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	数据科学与计算机学院	班 级	16 级计科教务 2 班	组长	钟哲灏
学号	16337331	16337327	16337341		
学生	钟哲灏	郑映雪	朱志儒		

实验分工			
钟哲灏	辅助实验、资料查找、主要负责数据分析，自评 96 分	朱志儒	主要负责抓包和建立 ftp，参与数据分析并撰写实验报告中 ftp 部分，自评 96 分
郑映雪	辅助实验、数据分析、主要负责撰写实验报告中 http 和 telnet 部分及负责实验报告总体排版，自评 96 分		

【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】

（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.回答实验提出的问题。（P72）

【实验要求】

一些重要的信息需给出截图，注意实验前后的对比。

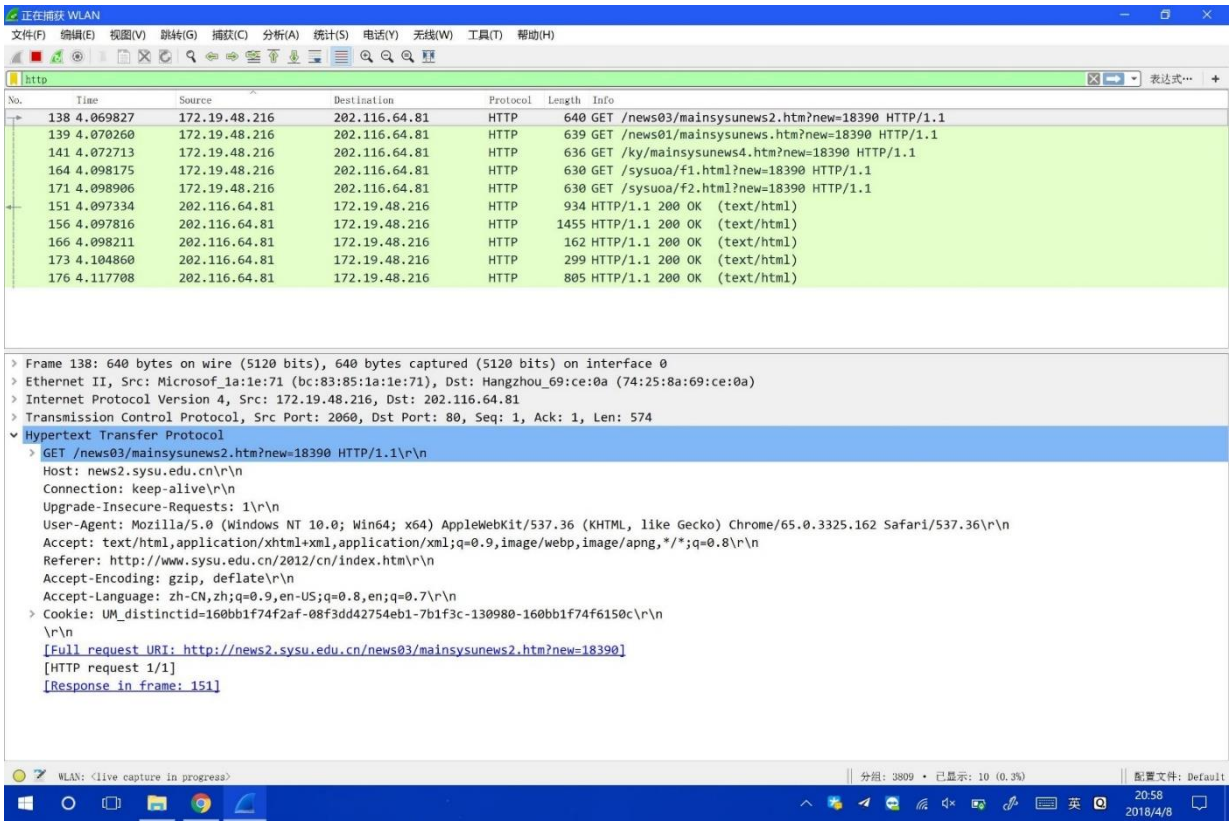
【实验记录】

一、HTTP 协议分析：

我们访问校园官网 sysu.edu.cn 进行抓包并将相关内容截图演示在下列的问题解答中。

（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连接？服务器和客户机分别使用了哪些端口？

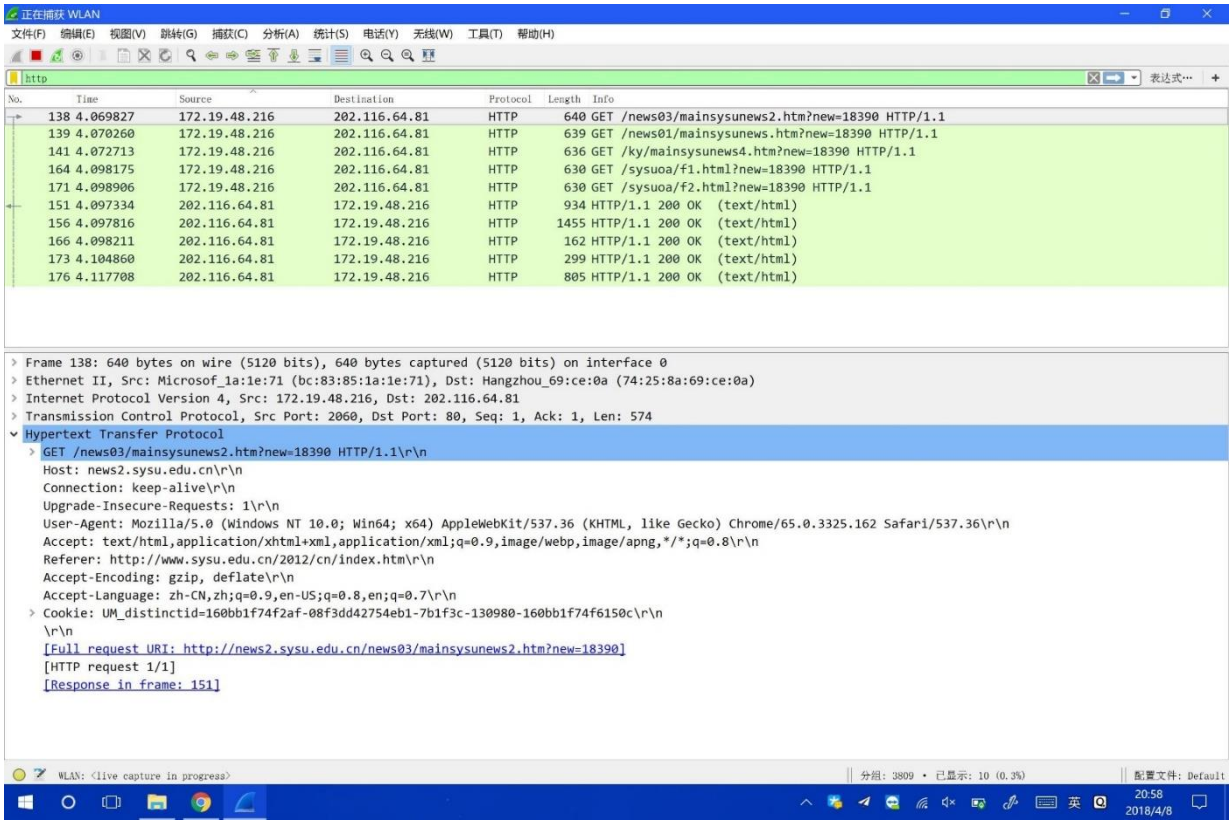
答：我们将捕获到的报文进行分析，截图如下：



我们将捕获的报文限制为 http，发现只有两种报文：GET 和 OK 报文，客户机与服务器建立了 5 个链接。其中服务器端口为 80，客户机的端口为 2059、2060、2061、2076、2077。

（2）在捕获的 HTTP 报文中，选择一个 HTTP 请求报文和 HTTP 应答报文，分析它们的字段，并将分析结果填表。

答：①请求报文。我们选择如下图所示的报文：



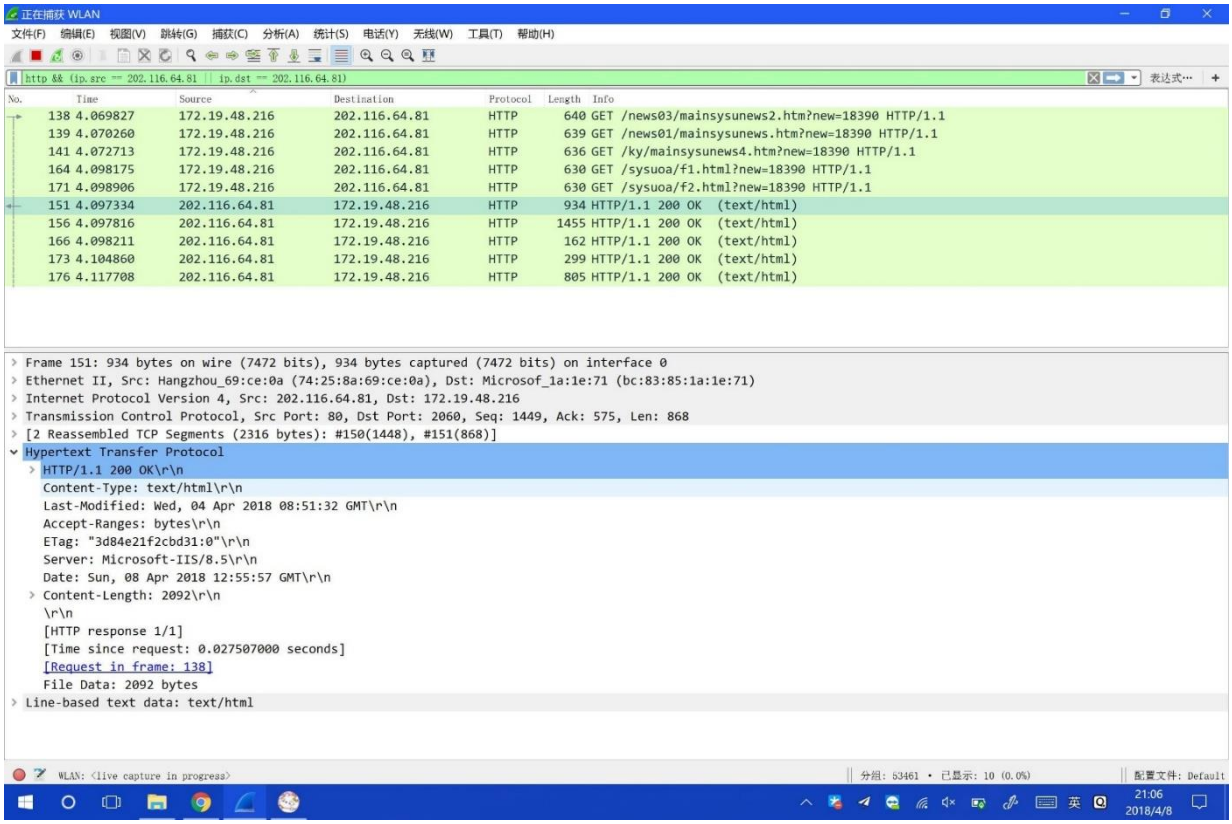
选择第一条请求报文，下方可以看到具体的报文信息。我们可以根据具体的信息填写如下的表格：

HTTP 请求报文

方法	GET	版本	1.1	URL	/news03/mainsysunews2.htm?new=18390
首部字段名	**字段值**	**字段所表达的信息**			
Host	new2.sysu.edu.cn	主机进行链接			
