

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：21 级

上机实践成绩：

指导教师：张召

姓名：杨茜雅

学号：10215501435

上机实践名称：Lab06

上机实践日期：2023.4.7

上机实践编号：6

组号：

上机实践时间：4.7-4.14

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、使用环境

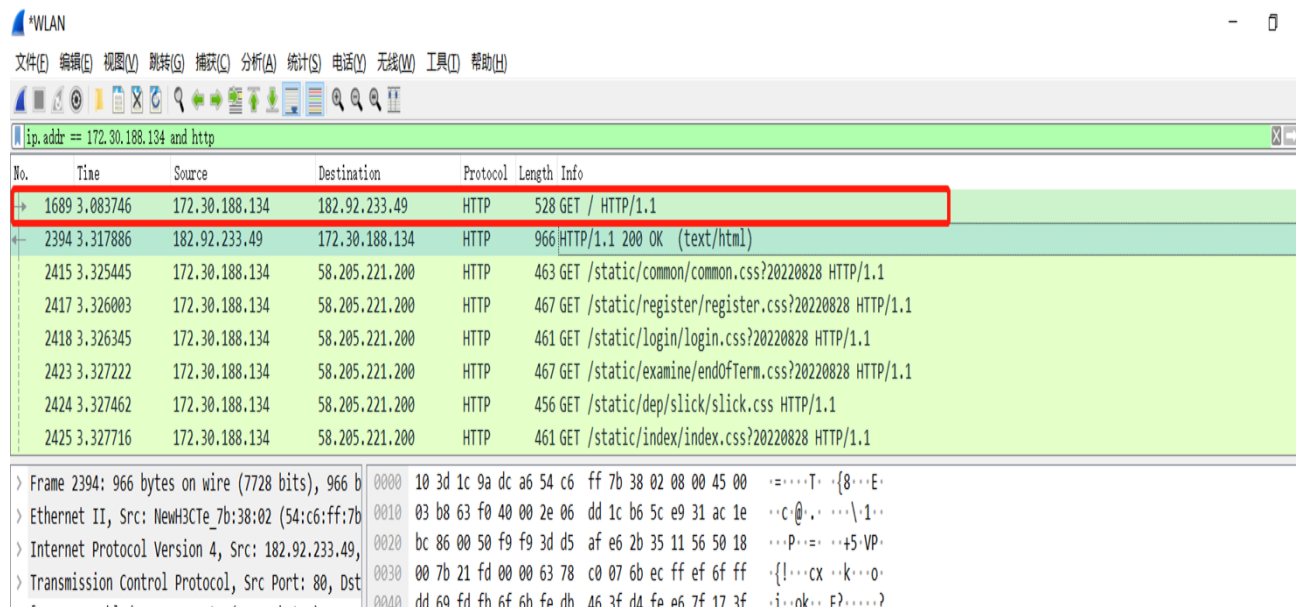
Wireshark

四、实验过程

Task 1:

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。

为避免过多网络包影响分析，在显示过滤器栏输入ip.addr == 172.30.188.134 and http，此时Wireshark 会按照条件过滤网络包，我选择的 HTTP 请求网络包如下：



可以看出 info 为请求行，其中方法字段为 GET，是绝大部分 HTTP 请求报文使用的方法；URL 代表请求的对象，这里其实就是我们输入的网址 <http://www.chinesemooc.org>；HTTP 版本是 HTTP/1.1 版本。

