

中国矿业大学计算机学院

2019 级本科生计算机网络实验报告

实验内容 协议报文分析

学生姓名 李春阳 学 号 10193657

专业班级 信息安全 2019-1 班

学 院 计算机科学与技术学院

任课教师 姜秀柱

课程基础理论掌握程度	熟练 <input type="checkbox"/>	较熟练 <input type="checkbox"/>	一般 <input type="checkbox"/>	不熟练 <input type="checkbox"/>
综合知识应用能力	强 <input type="checkbox"/>	较强 <input type="checkbox"/>	一般 <input type="checkbox"/>	差 <input type="checkbox"/>
报告内容	完整 <input type="checkbox"/>	较完整 <input type="checkbox"/>	一般 <input type="checkbox"/>	不完整 <input type="checkbox"/>
报告格式	规范 <input type="checkbox"/>	较规范 <input type="checkbox"/>	一般 <input type="checkbox"/>	不规范 <input type="checkbox"/>
实验完成状况	好 <input type="checkbox"/>	较好 <input type="checkbox"/>	一般 <input type="checkbox"/>	差 <input type="checkbox"/>
工作量	饱满 <input type="checkbox"/>	适中 <input type="checkbox"/>	一般 <input type="checkbox"/>	欠缺 <input type="checkbox"/>
学习、工作态度	好 <input type="checkbox"/>	较好 <input type="checkbox"/>	一般 <input type="checkbox"/>	差 <input type="checkbox"/>
抄袭现象	无 <input type="checkbox"/>	有 <input type="checkbox"/> 姓名:		
存在问题				
总体评价				

综合成绩:

任课教师签字:

年 月 日

实验编号：02

项目名称：协议报文分析

实验内容：

- (1) 拓扑结构探测：给出实验用机所在机房的局域网以及接入校园网的拓扑结构；
- (2) 测试互联网接入路径：运用 tracert 命令测试本机到互联网的接入路径；
- (3) 运用抓包工具，分别获取不同互联网访问情形下的本机网卡数据包；
- (4) 分别对不同互联网访问情形下的数据包进行逐层分析，给出各层协议的主要参数及意义；

要求分别获取 WWW 服务、Email 服务、QQ 通信和迅雷文件下载四种不同网络服务过程中的数据包。

实验要求：

- (1) 通过拓扑结构探测，懂得跨网连接的概念，以及跨网连接必须的设备；
- (2) 通过 tracert 命令应用，给出校园网连接互联网的接入网结构；
- (3) 运用抓包工具，实时抓包，记录包状态变化；
- (4) 给出不同应用情境下的不同层次数据包的分析结果。
- (5) 透过 Web 服务访问，分析 HTTP 协议工作过程，总结 HTTP 协议特点；透过 HTTP 工作过程分析，获取 TCP 协议的工作过程，验证连接建立的三次握手过程，以及滑动窗口工作机制（选做）。

预习要求：

提前通过互联网或在实验室开始实验前登录实验管理服务器，点击预习链接，阅览或下载实验指导书——预习\网络协议\进阶-IP 分组基本报文分析。

操作与观察：

正确按照实验指导书步骤操作，观察记录下操作结果。

实验报告要求：

- (1) 按照实验要求，完成全部实验内容
- (2) 在标准实验报告书上填写全部实验操作记录和观察结果
- (3) 登录实验管理服务器，提交实验报告电子档。

实验步骤

3.2.1 拓扑结构探测

给出实验用机所在机房的局域网以及接入校园网的拓扑结构；

3.2.2 测试互联网接入路径

运用 tracert 命令测试本机到互联网的接入路径；

3.2.3 运用抓包工具，分别获取不同互联网访问情形下的本机网卡数据包；

3.2.4 包分析

分别对不同互联网访问情形下的数据包进行逐层分析，给出各层协议的主要参数及意义（要求分别获取 WWW 服务、Email 服务、QQ 通信和迅雷文件下载四种不同网络服务过程中的数据包）。

实验报告内容：

1 拓扑结构探测

步骤一：进入计算机桌面，打开资源管理器，展开左侧“网络”。从展开的网络目录树中，看到处在同一个网络内的相邻的主机数及其主机名，记录下同网的的这些主机。



步骤二：使用快捷键 Win+R 打开运行，输入 cmd 打开命令提示符。用 tracert 命令，探测到达任一相邻主机的中间节点。记下通往所探测节点的中间节点数，给出本机到达探测主机的路径。对不同主机，重复此步，可给出本网内主机的拓扑结构。

```
C:\Users\i>tracert 192.168.42.110

通过最多 30 个跃点跟踪
到 27-20 [192.168.42.110] 的路由:

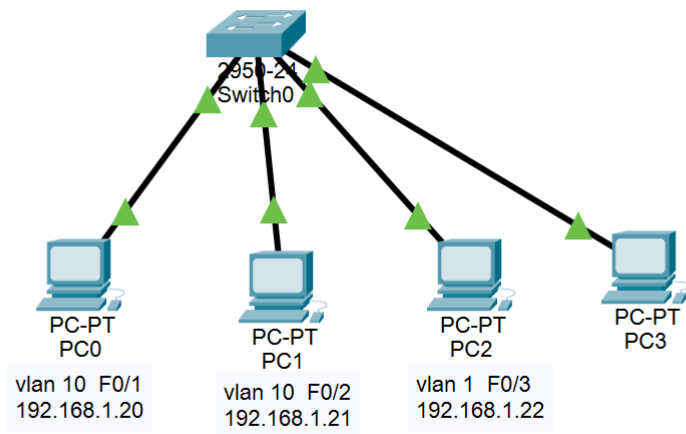
    1    <1 毫秒    <1 毫秒    <1 毫秒  27-20 [192.168.42.110]
跟踪完成。

C:\Users\i>tracert 192.168.42.139

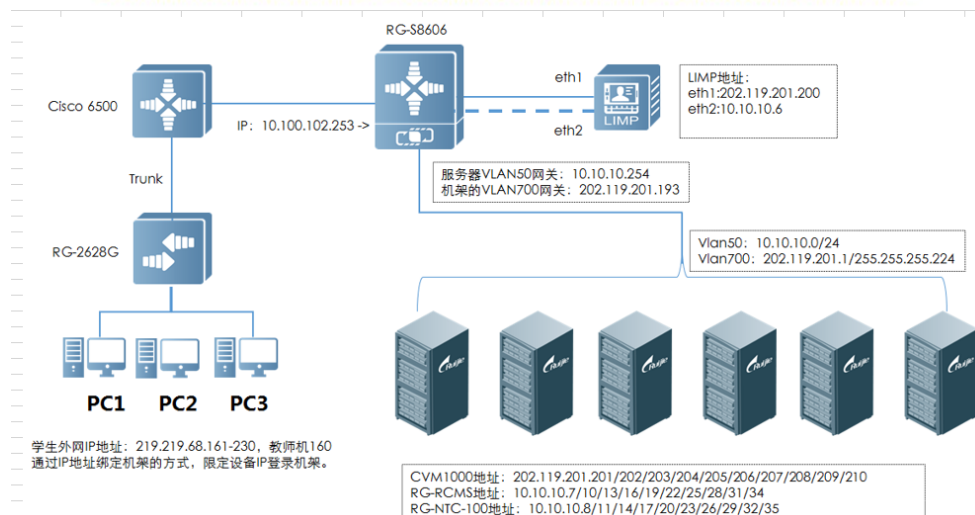
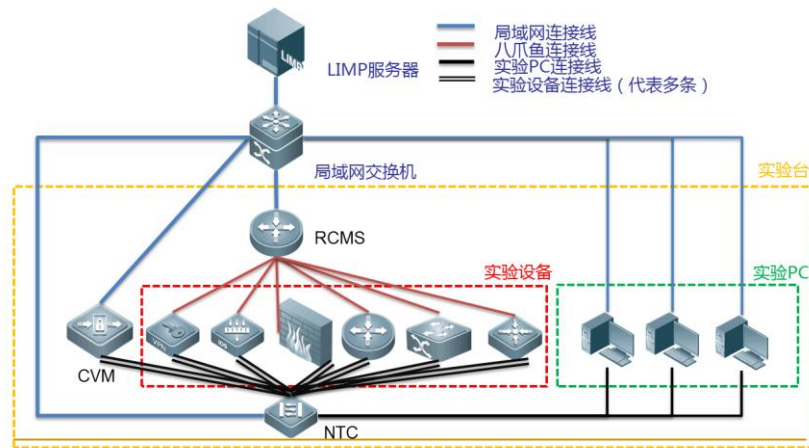
通过最多 30 个跃点跟踪
到 27-49 [192.168.42.139] 的路由:

    1    <1 毫秒    <1 毫秒    <1 毫秒  27-49 [192.168.42.139]
跟踪完成。
```

用 tracert 命令，探测到达任一相邻主机的中间节点，观察发现中间节点数均为 1。故下图为拓扑图。



■ 中国矿业大学 锐捷网络实验室典型拓扑



2 实验用机接入校园网的拓扑结构：

2.1 测试校园网接入路径

进入 cmd，用 tracert 命令，探测访问校园网 Web 服务器、DNS 服务器的路径，记录下从本机到达两个不同服务器的前面共同路径，即为本机接入校园网的路径。

```
C:\Users\lenovo>tracert www.cumt.edu.cn
```

通过最多 30 个跃点跟踪

到 www.cumt.edu.cn [202.119.200.206] 的路由：

1	*	26 ms	*	10.3.255.254
2	*	*	*	请求超时。
3	*	*	6 ms	10.2.6.1
4	7 ms	6 ms	5 ms	172.35.1.2
5	*	*	*	请求超时。
6	3 ms	2 ms	3 ms	202.119.200.206

