计算机网络实验报告

警示

1.实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2.当次小组成员成绩只计学号、姓名登录在下表中的。
3.在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4.实验报告文件以 PDF 格式提交。

院系	电子与信息工程学院	班 级	通信工程一班	组长	刘渤
学号	16308073	16308161	16308091	16308015	
学生	刘渤	邹紫婧	彭肖文	陈瑞佳	
实验分工					
刘渤	Telnet 协议分析模块 贡献：25%		邹紫婧	FTP 协议分析模块 贡献：25%	
彭肖文	HTTP 协议分析模块 贡献：25%		陈瑞佳	HTTP 协议分析模块 贡献：25%	

【实验题目】网络嗅探与协议分析实验

【实验目的】 通过网络嗅探了解网络数据类型、了解网络工作原理；学习相关工具的使用。

【实验内容】

第二版书：

(1)HTTP 协议分析：完成实验教程实例 2-3 的实验，回答实验提出的问题及实验思考。(P62)

(2)FTP 协议分析：完成实验教程实例 2-4。回答实验提出的问题及实验思考。(P66)

(3) telnet 协议分析：完成实验教程实例 2-5 实验 (P71)

【实验要求】

一些重要信息信息需给出截图。

注意实验步骤的前后对比！

【实验记录】(如有实验拓扑请自行画出，要求自行画出拓扑图)

（一）HTTP 协议分析

(1)在捕获的报文中，共有几种 HTTP 报文？客户机与服务器之间共建立了几个链接？服务器和客户机分别使用了哪几个端口

答：

1）从格式上来分，总共有两种 HTTP 报文：

① 请求报文

② 应答报文

从内容上来分，本次捕获的报文（主要是请求报文）有：

① GET / HTTP/1.1，方式是 GET，URI 是/，版本是 HTTP/1.1（持续连接）

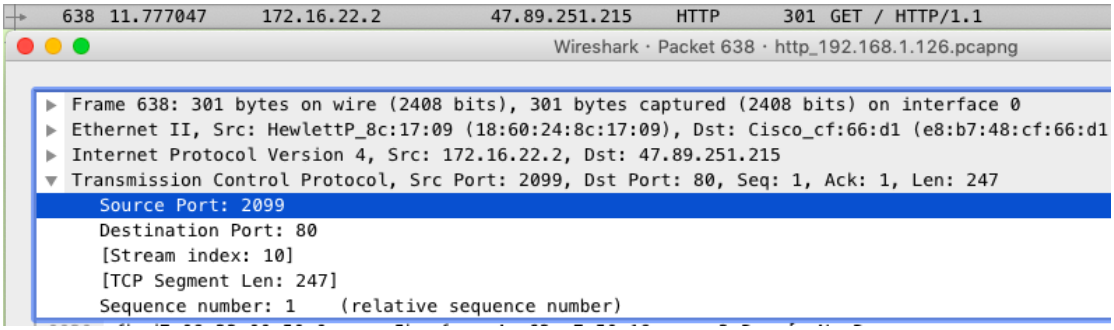
② GET /web.ico HTTP/1.1，方式是 GET，URI 是/web.ico，版本是 HTTP/1.1
这是服务器的图标。

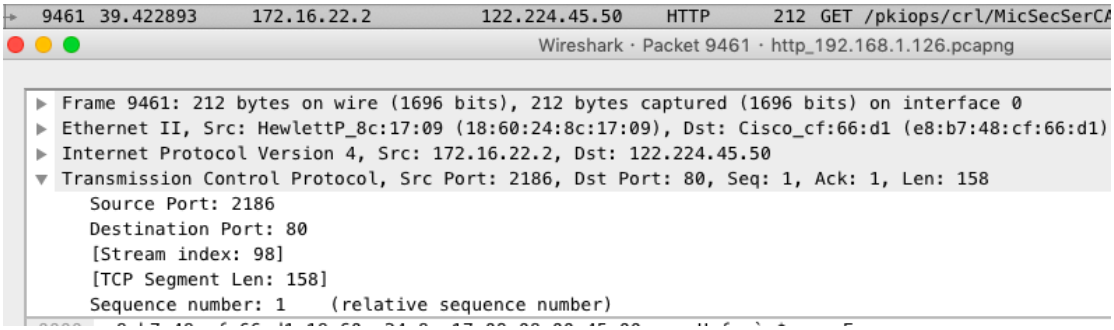
③ GET /favicon.ico HTTP/1.1，方式是 GET，URI 是/favicon.ico，版本是 HTTP/1.1
用于显示缩略的网站标志，一般是显示在浏览器地址栏、标签栏或者收藏夹上。