Connection	keep-alive	两者之间保持连接			
Upgrade_Insecure_Requests	1	用于让浏览器自动升级请求从 http 到 https，用于大量包含 http 资源的 http 网页直接升级到 https 而不会报错。			
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.162 Safari/537.36	用户代理，简称 UA，它是一个特殊字符串头，使得服务器能够识别客户使用的操作系统及版本、CPU 类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等。			

……

②应答报文。我们分析如下的报文：



由图中所示的报文信息显示的端口号可以得知它为上一条报文的应答报文，我们根据报文可以填写下表：

HTTP 应答报文

版本	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**			
Contect-Type	text/html	主要是用来指明报文主体部分内容属于何种类型，比如 html，json 或者 xml 等等。			
Last-Modified	Wed,04 Apr 2018 08:51:32 GMT	最近的修改时间。			
Accept-Ranges	bytes	表明服务器是否支持指定范围请求及哪种类型的分段请求，这里支持 bytes 类型的分段请求。			
ETag	“3d84e21f2cbd31:0”	被请求变量的实体值。			

（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。

答：我们将五对请求-应答报文结合在一起分析，并按照报文内容依次列出客户机端口号、服务器端口号、报文号、工作过程：

```
Transmission Control Protocol, Src Port: 2060, Dst Port: 80, Seq: 1, Ack: 1, Len: 574
    Source Port: 2060
    Destination Port: 80
    [Stream index: 7]
    [TCP Segment Len: 574]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 575    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