跟踪完成。

```
C:\Users\lenovo>tracert DNS.cumt.edu.cn
```

通过最多 30 个跃点跟踪

到 DNS.cumt.edu.cn [202.119.200.10] 的路由：

1	1 ms	1 ms	1 ms	10.3.255.254
2	*	16 ms	3 ms	10.2.4.2
3	6 ms	7 ms	4 ms	10.2.6.1
4	10 ms	15 ms	20 ms	172.35.1.2
5	*	*	*	请求超时。
6	7 ms	4 ms	3 ms	dns.cumt.edu.cn [202.119.200.10]

跟踪完成。

2.2.测试互联网接入路径

步骤四：进入 cmd，用 tracert 命令，反复探测到达互联网门户网站的中间节点，记录下这些节点，尤其记住前面几个节点。从而给出本机接入互联网的入网路径。

```
C:\Users\lenovo>tracert www.baidu.com
```

通过最多 30 个跃点跟踪
到 www.a.shifen.com [112.80.248.75] 的路由:

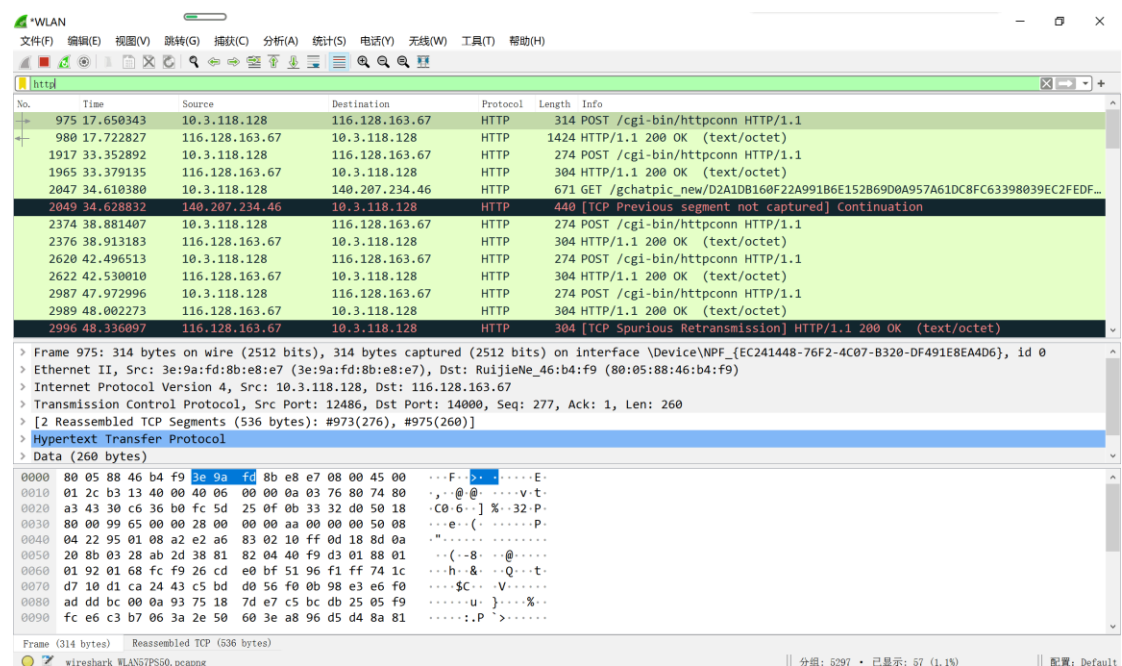
1	5 ms	4 ms	11 ms	10.3.255.254
2	*	9 ms	5 ms	10.2.1.1
3	14 ms	8 ms	9 ms	112.85.229.157
4	7 ms	7 ms	5 ms	112.85.230.46
5	16 ms	15 ms	15 ms	112.85.230.137
6	21 ms	19 ms	18 ms	122.96.66.102
7	23 ms	22 ms	20 ms	112.86.192.146
8	*	*	*	请求超时。
9	28 ms	22 ms	27 ms	112.80.248.75

跟踪完成。

3.抓包过程

运用抓包工具，分别获取不同互联网访问情形下的本机网卡数据包；过滤捕获和过滤显示不同条件的数据包

3.1.http 协议



3.2.UDP 协议

Wireshark capture of an HTTP POST request over UDP. The packet list shows a POST to /cgi-bin/httpconn. The packet details pane shows the structure of the HTTP message, including the status line 200 OK (text/plain). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
975	17.650343	10.3.118.128	116.128.163.67	HTTP	314	POST /cgi-bin/httpconn HTTP/1.1
980	17.722827	116.128.163.67	10.3.118.128	HTTP	1424	HTTP/1.1 200 OK (text/plain)
1917	33.352892	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
1965	33.379135	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2047	34.610380	10.3.118.128	140.207.234.46	HTTP	671	GET /gchatpic_new/D2A1D8160F22A991B6E152B69D0A957A61DC8FC63398039EC2FEDF...
2049	34.628832	140.207.234.46	10.3.118.128	HTTP	440	[TCP Previous segment not captured] Continuation
2374	38.881407	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
2376	38.913183	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2620	42.496513	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
2622	42.530010	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2987	47.972996	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
2989	48.002273	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2996	48.336097	116.128.163.67	10.3.118.128	HTTP	304	[TCP Spurious Retransmission] HTTP/1.1 200 OK (text/plain)

Frame 975: 314 bytes on wire (2512 bits), 314 bytes captured (2512 bits) on interface \Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6}, id 0
> Ethernet II, Src: 3e:9a:fd:8b:e8:e7 (3e:9a:fd:8b:e8:e7), Dst: RuijieNe_46:b4:f9 (80:05:88:46:b4:f9)
> Internet Protocol Version 4, Src: 10.3.118.128, Dst: 116.128.163.67
> Transmission Control Protocol, Src Port: 12486, Dst Port: 14000, Seq: 277, Ack: 1, Len: 260
> [2 Reassembled TCP Segments (536 bytes): #973(276), #975(260)]
> Hypertext Transfer Protocol
> Data (260 bytes)

```
0000  00 05 88 46 b4 f9 3e 9a fd 8b e8 e7 08 00 45 00  ...F...E-
0010  01 2c b3 13 40 00 40 06 00 00 0a 03 76 80 74 80  ,..@...v.t
0020  a3 43 30 c6 36 b0 fc 5d 25 0f 0b 33 32 d0 50 18  .C0.6...]-32.P
0030  00 00 99 65 00 00 28 00 00 00 aa 00 00 00 50 08  ...e...P
0040  04 22 95 01 08 a2 e2 a6 83 02 10 ff 0d 18 8d 0a  ."-.....
0050  20 8b 03 28 ab 2d 38 81 82 04 40 f9 d3 01 88 01  ..(-8...@...
0060  01 92 01 68 fc f9 26 cd e0 bf 51 96 f1 ff 74 1c  ..h-&...Q...t
0070  d7 10 d1 ca 24 43 c5 bd d0 56 f0 0b 98 e3 e6 f0  ...$C...V...
0080  ad dd bc 00 0a 93 75 18 7d e7 c5 bc db 25 05 f9  ....u...}%...
0090  fc e6 c3 b7 06 3a 2e 50 60 3e a8 96 d5 d4 8a 81  ....:..P`>.....
```

3.3.TCP 协议

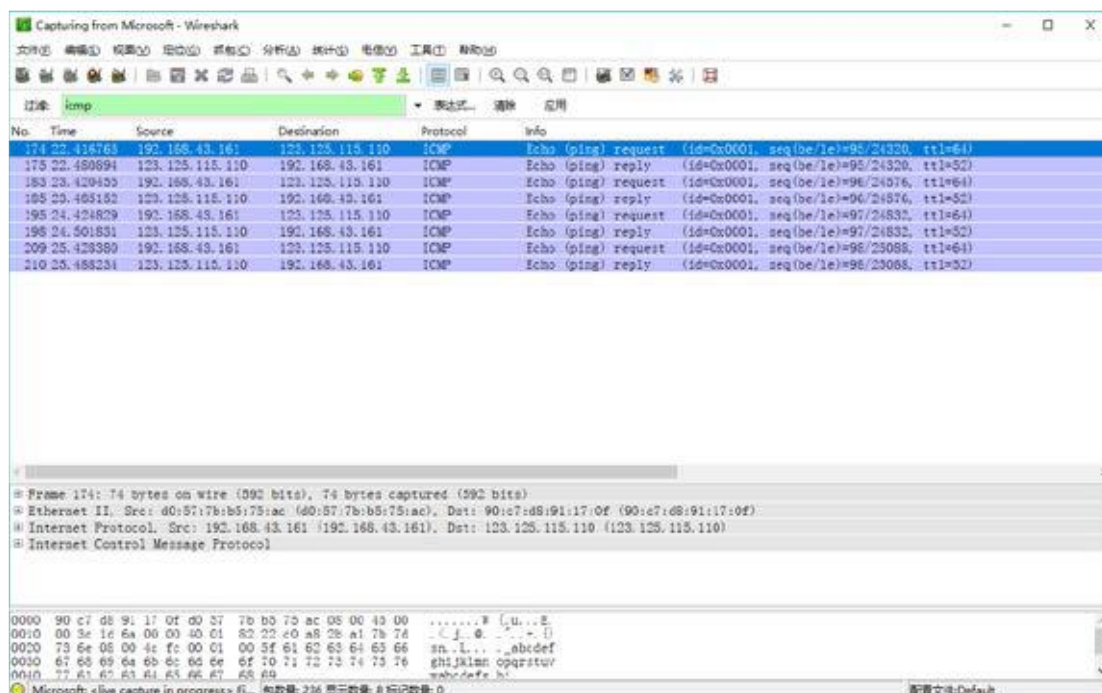
Wireshark capture of an HTTP POST request over TCP. The packet list shows a POST to /cgi-bin/httpconn. The packet details pane shows the structure of the HTTP message, including the status line 200 OK (text/plain). The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
975	17.650343	10.3.118.128	116.128.163.67	HTTP	314	POST /cgi-bin/httpconn HTTP/1.1
980	17.722827	116.128.163.67	10.3.118.128	HTTP	1424	HTTP/1.1 200 OK (text/plain)
1917	33.352892	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
1965	33.379135	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2047	34.610380	10.3.118.128	140.207.234.46	HTTP	671	GET /gchatpic_new/D2A1D8160F22A991B6E152B69D0A957A61DC8FC63398039EC2FEDF...
2049	34.628832	140.207.234.46	10.3.118.128	HTTP	440	[TCP Previous segment not captured] Continuation
2374	38.881407	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
2376	38.913183	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2620	42.496513	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
2622	42.530010	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2987	47.972996	10.3.118.128	116.128.163.67	HTTP	274	POST /cgi-bin/httpconn HTTP/1.1
2989	48.002273	116.128.163.67	10.3.118.128	HTTP	304	HTTP/1.1 200 OK (text/plain)
2996	48.336097	116.128.163.67	10.3.118.128	HTTP	304	[TCP Spurious Retransmission] HTTP/1.1 200 OK (text/plain)