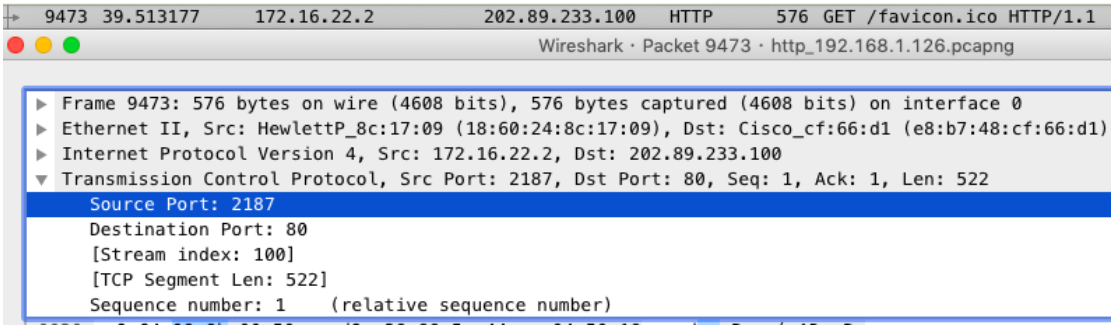
2）客户机和服务器之间建立了 6 个连接，因为客户机有六个不同的端口。

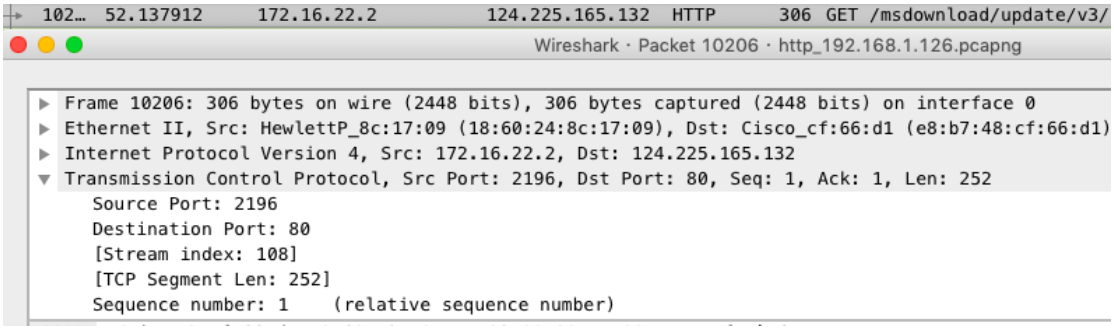
3）服务器只有一个端口：80，因为 HTTP 协议的端口都是 80。

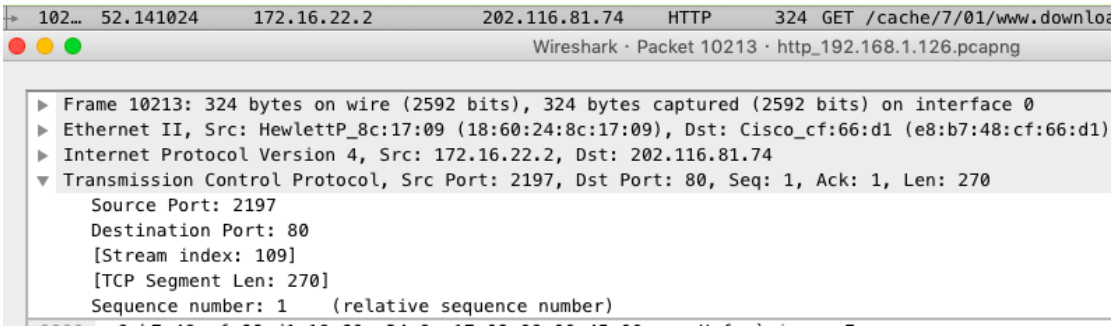
客户机的端口：2099、2186、2187、2196、2199、2205，下面是部分端口：

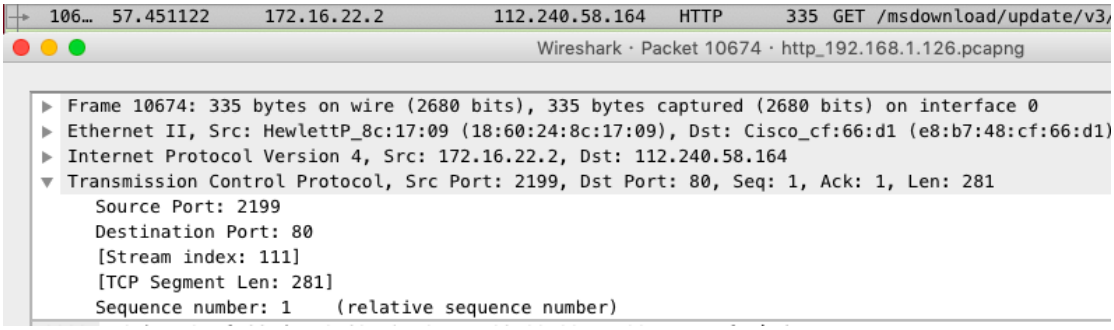

638 11.777047 172.16.22.2 47.89.251.215 HTTP 301 GET / HTTP/1.1
Wireshark · Packet 638 · http_192.168.1.126.pcapng
Frame 638: 301 bytes on wire (2408 bits), 301 bytes captured (2408 bits) on interface 0
Ethernet II, Src: HewlettP_8c:17:09 (18:60:24:8c:17:09), Dst: Cisco_cf:66:d1 (e8:b7:48:cf:66:d1
Internet Protocol Version 4, Src: 172.16.22.2, Dst: 47.89.251.215
Transmission Control Protocol, Src Port: 2099, Dst Port: 80, Seq: 1, Ack: 1, Len: 247
Source Port: 2099
Destination Port: 80
[Stream index: 10]
[TCP Segment Len: 247]
Sequence number: 1 (relative sequence number)


9461 39.422893 172.16.22.2 122.224.45.50 HTTP 212 GET /pkiops/crl/MicSecSerCA
Wireshark · Packet 9461 · http_192.168.1.126.pcapng
Frame 9461: 212 bytes on wire (1696 bits), 212 bytes captured (1696 bits) on interface 0
Ethernet II, Src: HewlettP_8c:17:09 (18:60:24:8c:17:09), Dst: Cisco_cf:66:d1 (e8:b7:48:cf:66:d1)
Internet Protocol Version 4, Src: 172.16.22.2, Dst: 122.224.45.50
Transmission Control Protocol, Src Port: 2186, Dst Port: 80, Seq: 1, Ack: 1, Len: 158
Source Port: 2186
Destination Port: 80
[Stream index: 98]
[TCP Segment Len: 158]
Sequence number: 1 (relative sequence number)


9473 39.513177 172.16.22.2 202.89.233.100 HTTP 576 GET /favicon.ico HTTP/1.1
Wireshark · Packet 9473 · http_192.168.1.126.pcapng
Frame 9473: 576 bytes on wire (4608 bits), 576 bytes captured (4608 bits) on interface 0
Ethernet II, Src: HewlettP_8c:17:09 (18:60:24:8c:17:09), Dst: Cisco_cf:66:d1 (e8:b7:48:cf:66:d1)
Internet Protocol Version 4, Src: 172.16.22.2, Dst: 202.89.233.100
Transmission Control Protocol, Src Port: 2187, Dst Port: 80, Seq: 1, Ack: 1, Len: 522
Source Port: 2187
Destination Port: 80
[Stream index: 100]
[TCP Segment Len: 522]
Sequence number: 1 (relative sequence number)


