

# 华东师范大学数据科学与工程学院实验报告

课程名称:计算机网络与编程	年级:22级	上机实践成绩:
指导教师:张召	姓名:郭夏辉	学号:10211900416
上机实践名称:HTTP、SMTP、POP3协议分析	上机实践日期:2023年4月7日	上机实践编号:No.06
组号:1-416	上机实践时间:2023年4月7日	

## 一、实验目的

- 熟悉HTTP协议的工作原理
- 了解HTTP协议在实际网络中的运行过程
- 熟悉SMTP和POP3协议的工作原理
- 了解SMTP和POP3协议在实际网络中的运行过程

## 二、实验任务

- 通过Wireshark分析HTTP协议
- 通过Wireshark分析SMTP和POP3协议

## 三、实验环境

- IntelliJ IDEA 2022.3.2
- JDK 19

## 四、实验过程

### task1

利用Wireshark抓取一条HTTP请求网络包，分析HTTP请求网络包的组成（要求根据报文结构正确标识每个部分），请将实验结果附在实验报告中。

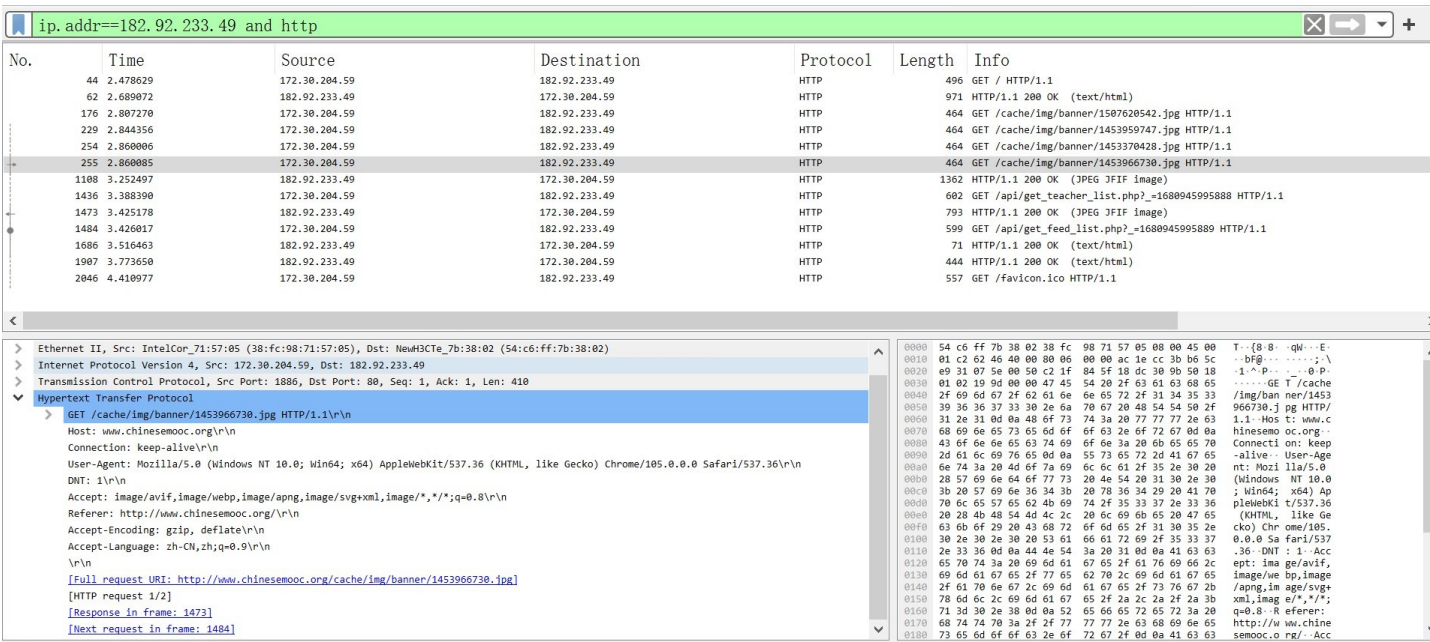
首先，我访问了网址: <http://www.chinesemooc.org>

然后通过ping看到了华文慕课的ip地址: 182.92.233.49

```
C:\Users\tom>ping chineseMOOC.org

正在 Ping chineseMOOC.org [182.92.233.49] 具有 32 字节的数据:
来自 182.92.233.49 的回复: 字节=32 时间=35ms TTL=45
来自 182.92.233.49 的回复: 字节=32 时间=30ms TTL=45
来自 182.92.233.49 的回复: 字节=32 时间=33ms TTL=45
来自 182.92.233.49 的回复: 字节=32 时间=95ms TTL=45
```

接着我打开Wireshark抓取相关的HTTP请求网络包如下图所示。注意在Wireshark之中HTTP请求报文会在左侧的标号处标一个向右的箭头，而与之对应的响应报文会标注一个向左的箭头。