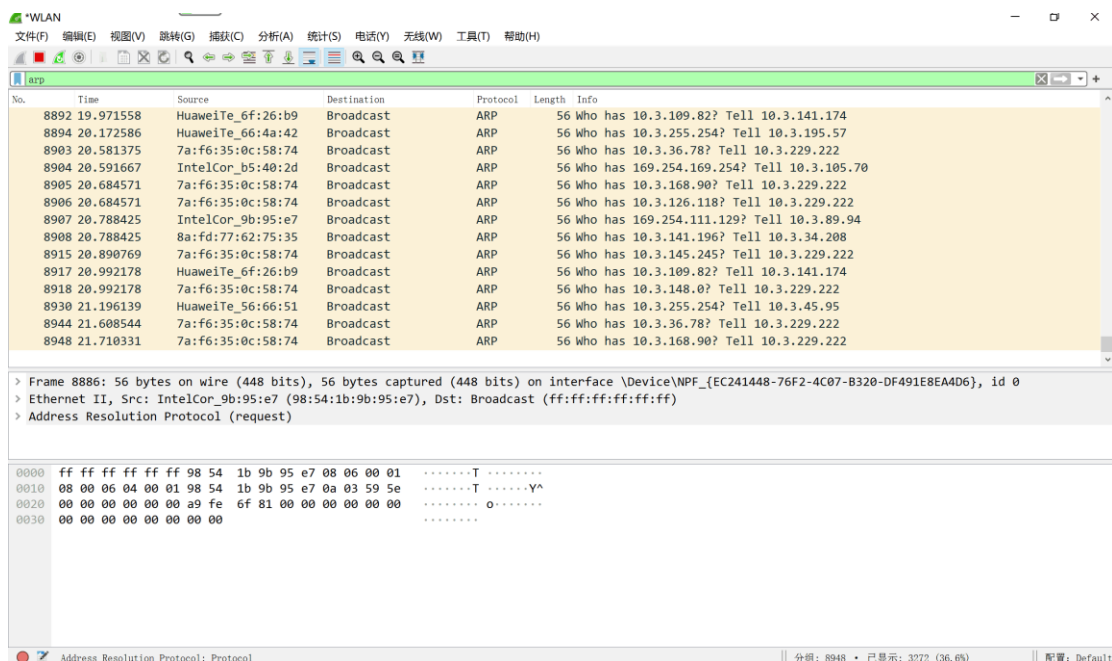
Frame 975: 314 bytes on wire (2512 bits), 314 bytes captured (2512 bits) on interface \Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6}, id 0
> Ethernet II, Src: 3e:9a:fd:8b:e8:e7 (3e:9a:fd:8b:e8:e7), Dst: RuijieNe_46:b4:f9 (80:05:88:46:b4:f9)
> Internet Protocol Version 4, Src: 10.3.118.128, Dst: 116.128.163.67
> Transmission Control Protocol, Src Port: 12486, Dst Port: 14000, Seq: 277, Ack: 1, Len: 260
> [2 Reassembled TCP Segments (536 bytes): #973(276), #975(260)]
> Hypertext Transfer Protocol
> Data (260 bytes)

```
0000  00 05 88 46 b4 f9 3e 9a fd 8b e8 e7 08 00 45 00  ...F...E-
0010  01 2c b3 13 40 00 40 06 00 00 0a 03 76 80 74 80  ,..@...v.t
0020  a3 43 30 c6 36 b0 fc 5d 25 0f 0b 33 32 d0 50 18  .C0.6...]-32.P
0030  00 00 99 65 00 00 28 00 00 00 aa 00 00 00 50 08  ...e...P
0040  04 22 95 01 08 a2 e2 a6 83 02 10 ff 0d 18 8d 0a  ."-.....
0050  20 8b 03 28 ab 2d 38 81 82 04 40 f9 d3 01 88 01  ..(-8...@...
0060  01 92 01 68 fc f9 26 cd e0 bf 51 96 f1 ff 74 1c  ..h-&...Q...t
0070  d7 10 d1 ca 24 43 c5 bd d0 56 f0 0b 98 e3 e6 f0  ...$C...V...
0080  ad dd bc 00 0a 93 75 18 7d e7 c5 bc db 25 05 f9  ....u...}%...
0090  fc e6 c3 b7 06 3a 2e 50 60 3e a8 96 d5 d4 8a 81  ....:..P`>.....
```

3.4.ICMP 协议



3.5.ARP 协议



4.解析数据包

分别对不同互联网访问情形下的数据包进行逐层分析, 给出各层协议的主要参数及意义; 要求分别获取 WWW 服务、Email 服务、QQ 通信和迅雷文件下载四种不同网络服务过程中的数据包。

4.1.WWW 服务数据包:

The image shows a Wireshark packet capture of HTTP traffic on the wlan interface. The packet list shows several HTTP requests and responses, including a GET request for /test/latency?_MSPROXY_NAME=58957698840791034ceccb138_MSPROXY=DIRECT... and a POST request. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
34006	275.290425	140.207.234.35	10.3.118.128	HTTP	1278	[TCP Previous segment not captured] Continuation
34019	275.307948	140.207.234.35	10.3.118.128	HTTP	1454	[TCP Spurious Retransmission] Continuation
34020	275.307948	140.207.234.35	10.3.118.128	HTTP	1454	[TCP Spurious Retransmission] Continuation
34021	275.307948	140.207.234.35	10.3.118.128	HTTP	1454	[TCP Spurious Retransmission] Continuation
34022	275.307948	140.207.234.35	10.3.118.128	HTTP	1454	[TCP Spurious Retransmission] Continuation
34027	275.310335	140.207.234.35	10.3.118.128	HTTP	1454	[TCP Spurious Retransmission] Continuation
34119	278.386175	10.3.118.128	122.96.96.195	HTTP	768	POST / HTTP/1.1
34120	278.412910	122.96.96.195	10.3.118.128	HTTP	262	HTTP/1.1 200 OK (application/multipart-formdata)
34172	281.341686	10.3.118.128	117.132.2.19	HTTP	464	GET /test/speed?_MSPROXY_NAME=58809c7a8403915d92aba111&s=320000&v=b2c037...
34177	281.364921	10.3.118.128	101.230.86.30	HTTP	474	GET /test/latency?_MSPROXY_NAME=58957698840791034ceccb138_MSPROXY=DIRECT...
34227	281.453714	101.230.86.30	10.3.118.128	HTTP	212	HTTP/1.1 200 OK (text/plain)
34505	281.579406	117.132.2.19	10.3.118.128	HTTP	1218	HTTP/1.1 200 OK (text/plain)
34895	298.620958	10.3.118.128	116.128.163.67	HTTP	282	POST /cgi-bin/httpconn HTTP/1.1
34897	298.656139	116.128.163.67	10.3.118.128	HTTP	303	HTTP/1.1 200 OK (text/octet)

Frame 296: 473 bytes on wire (3784 bits), 473 bytes captured (3784 bits) on interface \Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6}, id 0
> Ethernet II, Src: 3e:9a:fd:8b:e8:e7 (3e:9a:fd:8b:e8:e7), Dst: RuijieNe_46:b4:f9 (08:05:88:46:b4:f9)
> Internet Protocol Version 4, Src: 10.3.118.128, Dst: 117.132.2.19
> Transmission Control Protocol, Src Port: 7386, Dst Port: 9611, Seq: 1, Ack: 1, Len: 419
> Hypertext Transfer Protocol