```
Transmission Control Protocol, Src Port: 80, Dst Port: 2060, Seq: 1449, Ack: 575, Len: 868
    Source Port: 80
    Destination Port: 2060
    [Stream index: 7]
    [TCP Segment Len: 868]
    Sequence number: 1449    (relative sequence number)
    [Next sequence number: 2317    (relative sequence number)]
    Acknowledgment number: 575    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

上面是第一对请求-应答报文，由报文内容可填写表格的第一行。

```
Transmission Control Protocol, Src Port: 2059, Dst Port: 80, Seq: 1, Ack: 1, Len: 573
    Source Port: 2059
    Destination Port: 80
    [Stream index: 8]
    [TCP Segment Len: 573]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 574    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

```
Transmission Control Protocol, Src Port: 80, Dst Port: 2059, Seq: 1449, Ack: 574, Len: 1389
    Source Port: 80
    Destination Port: 2059
    [Stream index: 8]
    [TCP Segment Len: 1389]
    Sequence number: 1449    (relative sequence number)
    [Next sequence number: 2838    (relative sequence number)]
    Acknowledgment number: 574    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

上面是第二对请求-应答报文，由报文内容可填写表格的第二行。

```
Transmission Control Protocol, Src Port: 2061, Dst Port: 80, Seq: 1, Ack: 1, Len: 570
    Source Port: 2061
    Destination Port: 80
    [Stream index: 10]
    [TCP Segment Len: 570]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 571    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

```
Transmission Control Protocol, Src Port: 80, Dst Port: 2061, Seq: 1449, Ack: 571, Len: 96
    Source Port: 80
    Destination Port: 2061
    [Stream index: 10]
    [TCP Segment Len: 96]
    Sequence number: 1449    (relative sequence number)
    [Next sequence number: 1545    (relative sequence number)]
    Acknowledgment number: 571    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

上面是第三对请求-应答报文，由报文内容可填写表格的第三行。

```
Transmission Control Protocol, Src Port: 2076, Dst Port: 80, Seq: 1, Ack: 1, Len: 564
    Source Port: 2076
    Destination Port: 80
    [Stream index: 9]
    [TCP Segment Len: 564]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 565    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

```
Transmission Control Protocol, Src Port: 80, Dst Port: 2076, Seq: 1449, Ack: 565, Len: 739
    Source Port: 80
    Destination Port: 2076
    [Stream index: 9]
    [TCP Segment Len: 739]
    Sequence number: 1449    (relative sequence number)
    [Next sequence number: 2188    (relative sequence number)]
    Acknowledgment number: 565    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

上面是第四对请求-应答报文，由报文内容可填写表格的第四行。

```
Transmission Control Protocol, Src Port: 2077, Dst Port: 80, Seq: 1, Ack: 1, Len: 564
    Source Port: 2077
    Destination Port: 80
    [Stream index: 11]
    [TCP Segment Len: 564]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 565    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

```
Transmission Control Protocol, Src Port: 80, Dst Port: 2077, Seq: 1449, Ack: 565, Len: 233
    Source Port: 80
    Destination Port: 2077
    [Stream index: 11]
    [TCP Segment Len: 233]
    Sequence number: 1449    (relative sequence number)
    [Next sequence number: 1682    (relative sequence number)]
    Acknowledgment number: 565    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
```

上面是第五对请求-应答报文，由报文内容可填写表格的第五行。（可以与前面第一张图对比发现这条其实是有延时的响应）

由以上所示的报文信息，我们可以填写如下表格：

客户机端口号	服务器端口号	包括的报文号	工作过程
2060	80	1、1449	客户机向服务器发送数据请求，服务器进行响应回传。
2059	80	1、1449	客户机向服务器发送数据请求，服务器进行响应回传。
2061	80	1、1449	客户机向服务器发送数据请求，服务器进行响应回传。
2076	80	1、1449	客户机向服务器发送数据请求，服务器进行响应回传。
2077	80	1、1449	客户机向服务器发送数据请求，服务器进行响应回传。

（4）在第 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：由第一张总览图可以得知，在本次实验中所有 5 条 GET 请求的响应都是 OK，则第 1 个和第 3 个会话中，响应均为 OK。

【HTTP 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：由第一张总览图得知，是客户机启动的，查阅资料和课本我们得知是靠 TCP 三次握手启动了 HTTP 会话。

（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：在本实验中我们组实际操作中无法知道是谁先发出了结束 HTTP 的会话的信号，因为它显示 keep-alive 状态，所以还没有结束连接。

（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：我们访问新浪博客，尝试提交一篇博文，并在此时开启捕捉。

对于 GET 方式，浏览器会把头部字段和数据一并发送出去，服务器返回数据；而对于 POST，根据我们查阅资料，浏览器先发送头部字段，服务器响应 continue，浏览器再发送数据，服务器返回数据。简单地说，就是 GET 产生一个 TCP 数据包，POST 产生 2 个 TCP 数据包。但要注意有的浏览器对于 POST 也是发送 1 次数据包（比如我们小组在做实验时使用的 Chorme 浏览器……）。

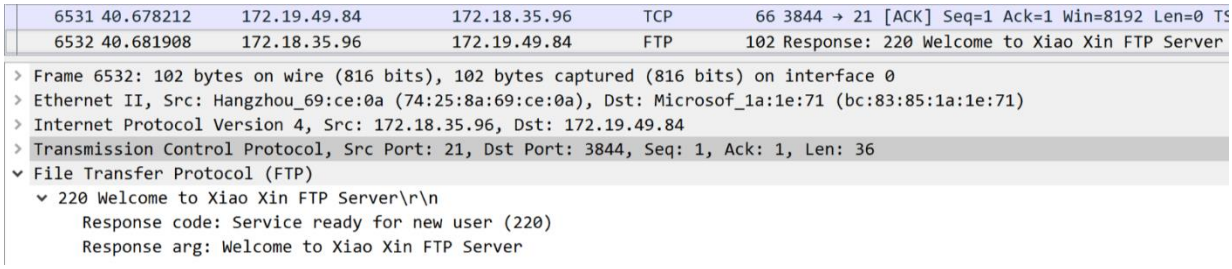
我们的另外一个发现是 POST 多了三个重组的 TCP 报文段，如下图所示。而这是在 GET 报文中没有发现的。

