深 圳 大 学 实 验 报 告

课程名称：计算机网络

实验项目名称：数据包抓取与分析

学院：计算机与软件学院

专业：计算机科学与技术

指导教师：谢瑞桃

报告人：王云舒 学号：2018152044 班级：计科 03

实验时间：2021/3/29

实验报告提交时间：2021/5/29

教务处制

实验目的：

学习安装、使用协议分析软件，掌握基本的数据报抓取、过滤和分析方法，能分析 HTTP、TCP、ICMP 等协议。

实验环境：

使用具有 Internet 连接的 Windows 操作系统；抓包软件 Wireshark。

实验内容：

（1）安装学习 Wireshark 软件；

（2）抓包与分析 HTTP 协议；

（3）分析 TCP 协议；

（4）分析 TCP 三次握手；

（5）分析 ICMP 协议。

实验步骤：

1 安装学习 Wireshark 软件

Wireshark 是世界上最广泛使用的网络协议分析器，下载官网地址：https://www.wireshark.org/ 。软件使用手册：https://www.wireshark.org/download/docs/user-guide.pdf。

1.1 运行软件并捕获分组

运行 Wireshark，初始界面如图 1.1.1 所示。从接口列表中选择要捕获的接口，双击即可开始捕获。（实验时根据哪个网络有后面的流量，选择具体捕获哪个接口）