0000 80 05 88 46 b4 f9 3e 9a fd 8b e8 e7 08 00 45 00 ...F-->E-
0010 01 cb 49 26 40 00 40 06 00 00 0a 03 76 80 75 84 ...I&@_@ ...-V u
0020 02 13 1c da 25 8b ab c3 f5 eb b6 24 5a a3 50 18 ...% ...-\$Z P
0030 02 02 f9 d7 00 00 47 45 54 20 2f 74 65 73 74 2fGE T /test/
0040 6c 61 74 65 6e 63 79 3f 5f 4d 53 50 52 4f 58 59 latency? _MSPROXY
0050 5f 4e 41 4d 45 3d 35 38 39 35 31 33 39 33 31 32 _NAME=58 95139312
0060 30 37 32 32 30 33 34 63 65 65 34 64 34 34 26 5f 0722034c eed444&
0070 4d 53 50 52 4f 58 59 3d 44 49 52 45 43 54 26 76 MSPROXY= DIRECT&v
0080 3d 62 32 63 30 33 37 66 66 30 66 33 34 65 61 37 =b2c037f f0f34ea7
0090 37 62 38 35 36 38 64 35 34 65 65 32 39 31 37 37 7b8568d5 4ee29177
00a0 38 26 74 3d 31 36 32 35 31 38 39 32 35 33 20 48 8&t=1625 189253 H

4.2.Email 服务数据包

The image shows a Wireshark packet capture of SMTP traffic on the wlan interface. The packet list shows several SMTP messages, including a MAIL FROM: <comydream@163.com> and a DATA fragment. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Simple Mail Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
69049	611.646625	123.125.50.135	192.168.0.100	SMTP	119	S: 220 163.com Anti-spam GT for Coremail System (163com[20141201])
69050	611.647284	192.168.0.100	123.125.50.135	SMTP	76	C: EHLO LAPTOP-69B0GHJL
69052	611.655107	123.125.50.135	192.168.0.100	SMTP	239	S: 250 mail 250 PIPELINING 250 AUTH LOGIN PLAIN 250 AUTH=LOGIN PLAIN 250 coremail 10vrr2k37kG8xk117GvU7I0s8FY2U3UJ8Cz28x1U00U07ic2I0V2UvBu3Y7UCa8v8vL
69053	611.685445	192.168.0.100	123.125.50.135	SMTP	66	C: AUTH LOGIN
69054	611.722856	123.125.50.135	192.168.0.100	SMTP	72	S: 334 dXNlcwShbkljG
69055	611.723145	192.168.0.100	123.125.50.135	SMTP	80	C: User: Y29teklRyDaf1QDE2Hy5jb20=
69056	611.762081	123.125.50.135	192.168.0.100	SMTP	72	S: 334 dGZac3dvcmQ0
69057	611.762989	192.168.0.100	123.125.50.135	SMTP	68	C: Pass:
69059	611.860889	123.125.50.135	192.168.0.100	SMTP	85	S: 235 Authentication successful
69061	611.921743	192.168.0.100	123.125.50.135	SMTP	86	C: MAIL FROM: <comydream@163.com>
69063	611.970137	123.125.50.135	192.168.0.100	SMTP	67	S: 250 Mail OK
69064	611.970421	192.168.0.100	123.125.50.135	SMTP	88	C: RCPT TO: <comydream@outlook.com>
69065	612.009254	123.125.50.135	192.168.0.100	SMTP	67	S: 250 Mail OK
69066	612.010124	192.168.0.100	123.125.50.135	SMTP	60	C: DATA
69067	612.051892	123.125.50.135	192.168.0.100	SMTP	91	S: 354 End data with <R><F>,<R><F>
69068	612.052142	192.168.0.100	123.125.50.135	SMTP	430	C: DATA fragment, 376 bytes
69071	612.142896	192.168.0.100	123.125.50.135	IPF	1110	From: "comydream@163.com" <comydream@163.com>, subject: =?GB2312?B?uK1&=??-, (text/plain) (text/html)
69073	612.219632	123.125.50.135	192.168.0.100	SMTP	126	S: 250 Mail OK queued as smtp5,D9Gou4QHjovR4EB0uEUAQ--999452 1530978378
69074	612.220437	192.168.0.100	123.125.50.135	SMTP	60	C: QUIT
69075	612.261701	123.125.50.135	192.168.0.100	SMTP	63	S: 221 Bye

Frame 69075: 63 bytes on wire (504 bits), 63 bytes captured (504 bits) on interface 0
> Ethernet II, Src: Shenzhen 12:ab:8a (78:c3:30:12:ab:8a), Dst: IntelCor_b5:75:ac (08:57:7b:b5:75:ac)
> Internet Protocol Version 4, Src: 123.125.50.135, Dst: 192.168.0.100
> Transmission Control Protocol, Src Port: 25, Dst Port: 45059, Seq: 453, Ack: 1585, Len: 9
> Simple Mail Transfer Protocol
Response: 221 Bye\r\n
Response code: 220mail> Service closing transmission channel (221)
Response parameter: Bye

4.3.QQ 通信数据包:

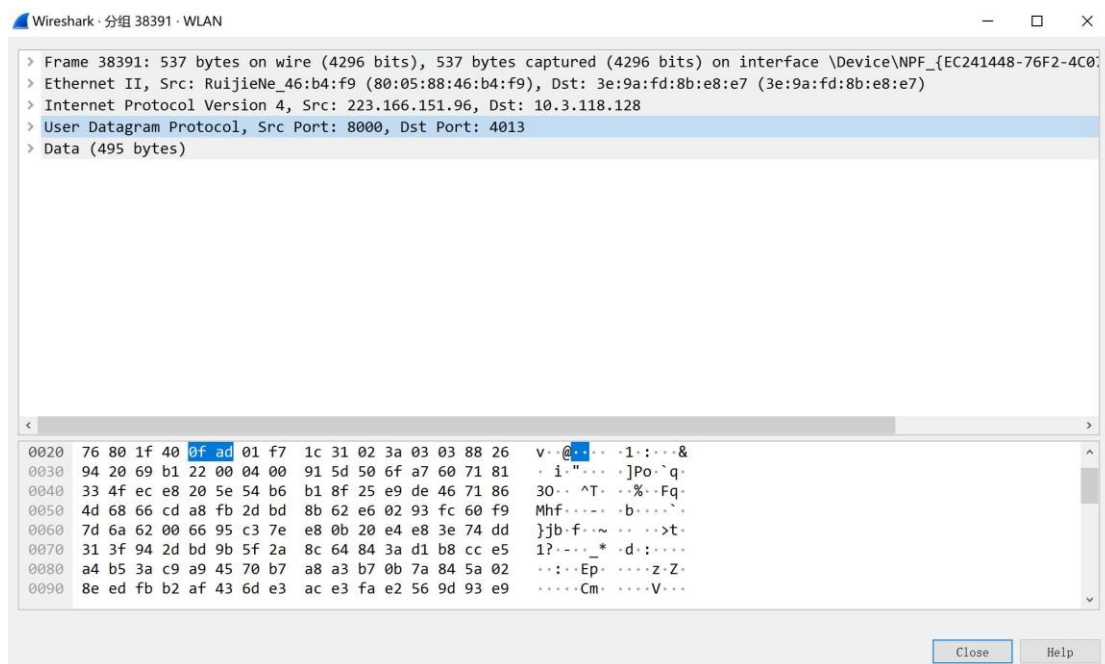
Wireshark interface showing a capture of QQ traffic. The packet list shows multiple OICQ (8000) and OICQ (4013) packets. The packet details pane shows the structure of an OICQ packet, including Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and OICQ - IM software, popular in China. The packet bytes pane shows the raw data in hexadecimal and ASCII.

4.4. 迅雷文件下载服务数据包

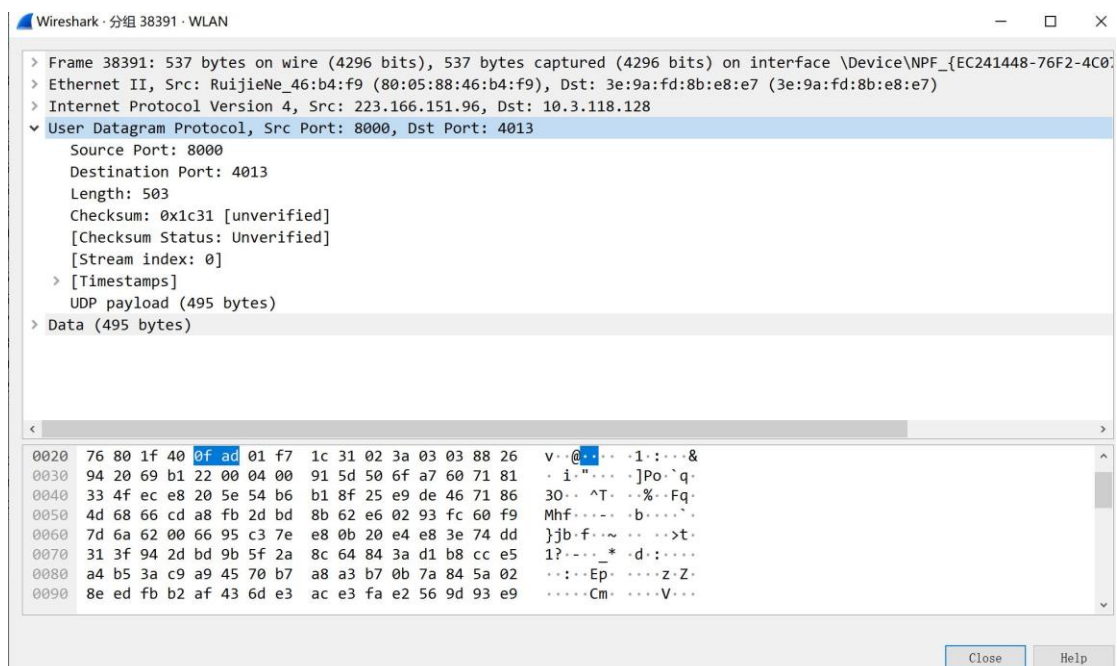
Wireshark interface showing a capture of迅雷 (Xunlei) file download service traffic. The packet list shows multiple HTTP (80) and HTTP (443) packets. The packet details pane shows the structure of an HTTP packet, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