接下来让我们看看网络包的组成：

HTTP 请求报文的全部内容为：

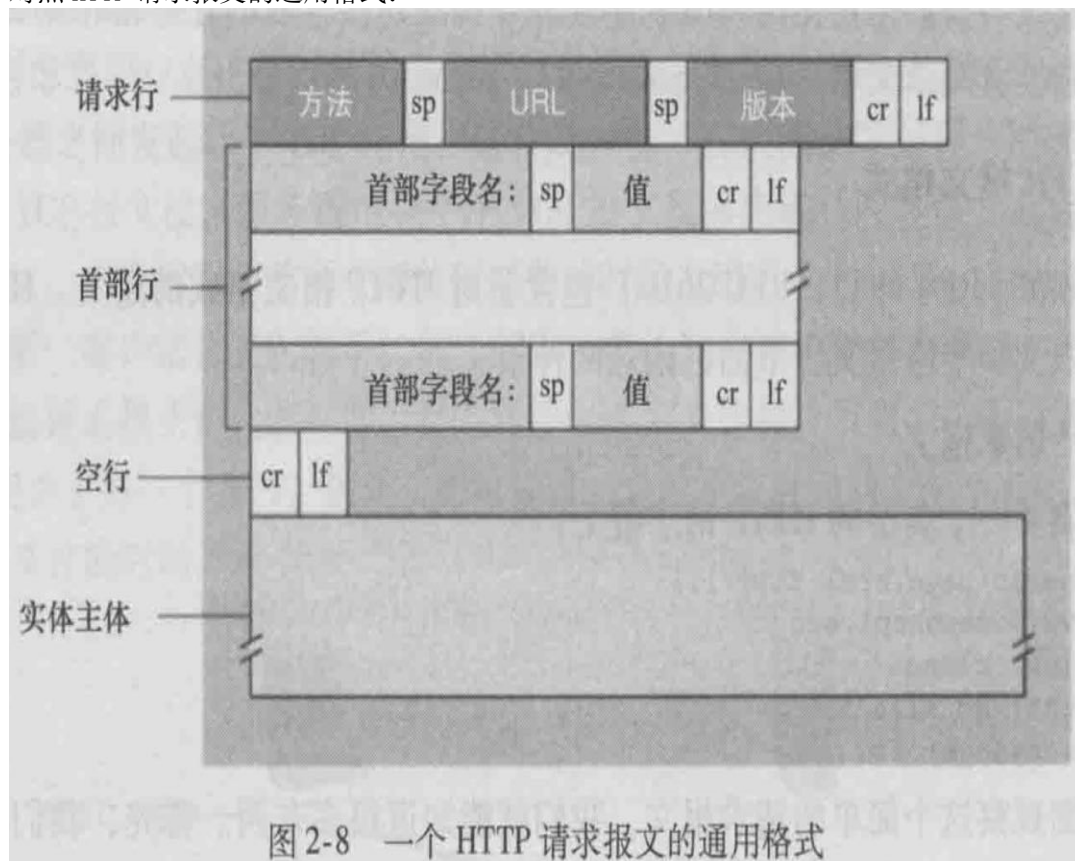
```

Wireshark · 分组 1689 · WLAN

> Frame 1689: 528 bytes on wire (4224 bits), 528 bytes captured (4224 bits) on interface \Device\NPF_{90CB2279-4FDD-49D7-8293-23DC640088EA}, id 0
> Ethernet II, Src: IntelCor_9a:dc:a6 (10:3d:1c:9a:dc:a6), Dst: NewH3CTe_7b:38:02 (54:c6:ff:7b:38:02)
> Internet Protocol Version 4, Src: 172.30.188.134, Dst: 182.92.233.49
> Transmission Control Protocol, Src Port: 63993, Dst Port: 80, Seq: 1, Ack: 1, Len: 474
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.chinesemooc.org\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.62\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
    \r\n
    [Full request URI: http://www.chinesemooc.org/]
    [HTTP request 1/2]
    [Response in frame: 2394]
    [Next request in frame: 2709]

0030  02 00 0a 28 00 00 47 45 54 20 2f 20 48 54 54 50  ...(..GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e  /1.1..Ho st: www.
0050  63 68 69 6e 65 73 65 6d 6f 6f 63 2e 6f 72 67 0d  chinesem ooc.org-
0060  0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65  -connect ion: kee
0070  70 2d 61 6c 69 76 65 0d 0a 55 70 67 72 61 64 65  p-alive- ·Upgrade
  
```

对照 HTTP 请求报文的通用格式：



GET 方法的 HTTP 请求报文：

```

> Frame 1689: 528 bytes on wire (4224 bits), 528 bytes captured (4224 bits) on interface \Device\NPF_{90CB2279-4
> Ethernet II, Src: IntelCor_9a:dc:a6 (10:3d:1c:9a:dc:a6), Dst: NewH3CTe_7b:38:02 (54:c6:ff:7b:38:02)
> Internet Protocol Version 4, Src: 172.30.188.134, Dst: 182.92.233.49
> Transmission Control Protocol, Src Port: 63993, Dst Port: 80, Seq: 1, Ack: 1, Len: 474
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n } 请求行
    Host: www.chinesemoooc.org\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/s
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n
    \r\n
  [Full request URI: http://www.chinesemoooc.org/]
  [HTTP request 1/2]
  [Response in frame: 2394]
  [Next request in frame: 2709]
  