HTTP的请求报文格式如下：

HTTP请求报文分析



图1 HTTP请求报文

然后我具体地对这个报文进行分析：

> Ethernet II, Src: IntelCor\_71:57:05 (38:fc:98:71:57:05), Dst: NewH3CTe\_7b:38:02 (54:c6:ff:7b:38:02)

> Internet Protocol Version 4, Src: 172.30.204.59, Dst: 182.92.233.49

> Transmission Control Protocol, Src Port: 1886, Dst Port: 80, Seq: 1, Ack: 1, Len: 410

> Hypertext Transfer Protocol

> GET /cache/img/banner/1453966730.jpg HTTP/1.1\r\n

Host: www.chinesemooc.org\r\n

Connection: keep-alive\r\n

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36\r\n

DNT: 1\r\n

Accept: image/avif,image/webp,image/apng,image/svg+xml,image/\*,\*/\*;q=0.8\r\n

Referer: http://www.chinesemooc.org/\r\n

Accept-Encoding: gzip, deflate\r\n

Accept-Language: zh-CN,zh;q=0.9\r\n

[Full request URI: http://www.chinesemooc.org/cache/img/banner/1453966730.jpg]

[HTTP request 1/2]

[Response in frame: 1473]

[Next request in frame: 1484]

请求行

请求方法(空格)URL(空格)协议版本(回车)(换行)

请求头

请求正文

task2

利用Wireshark找到上述请求网络包相对应的HTTP响应网络包，然后对比分析两个网络包的组成，请在实验报告中说明两者之间的区别。

通过Task1中的操作，我只需找到Wireshark中与那个请求报文相对应的箭头即可，这便是相对应的响应报文。

ip.addr==182.92.233.49 and http

No.	Time	Source	Destination	Protocol	Length	Info
44	2.478629	172.30.204.59	182.92.233.49	HTTP	496	GET / HTTP/1.1
62	2.689072	182.92.233.49	172.30.204.59	HTTP	971	HTTP/1.1 200 OK (text/html)
176	2.887270	172.30.204.59	182.92.233.49	HTTP	464	GET /cache/img/banner/1507620542.jpg HTTP/1.1
229	2.844356	172.30.204.59	182.92.233.49	HTTP	464	GET /cache/img/banner/1453959747.jpg HTTP/1.1
254	2.860006	172.30.204.59	182.92.233.49	HTTP	464	GET /cache/img/banner/1453370428.jpg HTTP/1.1
255	2.860005	172.30.204.59	182.92.233.49	HTTP	464	GET /cache/img/banner/1453966730.jpg HTTP/1.1
1108	3.252497	182.92.233.49	172.30.204.59	HTTP	1362	HTTP/1.1 200 OK (JPEG JFIF image)
1436	3.388390	172.30.204.59	182.92.233.49	HTTP	602	GET /api/get_teacher_list.php?_1680945995888 HTTP/1.1
1473	3.425178	182.92.233.49	172.30.204.59	HTTP	793	HTTP/1.1 200 OK (JPEG JFIF image)
1484	3.426017	172.30.204.59	182.92.233.49	HTTP	509	GET /api/get_feed_list.php?_1680945995889 HTTP/1.1
1686	3.516463	182.92.233.49	172.30.204.59	HTTP	71	HTTP/1.1 200 OK (text/html)
1907	3.773650	182.92.233.49	172.30.204.59	HTTP	444	HTTP/1.1 200 OK (text/html)

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Server: nginx\r\n

Date: Sat, 08 Apr 2023 09:13:41 GMT\r\n

Content-Type: image/jpeg\r\n

Content-Length: 206359\r\n

Last-Modified: Fri, 19 Jan 2018 10:17:58 GMT\r\n

Connection: keep-alive\r\n

Etag: "5a61c5d6-324e2"\r\n

Expires: Mon, 08 May 2023 09:13:41 GMT\r\n

Cache-Control: max-age=2592000\r\n

Accept-Ranges: bytes\r\n

\r\n

[HTTP response 1/2]

[Time since request: 0.565893000 seconds]

[Request in frame: 255]

[Next request in frame: 1484]

[Next response in frame: 1686]

[Request URI: http://www.chinesemooc.org/cache/img/banner/1453966730.jpg]

Frame (793 bytes) Reassembled TCP (206359 bytes)

HTTP的响应报文格式如下：

HTTP响应报文分析



图2 HTTP响应报文

然后我具体地对这个报文进行分析:

▼ Hypertext Transfer Protocol

> HTTP/1.1 200 OK\r\n

Server: nginx\r\n

Date: Sat, 08 Apr 2023 09:13:41 GMT\r\n

Content-Type: image/jpeg\r\n

Content-Length: 206050\r\n

Last-Modified: Fri, 19 Jan 2018 10:17:58 GMT\r\n

Connection: keep-alive\r\n

ETag: "5a61c5d6-324e2"\r\n

Expires: Mon, 08 May 2023 09:13:41 GMT\r\n

Cache-Control: max-age=2592000\r\n

Accept-Ranges: bytes\r\n

\r\n

[HTTP response 1/2]

