server response
const char *QUIT = "quit\r\n";
send(s_fd, QUIT, strlen(QUIT), 0);
printResponse(s_fd);
close(s_fd);
```

图 2.6 发送 quit 命令断开连接

二、实验结果截图及分析

1. 邮件发送客户端实验结果及分析

如图所示，调用 send 函数发送邮件，主题为 subject，消息为 test.txt 里的内容（test attachment function），附件为 test.txt。程序打印出与服务器交互时的响应消息，可以看到，Authentication successful 表示认证成功，Bye 表示发送邮件后成功断开。

```
ming3@ming3-virtual-machine:~/mailab$ ./send 1978984675@qq.com -s "subject" -m "test.txt" -a "test.txt"
220 newxmesmtplgicsvrsza8.qq.com XMail Esmtp QQ Mail Server.
250-newxmesmtplgicsvrsza8.qq.com
250-PIPELINING
250-SIZE 73400320
250-STARTTLS
250-AUTH LOGIN PLAIN XOAUTH XOAUTH2
250-AUTH=LOGIN
250-MAILCOMPRESS
250 8BITMIME
334 VXNlcm5hbnU6
334 UGFzc3dvcmQ6
235 Authentication successful
250 OK
250 OK
354 End data with <CR><LF>.<CR><LF>.
250 OK: queued as.
221 Bye.
```

图 3.1 打印发送邮件时服务器的响应信息

如图 3.2 所示，利用 wireshark 抓包可以看到邮件发送客户端与 SMTP 服务器的交互情况。发送 EHLO 命令时，可以看到 SMTP 服务器的 IP 地址为 183.47.101.192，端口号为 25。其中发送内容如图 3.3 所示，发送的主题为 subject，由 1604350283@qq.com 发送，采用的 MIME 版本号为 1.0，类型为 multipart/mixed，以 #BOUNDARY# 为界限标志，正文部分采用 text/plain，内容为 test attachment function。附件采用 application/octet-stream，编码

为 base64，内容为 base64 编码后的 test attachment function。

No.	Time	Source	Destination	Protocol	Length	Info
53	10.463966	183.47.101.192	10.249.71.202	SMTP	117 S:	220 newmesmtplgicvsrzs8.qq.com XMail Esmtpp QQ Mail Server.
54	10.464912	10.249.71.202	183.47.101.192	SMTP	67 C:	EHL0 qq.com
56	10.484189	183.47.101.192	10.249.71.202	SMTP	223 S:	250-newmesmtplgicvsrzs8.qq.com PIPELINING SIZE 73400320 STARTTLS AUTH LOGIN PLAIN XOAUTH XO...
57	10.484802	10.249.71.202	183.47.101.192	SMTP	66 C:	AUTH login
58	10.505003	183.47.101.192	10.249.71.202	SMTP	72 S:	334 VX0lcn5hbmU6
59	10.505083	10.249.71.202	183.47.101.192	SMTP	79 C:	User: MTYwMDM1MDIAP08xc55jb20=
61	10.563773	10.249.71.202	183.47.101.192	SMTP	56 C:	
63	10.582518	183.47.101.192	10.249.71.202	SMTP	72 S:	334 UGFzc3dvcmQ6
64	10.583020	10.249.71.202	183.47.101.192	SMTP	79 C:	Pass: b2tjd2xxaG1cnhzYmFmZQ==
66	10.640185	10.249.71.202	183.47.101.192	SMTP	56 C:	
68	10.859068	183.47.101.192	10.249.71.202	SMTP	85 S:	235 Authentication successful
69	10.859741	10.249.71.202	183.47.101.192	SMTP	85 C:	MAIL FROM: c1604350283@qq.com>
71	10.921243	183.47.101.192	10.249.71.202	SMTP	62 S:	250 OK
72	10.921954	10.249.71.202	183.47.101.192	SMTP	83 C:	RCPT TO: c1978984675@qq.com>
74	11.001576	183.47.101.192	10.249.71.202	SMTP	62 S:	250 OK

> Frame 54: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface \Device\NPF_{918E7C2F-BD42-49B9-A41F-D5AB77F2AB08}, id 0
> Ethernet II, Src: LCF0HeFe17:cf:d4 (e8:6a:64:17:cf:d4), Dst: IETF-VRRP-VRID_03 (00:00:5e:00:01:03)
> Internet Protocol Version 4, Src: 10.249.71.202, Dst: 183.47.101.192