```

Task 2:

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。

找到相对应的 HTTP 响应网络包：

ip.addr == 172.30.188.134 and http

No.	Time	Source	Destination	Protocol	Length	Info
1689	3.083746	172.30.188.134	182.92.233.49	HTTP	528	GET / HTTP/1.1
2394	3.317886	182.92.233.49	172.30.188.134	HTTP	966	HTTP/1.1 200 OK (text/html)
2415	3.325445	172.30.188.134	58.205.221.200	HTTP	463	GET /static/common/common.css?20220828 HTTP/1.1
2417	3.326003	172.30.188.134	58.205.221.200	HTTP	467	GET /static/register/register.css?20220828 HTTP/1.1
2418	3.326345	172.30.188.134	58.205.221.200	HTTP	461	GET /static/login/login.css?20220828 HTTP/1.1
2423	3.327222	172.30.188.134	58.205.221.200	HTTP	467	GET /static/examine/endOfTerm.css?20220828 HTTP/1.1
2424	3.327462	172.30.188.134	58.205.221.200	HTTP	456	GET /static/dep/slick/slick.css HTTP/1.1
2425	3.327716	172.30.188.134	58.205.221.200	HTTP	461	GET /static/index/index.css?20220828 HTTP/1.1

对照 HTTP 相应报文的通用格式：

协议版本	空格	状态码	空格	状态码描述	回车符	换行符	状态行
头部字段名	:	值	回车符	换行符	} 响应头部		
...							
头部字段名	:	值	回车符	换行符			
回车符	换行符						} 响应正文

图2 HTTP响应报文

标识报文结构的每个部分：

```
> Frame 2394: 966 bytes on wire (7728 bits), 966 bytes captured (7728 bits) on interface \Device\NPF_{90CB2279-4FDD-49D7-8293-23DC64008BEA}, id 0
> Ethernet II, Src: NewH3CTe_7b:38:02 (54:c6:ff:7b:38:02), Dst: IntelCor_9a:dc:a6 (10:3d:1c:9a:dc:a6)
> Internet Protocol Version 4, Src: 182.92.233.49, Dst: 172.30.188.134
> Transmission Control Protocol, Src Port: 80, Dst Port: 63993, Seq: 16561, Ack: 475, Len: 912
> [13 Reassembled TCP Segments (17472 bytes): #2277(1380), #2278(1380), #2280(1380), #2281(1380), #2282(1380), #2283(1380), #2284(1380), #2285(1380)]
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n 状态行：协议版本+状态码+状态码描述
    Server: nginx\r\n      HTTP/1.1 200 OK
    Date: Mon, 10 Apr 2023 02:38:08 GMT\r\n
    Content-Type: text/html\r\n 头部字段名+值+回车符+换行符
    Transfer-Encoding: chunked\r\n
    Connection: keep-alive\r\n
    Set-Cookie: pku_auth=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/\r\n
    Content-Encoding: gzip\r\n
    \r\n 回车符+换行符
    [HTTP response 1/2]
    [Time since request: 0.234140000 seconds]
    [Request in frame: 1689]
    [Next request in frame: 2709]
    [Request URI: http://www.chinesemoooc.org/] 与请求英文的URL一致
    HTTP chunked response
    Content-encoded entity body (gzip): 17210 bytes -> 71759 bytes
    File Data: 71759 bytes
  > Line-based text data: text/html (1084 lines)
```

对比两个网络包的组成，分析区别：

对比发现两者的主要区别在于请求行和状态行

请求行：

请求方法	空格	URL	空格	协议版本	回车符	换行符	请求行
GET		/		HTTP/1.1			GET / HTTP/1.1\r\n

状态行

协议版本	空格	状态码	空格	状态码描述	回车符	换行符	状态行
HTTP/1.1		200		OK (text/html)			HTTP/1.1 200 OK (text/html)\r\n