102… 52.137912 172.16.22.2 124.225.165.132 HTTP 306 GET /msdownload/update/v3/
Wireshark · Packet 10206 · http_192.168.1.126.pcapng
Frame 10206: 306 bytes on wire (2448 bits), 306 bytes captured (2448 bits) on interface 0
Ethernet II, Src: HewlettP_8c:17:09 (18:60:24:8c:17:09), Dst: Cisco_cf:66:d1 (e8:b7:48:cf:66:d1)
Internet Protocol Version 4, Src: 172.16.22.2, Dst: 124.225.165.132
Transmission Control Protocol, Src Port: 2196, Dst Port: 80, Seq: 1, Ack: 1, Len: 252
Source Port: 2196
Destination Port: 80
[Stream index: 108]
[TCP Segment Len: 252]
Sequence number: 1 (relative sequence number)


102… 52.141024 172.16.22.2 202.116.81.74 HTTP 324 GET /cache/7/01/www.downloa
Wireshark · Packet 10213 · http_192.168.1.126.pcapng
Frame 10213: 324 bytes on wire (2592 bits), 324 bytes captured (2592 bits) on interface 0
Ethernet II, Src: HewlettP_8c:17:09 (18:60:24:8c:17:09), Dst: Cisco_cf:66:d1 (e8:b7:48:cf:66:d1)
Internet Protocol Version 4, Src: 172.16.22.2, Dst: 202.116.81.74
Transmission Control Protocol, Src Port: 2197, Dst Port: 80, Seq: 1, Ack: 1, Len: 270
Source Port: 2197
Destination Port: 80
[Stream index: 109]
[TCP Segment Len: 270]
Sequence number: 1 (relative sequence number)



（2）在捕获的 HTTP 报文中，选择一个 HTTP 请求报文和对应的 HTTP 应答报文，按图 2-8 分析它们的字段，并将分析结果填入表 2-4 和表 2-5 中

表 2-4 HTTP 请求报文

方法	GET	版本	HTTP/1.1	URI	http://e.543891.com/
首部字段名	字段值	字段所表达的信息			
Accpet	Text/html,application/xhtml+xml,*/*	客户端可是别的内容类型			
Accept-Language	zh-CN	客户端能解释的语言种类-简体中文			
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko	客户机浏览器的版本号及操作系统等信息			
Connection	Keep-Alive	连接方式-持久连接			
Accept-Encoding	gzip, deflate	浏览器接收的编码格式，默认 gzip 压缩格式			

表 2-5 HTTP 应答报文

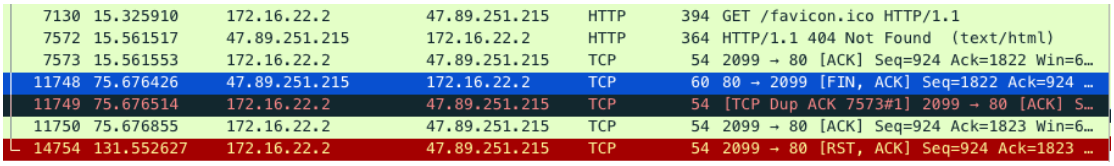
版本	HTTP/1.1	状态码	200	短语	OK
首部字段名	字段值	字段所表达的信息			
Server	nginx\r\n	服务器名称			
Date	Thu, 15 Nov 2018 07:31:45 GMT	时间信息:2018 年 9 月 15 日，星期四，07：31：45，说明报文是什么时候创建的			
Content-Type	text/html	服务器告诉浏览器，自己需要响应的内容形式：文本			
Content-Length	970	内容长度：970			
Last-Modified	Tue, 13 Mar 2018 03:29:18 GMT	最后一次被修改的时间和日期			
Connection	keep-alive	连接方式-持久连接			
ETag	"5aa7458e-3ca"	进程实体的标记			
Accept-Ranges	bytes	服务器可以接受的范围类型			

（3）综合分析捕获的报文，理解 HTTP 协议的工作过程，将结果填入表 2-6 中。

表 2-6　HTTP 协议的工作过程

客户机端口号	服务器端口号	所包括的报文号	工作过程
2099	80	623、634、635	客户机和服务器建立连接-三次握手
2099	80	652、653、654	客户机发送请求
80	2099	655、656	服务器响应请求
80	2099	11748、11750 11754（异常）	TCP 释放连接-两次握手后出现 RST=1 终止释连接。

但是我们没有出现理想的四次握手释放连接的过程，我们仔细查看了这段报文：



7130：客户端发出 GET 指令，请求数据。

7572：服务器返回响应，但是中间没有 TCP 传输，分析响应报文，可以看出“404 Not Found“，说明服我们有可能输入了一个错误的地址链接、服务器将过期的页面删除或者访问不到 DNS 阻止的服务器。

7573：数据传输，客户端想要传一条数据至服务器。

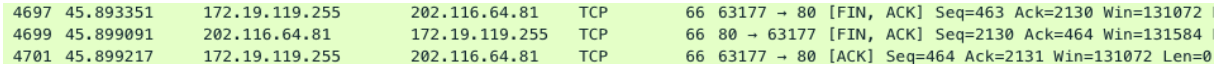
11748：从时间上可以看出，上一条是 15s，这条是 75s，服务器发出终止连接的请求，FIN=1。

11749：这条黑色的是 7573 客户端到服务器没有传输成功，再次重传。

11750：客户端收到并返回一条数据，第二次握手。

14754：这条数据和上一条也差了 60s，而且出现 RST=1，说明等待了很久，服务器一直没有作出应答，那么释放连接失败，直接关闭了连接，没有成功进行四次握手。

所以我再次自己抓取了 http://sysu.edu.cn 的包进行分析：



4967：客户端向服务器发送 FIN=1，请求释放连接，

4699：服务器向客户端发送 FIN，同意释放连接，这里服务器没有发回客户端“确认释放连接的请求”报文信息，我猜想是：客户端不需要服务器的确认释放而直接释放了 TCP 连接，客户到服务器的连接断开。然后服务器直接发回了 FIN=1，请求释放服务器与客户端的连接