> Transmission Control Protocol, Src Port: 64131, Dst Port: 25, Seq: 1, Ack: 64, Len: 13
> Simple Mail Transfer Protocol

0000 00 00 5e 00 01 03 e8 6a 64 17 cf d4 08 00 45 00 ...d....E-
0010 00 35 39 70 40 00 00 00 00 0a f9 47 ca b7 2f 59 9b @ ---G-/
0020 65 c0 fa 83 00 19 21 6e 6f da 84 12 e3 15 50 18 e.....In o....P-
0030 02 05 6f da 00 00 45 48 4c 4f 20 71 71 2e 63 6f ...o...EH LO qq.co
0040 6d 0d 0a m...

图 3.2 邮件发送客户端抓包信息

No.	Time	Source	Destination	Protocol	Length	Info
72	10.921954	10.249.71.202	183.47.101.192	SMTP	83 C:	RCPT TO: c1978984675@qq.com>
74	11.001576	183.47.101.192	10.249.71.202	SMTP	62 S:	250 OK
75	11.003096	10.249.71.202	183.47.101.192	SMTP	60 C:	DATA
77	11.023005	183.47.101.192	10.249.71.202	SMTP	92 S:	354 End data with <CR><LF>.<CR><LF>.
80	11.079804	10.249.71.202	183.47.101.192	SMTP/I...	424	subject: subject, from: 1604350283@qq.com, (text/plain)
82	11.309221	183.47.101.192	10.249.71.202	SMTP	74 S:	250 OK: queued as.
83	11.312240	10.249.71.202	183.47.101.192	SMTP	60 C:	quit
85	11.332263	183.47.101.192	10.249.71.202	SMTP	64 S:	221 Bye.

Internet Message Format
Subject: subject
> From: 1604350283@qq.com, 1 item
MIME-Version: 1.0
> Content-Type: multipart/mixed;boundary=#BOUNDARY#
> MIME multipart Media Encapsulation, Type: multipart/mixed, Boundary: "#BOUNDAR"
[Type: multipart/mixed]
First boundary: --#BOUNDARY#\r\n
Encapsulated multipart part: (text/plain)
Content-Type: text/plain\r\n
Content-Description: body\r\n\r\n
Line-based text data: text/plain (1 lines)
test attachment function

0000 73 63 72 69 70 74 69 6f 6e 3a 62 6f 64 79 0d 0a scrip tio n:body--
0010 0d 0a 74 65 73 74 20 61 74 74 61 63 68 6d 65 6e .test a ttachmen
0020 74 20 66 75 6e 63 74 69 6f 6e 0d 0a 2d 2d 23 42 t functi on---#B
0030 4f 55 4e 44 41 52 59 23 0d 0a 43 6f 6e 74 65 6e OUNDARY# --Conen
0040 74 2d 54 79 70 65 3a 61 70 70 6c 69 63 61 74 69 t-Type: a plicati
0050 6f 6e 2f 6f 63 74 65 74 2d 73 74 72 65 61 6d 3b on/octet -stream;
0060 6e 61 6d 65 3d 74 65 73 74 2e 74 78 74 0d 0a 43 name=tes t.txt--C
0070 6f 6e 74 65 6e 74 2d 54 72 61 6e 73 66 65 72 2d ontent-T ransfer-
0080 45 6e 63 6f 64 69 6e 67 3a 20 62 61 73 65 36 34 Encoding : base64
0090 0d 0a 43 6f 6e 74 65 6e 74 2d 44 65 73 63 72 69 .Content -Descri
0100 70 74 69 6f 6e 3a 61 74 74 61 63 68 6d 65 6e 74 ption:at tachment
0110 0d 0a 0d 0a 64 47 56 7a 64 43 42 68 64 48 52 68 ...dGVZ dCBhdHRh
0120 59 32 68 74 5a 57 35 30 49 47 5a 31 62 6d 4e 30 YZhtZW50 IGZ1bmN0
0130 61 57 39 75 0a aW9u.

图 3.3 邮件发送客户端发送内容信息

打开 QQ 邮箱可以查看邮件内容以及预览附件，如图 3.4 和图 3.5 所示



图 3.4 QQ 邮箱查看发送的邮件

test.txt

test attachment function

图 3.5 附件预览

2. 邮件接收客户端实验结果及分析

交互截图如图 3.6 所示