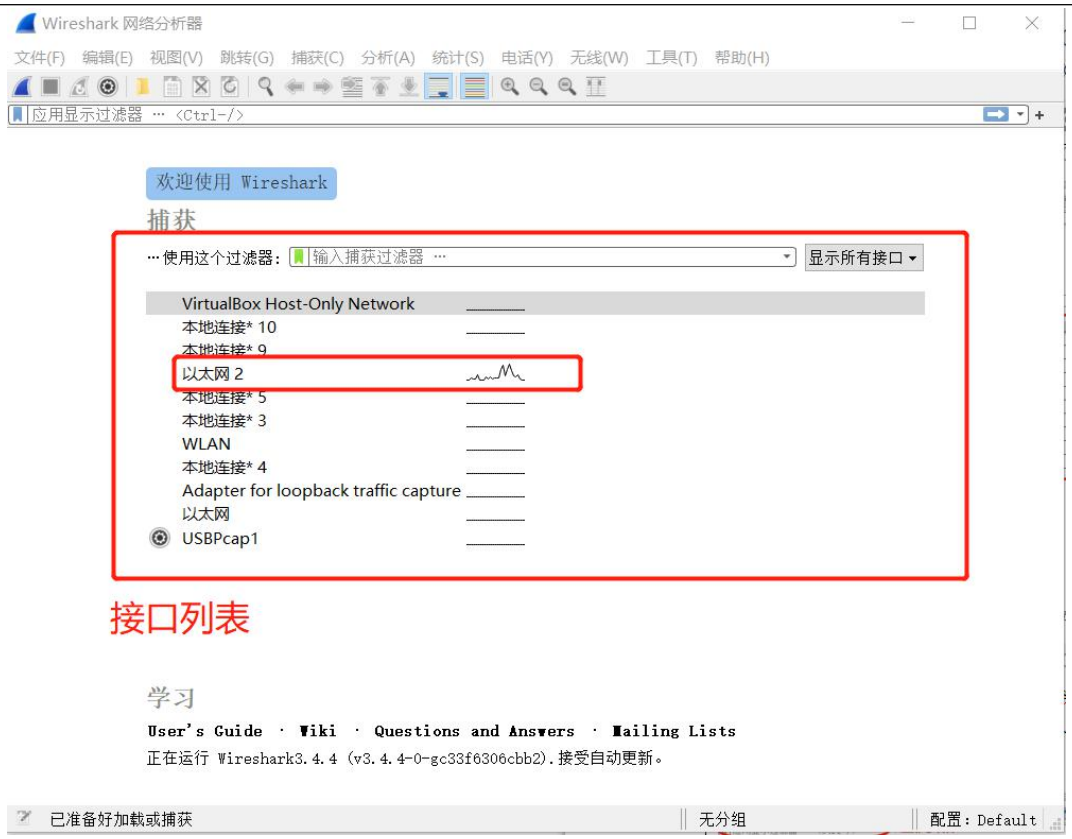


图 1.1.1 Wireshark 的主界面

选择以太网 2 并开始捕获分组，捕获分组的状态如图 1.1.2 所示。

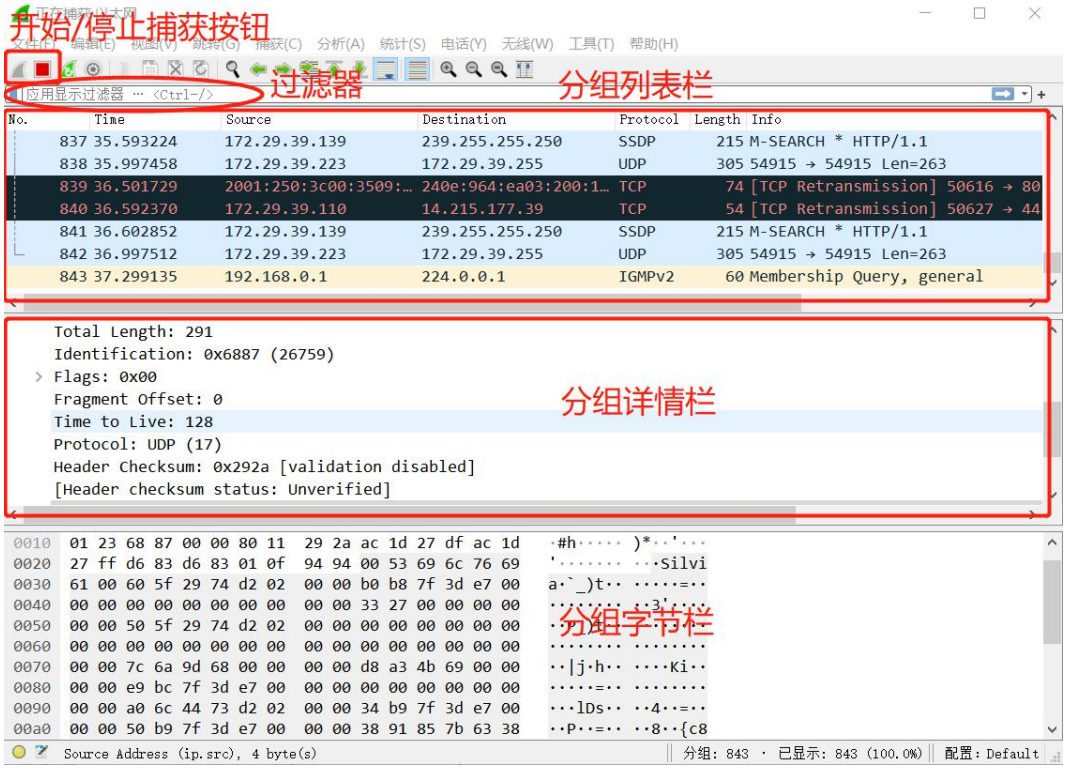


图 1.1.2 Wireshark 捕获分组

在分组列表栏里，单击某一行即可在分组详情栏和分组字节栏中看到该行的具体信息。如图 1.1.3 所示，图 1.1.3 展示了随机选择的一个的分组其分组详情栏与分组字节栏内容，这条分组没有应用层信息，应用层信息可以为 Hypertext Transfer Protocol，后面在抓包与分析 HTTP 协议时会更详细的分析。