5.包分析

5.1UDP 数据包详细分析

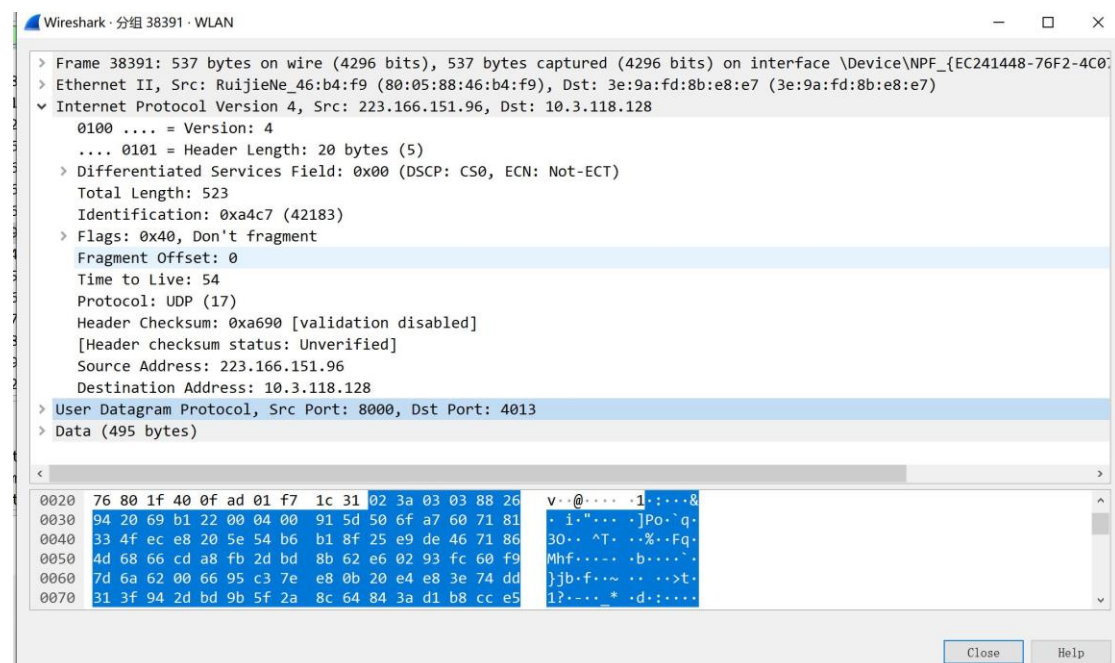


5.1.1.UDP 层



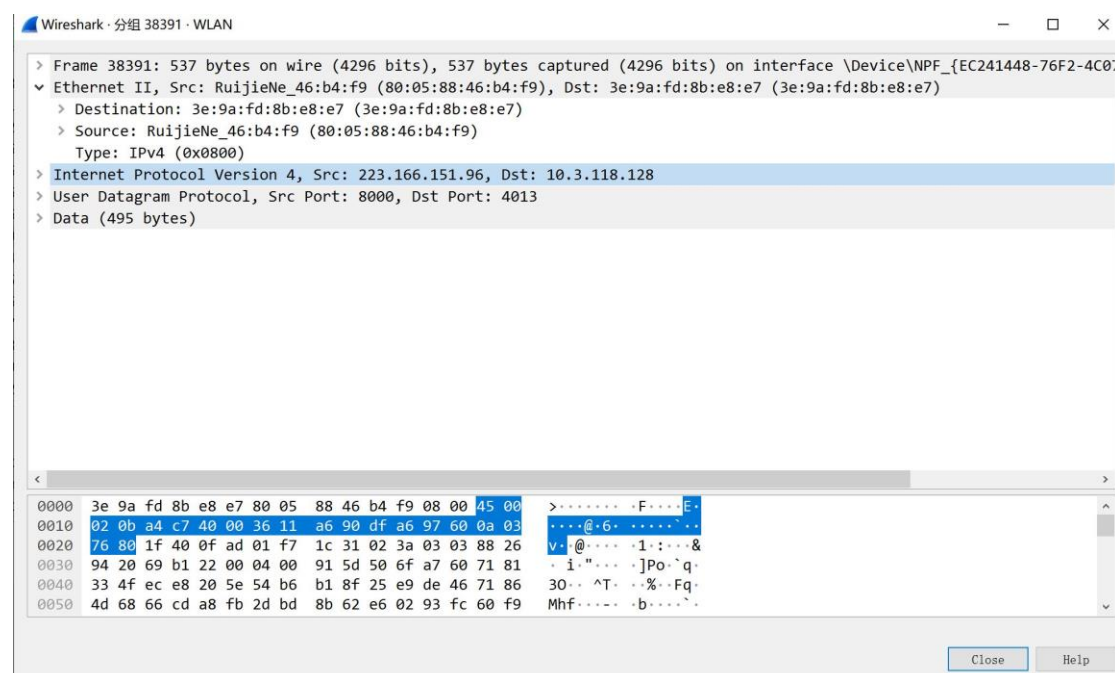
从 UDP 的报文格式可以得到 UDP 的首部信息，源端口为 8000，目的端口为 4013，UDP 长度为 503，检验和为 0x1c31

5.1.2.IP 层



分析出他的首部信息，首部长为 20 字节，还有标识位 42183，5 位没有设置的标志位以及总长度偏移量，TTL 为 54，协议字段 17，代表了上层使用 UDP，下面就是源 IP 为 223.166.151.96，目的 IP 为 10.3.118.128

5.1.3.数据链路层



上层 IP 类型为 IPv4，源 MAC 地址: 80:05:88:46:b4:f9 目的 MAC 地址 3e:9a:fd:8b:e8:e7

5.1.4.物理层

Wireshark · 分组 38391 · WLAN

▼ Frame 38391: 537 bytes on wire (4296 bits), 537 bytes captured (4296 bits) on interface \Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6}

- ▼ Interface id: 0 (\Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6})
 - Interface name: \Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6}
 - Interface description: WLAN
 - Encapsulation type: Ethernet (1)
 - Arrival Time: Jul 2, 2021 09:33:49.545271000 中国标准时间
 - [Time shift for this packet: 0.000000000 seconds]
 - Epoch Time: 1625189629.545271000 seconds
 - [Time delta from previous captured frame: 0.029331000 seconds]
 - [Time delta from previous displayed frame: 0.029331000 seconds]
 - [Time since reference or first frame: 376.899273000 seconds]
 - Frame Number: 38391
 - Frame Length: 537 bytes (4296 bits)
 - Capture Length: 537 bytes (4296 bits)
 - [Frame is marked: False]
 - [Frame is ignored: False]
 - [Protocols in frame: eth:ethertype:ip:udp:data]
 - [Coloring Rule Name: UDP]
 - [Coloring Rule String: udp]

0000	3e 9a fd 8b e8 e7 80 05	88 46 b4 f9 08 00 45 00	>.....F...E..
0010	02 0b a4 c7 40 00 36 11	a6 90 df a6 97 60 0a 03@.6.
0020	76 80 1f 40 0f ad 01 f7	1c 31 02 3a 03 03 88 26	v..@....-1:....&
0030	94 20 69 b1 22 00 04 00	91 5d 50 6f a7 60 71 81	.i"....]Po`q.
0040	33 4f ec e8 20 5e 54 b6	b1 8f 25 e9 de 46 71 86	30..^T.-%..Fq.
0050	4d 68 66 cd a8 fb 2d bd	8b 62 e6 02 93 fc 60 f9	Mhf....b....`.

Close

接口 id 为 0 捕获日期为: Jul 2,2021 09:33:49, 帧序号为 38391, 帧长度为 537 字节, 捕获了 537 字节, 帧内封装的协议层次结构: eth:ethertype:ip:udp:data

5.2.HTTP 数据包详细分析

5.2.1.传输层

Wireshark · 分组 37342 · WLAN

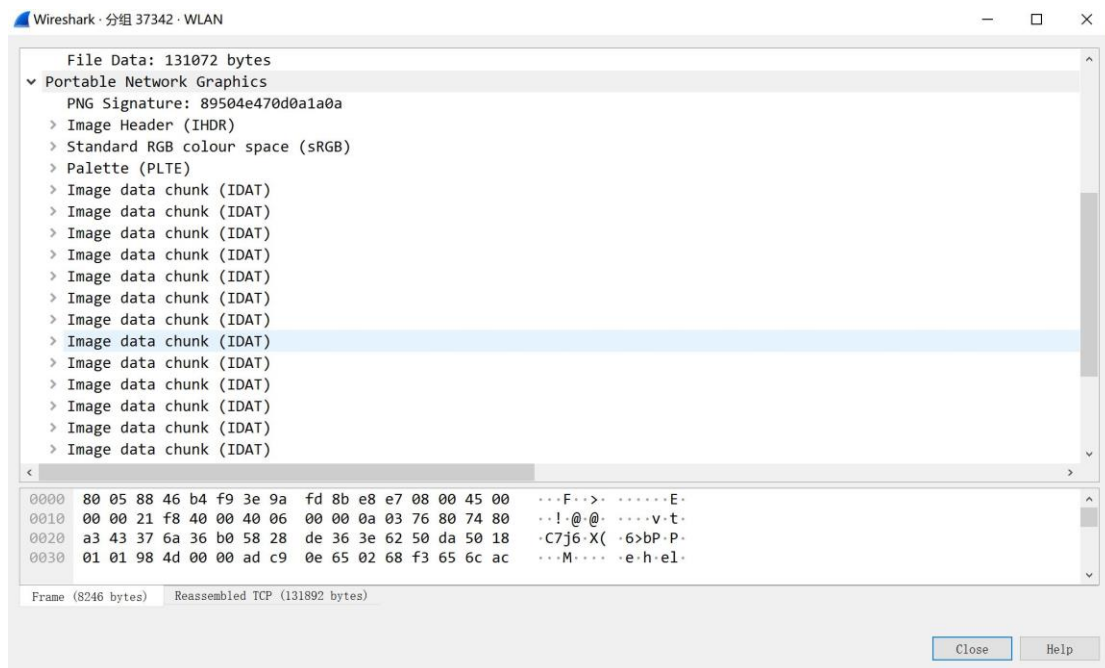
> Frame 37342: 8246 bytes on wire (65968 bits), 8246 bytes captured (65968 bits) on interface \Device\NPF_{EC241448-76F2-4C07-B320-DF491E8EA4D6}