剩下的请求头部、请求正文以及响应头部、响应正文组成类似

头部字段名	:	值	回车符	换行符
...				
头部字段名	:	值	回车符	换行符
回车符	换行符			

Task 3: 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。

POST 方法的请求和响应报文

No.	Time	Source	Destination	Protocol	Length	Info
1006	8.352774	172.30.188.134	120.52.183.129	HTTP	103	POST /squery_v2?166585906 HTTP/1.0
1009	8.477682	120.52.183.129	172.30.188.134	HTTP	516	HTTP/1.1 200 OK
1398	8.916538	172.30.188.134	120.92.73.121	HTTP	97	POST /query3?163812 HTTP/1.0
1401	8.916908	172.30.188.134	120.92.73.121	HTTP	97	POST /query3?184000 HTTP/1.0
1406	8.949055	120.92.73.121	172.30.188.134	HTTP	346	HTTP/1.1 200 OK (text/plain)
1408	8.950540	120.92.73.121	172.30.188.134	HTTP	346	HTTP/1.1 200 OK (text/plain)
1467	9.101673	172.30.188.134	182.92.233.49	HTTP	588	GET / HTTP/1.1
1577	9.344176	182.92.233.49	172.30.188.134	HTTP	965	HTTP/1.1 200 OK (text/html)

请求报文内容如下:

```
> Frame 1006: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface \Device\NPF_{90CB2279-4FDD-49D7-8293-23DC64008BEA}, id 0
> Ethernet II, Src: IntelCor_9a:dc:a6 (10:3d:1c:9a:dc:a6), Dst: NewH3CTe_7b:38:02 (54:c6:ff:7b:38:02)
> Internet Protocol Version 4, Src: 172.30.188.134, Dst: 120.52.183.129
> Transmission Control Protocol, Src Port: 53395, Dst Port: 80, Seq: 287, Ack: 1, Len: 49
> [2 Reassembled TCP Segments (335 bytes): #1005(286), #1006(49)]
< Hypertext Transfer Protocol
  > POST /squery_v2?166585906 HTTP/1.0\r\n
    Host: 120.52.183.129\r\n
    Accept: */*\r\n
    Content-Type: multipart/form-data; boundary=-VisualSeawind-\r\n
    Accept: */*\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Microsoft-ATL-Native/8.00\r\n
    Connection: keep-alive\r\n
    Accept-Language: zh-CN\r\n
  > Content-Length: 49\r\n
    \r\n
    [Full request URI: http://120.52.183.129/squery_v2?166585906]
    [HTTP request 1/1]
    [Response in frame: 1009]
    File Data: 49 bytes
  > MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-VisualSeawind-"
```

对比 GET 和 POST 方法的请求报文, 可以发现:

- 1、最直观的区别就是 GET 把参数包含在 URL 中, POST 通过 request body 传递参数。
- 2、get 请求只能进行 url 编码 (appliacation-x-www-form-urlencoded), post 请求支持多种 (multipart/form-data 等)
- 3、GET 比 POST 更不安全, 因为参数直接暴露在 URL 上, 所以不能用来传递敏感信息。

POST 方法的响应报文:

```
> Frame 1009: 516 bytes on wire (4128 bits), 516 bytes captured (4128 bits) on interface \Device\NPF_{90CB2279-4FDD-49D7-8293-23DC64008BEA}, id 0
> Ethernet II, Src: NewH3CTe_7b:38:02 (54:c6:ff:7b:38:02), Dst: IntelCor_9a:dc:a6 (10:3d:1c:9a:dc:a6)
> Internet Protocol Version 4, Src: 120.52.183.129, Dst: 172.30.188.134
> Transmission Control Protocol, Src Port: 80, Dst Port: 53395, Seq: 1, Ack: 336, Len: 462
< Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: Tengine/1.5.2\r\n
    Date: Mon, 10 Apr 2023 06:35:49 GMT\r\n
    Content-Type: application/octet-stream\r\n
  > Content-Length: 273\r\n
    Connection: keep-alive\r\n
    Content-Tag: 1936292724\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.124908000 seconds]
    [Request in frame: 1006]
    [Request URI: http://120.52.183.129/squery_v2?166585906]
    File Data: 273 bytes
  > Data (273 bytes)
```

对比 GET 方法和 POST 方法的响应报文，可以发现：

- 1、GET 表示从服务器获取资源，而POST 表示向指定的服务器资源提交数据。
- 2、get 传送的数据量较小，不能大于2KB。post 传送的数据量较大，一般被默认为不受限制。