4701：最后一条是客户端接收到 FIN 报文之后对服务器发送“请求释放服务器和客户端的连接确认”报文，释放完成，“四次握手”简化成“三次握手”。

（4）第 1 个和第 3 个 HTTP 会话中，Web 服务器对 Web 客户端 GET 请求的响应是什么。

答：第一个 HTTP 会话：

HTTP/1.1 200 OK\r\n

说明连接已经建立，200 表示正常，OK 表示响应成功。

第三个 HTTP 会话：

HTTP/1.1 404 Not Found\r\n

404 说明请求的资源不存在，访问失败

【实验思考】

（1）实验中哪台计算机启动了 HTTP 会话？是如何启动的？

答：176.16.22.2 启动 HTTP 会话。通过 TCP 三次握手建立连接。

（2）哪台计算机首先发出了结束 HTTP 会话的信号，是如何发出的？

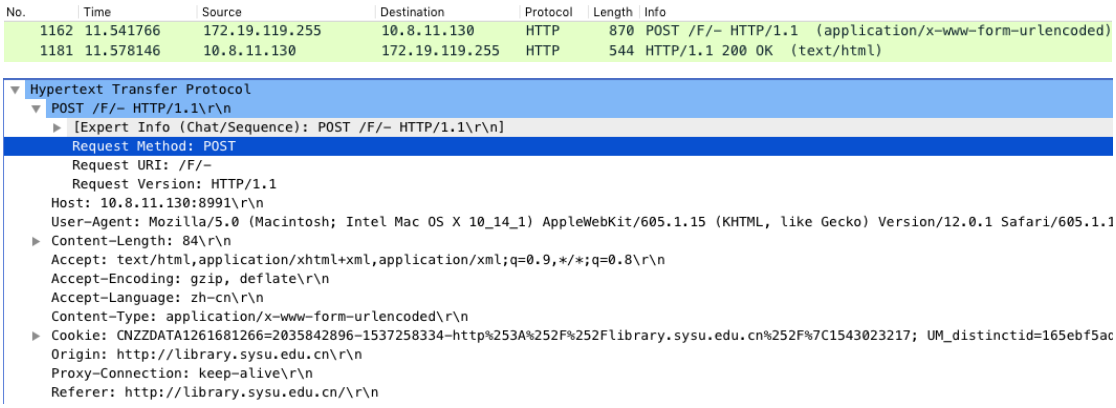
答：47.89.251.215 结束 HTTP 会话。远程的服务器 80 端口发出的结束 HTTP 会话的信号，标志是 FIN=1。

（3）GET 方法取回由 Request-URI 标识的信息，POST 方法可以用于提交表单。请寻找一个有表单提交特征的网页，访问该网页，捕获数据包并分析请求方法中的 GET 和 POST 方法。

答：之前已经分析过 GET 方法，用于取回网站的信息，下面分析两个 POST 方法：

尝试进入中山大学图书馆登录界面：

1）



No.	Time	Source	Destination	Protocol	Length	Info
1162	11.541766	172.19.119.255	10.8.11.130	HTTP	870	POST /F/- HTTP/1.1 (application/x-www-form-urlencoded)
1181	11.578146	10.8.11.130	172.19.119.255	HTTP	544	HTTP/1.1 200 OK (text/html)

```
▼ Hypertext Transfer Protocol
  ▼ POST /F/- HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): POST /F/- HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /F/-
      Request Version: HTTP/1.1
    Host: 10.8.11.130:8991\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_1) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0.1 Safari/605.1.1
  ▶ Content-Length: 84\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-cn\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  ▶ Cookie: CNZZDATA1261681266=2035842896-1537258334-http%253A%252F%252Flibrary.sysu.edu.cn%252F%7C1543023217; UM_distinctid=165ebf5ac
    Origin: http://library.sysu.edu.cn\r\n
    Proxy-Connection: keep-alive\r\n
    Referer: http://library.sysu.edu.cn/\r\n
```

```
    Upgrade-Insecure-Requests: 1\r\n
    X-Lantern-Version: 4.9.0\r\n
    \r\n
    [Full request URI: http://10.8.11.130:8991/F/-]
    [HTTP request 1/1]
    [Response in frame: 1181]
    File Data: 84 bytes
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▶ Form item: "func" = "login-session"
  ▶ Form item: "login_source" = "bor-info"
  ▶ Form item: "bor_library" = "ZSU50"
  ▶ Form item: "bor_id" = ""
  ▶ Form item: "bor_verification" = ""
```