```
ming3@ming3-virtual-machine:~/maillab$ ./recv
+OK XMail POP3 Server v1.0 Service Ready(XMail v1.0)
+OK
+OK
+OK 4 25694
+OK
1 1629
2 11217
3 1609
4 11239
.
+OK 1629
Received: from qq.com ([223.73.111.169])
    by newxmesmtplogicsvrsza.qq.com (NewEsmtp) with SMTP
    id BB4B602A; Tue, 10 May 2022 09:46:52 +0800
X-QQ-mid: xmsmtp1652147212ty18g1s14
Message-ID: <tencent_0AE7D5A2F24E772002624C5C852F32075005@qq.com>
X-QQ-XMAILINFO: 0UrMHMu9XZHVdoytmJmA/cpHarR3W09shNYsrwxJ+OgE3Wfc2B5MNoY8fOZEoQ
8KIn9vp5spoRiAZTMiBnAR0W7yig0ZvXT2JNOU/+rdrnRgOum6w0bdGISiZrHilSO5IuIAEvQG2is
Pk4UPWi/RrYPnZo4KrFhyZ2Eyi4r5sDst8mAkxTtfPvLC/x9Z/JqDq176ZYBxq62+xF3bq3Q2+A
UeVINeyc2rRyksu8/PHS9JGPlovFUaDqoDhdnpclTb6la9yzVE711qMHGQU7GnH8QFzF4H3048Wf
SqzHdvtnf70XWxfWEbsZJfapIyTdHF2baDxmdb/E8uuManB6ipcwGPhCnXfFGuillIFACDx+u6mpA
8QG1pz+9wc4D3vsZpNRoHVLUJ4bVbDfoambMS/nbAhhgEJbdBNa2ajHy9m1yHa9YTJ/coHmdHsAU
yUVvEMMnB0Gb7e/LEpOFRea5vX6Qyi5z1BD0002ep7Hlq/ggLRioqpJE6zjLQgfJ9U17qmQ70uK
KL9byVJ6PeZ8ld9QbRDDzXf2zcmNkecsfq1xNmou9ihxD5nsXlRaP/j9/X0bm1Sff+wtu35gs8Yx
fOu9sip8KJw/Yy2W0RC255yxDQ+A+upqMn6FsCE8LDMaFaTG3o4zb0BYshjKZYw2bLEfC387xxje
naWfbbxAU994f/mj7PLubq9ZpZwY+QF2JXgqICYjCxyVe7m/9eW0rloSgwi9FLpqWOKL6eMcvNZe
qKK+iRwtNuwCorqYRQtz3UMq+k+IEQ3a/TpXqqjhiEuK8acV/lJynnleEGW+xfVSCzbBg1UiUXC1
15SxCaSoTjE7uiS4B04YNGm9qQoR0ahdLDgz7bkNwWGA+RRiVPSnbZGO+HJJ0ynbBdaNabhKwn2T
OMaQT/GGP6TNvm3/5F/0wKdpb+aacQuNOUu8oPbt1b
subject:subject
from:1604350283@qq.com
MIME-Version:1.0
Content-Type:multipart/mixed;boundary=#BOUNDARY#

--#BOUNDARY#
Content-Type:text/plain
Content-Description:body

test attachment function
--#BOUNDARY#
Content-Type:application/octet-stream;name=test.txt
Content-Transfer-Encoding: base64
Content-Description:attachment

dGVzdCBhdHRhY2htZW50IGZ1bmN0aw9u

.
+OK Bye
```

图 3.6-1 邮件接受客户端收到服务器的响应消息

```
KL9byVJ6PeZ8ld9QbRDDzXf2zcmNkecsfq1xNmou9ihxD5nsXlRaP/j9/X0bm1Sff+wtu35gs8Yx
fOu9sip8KJw/Yy2W0RC255yxDQ+A+upqMn6FsCE8LDMaFaTG3o4zb0BYshjKZYw2bLEfC387xxje
naWfbbxAU994f/mj7PLubq9ZpZwY+QF2JXgqICYjCxyVe7m/9eW0rloSgwi9FLpqWOKL6eMcvNZe
qKK+iRwtNuwCorqYRQtz3UMq+k+IEQ3a/TpXqqjhiEuK8acV/lJynnleEGW+xfVSCzbBg1UiUXC1
15SxCaSoTjE7uiS4B04YNGm9qQoR0ahdLDgz7bkNwWGA+RRiVPSnbZGO+HJJ0ynbBdaNabhKwn2T
OMaQT/GGP6TNvm3/5F/0wKdpb+aacQuNOUu8oPbt1b
subject:subject
from:1604350283@qq.com
MIME-Version:1.0
Content-Type:multipart/mixed;boundary=#BOUNDARY#