Task 4: 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。

发送邮件后，利用Wireshark过滤SMTP和POP3协议的数据包

No.	Time	Source	Destination	Protocol	Length	Info
186	14.050980	220.181.15.161	172.30.188.134	SMTP	119 S: 220	*****
187	14.055140	172.30.188.134	220.181.15.161	SMTP	76 C: EHLO	DESKTOP-786SIJ7
190	14.087464	220.181.15.161	172.30.188.134	SMTP	263 S: 250-mail	PIPELINING AUTH LOGIN PLAIN XOAUTH2 AUTH=LOGIN PLAIN XOAUTH2 XXXXXXXXXXXXXXXXXXXXXXXX...
191	14.088824	172.30.188.134	220.181.15.161	SMTP	66 C: AUTH	LOGIN
192	14.116404	220.181.15.161	172.30.188.134	SMTP	72 S: 334	dXNlcm5hbWU6
193	14.117295	172.30.188.134	220.181.15.161	SMTP	88 C: User:	TGlseTEzODE3Wzc2MDgzQDE2My5jb20=
194	14.145530	220.181.15.161	172.30.188.134	SMTP	72 S: 334	UGFzc3dvcmQ6
195	14.146389	172.30.188.134	220.181.15.161	SMTP	80 C: Pass:	Q0NFVfdlVFBCwkbVBTJVSQ==
198	14.222846	220.181.15.161	172.30.188.134	SMTP	85 S: 235	Authentication successful
199	14.226015	172.30.188.134	220.181.15.161	SMTP	92 C: MAIL	FROM: <Lily13817776083@163.com>
201	14.254195	220.181.15.161	172.30.188.134	SMTP	67 S: 250	Mail OK
202	14.255156	172.30.188.134	220.181.15.161	SMTP	92 C: RCPT	TO: <math_modeling2023@163.com>
203	14.284973	220.181.15.161	172.30.188.134	SMTP	67 S: 250	Mail OK
204	14.285993	172.30.188.134	220.181.15.161	SMTP	60 C: DATA	
205	14.313108	220.181.15.161	172.30.188.134	SMTP	91 S: 354	End data with <CR><LF>.<CR><LF>
206	14.316701	172.30.188.134	220.181.15.161	SMTP	1078 C: DATA	fragment, 1024 bytes
210	14.385406	172.30.188.134	220.181.15.161	SMTP/I...	377 from:	"Lily13817776083@163.com" <Lily13817776083@163.com>, subject: check, (text/plain) (text/html)
212	14.425534	220.181.15.161	172.30.188.134	SMTP	140 S: 250	Mail OK queued as zwqz-smtp-mta-g5-1, _____wB3Huv3xzNkm5FgBA--..35165S2 1681115127
213	14.427135	172.30.188.134	220.181.15.161	SMTP	60 C: QUIT	
214	14.454300	220.181.15.161	172.30.188.134	SMTP	63 S: 221	Bye
296	17.819201	121.195.178.52	172.30.188.134	POP	141 S: +OK	Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf5cs])
297	17.822389	172.30.188.134	121.195.178.52	POP	84 C: USER	Lily13817776083@163.com
299	17.850549	121.195.178.52	172.30.188.134	POP	69 S: +OK	core mail
300	17.851196	172.30.188.134	121.195.178.52	POP	77 C: PASS	CCETWHTPBZEAMRUI
304	18.154889	121.195.178.52	172.30.188.134	POP	91 S: +OK	27 message(s) [4804180 byte(s)]