[Time since request: 0.565093000 seconds]

[Request in frame: 255]

[Next request in frame: 1484]

[Next response in frame: 1686]

[Request URI: http://www.chinesemooc.org/cache/img/banner/1453966730.jpg]

响应头部

响应正文

状态行 协议版本(空格)状态码(空格)状态码描述(回车)(换行)

头部字段名:值(回车)(换行)

可以看到这个和请求报文有很多的类似之处，响应头部和请求头部的结构类似，都是字段名:值(回车)(换行);它们的正文部分也是类似的，最明显的差异在于首行不同:

请求报文的首行是请求行，结构是 请求方法(空格)URL(空格)协议版本(回车)(换行)

但是响应报文的首行是状态行，结构是 协议版本(空格)状态码(空格)状态描述(回车)(换行)

task3

学习了解GET和POST方法，请在实验报告中分析对比GET和POST方法的请求报文，以及GET和POST方法的和响应报文之间的区别。

最首要的任务当然还是先了解GET和POST方法的请求和响应报文之基本格式:



GET	空格	/	空格	HTTP/1.1	\r	\n
Accept	:	text/html,application/xhtml+xml,application/xml			\r	\n
...						
Connection	:	keep-alive			\r	\n
\r				\n		
Full request URI: http://10.1.1.33:8080/						

图3 GET方法的HTTP请求报文

HTTP/1.1	空格	200	空格	OK	\r	\n
Content-Type	:	text/html		\r		\n
...						
Content-Encoding	:	gzip		\r		\n
\r			\n			
省略						

图 4 GET方法的HTTP响应报文

POST	空格	/hfs2_3b287/	空格	HTTP/1.1	\r	\n
Accept	:	text/html,application/xhtml+xml,application/xml		\r		\n
...						
Content-Length	:	367		\r		\n
\r			\n			
忽略						

图 5 POST方法的HTTP请求报文

HTTP/1.1	空格	200	空格	OK	\r	\n
Server	:	HFS 2.3 beta		\r		\n
...						
Content-Encoding	:	gzip		\r		\n
\r			\n			
省略						

图 6 POST方法的HTTP响应报文

可以看到GET 和 POST 的请求报文结构上类似，但差异在于 GET 请求报文一般没有请求体，数据放在 URL 中；而 POST 请求报文有请求体，要传递的数据一般放在这里。同时GET一般用于从服务器获取资源，但是POST 表示向指定服务器提交数据。GET方法传递的数据量较小，最大不超过2KB（因为受URL长度限制）而Post方法传递的数据量较大，一般不受限制（大小取决于服务器的处理能力）。

还有就是GET会产生一个TCP数据包，浏览器会把Header和Data一并发送出去，服务器响应200（OK），并回传相应的数据。POST方法会产生两个TCP数据包，浏览器会先将Header发送出去，服务器响应100（Continue）后，浏览器再发送数据，服务器响应200（OK），并回传相应的数据。

为了具体的分析，我抓取了一段POST方法的请求报文和响应报文

这是POST方法的请求报文：

```

> Frame 7648: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface \Device\NPF_{B932FD84-A443-440B-B308-6C183DE22E67}, id 0
> Ethernet II, Src: IntelCor_71:57:05 (38:fc:98:71:57:05), Dst: NewH3CTe_7b:38:02 (54:c6:ff:7b:38:02)
> Internet Protocol Version 4, Src: 172.30.204.59, Dst: 59.111.19.33
> Transmission Control Protocol, Src Port: 1978, Dst Port: 80, Seq: 2378, Ack: 1, Len: 64
> [3 Reassembled TCP Segments (2441 bytes): #7646(917), #7647(1460), #7648(64)]
▼ Hypertext Transfer Protocol
  > POST /api/feedback/client/log HTTP/1.1\r\n
    Host: clientlog.music.163.com\r\n
    Connection: keep-alive\r\n
  > Content-Length: 1524\r\n
    Content-Type: multipart/form-data;boundary=0xKhTmLbOuNdArY-0EACC579-4631-4F5D-AC62-21DF87C1674B\r\n
  > [truncated]Cookie: userid=277772837;MUSIC_A_T=1463238104224;MUSIC_R_T=1463238125044;MUSIC_U=e505cfcea9d38212ba457bc71a630f3e0b50b01f699
    Referer: http://music.163.com/di\r\n
    User-Agent:\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Language: zh-CN,zh;q=0.8\r\n
    \r\n
    [Full request URI: http://clientlog.music.163.com/api/feedback/client/log]
    [HTTP request 1/1]