```
> Frame 5185: 606 bytes on wire (4848 bits), 606 bytes captured (4848 bits) on interface 0
> Ethernet II, Src: Microsof_1a:1e:71 (bc:83:85:1a:1e:71), Dst: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a)
> Internet Protocol Version 4, Src: 172.19.48.216, Dst: 219.142.118.113
> Transmission Control Protocol, Src Port: 4785, Dst Port: 80, Seq: 1478, Ack: 1, Len: 552
v [3 Reassembled TCP Segments (2029 bytes): #5183(1460), #5184(17), #5185(552)]
    [Frame: 5183, payload: 0-1459 (1460 bytes)]
    [Frame: 5184, payload: 1460-1476 (17 bytes)]
    [Frame: 5185, payload: 1477-2028 (552 bytes)]
    [Segment count: 3]
    [Reassembled TCP length: 2029]
    [Reassembled TCP Data: 504f5354202f61646d696e2f61727469636c652f61727469...]
> Hypertext Transfer Protocol
> HTML Form URL Encoded: application/x-www-form-urlencoded
```

二、FTP 协议分析

（1）对 FTP-DOS 报文进行分析，找到 TCP 三次握手后的第一个 FTP 报文，分析并填写表

答：我们根据下面的报文内容填写表格：



FTP 报文格式分析

源 IP 地址	172.18.35.96	源端口	21
目的 IP 地址	172.19.49.84	目的端口	3844
FTP 字段	字段值	字段所表达的意思	
Response Code	Service ready for new user	为新用户准备好服务	
Response Arg	Welcome to Xiao Xin FTP Server	欢迎来到小新 FTP 服务器	

（2）在 FTP-DOS 中找出 FTP 指令传送和响应报文，分析并填写表格。

答：根据报文填写下表，报文截图附在此表之后。

FTP 指令和响应过程分析

过程	指令/响应	报文号	报文信息
User	Request	15	Request command: USER Request arg: user
	Response	101	Response code: User name okay, need password (331) Response arg: Password required for user
Password	Request	26	Request command: PASS Request arg: user
	Response	133	Response code: User logged in, proceed(230) Response arg: Logged on

Quit	Request	37	Request command: QUIT
	Response	148	Response code: Service closing control connection (221) Response arg: Goodbye

附截图：

User Request:

```
   7966 49.093935     172.19.49.84          172.18.35.96          FTP         77 Request: USER user
> Frame 7966: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
> Ethernet II, Src: Microsof_1a:1e:71 (bc:83:85:1a:1e:71), Dst: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a)
> Internet Protocol Version 4, Src: 172.19.49.84, Dst: 172.18.35.96
> Transmission Control Protocol, Src Port: 3844, Dst Port: 21, Seq: 15, Ack: 101, Len: 11
v File Transfer Protocol (FTP)
  v USER user\r\n
      Request command: USER
      Request arg: user
```

User Response:

```
   7971 49.118232     172.18.35.96          172.19.49.84          FTP         98 Response: 331 Password required for user
> Frame 7971: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
> Ethernet II, Src: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a), Dst: Microsof_1a:1e:71 (bc:83:85:1a:1e:71)
> Internet Protocol Version 4, Src: 172.18.35.96, Dst: 172.19.49.84
> Transmission Control Protocol, Src Port: 21, Dst Port: 3844, Seq: 101, Ack: 26, Len: 32
v File Transfer Protocol (FTP)
  v 331 Password required for user\r\n
      Response code: User name okay, need password (331)
      Response arg: Password required for user
```

Password Request:

```
   8261 51.873719     172.19.49.84          172.18.35.96          FTP         77 Request: PASS user
> Frame 8261: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0
> Ethernet II, Src: Microsof_1a:1e:71 (bc:83:85:1a:1e:71), Dst: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a)
> Internet Protocol Version 4, Src: 172.19.49.84, Dst: 172.18.35.96
> Transmission Control Protocol, Src Port: 3844, Dst Port: 21, Seq: 26, Ack: 133, Len: 11
v File Transfer Protocol (FTP)
  v PASS user\r\n
      Request command: PASS
      Request arg: user
```

Password Response:

```
   8267 51.918800     172.18.35.96          172.19.49.84          FTP         81 Response: 230 Logged on
> Frame 8267: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
> Ethernet II, Src: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a), Dst: Microsof_1a:1e:71 (bc:83:85:1a:1e:71)
> Internet Protocol Version 4, Src: 172.18.35.96, Dst: 172.19.49.84
> Transmission Control Protocol, Src Port: 21, Dst Port: 3844, Seq: 133, Ack: 37, Len: 15
v File Transfer Protocol (FTP)
  v 230 Logged on\r\n
      Response code: User logged in, proceed (230)
      Response arg: Logged on
```

Quit Request:

```
11782 79.463847    172.19.49.84      172.18.35.96      FTP       72 Request: QUIT
Frame 11782: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
Ethernet II, Src: Microsof_1a:1e:71 (bc:83:85:1a:1e:71), Dst: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a)
Internet Protocol Version 4, Src: 172.19.49.84, Dst: 172.18.35.96
Transmission Control Protocol, Src Port: 3844, Dst Port: 21, Seq: 37, Ack: 148, Len: 6
File Transfer Protocol (FTP)
  QUIT\r\n
      Request command: QUIT
```

Quit Response:

```
11783 79.484682    172.18.35.96      172.19.49.84      FTP       79 Response: 221 Goodbye
Frame 11783: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0
Ethernet II, Src: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a), Dst: Microsof_1a:1e:71 (bc:83:85:1a:1e:71)
Internet Protocol Version 4, Src: 172.18.35.96, Dst: 172.19.49.84
Transmission Control Protocol, Src Port: 21, Dst Port: 3844, Seq: 148, Ack: 43, Len: 13
File Transfer Protocol (FTP)
  221 Goodbye\r\n
      Response code: Service closing control connection (221)
      Response arg: Goodbye
```

（3）对 FTP-WEB 捕获的报文进行综合分析，观察 FTP 协议的工作过程。特别观察两种连接的建立过程和释放过程，以及这两种连接建立和释放的先后顺序，将结果填入表。

答：下面用报文截图表示在此实验中 FTP 协议的工作过程：

①控制连接的建立：

```
4996 27.635104    172.19.49.84      172.18.35.96      TCP       74 4029 → 21 [SYN] Seq=0 Win=64240 Len=0 …
4997 27.641593    172.18.35.96      172.19.49.84      TCP       74 21 → 4029 [SYN, ACK] Seq=0 Ack=1 Win=6…
4998 27.641695    172.19.49.84      172.18.35.96      TCP       66 4029 → 21 [ACK] Seq=1 Ack=1 Win=66560 …
```

②数据连接的建立：

```
5023 27.677347    172.19.49.84      172.18.35.96      TCP       74 4030 → 59243 [SYN] Seq=0 Win=64240 Len…
5024 27.680011    172.18.35.96      172.19.49.84      TCP       74 59243 → 4030 [SYN, ACK] Seq=0 Ack=1 Wi…
5025 27.680124    172.19.49.84      172.18.35.96      TCP       66 4030 → 59243 [ACK] Seq=1 Ack=1 Win=665…
```

③FTP 数据传送：