通过追踪流得到具体的交互信息：

203	14.284973	220.181.15.161	172.30.188.134	SMTP	67 S: 250	Mail OK	编辑解析的名称	
204	14.285993	172.30.188.134	220.181.15.161	SMTP	60 C: DATA		作为过滤器应用	
205	14.313108	220.181.15.161	172.30.188.134	SMTP	91 S: 354	End data with <<	准备作为过滤器	
206	14.316701	172.30.188.134	220.181.15.161	SMTP	1078 C: DATA	fragment, 1024	对话过滤器	
210	14.385406	172.30.188.134	220.181.15.161	SMTP/I...	377 from:	"Lily13817776083@163.com" <Lily13817776083@163.com>, subject: check, (text/plain) (text/html)	对话着色	
212	14.425534	220.181.15.161	172.30.188.134	SMTP	140 S: 250	Mail OK queued as zwqz-smtp-mta-g5-1, _____wB3Huv3xzNkm5FgBA--..35165S2 1681115127	SCTP	
213	14.427135	172.30.188.134	220.181.15.161	SMTP	60 C: QUIT		追踪流	TCP 流 Ctrl+Alt+Shift+T
214	14.454300	220.181.15.161	172.30.188.134	SMTP	63 S: 221	Bye	复制	UDP 流 Ctrl+Alt+Shift+U
296	17.819201	121.195.178.52	172.30.188.134	POP	141 S: +OK	Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf5cs])		DCCP Stream Ctrl+Alt+Shift+E
297	17.822389	172.30.188.134	121.195.178.52	POP	84 C: USER	Lily13817776083@163.com	协议首选项	TLS 流 Ctrl+Alt+Shift+S
299	17.850549	121.195.178.52	172.30.188.134	POP	69 S: +OK	core mail	Decode As...	HTTP 流 Ctrl+Alt+Shift+H
300	17.851196	172.30.188.134	121.195.178.52	POP	77 C: PASS	CCETWHTPBZEAMRUI	在新窗口显示分组(W)	HTTP/2 Stream
304	18.154889	121.195.178.52	172.30.188.134	POP	91 S: +OK	27 message(s) [4804180 byte(s)]		QUIC Stream

抓取POP3网络包：

293	17.761488	172.30.188.134	121.195.178.52	TCP	66 60457 → 110 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
294	17.790952	121.195.178.52	172.30.188.134	TCP	66 110 → 60457 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PERM WS=128
295	17.791075	172.30.188.134	121.195.178.52	TCP	54 60457 → 110 [ACK] Seq=1 Ack=1 Win=131072 Len=0
296	17.819201	121.195.178.52	172.30.188.134	POP	141 S: +OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf5cs])
297	17.822389	172.30.188.134	121.195.178.52	POP	84 C: USER Lily13817776083@163.com
298	17.850016	121.195.178.52	172.30.188.134	TCP	54 110 → 60457 [ACK] Seq=88 Ack=31 Win=14720 Len=0
299	17.850549	121.195.178.52	172.30.188.134	POP	69 S: +OK core mail

具体组成+分析如下:

Wireshark - 追踪 TCP 流 (tcp.stream eq 32) - WLAN

```

+OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d26d71b52404dcf5cs])
USER Lily13817776083@163.com 认证用户名
+OK core mail
PASS CCETWHTPBZEAMRUI 认证用户密码 } 以明文形式鉴别用户
+OK 27 message(s) [4804180 byte(s)]
STAT 处理请求 serve 回送邮箱统计资料
+OK 27 4804180
LIST 返回指定邮件大小
+OK 27 4804180
1 28931
2 56712
3 3793517
4 228649
5 41650
6 6637
7 29378
8 54084
9 8037
10 67322
11 67868
12 56301
13 32148
14 28902
15 49236
16 29280
17 8719
18 6600
19 67599
20 5778
21 67473
22 53359
23 3547
24 3591
25 3528
26 3529
27 1805
.
UIDL 返回用于指定邮件的唯一错误
+OK 27 4804180
1 1tbiPR4u9GI0VeGwIgAAsJ
2 1tbiPQew9GI0Ve+ogAAAs8
3 1tbizGUz9FaDqsTbHwAAsz
4 1tbiyAU0l1p7JvDSbgAAml
5 xtbBFQY39GB9mHusQQAAs7
6 1tbiVxAS9FetqWxzNwAAAsx
7 xtbBFRg59GB9mJPisGAAsG
8 1tbiyBU59Fp7JyEoYwAAs8
9 1tbivGI69FZcgwsu+wAAAsl
10 1tbiVvw69FetqXh7SgAAAsP
11 1tbiPR099GI0VoNQQAAsD
12 1tbivg4+9FZcgZDcfwAAAsg
13 1tbivg4+9FZcgZDcfwABsh
14 1tbiPQs-9GI0VpjXbwAAAsd
15 1tbiyANC9Fp7J4LT0AAAs6
16 1tbiyBRC9Fp7J4X0+wAAAsm
17 xtbBFQJC9GB9mPrGLwAAAs3
18 xtbBFQJC9GB9mPrGLwABs2
19 1tbiPQRE9GI0VtGywgAAAsO
20 1tbiVw9H9FetqgW2twAAAs-
21 xtbBFRRL9GB9mVINIAAAsx
22 xtbBFRRL9GB9mVINIAABsw
23 xtbBFQpN9GB9mWLGaAAAsu
24 1tbiyB5N9Fp7J+6wEQAAAsm
25 xtbBFRJN9GB9mWLPAAAsT
26 xtbBFRJN9GB9mWLPAAAsS
27 1tbiVhxN9FZcg8+b1AAAsS
.
QUIT 结束会话
+OK core mail
  