```

这是POST方法的响应报文:

```

▼ Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: Tengine\r\n
    Content-Type: image/png\r\n
  > Content-Length: 1509\r\n
    Connection: keep-alive\r\n
    Date: Wed, 12 Apr 2023 01:14:41 GMT\r\n
    Last-Modified: Wed, 11 Feb 2015 13:53:59 GMT\r\n
    ETag: "54db5ef7-5e5"\r\n
    Accept-Ranges: bytes\r\n
    Ali-Swift-Global-Savetime: 1681262880\r\n
    Via: cache45.l2cn2633[0,0,304-0,H], cache22.l2cn2633[0,0], cache2.cn1105[99,98,200-0,H], cache7.cn1105[101,0]\r\n
    Age: 980\r\n
    X-Cache: HIT TCP_REFRESH_HIT dirn:0:391554150\r\n
    X-Swift-SaveTime: Wed, 12 Apr 2023 01:44:20 GMT\r\n
    X-Swift-CacheTime: 3600\r\n
    Timing-Allow-Origin: *\r\n
    EagleId: 79c20a4b16812638606658190e\r\n
    \r\n
    [HTTP response 8/9]

```

这是GET方法的请求报文作为对比:

http.request.method==GET						
No.	Time	Source	Destination	Protocol	Length	Info
926	3.178597	172.30.204.59	58.205.221.206	HTTP	502	GET /static/index/img/home-section-tit4.png HTTP/1.1
1020	3.197968	172.30.204.59	58.205.221.206	HTTP	502	GET /static/index/img/home-section-tit5.png HTTP/1.1
1031	3.202323	172.30.204.59	58.205.221.206	HTTP	499	GET /static/common/img/header-logo.png HTTP/1.1
1097	3.246402	172.30.204.59	58.205.221.206	HTTP	508	GET /static/widget/footer/img/footer-sns-wx.png HTTP/1.1
1119	3.255079	172.30.204.59	58.205.221.206	HTTP	508	GET /static/widget/footer/img/footer-sns-wb.png HTTP/1.1
1395	3.378362	172.30.204.59	58.205.221.206	HTTP	604	GET /static/dep/slick/ajax-loader.gif HTTP/1.1
1436	3.388390	172.30.204.59	182.92.233.49	HTTP	602	GET /api/get_teacher_list.php?_1680945995888 HTTP/1.1
1484	3.426017	172.30.204.59	182.92.233.49	HTTP	599	GET /api/get_feed_list.php?_1680945995889 HTTP/1.1
1578	3.464216	172.30.204.59	58.205.209.185	HTTP	596	GET /cover/20220401/472f5ddf31dd99b594b1c8a3273860b2.jpg HTTP/1.1
1579	3.464332	172.30.204.59	58.205.209.185	HTTP	596	GET /cover/20220401/1266ff7648497e1c25dd022e7726d196.jpg HTTP/1.1
1923	3.778655	172.30.204.59	58.205.221.206	HTTP	632	GET /static/widget/teacher_list/img/teachers-staff-icon1.png HTTP/1.1
1929	3.779678	172.30.204.59	58.205.221.206	HTTP	632	GET /static/widget/teacher_list/img/teachers-staff-icon3.png HTTP/1.1