- > Ethernet II, Src: 3e:9a:fd:8b:e8:e7 (3e:9a:fd:8b:e8:e7), Dst: RuijieNe_46:b4:f9 (80:05:88:46:b4:f9)
- > Internet Protocol Version 4, Src: 10.3.118.128, Dst: 116.128.163.67
- > Transmission Control Protocol, Src Port: 14186, Dst Port: 14000, Seq: 123701, Ack: 1, Len: 8192
- ▼ Hypertext Transfer Protocol
 - > [truncated]POST /cgi-bin/httpconn?htcmd=0x6ff0070&ver=5803&ukey=81E7B7D9FA9D261DFD98AE7014B40429837EC2417300C758401
 - Host:116.128.163.67\r\n
 - > Content-Length:131072\r\n\r\n
 - [Full request URI [truncated]: http://116.128.163.67/cgi-bin/httpconn?htcmd=0x6ff0070&ver=5803&ukey=81E7B7D9FA9D261DFD98AE7014B40429837EC2417300C758401]
 - [HTTP request 1/2]
 - [Response in frame: 37355]
 - [Next request in frame: 37358]
 - File Data: 131072 bytes
- > Portable Network Graphics
 - > [Malformed Packet: PNG]

00000000	50 4f 53 54 20 2f 63 67	69 2d 62 69 6e 2f 68 74	POST /cg i-bin/ht
00000010	74 70 63 6f 6e 6e 3f 68	74 63 6d 64 3d 30 78 36	tpconn?h tcmd=0x6
00000020	66 66 30 30 37 30 26 76	65 72 3d 35 38 30 33 26	ff0070&v er=5803&
00000030	75 6b 65 79 3d 38 31 45	37 42 37 44 39 46 41 39	ukey=81E 7B7D9FA9

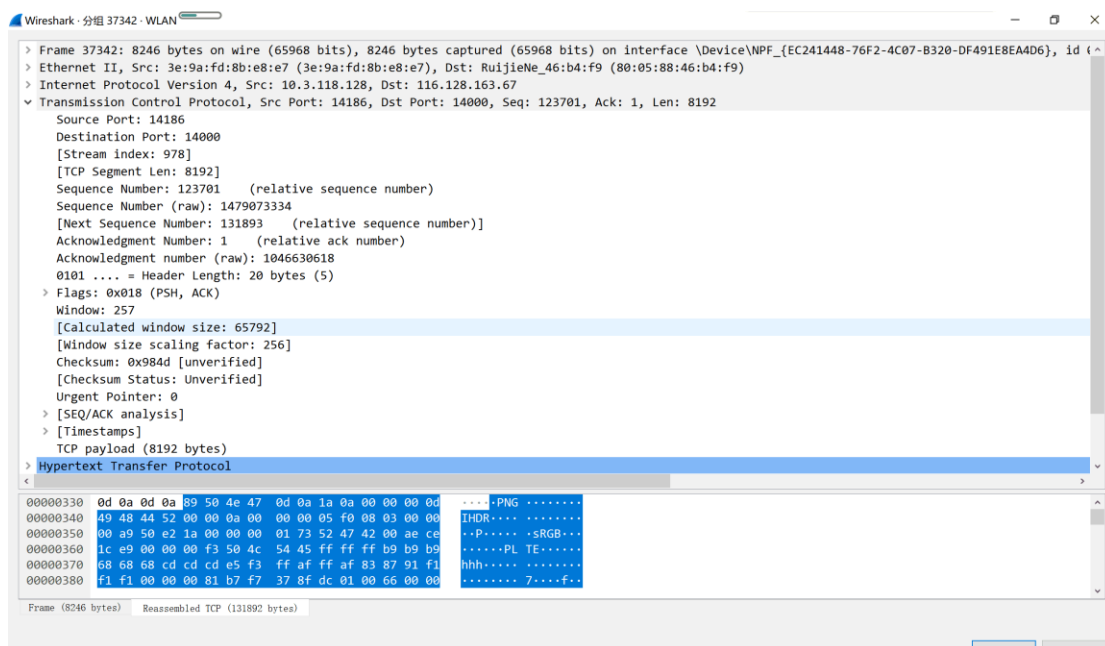
Frame (8246 bytes) Reassembled TCP (131892 bytes)

Close Help



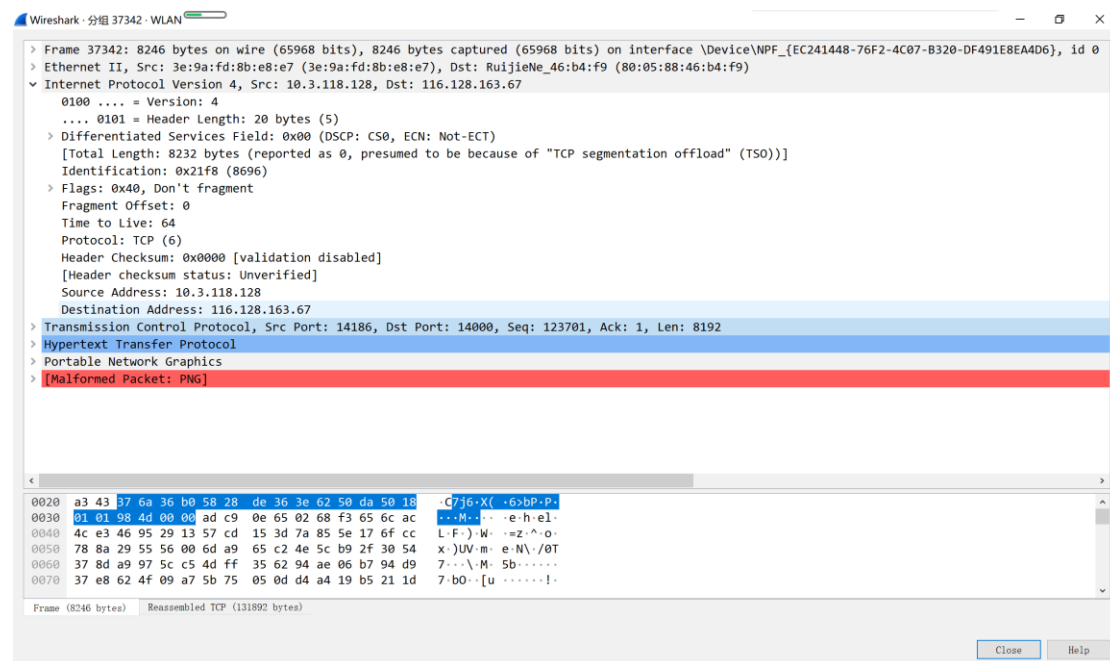
这一次应用层以 http 报文显示, 里面传递的东西很明确, 通过 GET 方式访问网站资源, 访问的主机名为 suggestion.baidu.com, 还有其他的一些浏览器相关的信息等等

5.2.2.TCP 层



可以看出 TCP 首部信息, 我们也可以从另一个方面推出 http 协议使用的运输层协议是 TCP, 源端口为 14186, 目的端口为 14000, 还有它的序号 131892 以及确认信号, 还可以看到标志位 Flags: 0x018, 窗口大小: 256, 首部长度为 20, 检验和是 0x8fd9, 紧急指针 Urgent pointer 置 0