--#BOUNDARY#
Content-Type:text/plain
Content-Description:body

test attachment function
--#BOUNDARY#
Content-Type:application/octet-stream;name=test.txt
Content-Transfer-Encoding: base64
Content-Description:attachment

dGVzdCBhdHRhY2htZW50IGZ1bmN0aw9u

.
+OK Bye
```

图 3.6-2 邮件接受客户端收到服务器的响应消息

如图 3.7 所示,使用 wireshark 抓包可以看到邮件接收客户端与 POP 服务器的交互流程。可以看到 POP 服务器的 IP 地址为 183.47.101.192,端口号为 110。如图 3.8 所示,可以看到邮件接收客户端收到的第一封邮件的内容。

pop						
No.	Time	Source	Destination	Protocol	Length	Info
63	8.638362	183.47.101.192	10.249.71.202	POP	108	S: +OK XMail POP3 Server v1.0 Service Ready(XMail v1.0)
63	8.639137	10.249.71.202	183.47.101.192	POP	78	C: user 1978984675@qq.com
65	8.678399	183.47.101.192	10.249.71.202	POP	60	S: +OK
66	8.679072	10.249.71.202	183.47.101.192	POP	77	C: pass zbihashjciagchib
68	9.083618	183.47.101.192	10.249.71.202	POP	60	S: +OK
69	9.085017	10.249.71.202	183.47.101.192	POP	60	C: stat
71	9.111054	183.47.101.192	10.249.71.202	POP	67	S: +OK 4 25694
72	9.111924	10.249.71.202	183.47.101.192	POP	60	C: list
74	9.207186	183.47.101.192	10.249.71.202	POP	96	S: +OK
75	9.207771	10.249.71.202	183.47.101.192	POP	62	C: retr 1
77	9.388855	183.47.101.192	10.249.71.202	POP	1478	S: +OK 1629
78	9.388855	183.47.101.192	10.249.71.202	POP	274	S: DATA fragment, 220 bytes
80	9.389491	10.249.71.202	183.47.101.192	POP	60	C: quit
82	9.494780	183.47.101.192	10.249.71.202	POP	63	S: +OK Bye

> Frame 63: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface \Device\NPF_{918E7C2F-BD42-49B9-A41F-D5AB7FF2AB0B}, id 0
> Ethernet II, Src: LCfCHeFe_17:cf:d4 (e8:6a:64:17:cf:d4), Dst: IETF-VRRP-VRID_03 (00:00:5e:00:01:03)
> Internet Protocol Version 4, Src: 10.249.71.202, Dst: 183.47.101.192
> Transmission Control Protocol, Src Port: 64493, Dst Port: 110, Seq: 1, Ack: 55, Len: 24
> Post Office Protocol

0000 00 00 5e 00 01 03 e8 6a 64 17 cf d4 08 00 45 00 --^-----j d-----E-
0010 00 40 39 80 40 00 00 06 00 00 0a f9 47 ca b7 2f -@9-@-@- -G-/-
0020 65 c0 fb ed 00 6e 5d e8 21 6b dc eb cc 0e 50 18 e-----n] lk---P-
0030 02 05 6f e5 00 00 75 73 65 72 20 31 39 37 38 39 --o-----us er 19789
0040 38 34 36 37 35 40 71 71 2e 63 6f 6d 0d 0a 84675@qq .com--

图 3.7 邮件接受客户端抓包信息

pop						
No.	Time	Source	Destination	Protocol	Length	Info
62	8.638362	183.47.101.192	10.249.71.202	POP	108	S: +OK XMail POP3 Server v1.0 Service Ready(XMail v1.0)
63	8.639137	10.249.71.202	183.47.101.192	POP	78	C: user 1978984675@qq.com
65	8.678399	183.47.101.192	10.249.71.202	POP	60	S: +OK
66	8.679072	10.249.71.202	183.47.101.192	POP	77	C: pass zbihashjciagchib