```
5028 27.685816    172.18.35.96      172.19.49.84      FTP-DATA  1008 FTP Data: 942 bytes
Frame 5028: 1008 bytes on wire (8064 bits), 1008 bytes captured (8064 bits) on interface 0
Ethernet II, Src: Hangzhou_69:ce:0a (74:25:8a:69:ce:0a), Dst: Microsof_1a:1e:71 (bc:83:85:1a:1e:71)
Internet Protocol Version 4, Src: 172.18.35.96, Dst: 172.19.49.84
Transmission Control Protocol, Src Port: 59243, Dst Port: 4030, Seq: 1, Ack: 1, Len: 942
FTP Data (drwxr-xr-x 1 ftp ftp              0 Mar 15 23:05 $RECYCLE.BIN\r\n-r--r--r-- 1 ftp ftp
```

④FTP 指令传送和响应：

```
5000 27.645448    172.19.49.84      172.18.35.96      FTP       77 Request: USER user
5008 27.651242    172.18.35.96      172.19.49.84      FTP       98 Response: 331 Password required for user
5009 27.651522    172.19.49.84      172.18.35.96      FTP       77 Request: PASS user
5010 27.654843    172.18.35.96      172.19.49.84      FTP       81 Response: 230 Logged on
5011 27.655133    172.19.49.84      172.18.35.96      FTP       72 Request: SYST
5012 27.657889    172.18.35.96      172.19.49.84      FTP       98 Response: 215 UNIX emulated by FileZilla
5013 27.658259    172.19.49.84      172.18.35.96      FTP       71 Request: PWD
5014 27.661500    172.18.35.96      172.19.49.84      FTP       97 Response: 257 "/" is current directory.
5015 27.661810    172.19.49.84      172.18.35.96      FTP       74 Request: TYPE I
5016 27.664099    172.18.35.96      172.19.49.84      FTP       85 Response: 200 Type set to I
5017 27.664316    172.19.49.84      172.18.35.96      FTP       74 Request: SIZE /
5018 27.667903    172.18.35.96      172.19.49.84      FTP       86 Response: 550 File not found
5019 27.668274    172.19.49.84      172.18.35.96      FTP       73 Request: CWD /
5020 27.671229    172.18.35.96      172.19.49.84      FTP      113 Response: 250 CWD successful. "/" is current director…
5021 27.671472    172.19.49.84      172.18.35.96      FTP       72 Request: PASV
5022 27.676796    172.18.35.96      172.19.49.84      FTP      116 Response: 227 Entering Passive Mode (172,18,35,96,231…
```

5026 27.680413	172.19.49.84	172.18.35.96	FTP	75 Request: LIST -l
5027 27.685400	172.18.35.96	172.19.49.84	FTP	121 Response: 150 Opening data channel for directory list…

5034 27.688453	172.19.49.84	172.18.35.96	FTP	72 Request: QUIT
5036 27.690289	172.18.35.96	172.19.49.84	FTP	79 Response: 221 Goodbye

⑤数据连接的释放：

5029 27.685816	172.18.35.96	172.19.49.84	TCP	66 59243 → 4030 [FIN, ACK] Seq=943 Ack=1 Win=66560 Len=0…
5030 27.685893	172.19.49.84	172.18.35.96	TCP	66 4030 → 59243 [ACK] Seq=1 Ack=944 Win=65536 Len=0 TSva…