```

+OK 命令正常 -ERR 命令出差错

UIDL 返回用于指定邮件的唯一错误

QUIT 结束会话

6 客户端 分组, 7 服务器 分组, 12 sum(x)

整个对话 (1259 bytes)

Show data as ASCII

查找: 查找下一个 (N)

抓取SMTP网络包:

187	14.055140	172.30.188.134	220.181.15.161	SMTP	76 C: EHLO DESKTOP-786SIJ7
189	14.081011	220.181.15.161	172.30.188.134	TCP	54 25 → 60451 [ACK] Seq=66 Ack=23 Win=29312 Len=0
190	14.087464	220.181.15.161	172.30.188.134	SMTP	263 S: 250-mail PIPELINING AUTH LOGIN PLAIN XOAUTH2 AUTH=LOGIN PLAIN XOAUTH2 XXXXXXXXXXXXXXXXXXXXXXXX...
191	14.088824	172.30.188.134	220.181.15.161	SMTP	66 C: AUTH LOGIN
192	14.116404	220.181.15.161	172.30.188.134	SMTP	72 S: 334 dXNlcm5hbWU6

具体组成+分析如下:

Wireshark · 追踪 TCP 流 (tcp.stream eq 24) · WLAN

```

220 *****
EHLO DESKTOP-786SIJ7
250-mail 标识用户身份
250-PIPELINING 250-请等待命令完成
250-AUTH LOGIN PLAIN XOAUTH2
250-AUTH=LOGIN PLAIN XOAUTH2
250-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
250-STARTTLS
250-XB
250 8BITMIME
AUTH LOGIN 认证连接
334 dXNlcm5hbWU6
TGlseTEzODE3Nzc2MDgzQDE2My5jb20=
334 UGFzc3dvcmQ6
235 Authentication successful
MAIL FROM: <Lily1381776083@163.com> 发件人地址
250 Mail OK
RCPT TO: <math_modeling2023@163.com> 接收人地址
250 Mail OK
DATA 消息内容
354 End data with <CR><LF>.<CR><LF>
Date: Mon, 10 Apr 2023 16:25:27 +0800
From: "Lily1381776083@163.com" <Lily1381776083@163.com>
To: math_modeling2023 <math_modeling2023@163.com>
Subject: check
X-Priority: 3
X-Has-Attach: no
X-Mailer: Foxmail 7.2.25.213[cn]
Mime-Version: 1.0
Message-ID: <2023041016252699397613@163.com>
Content-Type: multipart/alternative;
        boundary="-----_001_NextPart450665583770_-----"

This is a multi-part message in MIME format.

-----_001_NextPart450665583770_-----
Content-Type: text/plain;
        charset="us-ascii"
Content-Transfer-Encoding: base64

DQpoZWxsbnw0KDQoNCkxpbHkxMzgxNzc3NjA4MDAxNjMuY29tDQo=

-----_001_NextPart450665583770_-----
Content-Type: text/html;
        charset="us-ascii"
Content-Transfer-Encoding: quoted-printable

<html><head><meta http-equiv=3D"content-type" content=3D"text/html; charse=
t=3Dus-ascii"><style>body { line-height: 1.5; }body { font-size: 14px; fon=
t-family: "Microsoft YaHei UI"; color: rgb(0, 0, 0); line-height: 1.5; }<=
style></head><body>=0A<div><span></span></div>=0A<div>hello</div><hr s=
tyle=3D"width: 210px; height: 1px;" color=3D"#b5c4df" size=3D"1" align=3D"=
left">=0A<div><span><div style=3D"MARGIN: 10px; FONT-FAMILY: verdana; FONT=
-SIZE: 10pt"><div>Lily1381776083@163.com</div></div></span></div>=0A</bod=
y></html>

-----_001_NextPart450665583770_-----
.
250 Mail OK queued as zwqz-smtp-mta-g5-1,_____wB3Huv3xzNkm5FgBA--.35165S2 1681115127
QUIT 关闭连接
221 Bye
221-服务关闭传输通道
    
```

10 客户端 分组, 10 服务器 分组, 10 run(s).