68	9.083618	183.47.101.192	10.249.71.202	POP	60	S: +OK
69	9.085017	10.249.71.202	183.47.101.192	POP	60	C: stat
71	9.111054	183.47.101.192	10.249.71.202	POP	67	S: +OK 4 25694
72	9.111924	10.249.71.202	183.47.101.192	POP	60	C: list
74	9.207186	183.47.101.192	10.249.71.202	POP	96	S: +OK
75	9.207771	10.249.71.202	183.47.101.192	POP	62	C: retr 1
77	9.388855	183.47.101.192	10.249.71.202	POP	1478	S: +OK 1629
78	9.388855	183.47.101.192	10.249.71.202	POP	274	S: DATA fragment, 220 bytes
80	9.389491	10.249.71.202	183.47.101.192	POP	60	C: quit
82	9.494780	183.47.101.192	10.249.71.202	POP	63	S: +OK Bye

> Frame 78: 274 bytes on wire (2192 bits), 274 bytes captured (2192 bits) on interface \Device\NPF_{918E7C2F-BD42-49B9-A41F-D5AB7FF2AB0B}, id 0
> Ethernet II, Src: IETF-VRRP-VRID_03 (00:00:5e:00:01:03), Dst: LCfCHeFe_17:cf:d4 (e8:6a:64:17:cf:d4)
> Internet Protocol Version 4, Src: 183.47.101.192, Dst: 10.249.71.202
> Transmission Control Protocol, Src Port: 110, Dst Port: 64493, Seq: 1544, Ack: 68, Len: 220
> Post Office Protocol

0030 00 e5 95 5f 00 00 44 65 73 63 72 69 70 74 69 6f ... --de scriptio
0040 6e 3a 62 6f 64 79 0d 0a 0d 0a 74 65 73 74 20 61 n:body...test a
0050 74 74 61 63 68 6d 65 6e 74 20 66 75 6e 63 74 69 ttachmen t functi
0060 6f 6e 0d 0a 2d 2d 23 42 4f 55 4e 44 41 52 59 23 on---#B OUNDARY#
0070 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 61 --Conten t-type:a
0080 70 70 6c 69 63 61 74 69 6f 6e 2f 6f 63 74 65 74 pplicati on/octet
0090 2d 73 74 72 65 61 6d 3b 6e 61 6d 65 3d 74 65 73 -stream; name=tes
00a0 74 2e 74 78 74 0d 0a 43 6f 6e 74 65 6e 74 2d 54 t.txt--C ontent-T
00b0 72 61 6e 73 66 65 72 2d 45 6e 63 6f 64 69 6e 67 ransfer- Encoding
00c0 3a 20 62 61 73 65 36 34 0d 0a 43 6f 6e 74 65 6e : base64 --Conten
00d0 74 2d 44 65 73 63 72 69 70 74 69 6f 6e 3a 61 74 t-Descri ption:at
00e0 74 61 63 68 6d 65 6e 74 0d 0a 0d 0a 64 47 56 7a tachmentdGV2
00f0 64 43 42 68 64 48 52 68 59 32 68 74 5a 57 35 30 dCBhdHRh Y2htZW50
0100 49 47 5a 31 62 6d 4e 30 61 57 39 75 0a 0d 0a 2e IGZ1bmN0 aW9u....
0110 0d 0a

图 3.8 邮件发送客户端发送内容信息

三、 实验中遇到的问题及解决方法

使用 multipart/mixed 格式发送正文时, 若 MIME 头最后只有一个<CR><LF>后直接发送正文的话, 在接受邮件时, 会吞掉正文消息。发送时多添加一个<CR><LF>即可解决。

四、实验收获和建议

在配置验证实验中，用思科的仿真功能，我较好的理解了通过交互机、路由器进行网络传送、接受数据包的流程，直观地理解了路由转发功能和 NAT 技术的作用。通过实现了一套简单的协议栈，我更好的掌握了理论课上抽象的不同层次的协议内容，而 **Socket** 编程实验则让我更好地理解网络应用的典型例子，邮件接受、发送客户端的实现，体验了 **SMTP** 和 **POP3** 协议的使用。

在协议栈的实验过程中，感觉测试程序不太完善，有一些盲区测试不到，虽然实验结果正确，但很多细节并未考虑到。**Socket** 编程实验较流水化，主要是发送命令即可，涉及知识点较少，希望后面能完善。