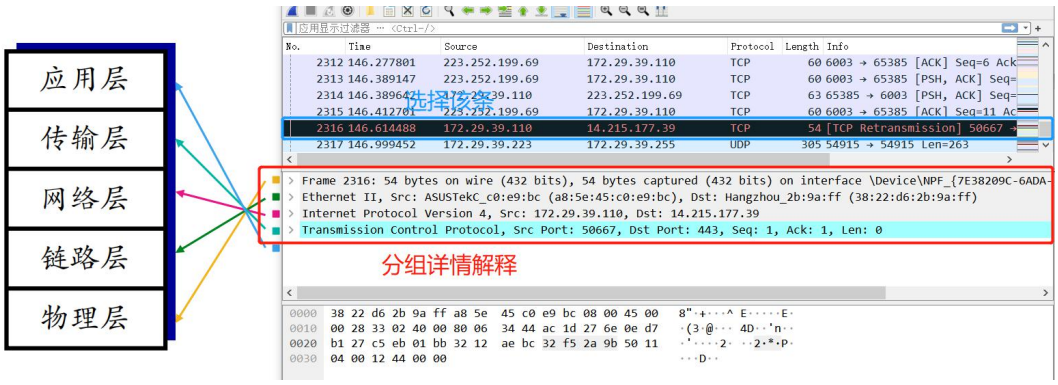


图 1.1.3 分组详情栏的解释

1.2 学习使用过滤器

过滤包括协议过滤、IP 地址过滤、模式过滤和端口过滤等。

协议过滤以 HTTP 为例，捕获结果如图 1.2.1 所示：

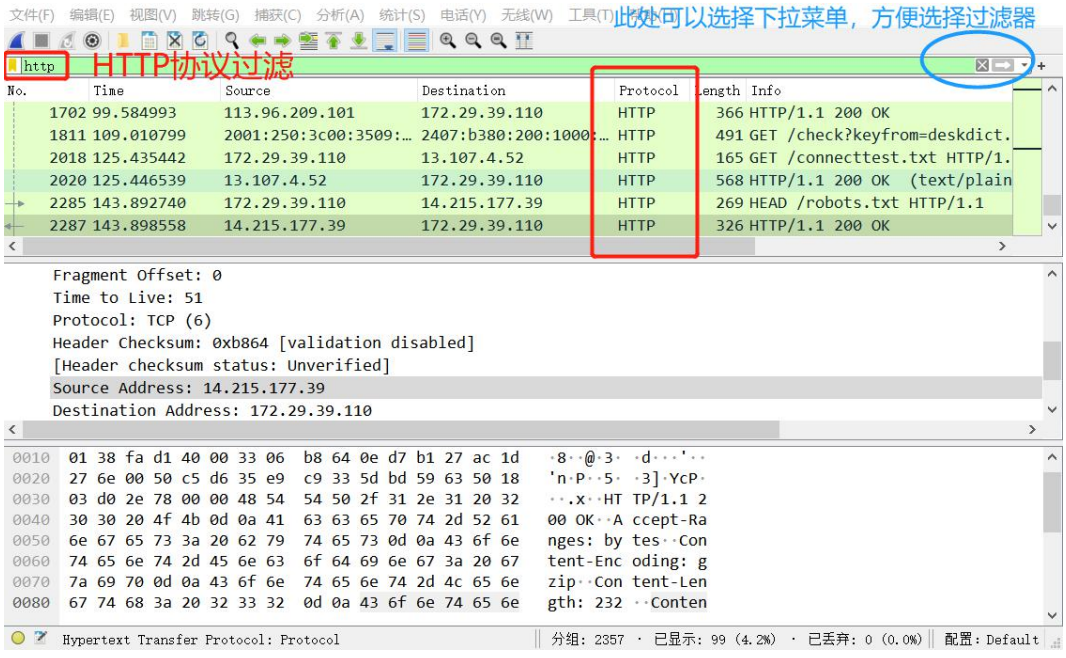


图 1.2.1 HTTP 协议过滤

IP 地址过滤以 ip.src== 172.29.39.110 and ip.dst==14.215.177.39 为例，含义为过滤源地址是 172.29.39.110 并且目的地址是 14.215.177.39（百度的 IP）的分组。捕获的结果如图 1.2.2 所示。