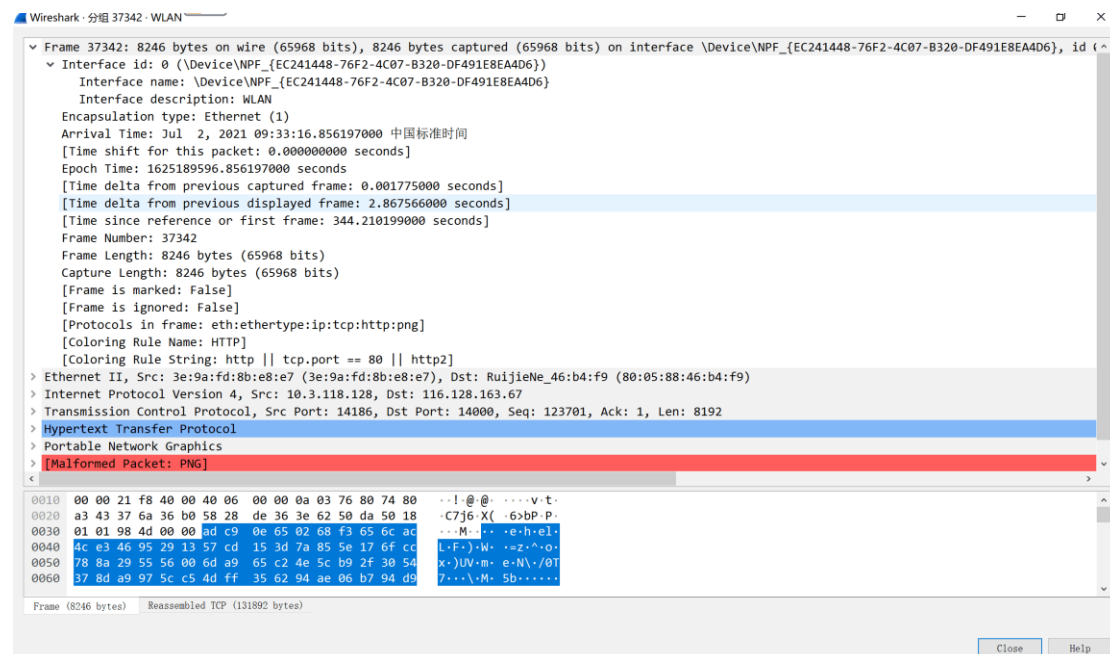
5.2.3.IP 层



IP 层的内容，可以清楚分析出他的首部信息，版本号为 4，首部长为 20 字节，还有标识 8696，

TTL 为 64，协议字段 6，代表了上层使用 TCP，下面就是源 IP 为 10.3.118.128，目的 IP 为 116.128.163.67

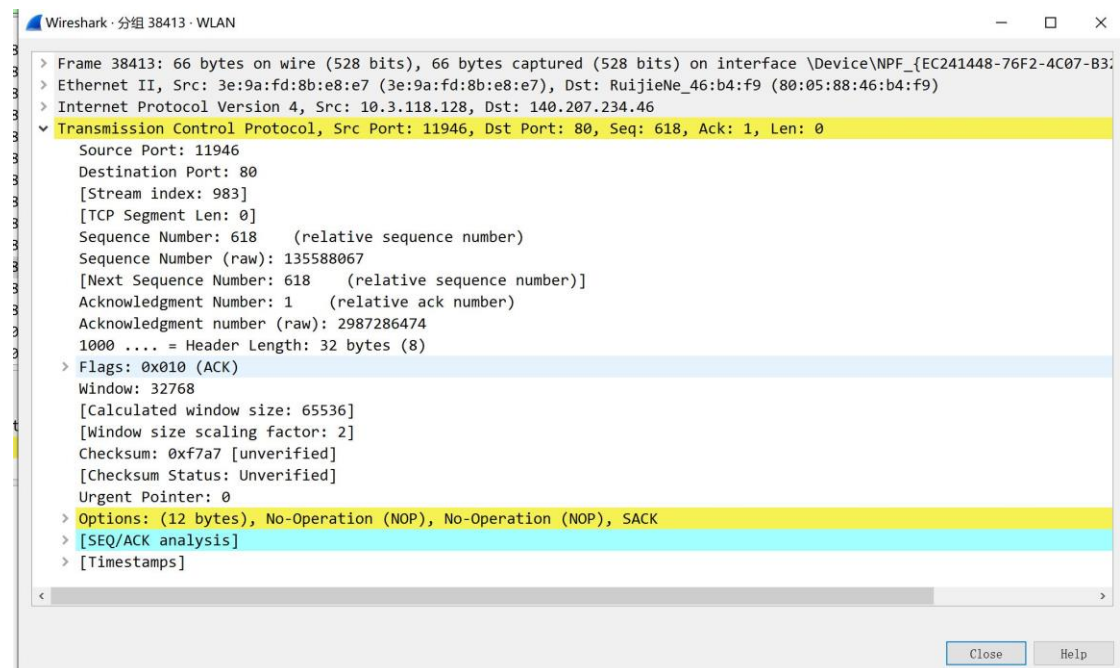
5.2.4.数据链路层



上层 IP 类型为 IPv4，源 MAC 地址: 3e:9a:fd:8b:e8:e7,目的 MAC 地址 80:05:88:46:b4:f9

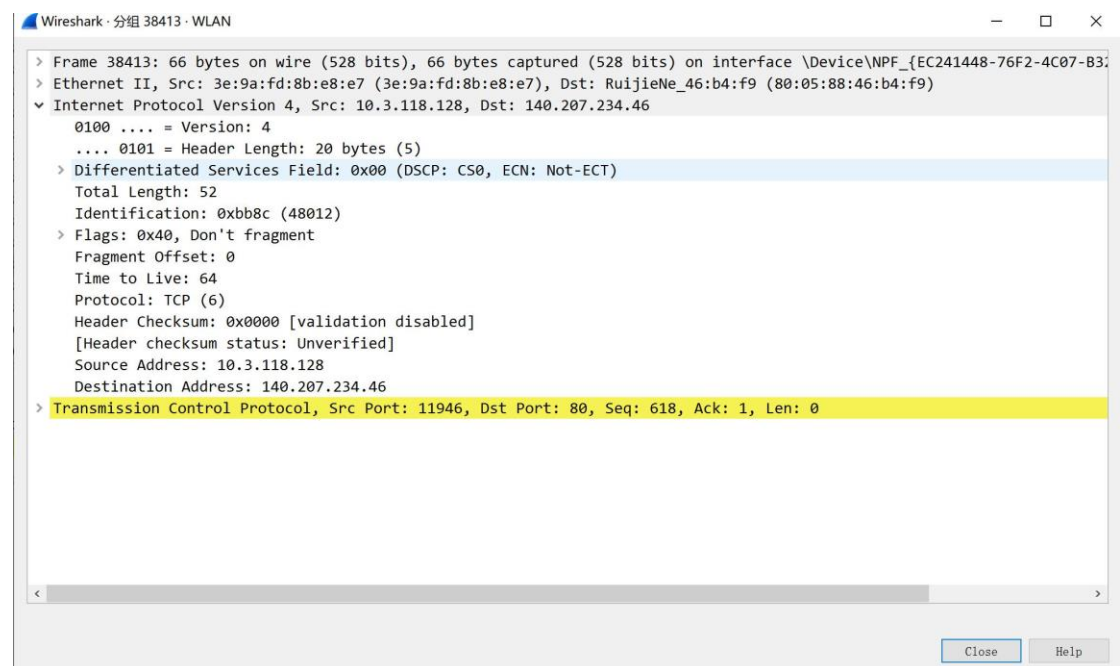
5.3.TCP 数据包详细分析

5.3.1.TCP 层



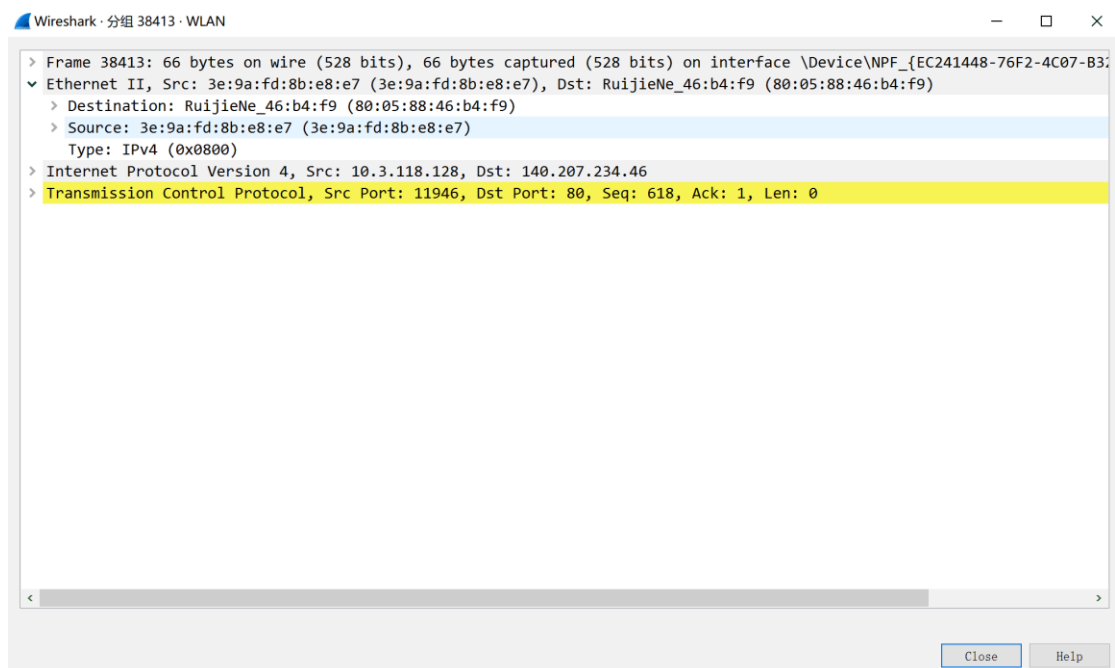
源端口为 11946，目的端口为 80，还有它的序号 0 以及确认信号 618，还可以看到标志位 Flags: 0x010，窗口大小:65536，首部长度为 32，检验和是 0x1577，紧急指针 Urgent pointer 置 0