- （抓取数据包 11s 之后进入该网站）加载网页的时候，使用 POST 提交数据，并使用 application/x-www-form-urlencoded 方式，说明请求体内容是纯文本或者纯二进制。
- POST 的内容为：/F/-，但是无法理解和找到这个 post 的含义，按照我的猜测，这是一个空提交，因为浏览器的智能记忆性或者确保连接有效，或者其他原因：在加载网页的一开始会预先 POST 一遍。

```
  1247 11.764429    172.19.119.255   106.11.92.6      HTTP   484 GET /app.gif?&cna=46p/FOzhQCICAXjsrquFGAQf HTTP/1.1
  1252 11.804227    106.11.92.6      172.19.119.255   HTTP   473 HTTP/1.1 200 OK  (GIF89a)
  3088 37.018750    172.19.119.255   10.8.11.130      HTTP   962 POST /F/3QIU7H6GYLQTSGSMQSUIN5VEXIYUHYIQFDNJHL9Q172TC5R1QM-32387 HTTP/1.1  (application/…
  3108 37.046820    10.8.11.130      172.19.119.255   HTTP   560 HTTP/1.1 200 OK  (text/html)
```

```
▼ Hypertext Transfer Protocol
  ▼ POST /F/3QIU7H6GYLQTSGSMQSUIN5VEXIYUHYIQFDNJHL9Q172TC5R1QM-32387 HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): POST /F/3QIU7H6GYLQTSGSMQSUIN5VEXIYUHYIQFDNJHL9Q172TC5R1QM-32387 HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /F/3QIU7H6GYLQTSGSMQSUIN5VEXIYUHYIQFDNJHL9Q172TC5R1QM-32387
      Request Version: HTTP/1.1
    Host: 10.8.11.130:8991\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_1) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0.1 Safari/605.1.1
  ▶ Content-Length: 123\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: zh-cn\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
  ▶ Cookie: CNZZDATA1261681266=2035842896-1537258334-http%253A%252F%252Flibrary.sysu.edu.cn%252F%7C1543023217; UM_distinctid=165ebf5ac
    Origin: http://10.8.11.130:8991\r\n
    Proxy-Connection: keep-alive\r\n
    Referer: http://10.8.11.130:8991/F/-\r\n
    Upgrade-Insecure-Requests: 1\r\n
    X-Lantern-Version: 4.9.0\r\n
    \r\n
    [Full request URI: http://10.8.11.130:8991/F/3QIU7H6GYLQTSGSMQSUIN5VEXIYUHYIQFDNJHL9Q172TC5R1QM-32387]
    [HTTP request 1/1]
    [Response in frame: 3108]
    File Data: 123 bytes
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▶ Form item: "func" = "login-session"
  ▶ Form item: "login_source" = "bor-info"
  ▶ Form item: "bor_id" = "16308015"
  ▶ Form item: "bor_verification" = "[redacted]"
  ▶ Form item: "verifyCode" = "mvf0"
  ▶ Form item: "bor_library" = "ZSU50"
```

- （37s 是输入完账号和密码之后点击登录）捕获的第一个就是 POST 方法。POST 的内容为：/F/3QIU7H6GYLQTSGSMQSUIN5VEXIYUHYIQFDNJHL9Q172TC5R1QM-32387，代表一份文字表单。

  我们可以从下面的 HTML_form_URL_Encode 格式看到：我们输入的数据全部都被发送了

  bor_id = 学号

  bor_verification = 密码

  verifyCode = 验证码

  说明表单发送成功。

（二）FTP 协议分析

（1） FTP 报文格式分析

源 IP 地址	222.200.181.161	源端口	21
目的 IP 地址	172.16.22.2	目的端口	2038
FTP 字段	字段值	字段表示的信息	
Response code	220	Service ready for new user	
Response Arg	FileZilla Server 0.9.60 beta		

```
▷ Frame 1135: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface 0
▷ Ethernet II, Src: Cisco_cf:66:d1 (e8:b7:48:cf:66:d1), Dst: HewlettP_8c:17:09 (18:60:24:8c:17:09)
▷ Internet Protocol Version 4, Src: 222.200.181.161, Dst: 172.16.22.2
▷ Transmission Control Protocol, Src Port: 21, Dst Port: 2038, Seq: 1, Ack: 1, Len: 100
▷ File Transfer Protocol (FTP)
```

(2)FTP 指令与响应过程分析

过程	指令/响应	报文号（相对）	报文信息
User	Request	1	User net2018
	Response	101	331 Password required for net2018
Password	Request	15	Pass net2018
	Response	136	230 Logged on
Quit	Request	29	Quit
	Response	151	221 Goodbye

	Time	Source	Destination	Protocol	Length	Info
1135	25.878013	222.200.181.161	172.16.22.2	FTP	154	Response: 220-FileZilla Server 0.9.60 beta
1489	32.196561	172.16.22.2	222.200.181.161	FTP	68	Request: USER net2018
1490	32.198602	222.200.181.161	172.16.22.2	FTP	89	Response: 331 Password required for net2018
1661	35.540534	172.16.22.2	222.200.181.161	FTP	68	Request: PASS net2018
1662	35.542456	222.200.181.161	172.16.22.2	FTP	69	Response: 230 Logged on
2210	52.692286	172.16.22.2	222.200.181.161	FTP	60	Request: QUIT
2211	52.697398	222.200.181.161	172.16.22.2	FTP	67	Response: 221 Goodbye

(3)FTP 传送过程中的报文

报文类型	所包括的报文序号	客户端口	服务器端口
控制连接的建立	1	2057	21
数据连接的建立	69	2057	21
FTP 数据传送	408	62836	2058
FTP 指令传送和响应	1	2057	21
数据连接的释放	440	62836	2058
控制连接的释放	442	2057	21