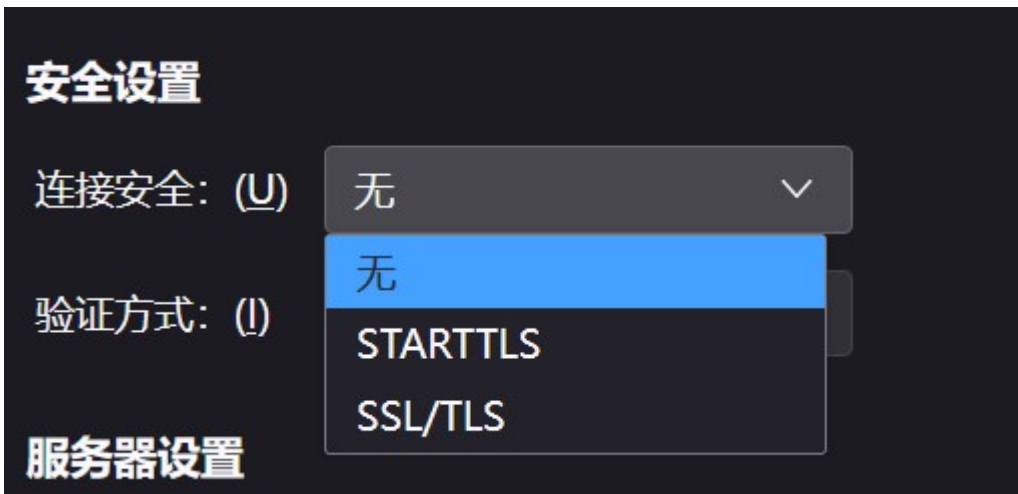
  

> Frame 926: 502 bytes on wire (4016 bits), 502 bytes captured (4016 bits) on interface \Device\NPF_{B932FD84-A443-440B-B308-6C183DE22E67}, id 0 > Ethernet II, Src: IntelCor_71:57:05 (38:fc:98:71:57:05), Dst: NewH3Cte_7b:38:02 (54:c6:ff:7b:38:02) > Internet Protocol Version 4, Src: 172.30.204.59, Dst: 58.205.221.206 > Transmission Control Protocol, Src Port: 1882, Dst Port: 80, Seq: 3629, Ack: 51650, Len: 448 > Hypertext Transfer Protocol > GET /static/index/img/home-section-tit4.png HTTP/1.1\r\n Host: img.chinesemoo.org\r\n Connection: keep-alive\r\n User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36\r\n DNT: 1\r\n Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n Referer: http://img.chinesemoo.org/static/index/index.css?20220828\r\n Accept-Encoding: gzip, deflate\r\n Accept-Language: zh-CN,zh;q=0.9\r\n \r\n [Full request URI: http://img.chinesemoo.org/static/index/img/home-section-tit4.png] [HTTP request 10/11] [Prev request in frame: 745] [Next request in frame: 1031]	0000 54 c6 ff 7b 38 02 38 fc 98 71 57 05 08 00 45 00 T..{8:..qm...E 0010 01 e8 25 01 40 00 80 06 00 00 ac 1e cc 3b 3a cd ..%@.....:.. 0020 dd ce 07 5a 00 50 40 4b 5d 17 7a f9 d7 b5 50 18 ...2P@K]z...P 0030 01 02 92 d0 00 00 47 45 54 20 2f 73 7a 61 74 69 .....GET /stati 0040 63 2f 69 6e 64 65 78 2f 69 6d 67 2f 68 6f 6d 65 c/index/ img/home 0050 2d 73 65 63 74 69 6f 6e 2d 74 69 74 34 2e 70 6e -section-tit4.pn 0060 67 20 48 54 54 50 2f 31 2e 31 00 0a 4b 6f 73 74 g HTTP/1.1 -Host 0070 3a 20 69 6d 67 2e 63 68 69 6e 65 73 65 6d 6f 6f : img.ch inesemoo 0080 63 2e 6f 72 67 00 0a 43 6f 6e 6e 65 63 74 69 6f c.org -C onnectio 0090 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 6d 0a 55 n: keep- alive -U 00a0 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6e ser-Agen t: Nozill 00b0 6c 61 2f 35 2e 30 20 28 57 69 6e 64 6f 77 73 20 la/5.0 ( Windows 00c0 4e 54 20 31 30 2e 30 3b 20 57 69 6e 36 34 3b 20 NT 10.0; Win64; 00d0 78 36 34 29 20 41 70 70 6c 65 57 65 62 4b 69 74 x64) App lewebkit 00e0 2f 35 33 37 2e 33 36 20 28 4b 48 54 4d 4c 2c 20 /537.36 (KHTML, 00f0 6c 69 6b 65 20 47 65 63 6b 6f 29 20 43 68 72 6f like Gec ko) Chro 0100 6d 65 2f 31 30 35 2e 30 2e 30 2e 30 20 53 61 66 me/105.0 .0.0 Saf 0110 61 72 69 2f 35 33 37 2e 33 36 00 0a 44 4e 54 3a ari/537. 36- DNT: 0120 20 31 00 0a 41 63 63 65 70 74 3a 20 69 6d 61 67 i- Acce pt: imag 0130 65 2f 61 76 69 6e 2c 69 6d 61 67 65 2f 77 65 62 e/avif, i mage/web 0140 70 2c 69 6d 61 67 65 2f 61 70 6e 67 2c 69 6d 61 p,image/ apng,ima 0150 67 65 2f 73 76 67 2b 78 6d 6c 2c 69 6d 61 67 65 ge/svg+ x ml,image 0160 2f 2a 2c 2a 2f 2a 3b 71 3d 30 2e 3b 0d 0a 52 65 /*,*/*;q=0.8- Re 0170 66 65 72 65 72 3a 20 68 74 74 70 3a 2f 2f 69 6d ferer: h ttp://im 0180 67 2e 63 68 69 6e 65 73 65 6d 6f 6f 63 2e 6f 72 g.chines emoo.or 0190 67 2f 73 74 61 74 69 63 2f 69 6e 64 65 78 2f 69 g/static /index/i 01a0 6e 64 65 78 2e 63 73 73 3f 32 30 32 32 30 38 32 ndex.css ?2022082
---	---

## task4

利用Wireshark抓取SMTP和POP3网络包，分析SMTP和POP3数据包组成（要求根据报文结构正确标识每个部分），请将实验结果附在实验报告中。

这个在实践过程中我这里出了一些问题。我用的不是实验操作手册里面讲道德Foxmail邮箱客户端而是自己常用的Thunderbird邮箱客户端。这个要进行一些类似的配置，否则Wireshark里面抓取不到任何报文。





## SMTP服务器

### 设置

描述(D):

服务器名称: (S) smtp.163.com

端口: (P) 25 默认: 587

### 安全及认证

连接安全: (N) 无

验证方式: (I) 无

用户名: (M) SSL/TLS

确定

取消

然后就是要注意在163邮箱里面开启SMTP和POP3服务，接着登录我的邮箱后刷新一下我就得到了POP3数据包:（这里为了保护隐私信息，抹去了相关的邮箱地址）

The image shows a Wireshark packet capture of POP3 traffic. The packet list on the left shows several packets, with packet 186187 selected. The packet details pane on the right shows the structure of the selected packet, including the POP3 protocol fields. A context menu is open over the packet list, showing various actions that can be performed on the selected packet, such as '追踪流' (Follow Stream), '复制' (Copy), and '解码为...' (Decode As...). The '追踪流' option is highlighted, and a submenu is visible showing the available stream types: 'TCP 流' (Follow TCP Stream), 'UDP 流' (Follow UDP Stream), 'DCCP Stream', 'TLS 流' (Follow TLS Stream), and 'HTTP 流' (Follow HTTP Stream).