5.3.2.IP 层



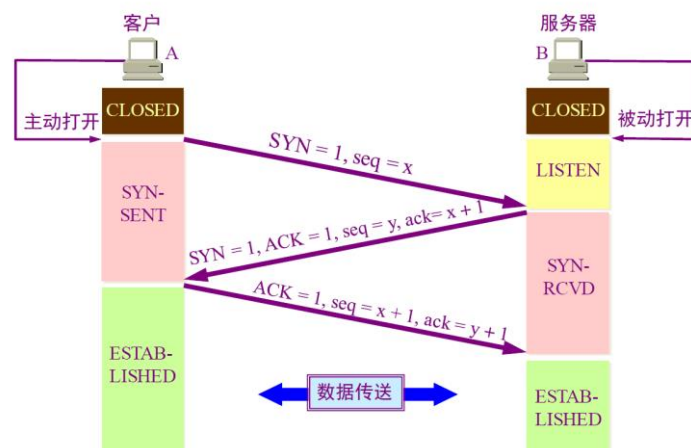
版本号为 4，首部长为 20 字节，还有标识 48012，TTL 为 64，协议字段 6，代表了上层使用 TCP，下面就是源 IP 为 10.3.118.128，目的 IP 为 140.207.234.46

5.3.3.数据链路层



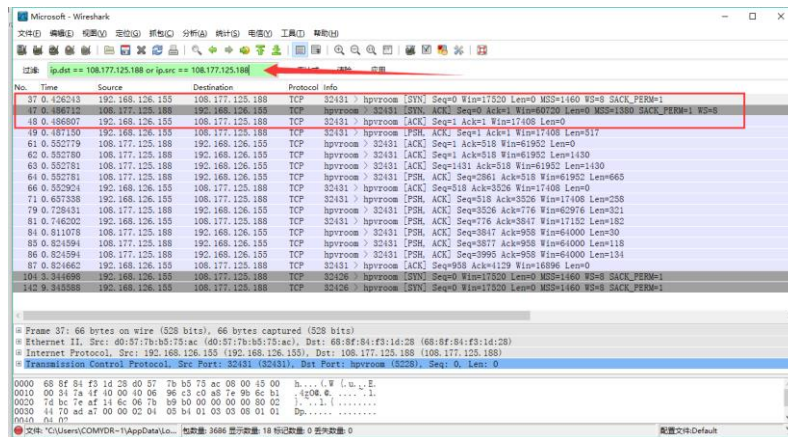
上层 IP 类型为 IPv4，源 MAC 地址: 3 e:9a:fd:8b:e8:e7,目的 MAC 地址 80:05:88:46:b4:f9

6.Transmission Control Protocol 连接建立三报文握手



用三报文握手建立 TCP 连接示意图

打开 Wireshark，选取网卡开始抓包。先在过滤填写 tcp，表示过滤出 TCP 协议的包，任取一个，例如本机与 108.177.125.188 的连接，则在过滤填写 ip.dst == 108.177.125.188 or ip.src == 108.177.125.188，这样，就可以过滤出（本机 → 108.177.125.188）或（108.177.125.188 → 本机）的包；



结合课程学习，可知前 3 个包为“三报文握手”，过程如下：

第一个 TCP 报文：客户端（本机）向服务器（108.177.125.188）发送连接请求包，标志位 SYN（同步序号）置为 1，序号 $seq = x = 0$ ；

```

Transmission Control Protocol, Src Port: 32431 (32431), Dst Port: hpvroom (5228), Seq: 0, Len: 0
  源 端口号: 32431(32431)
  目的端口号: hpvroom(5228)
  [Stream index: 20]
  Sequence number: 0 (relative sequence number)
  Header length: 32 bytes
  Flags: 0x02 (SYN)
  Window size: 17520
  Checksum: 0xad7 [validation disabled]
  Options: (12 bytes)
  
```

第二个 TCP 报文：服务器收到客户端发过来报文，由 $SYN=1$ 知道客户端要求建立联机。向客户端发送一个 SYN 和 ACK 都置为 1 的 TCP 报文，设置序号 $seq=y=0$ ，将确认号 ack 设置为客户端第一个 TCP 报文的序列号加 1，即 $ack = x+1 = 0+1 = 1$ ；

```

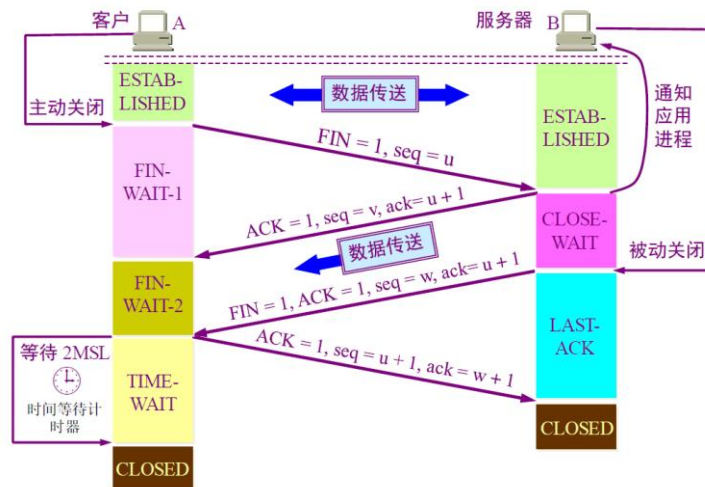
Transmission Control Protocol, Src Port: hpvroom (5228), Dst Port: 32431 (32431), Seq: 0, Ack: 1, Len: 0
  源 端口号: hpvroom(5228)
  目的端口号: 32431(32431)
  [Stream index: 20]
  Sequence number: 0 (relative sequence number)
  Acknowledgement number: 1 (relative ack number)
  Header length: 32 bytes
  Flags: 0x12 (SYN, ACK)
  Window size: 60720
  Checksum: 0x75f6 [validation disabled]
  Options: (12 bytes)
  [SEQ/ACK analysis]
  
```

第三个 TCP 报文：客户端收到服务器发来的包后检查确认号 ack 是否正确，即第一次发送的序号加 1 ($x+1=1$) 以及确认标志位 ACK 是否为 1，若正确，服务器再次发送确认包，ACK 标志位为 1，SYN 标志位为 0。确认号 $ack = y+1 = 0+1 = 1$ ，序号 $seq = x+1 = 1$ 。连接建立成功，可以传送数据了；

```

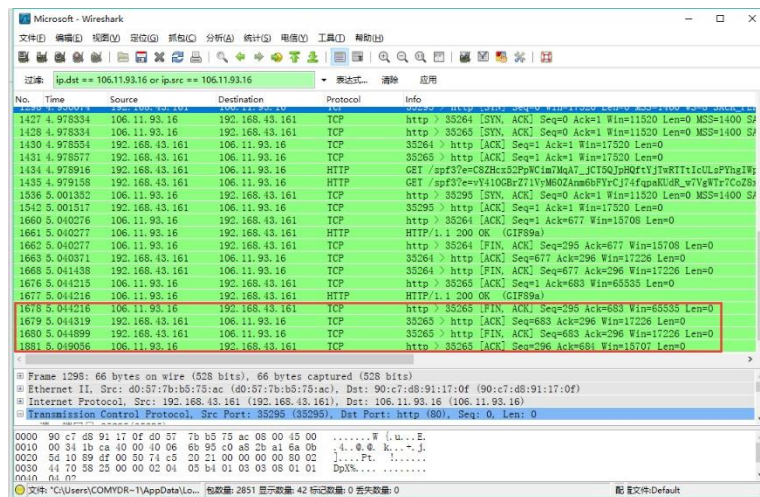
Transmission Control Protocol, Src Port: 32431 (32431), Dst Port: hpvroom (5228), Seq: 1, Ack: 1, Len: 0
  源 端口号: 32431(32431)
  目的端口号: hpvroom(5228)
  [Stream index: 20]
  Sequence number: 1 (relative sequence number)
  Acknowledgement number: 1 (relative ack number)
  Header length: 20 bytes
  Flags: 0x10 (ACK)
  Window size: 17408 (scaled)
  Checksum: 0xa366 [validation disabled]
  [SEQ/ACK analysis]
  
```

7. Transmission Control Protocol 连接释放四报文握手



用四报文握手释放 TCP 连接示意图

TCP 断开连接时，有四报文握手过程，如下图所示，Wireshark 截获到了四报文握手的四个数据包；



四报文握手过程如下：

- 第一个 TCP 报文：106.11.93.16 向本机发送连接释放请求，标志位 FIN 置为 1，序号 $seq = u = 295$ ；
- 第二个 TCP 报文：本机收到 FIN 报文后，发回一个 ACK 报文，序号 $seq = v = 683$ ，确认号 $ack = u + 1 = 296$ ；
- 第三个 TCP 报文：本机关闭与 106.11.93.16 的链接，发送一个 FIN，序号 $seq = v = 683$ ，确认号 $ack = u + 1 = 296$ ；
- 第四个 TCP 报文：106.11.93.16 收到本机 FIN 之后，发回 ACK，序号 $seq = u + 1 = 296$ ， $ack = v + 1 = 684$ ，连接释放。

实验体会：

通过本次实验，我对 Wireshark 软件的基本操作、过滤器填写、追踪流有了更深入的了解；对 IP、UDP、TCP、HTTP 等协议的首部有了更好的掌握；对 TCP 协议连接建立、数据传送、连接释放过程有了直观的了解。

通过 Wireshark，结合网络上的资料，我对 TCP 协议连接建立（三报文握手）和连接释放（四报文握手）过程有了更直观的认识。在抓包过程中，我遇到了很多问题，例如包的真正到达时间不是严格和 seq 顺序一致，还和网络环境等复杂因素有关系，由此有课本上的“超时重传”、“确认丢失”、“确认迟到”等问题

Wireshark 真的是一款非常强大的抓包工具，其功能是截取网络封包，并尽可能显示出最为详细的资料。Wireshark 使用 WinPCAP 作为接口，直接与网卡进行数据报文交换。它的很多功能我在本次实验中还没有用到，在将来，我会更深入地去学习和了解这些网络分析用的工具软件，提高网络分析、网络安全的能力。