整个对话 (2028 bytes) Show data as ASCII 流 24

查找: 查找下一个(N)

滤掉此流 打印 另存为... 返回 Close Help

Task 5: 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。

```

220 *****
EHLO DESKTOP-786SIJ7
250-mail
250-PIPELINING
250-AUTH LOGIN PLAIN XOAUTH2
250-AUTH=LOGIN PLAIN XOAUTH2
250-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
250-STARTTLS
250-XB
250 8BITMIME
AUTH LOGIN
334 dXNlcm5hbWU6
TGlseTEzODE3Nzc2MDgzQDE2My5jb20=
334 UGFzc3dvcmQ6
Q0NFVFcIVFBCWkVBTJVVSQ==
235 Authentication successful
MAIL FROM: <Lily13817776083@163.com>
250 Mail OK
RCPT TO: <math_modeling2023@163.com>
250 Mail OK
DATA
354 End data with <CR><LF>.<CR><LF>
Date: Mon, 10 Apr 2023 18:59:55 +0800
From: "Lily13817776083@163.com" <Lily13817776083@163.com>
To: math_modeling2023 <math_modeling2023@163.com>
Subject: hello
X-Priority: 3
X-Has-Attach: no
X-Mailer: Foxmail 7.2.25.213[cn]
Mime-Version: 1.0
Message-ID: <2023041018595492032714@163.com>
Content-Type: multipart/alternative;
        boundary="-----_001_NextPart152382466420_-----"

This is a multi-part message in MIME format.

-----=_001_NextPart152382466420_-----
Content-Type: text/plain;
        charset="us-ascii"
Content-Transfer-Encoding: base64

DQpoZWxsbw0KDQoNCkxpbHkxMzgXNzc3NjA4M0AxNjMuY29tDQo=

-----=_001_NextPart152382466420_-----
Content-Type: text/html;
        charset="us-ascii"
Content-Transfer-Encoding: quoted-printable

<html><head><meta http-equiv=3D"content-type" content=3D"text/html; charse=
t=3Dus-ascii"><style>body { line-height: 1.5; }body { font-size: 14px; fon=
t-family: "Microsoft YaHei UI"; color: rgb(0, 0, 0); line-height: 1.5; }<=
style></head><body>=0A<div><span></span><br></div>=0A<div>hello</div><hr s=
tyle=3D"width: 210px; height: 1px;" color=3D"#b5c4df" size=3D"1" align=3D"=
left">=0A<div><span><div style=3D"MARGIN: 10px; FONT-FAMILY: verdana; FONT=
-SIZE: 10pt"><div>Lily13817776083@163.com</div></div></span></div>=0A</bod=
y></html>

-----=_001_NextPart152382466420_-----

.
250 Mail OK queued as zwqz-smtp-mta-g5-2,_____wDn9aIr7DNkv29zBA--.34947S2 1681124395
QUIT
221 Bye
  
```

10 客户端 分送, 10 服务器 分送, 18 转送(s).

整个对话 (2028 bytes) Show data as ASCII 流 18

查找: 查找下一个(N)

滤掉此流 打印 另存为... 返回 Close Help

S: 220 *****

C: EHLO DESKTOP-786SIJ7

S: 250 Hello DESKTOP-786SIJ7, please to meet you

C: AUTH LOGIN

S: 334 dXNlcm5hbWU6

```
C: MAIL FROM: <Lily13817776083@163.com>
S: 250 Mail OK
C: RCPT TO: <math_modeling@163.com>
S: 250 Mail OK
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: .
S: 250 Mail OK queued as zwqz-smtp-mta-g5-2,wDn9aIr7DNkv29ZBA--.34947S2 1681124395
C: QUIT
S: 221 Bye
```

五、 总结

通过本次实验，我熟悉了HTTP协议的工作原理，了解了HTTP协议在实际网络中的运行过程，熟悉了SMTP和POP3协议的工作原理，了解了SMTP和POP3协议在实际网络中的运行过程。同时实践上手通过Wireshark分析HTTP协议、SMTP和POP3协议，对书本上的理论知识有了更深刻的见解。