5033 27.688225	172.19.49.84	172.18.35.96	TCP	66 4030 → 59243 [FIN, ACK] Seq=1 Ack=944 Win=65536 Len=0…
5035 27.690065	172.18.35.96	172.19.49.84	TCP	66 59243 → 4030 [ACK] Seq=944 Ack=2 Win=66560 Len=0 TSva…

⑥控制连接的释放：

5037 27.690443	172.19.49.84	172.18.35.96	TCP	66 4029 → 21 [FIN, ACK] Seq=78 Ack=385 Win=66048 Len=0 T…
5038 27.690448	172.18.35.96	172.19.49.84	TCP	66 21 → 4029 [FIN, ACK] Seq=385 Ack=78 Win=66304 Len=0 T…
5039 27.690487	172.19.49.84	172.18.35.96	TCP	66 4029 → 21 [ACK] Seq=79 Ack=386 Win=66048 Len=0 TSval=…
5087 27.992507	172.19.49.84	172.18.35.96	TCP	66 [TCP Retransmission] 4029 → 21 [FIN, ACK] Seq=78 Ack=…
5088 27.994789	172.18.35.96	172.19.49.84	TCP	60 [TCP ZeroWindow] 21 → 4029 [ACK] Seq=386 Ack=79 Win=0…

由此，我们可填写下表：

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	4996，4997，4998	4029	21
数据连接的建立	5023，5024，5025	4030	59243
FTP 数据传送	5028	4030	59243
FTP 指令传送和响应	5000，5008，5009，5010，5011，5012，5013，5014，5015，5016，5017，5018，5019，5020，5021，5022，5026，5027，5034，5036	4029	21
数据连接的释放	5029，5030，5033，5035	4030	59243
控制连接的释放	5037，5038，5039，5087，5088	4029	21

（4）从协议层面分析 FTP-DOS 与 FTP-WEB 的异同。

答：

异：在 FTP-WEB 中客户端向服务器发送一个 PASV 指令后，服务器进入被动模式，服务器打开 59243 端口并通知客户端在这个端口上传输数据，之后服务器就通过这个端口进行数据传送。而在 FTP-DOS 中，服务器在主动模式下工作，通过 20 端口与客户端传送数据。在 FTP-WEB 中控制连接的释放的过程中，客户端向服务器发送的[FIN, ACK]发生了重传，而在 FTP-DOS 中并没有出现这中情况。

同：FTP-DOS 和 FTP-WEB 均是使用 TCP 连接，它们的连接均是非持续连接。

（5）在步骤 5 中，FTP 中的匿名账号是什么？

答：匿名账号是 anonyous，密码是 chrome@example.com

（6）叙述 TCP 连接建立的三次握手的过程，四次挥手终止连接的过程。

答：

TCP 连接建立的三次握手的过程：

第一步，客户端 TCP 向服务器端的 TCP 发送一个 SYN 报文，其中 SYN 比特置为 1，还包括客户端选择的起始序号（client_isn）。

第二步，服务器的 TCP 收到客户端 TCP 的 SYN 报文后，向客户端 TCP 发送 SYNACK 报文段，其中 SYN 比特置为 1，确认号字段被置为 client_isn+1，还包括服务器端选择的初始序号（server_isn）。

第三步，客户端 TCP 接受 SYNACK 报文段后，向服务器端发送一个报文，其中确认号字段被置为 server_isn+1，SYN 比特置为 0。

四次挥手终止连接的过程：客户 TCP 向服务器进程发送一个特殊的 TCP 报文段，其中 FIN 比特被置为 1。当服务器受到该报文段后，向客户端回送一个确认报文段。然后服务器发送它自己的终止报文段，其中 FIN 比特被置为 1。最后，客户端接受该报文后返回一个确认报文段。

（7）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。

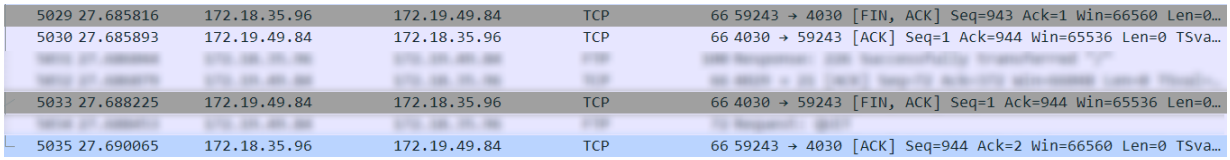
三次握手的过程：

5023	27.677347	172.19.49.84	172.18.35.96	TCP	74 4030 → 59243 [SYN] Seq=0 Win=64240 Len…
5024	27.680011	172.18.35.96	172.19.49.84	TCP	74 59243 → 4030 [SYN, ACK] Seq=0 Ack=1 Wi…
5025	27.680124	172.19.49.84	172.18.35.96	TCP	66 4030 → 59243 [ACK] Seq=1 Ack=1 Win=665…

从中可以看到客户端（172.19.49.84）向服务器（172.18.35.96）发送一个 SYN 报文

段，其中 Seq=0。服务器收到后向客户端发送一个 SYNACK 报文，其中 Seq=0，ACK=1。客户端收到后又向服务器发送一个 ACK 报文段，其中 Seq=1，ACK=1。

四次挥手终止连接的过程：



从中可以看到服务器向客户端发送一个 FIN 报文段，其中 Seq=943。客户端收到后返回一个 ACK 报文段，其中 ACK=944。然后客户端向服务器发送一个 FIN 报文段，其中 Seq=1，服务器收到后返回一个 ACK 报文段，其中 ACK=2。