No.	Time	Source	Destination	Protocol	Length	Info
186181	2449.650821	121.195.178.52	172.30.204.59	POP	141	S: +OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b524c
186182	2449.652475	172.30.204.59	121.195.178.52	POP	60	C: CAPA
186184	2449.680766	121.195.178.52	172.30.204.59	POP	157	S: +OK Capability list follows
186185	2449.682108	172.30.204.59	121.195.178.52	POP	81	C: USER m18435636781@163.com
186187	2449.713849	121.195.178.52	172.30.204.59	POP	69	S: +OK core mail
186188	2449.714516	172.30.204.59	121.195.178.52	POP	77	C: PASS NBVFVVBANAIKOBEX
186190	2449.791854	121.195.178.52	172.30.204.59	POP	88	S: +OK 1 message(s) [20093 byte(s)]
186191	2449.792489	121.195.178.52	172.30.204.59	POP	60	C: STAT
186193	2449.821713	121.195.178.52	172.30.204.59	POP	67	S: +OK 1 20093
186194	2449.822909	172.30.204.59	121.195.178.52	POP	60	C: LIST
186195	2449.850509	121.195.178.52	172.30.204.59	POP	79	S: +OK 1 20093
186196	2449.851853	172.30.204.59	121.195.178.52	POP	60	C: UIDL
186197	2449.879391	121.195.178.52	172.30.204.59	POP	96	S: +OK 1 20093
186198	2449.881589	172.30.204.59	121.195.178.52	POP	60	C: QUIT
186199	2449.916783	121.195.178.52	172.30.204.59	POP	69	S: +OK core mail

Context Menu Options:

- 标记/取消标记 分组(M) Ctrl+M
- 忽略/取消忽略 分组(I) Ctrl+D
- 设置/取消设置 时间参考 Ctrl+T
- 时间平移... Ctrl+Shift+T
- 分组注释
- 编辑解析的名称
- 作为过滤器应用
- 准备作为过滤器
- 对话过滤器
- 对话着色
- SCTP
- 追踪流
- 复制
- 协议首选项
- Decode As...

Follow Stream Submenu:

- TCP 流 Ctrl+Alt+Shift+T
- UDP 流 Ctrl+Alt+Shift+U
- DCCP Stream Ctrl+Alt+Shift+E
- TLS 流 Ctrl+Alt+Shift+S
- HTTP 流 Ctrl+Alt+Shift+H

Wireshark · 追踪 TCP 流 (tcp.stream eq 1121) · WLAN

+OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf5cs])  
CAPA  
+OK Capability list follows  
TOP  
USER  
PIPELINING  
UIDL  
LANG  
UTF8  
SASL PLAIN XOAUTH2  
STLS  
ID  
.  
USER m[REDACTED]@163.com  
+OK core mail  
PASS NBYFVYBANAIBOBEX  
+OK 1 message(s) [20093 byte(s)]  
STAT  
+OK 1 20093  
LIST  
+OK 1 20093  
1 20093  
.  
UIDL  
+OK 1 20093  
1 1tbiZR1LrF8ZXtbWGQAAsa  
.  
QUIT  
+OK core mail

7 客户端 分组, 8 服务器 分组, 14 turn(s).

整个对话 (414 bytes)

Show data as ASCII

流 1121

查找:

查找下一个 (N)

滤掉此流

打印

另存为...

返回

Close

Help

接下来是具体的分析：

Wireshark · 追踪 TCP 流 (tcp.stream eq 1121) · WLAN

```
+OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf5cs])
CAPA
+OK Capability list follows
TOP
USER
PIPELINING
UIDL
LANG
UTF8
SASL PLAIN XOAUTH2
STLS
ID
.
USER [REDACTED]@163.com 用户名
+OK core mail
PASS NBYFVYBANAIOBEX 密码
+OK 1 message(s) [20093 byte(s)]
STAT
+OK 1 20093
LIST 邮件数量
+OK 1 20093
1 20093
.
UIDL 返回用于指定邮件的唯一标识
+OK 1 20093
1 1tbiZRlLrF8ZXtbWQQAAsa
.
QUIT 结束会话
+OK core mail
```

7 客户端 分组, 8 服务器 分组, 14 turn(s).