1803 45.271753	172.16.22.2	222.200.181.161	TCP	66 2057 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1804 45.272952	222.200.181.161	172.16.22.2	TCP	66 21 → 2057 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1805 45.273031	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0
1806 45.274723	222.200.181.161	172.16.22.2	FTP	154 Response: 220-FileZilla Server 0.9.60 beta
1807 45.275402	172.16.22.2	222.200.181.161	FTP	68 Request: USER net2018
1808 45.276965	222.200.181.161	172.16.22.2	FTP	89 Response: 331 Password required for net2018
1809 45.277471	172.16.22.2	222.200.181.161	FTP	68 Request: PASS net2018
1810 45.278992	222.200.181.161	172.16.22.2	FTP	69 Response: 230 Logged on
1811 45.279402	172.16.22.2	222.200.181.161	FTP	60 Request: SYST
1812 45.280859	222.200.181.161	172.16.22.2	FTP	86 Response: 215 UNIX emulated by FileZilla
1813 45.281336	172.16.22.2	222.200.181.161	FTP	59 Request: PWD
1814 45.282847	222.200.181.161	172.16.22.2	FTP	85 Response: 257 "/" is current directory.
1815 45.283194	172.16.22.2	222.200.181.161	FTP	62 Request: TYPE I
1816 45.284732	222.200.181.161	172.16.22.2	FTP	73 Response: 200 Type set to I
1817 45.285085	172.16.22.2	222.200.181.161	FTP	62 Request: SIZE /
1818 45.286645	222.200.181.161	172.16.22.2	FTP	74 Response: 550 File not found
1819 45.287742	172.16.22.2	222.200.181.161	FTP	61 Request: CWD /
1820 45.289254	222.200.181.161	172.16.22.2	FTP	101 Response: 250 CWD successful. "/" is current directory.
1821 45.290101	172.16.22.2	222.200.181.161	FTP	60 Request: PASV
1822 45.291719	222.200.181.161	172.16.22.2	FTP	107 Response: 227 Entering Passive Mode (222,200,181,161,245,116)
1823 45.292148	172.16.22.2	222.200.181.161	TCP	66 2058 → 62836 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1824 45.293351	222.200.181.161	172.16.22.2	TCP	66 62836 → 2058 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1825 45.293391	172.16.22.2	222.200.181.161	TCP	54 2058 → 62836 [ACK] Seq=1 Ack=1 Win=65536 Len=0
1826 45.293651	172.16.22.2	222.200.181.161	FTP	63 Request: LIST -l
1827 45.295400	222.200.181.161	172.16.22.2	FTP	109 Response: 150 Opening data channel for directory listing of "/"
1828 45.296049	222.200.181.161	172.16.22.2	FTP-DA…	1514 FTP Data: 1460 bytes
1829 45.296050	222.200.181.161	172.16.22.2	FTP-DA…	525 FTP Data: 471 bytes
1830 45.296050	222.200.181.161	172.16.22.2	TCP	60 62836 → 2058 [FIN, ACK] Seq=1932 Ack=1 Win=65536 Len=0
1831 45.296051	222.200.181.161	172.16.22.2	FTP	88 Response: 226 Successfully transferred "/"
1832 45.296086	172.16.22.2	222.200.181.161	TCP	54 2058 → 62836 [ACK] Seq=1 Ack=1932 Win=65536 Len=0
1833 45.296091	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [ACK] Seq=78 Ack=442 Win=65024 Len=0
1834 45.296184	172.16.22.2	222.200.181.161	TCP	54 2058 → 62836 [ACK] Seq=1 Ack=1933 Win=65536 Len=0
1835 45.296628	172.16.22.2	222.200.181.161	TCP	54 2058 → 62836 [RST, ACK] Seq=1 Ack=1933 Win=0 Len=0
1836 45.296747	172.16.22.2	222.200.181.161	FTP	60 Request: QUIT
1837 45.298216	222.200.181.161	172.16.22.2	FTP	67 Response: 221 Goodbye
1838 45.298508	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [FIN, ACK] Seq=84 Ack=455 Win=65024 Len=0
1839 45.298869	222.200.181.161	172.16.22.2	TCP	60 21 → 2057 [FIN, ACK] Seq=455 Ack=84 Win=65536 Len=0
1840 45.298952	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [RST, ACK] Seq=85 Ack=456 Win=0 Len=0
2065 49.348421	172.16.22.2	203.208.40.64	TCP	55 [TCP Keep-Alive] 1866 → 443 [ACK] Seq=1 Ack=1 Win=257 Len=1
2066 49.374075	203.208.40.64	172.16.22.2	TCP	66 [TCP Keep-Alive ACK] 443 → 1866 [ACK] Seq=1 Ack=2 Win=115 Len=0 SLE=1 SRE=2
2632 63.089692	172.16.22.2	222.200.181.161	TCP	66 2059 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2633 63.091133	222.200.181.161	172.16.22.2	TCP	66 21 → 2059 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
2634 63.091217	172.16.22.2	222.200.181.161	TCP	54 2059 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0

(4)从协议层面分析 FTP-DOS 与 FTP-WEB 的异同

其他方面都是相同的，除了 FTP-DOS 使用的是 PORT 模式。而 FTP_WEB 使用的是 PASV 模式，因为从数据包看出，FTP-DOS 的数据连接是建立在服务器端口 20 上的，而 FTP-WEB 是建立在随机生成的一个端口。

(5)在步骤 5 中，FTP 的匿名账户是什么？

账户是：anonymous

密码是：chrome@example.com

2635 63.092959	222.200.181.161	172.16.22.2	FTP	154 Response: 220-FileZilla Server 0.9.60 beta
2636 63.093372	172.16.22.2	222.200.181.161	FTP	70 Request: USER anonymous
2637 63.094966	222.200.181.161	172.16.22.2	FTP	91 Response: 331 Password required for anonymous
2638 63.095554	172.16.22.2	222.200.181.161	FTP	79 Request: PASS chrome@example.com

(6)结合过程分析 TCP 连接的建立与终止连接的过程

(1)叙述 TCP 连接建立的三次握手过程，四次握手终止的过程

客户端向服务器发送一个请求建立连接，服务器返回一个确认信息建立连接，然后客户端向服务器发送一个消息确认收到确认连接的信息。

客户端向服务器端发送一个请求关闭连接，服务器端返回一个确认信息确认关闭连接，并向客户端发送一个信息确认是否关闭连接，客户端最后向服务端发送一个确认信息关闭连接。

三次握手过程:

1803 45.271753	172.16.22.2	222.200.181.161	TCP	66 2057 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
1804 45.272952	222.200.181.161	172.16.22.2	TCP	66 21 → 2057 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1805 45.273031	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [ACK] Seq=1 Ack=1 Win=65536 Len=0

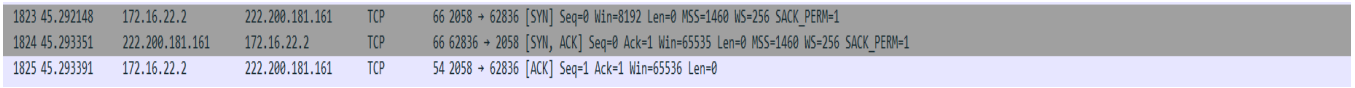
四次终止过程:

1838 45.298508	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [FIN, ACK] Seq=84 Ack=455 Win=65024 Len=0
1839 45.298869	222.200.181.161	172.16.22.2	TCP	60 21 → 2057 [FIN, ACK] Seq=455 Ack=84 Win=65536 Len=0
1840 45.298952	172.16.22.2	222.200.181.161	TCP	54 2057 → 21 [RST, ACK] Seq=85 Ack=456 Win=0 Len=0
2065 49.348421	172.16.22.2	203.208.40.64	TCP	55 [TCP Keep-Alive] 1866 → 443 [ACK] Seq=1 Ack=1 Win=257 Len=1
2066 49.374075	203.208.40.64	172.16.22.2	TCP	66 [TCP Keep-Alive ACK] 443 → 1866 [ACK] Seq=1 Ack=2 Win=115 Len=0 SLE=1 SRE=2

(2)从捕获的数据包分析三次握手的过程，四次挥手终止连接的过程

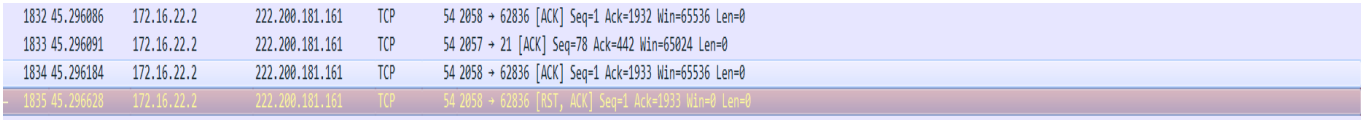
从这三个数据包可以看出，客户端首先向服务端发送一个 SYN 请求包，然后服务端返回一个 SYN+ACK 的应答，最后客户端返回一个 ACK 应答。

三次握手过程:



从这四个数据包可以看出，客户端首先向服务端发送一个 FIN+ACK 的请求包，然后服务器端也返回一个 FIN+ACK 的请求包，接着客户端返回给服务端一个 ACK 的应答包确认，然后服务器端也返回给客户端一个 ACK 应答包确认。

四次终止过程:



【实验思考】

1、 分析 FTP 使用的两条 TCP 连接，具体指出哪些情况下使用数据连接，那些情况下使用控制连接。

当登入的时候或者结束连接的时候，都是控制连接，就是凡是涉及到控制要求的都是控制连接，而使用 dir 命令，或者再下载一个文件时就使用数据连接。

2、 比较 FTP 协议与 HTTP 协议

HTTP 协议是无状态的,所以上一个请求与下一个请求无法确认是同一个用户发送的,所以无法维持请求的状态信息，而 FTP 协议是有状态的，在请求之前建立一个永久的连接,然后每次请求都是通过这个连接发送的,所以 FTP 能获得每个用户的状态信息。

3、 讨论 FTP 协议的安全性

由于 FTP 没有对信息进行任何的加密，所以信息是明文传输的，如果能够截取到信息，就可以直接得到用户的账号和密码。

4、 启用 FTP SERVER 的口令安全，通过捕获数据包分析它能否保证用户名和口令的安全？

不能了，因为已经对传输的报文经过了压缩与加密。

5、 同一台主机，既是客户端，又是服务端，如何捕获数据包？

在主机上建立一个服务端，连接的时候输入客户端 ip 地址，即主机地址。服务端地址为回送地址，127.0.0.1。该地址指本地机，一般用于测试使用，接着进行在 wireshark 软件中捕获报文。

（三）Telnet 协议分析

（1）TCP 连接建立后的第一个 Telnet 协议数据报的功能是进行选项协商的吗？在这个数据报中对哪些选项进行了协商，列出它们的选项名和选项代码。

```
5213 136.886484  172.16.22.2  172.16.22.3  TCP  66 2250 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5216 136.887854  172.16.22.3  172.16.22.2  TCP  66 23 → 2250 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5217 136.887962  172.16.22.2  172.16.22.3  TCP  54 2250 → 23 [ACK] Seq=1 Ack=1 Win=65536 Len=0
```

答：如上图，在经过 TCP 连接后的第一个 Telnet 协议数据报的功能是进行选项协商的，因为 Telnet 使用一条 TCP 连接，不像 FTP 使用 TCP 两个连接，所以 Telnet 连接时，要进行选项协商，一般出现的协商选项包括回显（1），抑制继续进行（3），终端类型（24），窗口大小（31），终端速率（32），远程流量控制（33），行方式（34），环境变量（36）等等，而在这个数据报中，出现了以下几种协商选项：Echo（1），Authentication option（25），suppress go ahead（03），new environment option（27），negotiate about window size（31），Binary transmission（0）。

可以看到下面是第一条报文的信息：

```
5213 136.886484  172.16.22.2  172.16.22.3  TCP     66 2250 → 23 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5216 136.887854  172.16.22.3  172.16.22.2  TCP     66 23 → 2250 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5217 136.887962  172.16.22.2  172.16.22.3  TCP     54 2250 → 23 [ACK] Seq=1 Ack=1 Win=65536 Len=0
5225 136.906937  172.16.22.3  172.16.22.2  TELNET  75 Telnet Data ...
```

```
▷ Frame 5225: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0
▷ Ethernet II, Src: HewlettP_8c:17:04 (18:60:24:8c:17:04), Dst: HewlettP_8c:17:09 (18:60:24:8c:17:09)
▷ Internet Protocol Version 4, Src: 172.16.22.3, Dst: 172.16.22.2
▷ Transmission Control Protocol, Src Port: 23, Dst Port: 2250, Seq: 1, Ack: 1, Len: 21
◢ Telnet
  ◢ Do Authentication Option
      Command: Do (253)
      Subcommand: Authentication Option
  ◢ Will Echo
      Command: Will (251)
      Subcommand: Echo
  ◢ Will Suppress Go Ahead
      Command: Will (251)
      Subcommand: Suppress Go Ahead
  ◢ Do New Environment Option
      Command: Do (253)
      Subcommand: New Environment Option
  ◢ Do Negotiate About Window Size
      Command: Do (253)
      Subcommand: Negotiate About Window Size
  ◢ Do Binary Transmission
      Command: Do (253)
      Subcommand: Binary Transmission
  ◢ Will Binary Transmission
      Command: Will (251)
      Subcommand: Binary Transmission
```