【FTP 实验思考】

（1）分析 FTP 使用的两个 TCP 连接，具体指出哪些情况下使用数据连接，哪些情况下使用控制连接。

答：在传送指令和响应指令的情况下使用控制连接，在传输数据的情况下使用数据连接。

（2）比较 FTP 协议和 HTTP 协议

答：

HTTP 协议用于访问网站，而 FTP 协议用于访问和传输文件。

HTTP 头包含了 metadata，例如最后更改的日期、编码方式、服务器名称版本还有其他的一些信息，而在 FTP 中不存在这些。

FTP 能传输 ACSII 数据或者二进制格式的数据，而 HTTP 只用二进制格式。

HTTP 支持流水线，即客户端可以在上一个请求处理完之前，发出下一个请求，多次请求数据省掉了部分服务器客户端往返时延。而 FTP 不支持流水线。

FTP 使用两个 TCP 连接，第一个 TCP 连接用于传送控制指令，第二个 TCP 连接用于接受或者发送数据。而 HTTP 在双向传输中使用动态端口。

HTTP 可建立持久连接，即对一个 HTTP 会话来讲，客户端可以维护一个单个的连接并使用它进行任意数量的数据传输。而 FTP 每次都创建一个新的连接来传输数据。

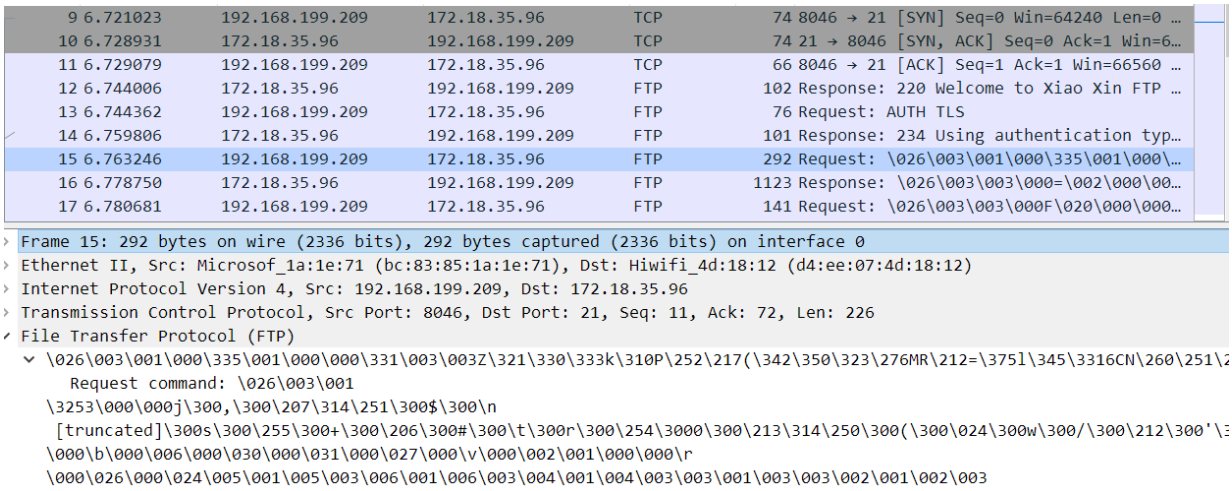
FTP 协议是直接面向文件级别的，即 FTP 可以通过命令列出远程服务器上的目录列表，而 HTTP 没有这个概念。

（3）讨论 FTP 协议的安全问题

答：FTP 协议是不安全的，因为 FTP 的用户名和密码是以明文 ASCII 码传送的。

（4）启用 FileZilla 创建的 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。

答：设置 TLS 加密后如图所示：



从图中可以看到，客户端和服务器发送的报文均被加密，没有以明文 ASCII 码传送用户名和密码，这样保证了用户名和口令的安全。

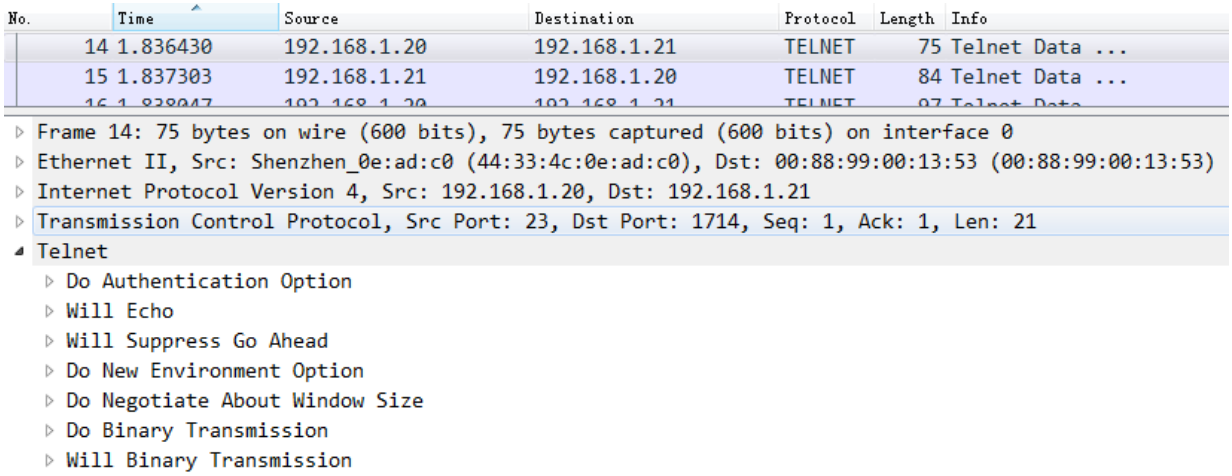
（5）同一台主机，既作为 FTP 服务器，有充当客户端（非虚拟机形式），如何捕获 FTP 数据包？

答：如果这台主机拥有两个网卡即可捕获 FTP 数据包。例如，主机拥有一个有线网卡和一个无线网卡，在分配给有线网卡的 IP 地址下建立一个 FTP 服务器，然后通过分配给无线网卡的 IP 访问该服务器并进行文件传送，这样就可以在同一台主机使用 wireshark 捕获 FTP 数据包。

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。

答：由报文内容可知是选项协商，且是单命令选项协商。如下图所示：



由图我们可以依次列出它们的选项名和代码（代码来源为查表，书上表中没有的为读取最下方 16 进制数据推算）：

选项名	选项代码
Authentication Option	37
Echo	1
Suppress Go Ahead	3
New Environment Option	39
Negotiate About Window size	31
Binary Transmission	0
Binary Transmission	0

（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，填入表中。

答：由上题的截图，结合书上的查表，我们可以继续完成此题的填表：

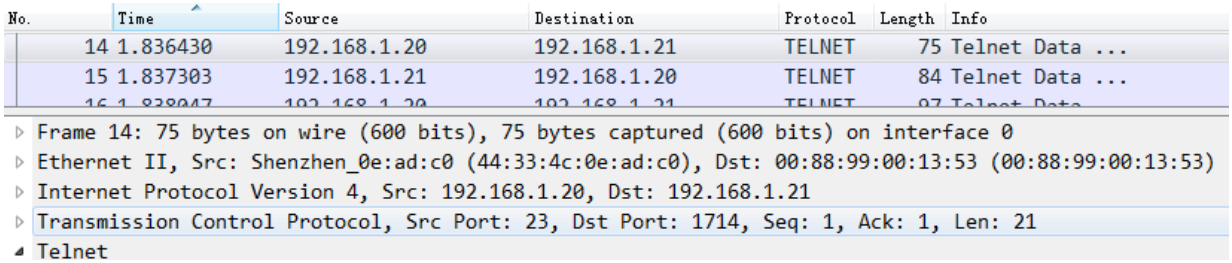
Telnet 报文分析

请求类型	请求类型代码	选项名称	选项代码	意义
Do	253	Authentication Option	37	发送希望接收方开始认证选项
Will	251	Echo	1	发送方希望开始执行回送选项
Will	251	Suppress Go Ahead	3	发送方希望开始执行抑制前进选项
Do	253	New Environment	39	发送方希望接收方

		Option		开始执行新环境选项
Do	253	Negotiate About Window size	31	发送方希望接收方执行窗口尺寸选项
Do	253	Binary Transmission	0	发送方希望接收方执行二进制文件传输选项
Will	251	Binary Transmission	0	发送方希望开始执行二进制文件传输选项

（3）在 TCP 连接时，Telnet 使用的端口号是多少？

答：如图所示

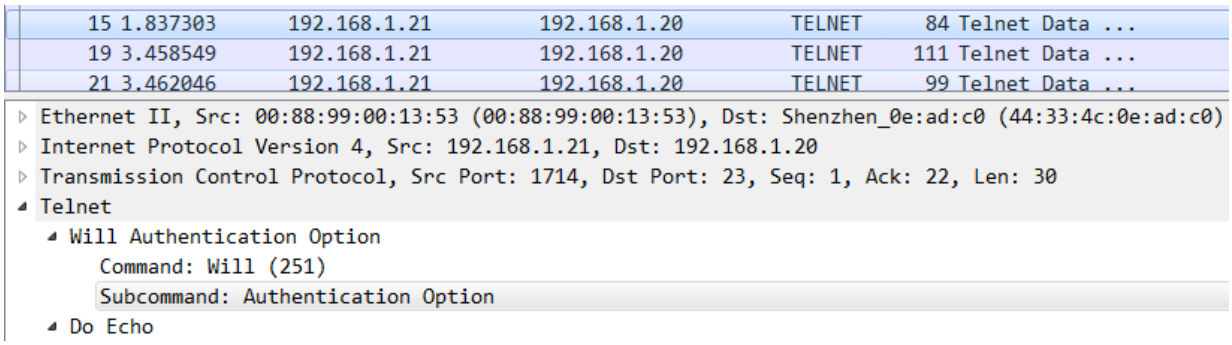


运输层信息显示，Telnet 使用的端口号为：服务器端口号 23，客户机端口号 1714。

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

答：将客户端输入命令的传输报文依次截图：

第一条报文为选项协商（单命令选项协商格式）。



第二条报文为选项协商（子协商选项格式）。

No.	Time	Source	Destination	Protocol	Length	Info
95	23.741664	192.168.1.20	192.168.1.21	TELNET	70	Telnet Data ...
97	23.885795	192.168.1.20	192.168.1.21	TELNET	70	Telnet Data ...
99	24.029135	192.168.1.20	192.168.1.21	TELNET	70	Telnet Data ...
15	1.837303	192.168.1.21	192.168.1.20	TELNET	84	Telnet Data ...
19	3.458549	192.168.1.21	192.168.1.20	TELNET	111	Telnet Data ...
21	3.462046	192.168.1.21	192.168.1.20	TELNET	99	Telnet Data ...