整个对话 (414 bytes) Show data as ASCII 流 1121

查找: 查找下一个 (N)

滤掉此流 打印 另存为... 返回 Close Help

STAT返回邮箱里面的邮件数量以及邮件占用的空间大小信息而LIST返回某一封邮件的统计信息.

然后我尝试着用这个邮箱发了一封邮件后略微等一下之后Wireshark中便抓取到了SMTP网络包了:

```
220 *****
EHLO [172.30.204.59]
250-mail
250-PIPELINING
250-AUTH LOGIN PLAIN XOAUTH2
250-AUTH=LOGIN PLAIN XOAUTH2
250-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
250-STARTTLS
250-XB
250 8BITMIME
AUTH PLAIN AG0xODQzNTYzNjc4MUAXNjMuY29tAE5CWUZWWUJBTkFJS09CRVg=
235 Authentication successful
MAIL FROM:<m18435636781@163.com> BODY=8BITMIME
250 Mail OK
RCPT TO:<541171475@qq.com>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
Message-ID: <17619098-5616-949d-8ddc-3cb469063ddc@163.com>
Date: Sat, 8 Apr 2023 18:19:34 +0800
MIME-Version: 1.0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Thunderbird/102.6.1
To: 184mail163 <m18435636781@163.com>
From: 184mail163 <m18435636781@163.com>
Subject: ComputerNetwork SMTP test.
Content-Type: text/plain; charset=UTF-8; format=flowed
Content-Transfer-Encoding: 7bit

ComputerNetwork SMTP test.

.
250 Mail OK queued as zwqz-smtp-mta-g2-3, _____wBXzBm4PzFk3NCQAw--.56096S2 1680949176
QUIT
221 Bye
```

分组 202199。8 客户端 分组, 8 服务器 分组, 14 turn(s)。点击选择。

整个对话 (1083 bytes)

Show data as ASCII

流 1266

查找:

查找下一个(N)

滤掉此流

打印

另存为...

返回

Close

Help

接下来是具体的分析：



Wireshark · 追踪 TCP 流 (tcp.stream eq 1266) · WLAN

```
220 *****
EHLO [172.30.204.59]
250-mail 标识用户身份
250-PIPELINING
250-AUTH LOGIN PLAIN XOAUTH2
250-AUTH=LOGIN PLAIN XOAUTH2
250-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
250-STARTTLS
250-XB
250 8BITMIME 250:完成请求命令
AUTH PLAIN AG0xODQzNTYzNjc4MUAXNjMuY29tAE5CWUZWWUJBtKfJS09CRVg= 连接认证
235 Authentication successful
MAIL FROM:<m18435636781@163.com> BODY=8BITMIME
250 Mail OK
RCPT TO:<541171475@qq.com>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF> 消息内容
Message-ID: <17619098-5616-949d-8ddc-3cb469063ddc@163.com>
Date: Sat, 8 Apr 2023 18:19:34 +0800
MIME-Version: 1.0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Thunderbird/102.6.1
To: 184mail163@qq.com 收件人
From: 184mail163 <m18435636781@163.com> 发件人
Subject: ComputerNetwork SMTP test.
Content-Type: text/plain; charset=UTF-8; format=flowed
Content-Transfer-Encoding: 7bit

ComputerNetwork SMTP test.

.
250 Mail OK queued as zwqz-smtp-mta-g2-3, wBXzBm4PzFk3NCQAw--56096S2 1680949176
QUIT 关闭连接
221 Bye 221:关闭传输通道
```