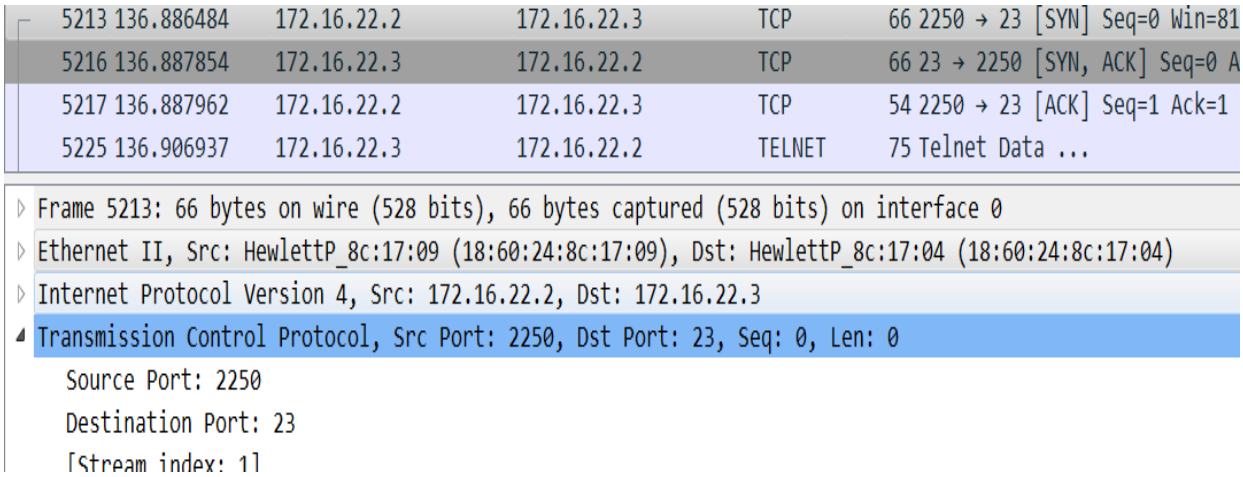
（2） 分析上述报文，写出所有选项的格式并指出格式中每一部分的意义，填入下表：

请求类型	请求类型代码	选项名称	选项代码	意义
DO	253	Authentication option	25	身份验证
WILL	251	Echo	1	回显
WILL	251	Suppress go ahead	3	抑制前进
DO	253	New	27	新环境选项

		environment option		
DO	253	Negotiate about window size	31	窗口尺寸选项
DO	253	Binary transmission	0	二进制传输选项
WILL	251	Binary transmission	0	二进制传输选项

（3） 在 TCP 连接时，Telnet 使用的端口号是多少？

答：源端口号是 2250，目的端口号是 23。如图



（4） 从 TCP 连接建立后开始分析捕获的报文，填写下表，Telnet 数据传输只填写客户端输入命令的传输报文：

过程	报文号	功能	信息以及参数	报文作用
选项协商	5225	选项协商	253（DO）——25（authentication） 251（WILL）——1（Echo） 251——3 253——27 等等	TCP 连接后的选项协商，包括窗口、回显、抑制前进、环境等等选项的请求允许
	5228	选项协商	Will authentication option	同意身份验证
	5229	选项协商	Suboption end	结束协商命令
	5230	选项协商	Do——Echo Do — — Suppress go ahead 等等	允许之前的协商请求
	5231	选项协商	250——27——01 240 等等	回应环境选项变量，子选项结束
	5528	选项协商	250——27——00 240 等等	子选项结束，回应环境选项变量
	5529	选项协商	250——25——1	拒绝身份验

			240	证、子选项结束
	5530	选项协商	250——27 240	子选项环境变量 子选项结束
	5535	选项协商	250——0——2 240	接收身份认证 子选项结束
数据传输	5794	数据传输	\r\n	回车
	5795	数据传输	\n \rlogin:	请求输入用户名
	6999	数据传输	John	输入用户名
	7137	数据传输	\r\n	回车
	7138	数据传输	\n \ rpassword:	请求输入密码
	7167	数据传输	123456	输入密码
	7215	数据传输	\r\n	回车

（5） 类似于 Telnet-Dos 的分析，分析 Telnet-Web 的报文，指出两者区别。

对于 Telnet 的 Web 方式报文，协议的过程与 Dos 的连接基本一致的，只是 Dos 方式的端口是固定的，对于服务器的端口总是取到 23，而对于 Web 行式的端口是随机分配的。

（6） 与远程桌面连接，Telnet 的连接方式与其有什么区别？

Telnet 的传输过程中，我们可以清晰地看到每一步的传输内容，它是一个明文传输，可以清楚看到数据，但是不安全，保密性不够，相比于远程桌面连接，它也没有图形用户界面，只有纯文字界面。

同时 Telnet 使用的是 TCP 协议传输，而远程桌面连接使用的是 RDP 协议，这个协议是位于 TCP\IP 之上的，速度非常快，在各方面的性能都优于 Telnet。

（7） Telnet 的口令是否为明文传输

答：是的

本次实验完成后，请根据组员在实验中的贡献，请实事求是，自评在实验中应得的分数。（按百分制）

学号	学生	自评分
16308073	刘渤	100
16308161	邹紫婧	100
16308091	彭肖文	100
16308015	陈瑞佳	100

计算机网络实验报告

【交实验报告】

上传实验报告：ftp://222.200.181.161/　　截止日期（不迟于）：1 周之内

上传包括两个文件：

（1）小组实验报告。上传文件名格式：小组号_ Ftp 协议分析实验.pdf （由组长负责上传）

例如: 文件名“10_ Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告

（2）小组成员实验体会。每个同学单独交一份只填写了实验体会的实验报告。只需填写自己的学号和姓名。

文件名格式：小组号_学号_姓名_ Ftp 协议分析实验.pdf （由组员自行上传）

例如: 文件名“10_05373092_张三_ Ftp 协议分析实验.pdf”表示第 10 组的 Ftp 协议分析实验报告。

注意：不要打包上传！