```
▷ Frame 19: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 31, Ack: 65, Len: 57
◢ Telnet
  ◢ Suboption Authentication Option
      Command: Suboption (250)
    ▷ Subcommand: Authentication Option
  ◢ Suboption End
      Command: Suboption End (240)
```

第三条报文为选项协商（子协商选项格式）。

No.	Time	Source	Destination	Protocol	Length	Info
15	1.837303	192.168.1.21	192.168.1.20	TELNET	84	Telnet Data ...
19	3.458549	192.168.1.21	192.168.1.20	TELNET	111	Telnet Data ...
21	3.462046	192.168.1.21	192.168.1.20	TELNET	99	Telnet Data ...
23	3.658917	192.168.1.21	192.168.1.20	TELNET	492	Telnet Data ...
27	5.984465	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
30	6.440483	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...

```
▷ Frame 21: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 88, Ack: 220, Len: 45
◢ Telnet
  ◢ Suboption New Environment Option
      Command: Suboption (250)
    ▷ Subcommand: New Environment Option
  ◢ Suboption End
      Command: Suboption End (240)
  ◢ Suboption New Environment Option
      Command: Suboption (250)
    ▷ Subcommand: New Environment Option
  ◢ Suboption End
      Command: Suboption End (240)
```

第四条报文为选项协商（子协商选项格式）。

21	3.462046	192.168.1.21	192.168.1.20	TELNET	99	Telnet Data ...
23	3.658917	192.168.1.21	192.168.1.20	TELNET	492	Telnet Data ...
27	5.984465	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
30	6.440483	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...

```
▷ Frame 23: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 133, Ack: 220, Len: 438
◢ Telnet
  ◢ Suboption Authentication Option
      Command: Suboption (250)
    ▷ Subcommand: Authentication Option
  ◢ Suboption End
      Command: Suboption End (240)
```

接下来的几条报文为数据传输。

No.	Time	Source	Destination	Protocol	Length	Info
23	3.658917	192.168.1.21	192.168.1.20	TELNET	492	Telnet Data ...
27	5.984465	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
30	6.440483	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
33	6.936429	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
35	7.040038	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
37	7.176147	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...

```
▷ Frame 27: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 571, Ack: 411, Len: 1
◢ Telnet
    Data: a
```

No.	Time	Source	Destination	Protocol	Length	Info
23	3.658917	192.168.1.21	192.168.1.20	TELNET	492	Telnet Data ...
27	5.984465	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
30	6.440483	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
33	6.936429	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
35	7.040038	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
37	7.176147	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...

```
▷ Frame 30: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 572, Ack: 412, Len: 1
◢ Telnet
    Data: d
```

No.	Time	Source	Destination	Protocol	Length	Info
23	3.658917	192.168.1.21	192.168.1.20	TELNET	492	Telnet Data ...
27	5.984465	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
30	6.440483	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
33	6.936429	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
35	7.040038	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
37	7.176147	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...

```
▷ Frame 33: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 573, Ack: 413, Len: 1
◢ Telnet
    Data: m
```

No.	Time	Source	Destination	Protocol	Length	Info
23	3.658917	192.168.1.21	192.168.1.20	TELNET	492	Telnet Data ...
27	5.984465	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
30	6.440483	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
33	6.936429	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
35	7.040038	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...
37	7.176147	192.168.1.21	192.168.1.20	TELNET	55	Telnet Data ...

```
▷ Frame 35: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0
▷ Ethernet II, Src: 00:88:99:00:13:53 (00:88:99:00:13:53), Dst: Shenzhen_0e:ad:c0 (44:33:4c:0e:ad:c0)
▷ Internet Protocol Version 4, Src: 192.168.1.21, Dst: 192.168.1.20
▷ Transmission Control Protocol, Src Port: 1714, Dst Port: 23, Seq: 574, Ack: 414, Len: 1
◢ Telnet
    Data: i
```

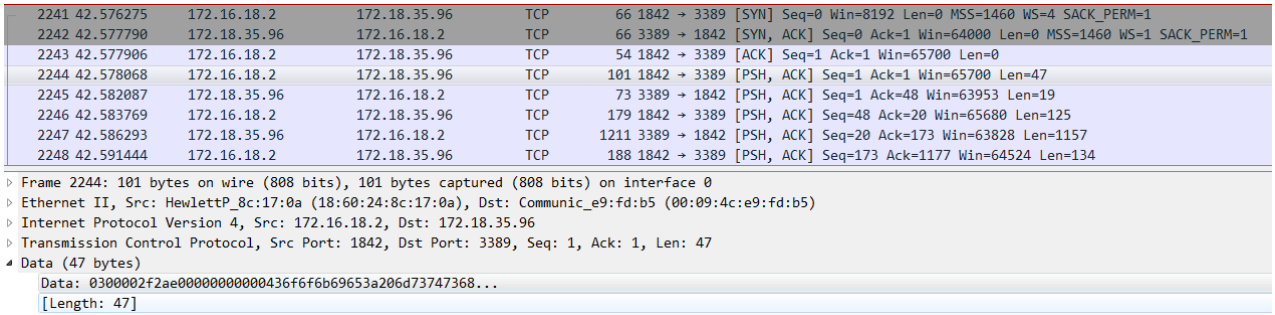
由上面报文的具体信息，我们可以填写下表：

Telnet 协议工作过程

过程	报文号	功能	信息及参数	报文作用
Telnet 选项协商	1	选项协商	Will Authenticatioin Option	发送方希望开始执行开始认证选项
	31	选项协商	Suboption Authenticatioin Option Suboption End	使用子协商将开始认证选项的内容发回服务器并结束
	88	选项协商	Suboption New Environment Option Suboption End	使用子协商将新环境选项的内容发回服务器并结束
	133	选项协商	Suboption Authentication Option	使用子协商将认证选项的内容发回服务器
Telnet 数据传输	571	数据传输	Data： a	传送数据‘a’
	572	数据传输	Data： d	传送数据‘d’
	573	数据传输	Data： m	传送数据‘m’
	574	数据传输	Data： i	传送数据‘i’

（6）进行“远程桌面连接”的实验，并捕获实验的数据包，通过对数据包的分析，指出这种方式与 Telnet 连接有什么异同？

答：捕获的数据包如下：



异：由图中灰色的部分可以看到，这种方式与 Telnet 相比是存在三次握手的，同时我们看到报文段中传送的数据，与之前截图中 Telnet 数据传输报文不同，是密文传输

的。

同：都是通过 TCP 连接传送数据的。

（7）Telnet 口令是否为明文传输？

答：是。理由如下 4 图所示，数据为一开始建立管理员的用户名，可以看到它们分别为 administrator 的前四个字母，显然 Telnet 口令是明文传输的。