分组 202199. 8 客户端 分组, 8 服务器 分组, 14 turn(s). 点击选择。

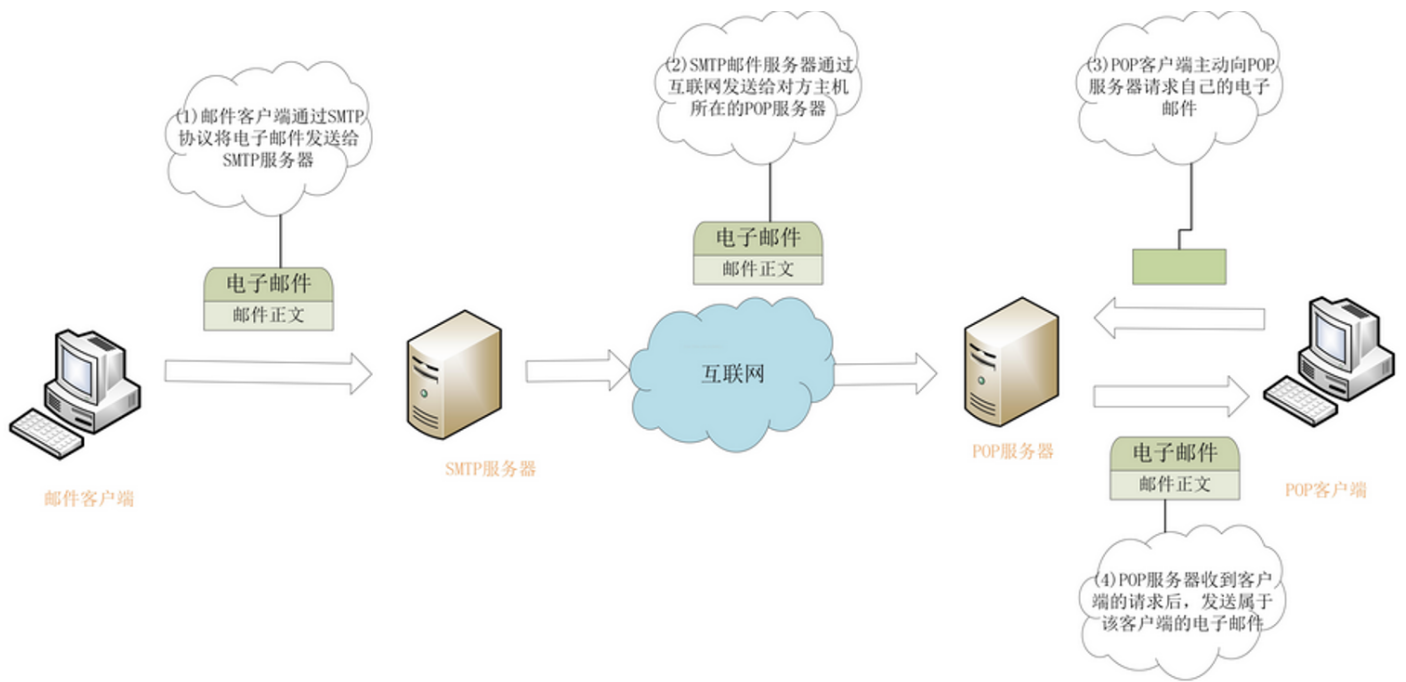
整个对话 (1083 bytes) Show data as ASCII 流 1266

查找: 查找下一个(N)

滤掉此流 打印 另存为... 返回 Close Help

- 220 代表连接 SMTP 服务器成功
- EHLO表明邮件带有身份验证，没有办法伪造的
- 250表示相关的请求命令完成了
- 221表示连接断开

为了更加深入地了解相关的原理，我上网查阅了SMTP和POP3协议的具体细节和流程。



## task5

利用Wireshark抓取SMTP网络包，分析一个在SMTP客户（C）和SMTP服务器（S）之间交换报文文本的例子（参考书本p77-78），请将实验结果附在实验报告中。

Wireshark · 追踪 TCP 流 (tcp.stream eq 1266) · WLAN

```
220 *****
EHLO [172.30.204.59]
250-mail
250-PIPELINING
250-AUTH LOGIN PLAIN XOAUTH2
250-AUTH=LOGIN PLAIN XOAUTH2
250-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXA
250-STARTTLS
250-XB
250 SBITIME
AUTH PLAIN AG0x0DQzNTYzNjc4MUAXNjMuY29tAE5CWUZWUJBTkFJS09CRVg=
235 Authentication successful
MAIL FROM: [REDACTED]@163.com BODY=SBITIME
250 Mail OK
RCPT TO: [REDACTED]@qq.com
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
Message-ID: <17619098-5616-949d-8ddc-3cb469063ddc@163.com>
Date: Sat, 8 Apr 2023 18:19:34 +0800
MIME-Version: 1.0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Thunderbird/102.6.1
To: [REDACTED]@qq.com
From: 184mail163 <[REDACTED]@163.com>
Subject: ComputerNetwork SMTP test.
Content-Type: text/plain; charset=UTF-8; format=flowed
Content-Transfer-Encoding: 7bit

ComputerNetwork SMTP test.

.
250 Mail OK queued as zwqz-smtp-mta-g2-3, [REDACTED]wBXzBm4PzFk3NCQAw--. 56096S2 1680949176
QUIT
221 Bye
```

分组 202199. 8 客户端 分组, 8 服务器 分组, 14 turn(s). 点击选择。

整个对话 (1083 bytes) Show data as ASCII 流 1266

查找: 查找下一个 (N)

滤掉此流 打印 另存为... 返回 Close Help

为了保护隐私，在这里我和上一个task一样隐去了相应的邮箱地址。

```
S:220 163.com(这个看不了具体的，但应该是163.com)
C:EHLO 172.30.204.59
S:Hello 172.30.204.59, pleased to meet you
C:AUTH AG...Vg=
S:235 Authentication successful
C:MAIL FROM:<...@163.com>
S:250 Mail OK
C:RCPT TO:<...@qq.com>
S:250 Mail OK
C:DATA
S:354 End data with <CR><LF>.<CR><LF>
C:.....(邮件内容)
C:..(结束了)
S:250 Mail OK queued as .....
C:QUIT
S:221 Bye
```

## 五、总结

这次实验整体上来说还是比较简单的，我通过抓取网络包并分析、观察其报文，熟悉了 HTTP、SMTP、POP3 的工作原理，明白了 GET 和 POST 方法的差异，直观地见识了它们在实际中的运行过程，巩固知识的同时为以后的理论学习和实验打下了坚实基础。