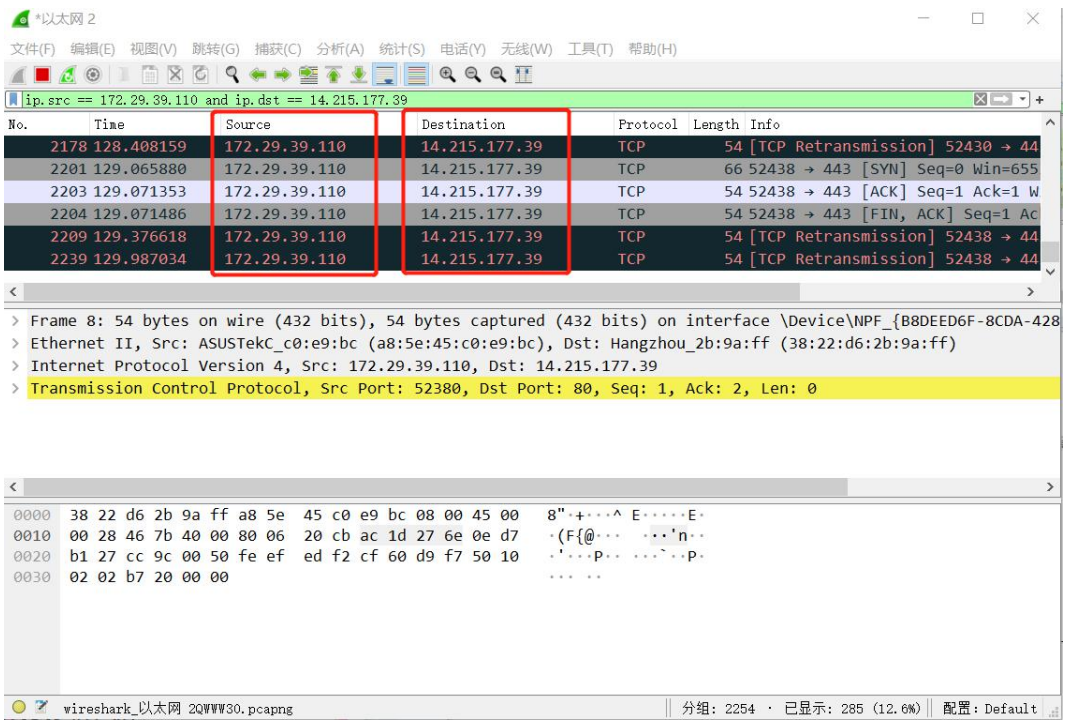


图 1.2.2 IP 地址过滤

模式过滤以 http.request.method==”GET”为例，含义为过滤 http 请求方法是 GET 的分组。结果如图 1.2.3 所示；除了 GET 方法外，还有其他方法如 POST 方法，如图 1.2.4 所示。

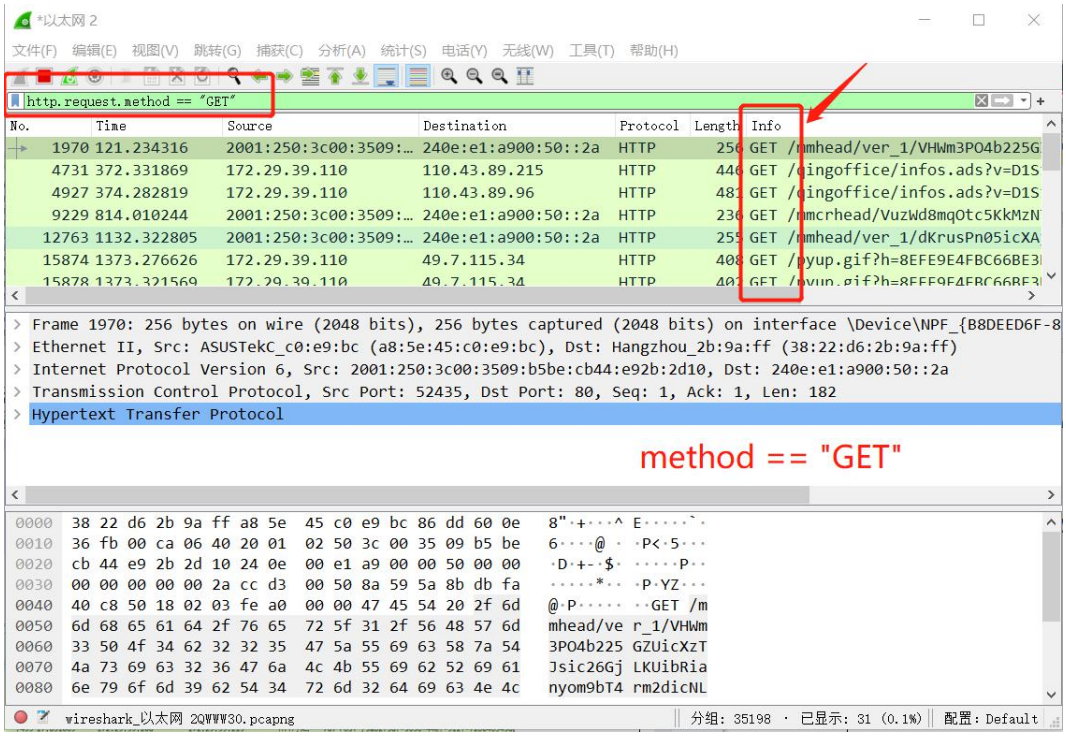


图 1.2.3 模式过滤（GET）

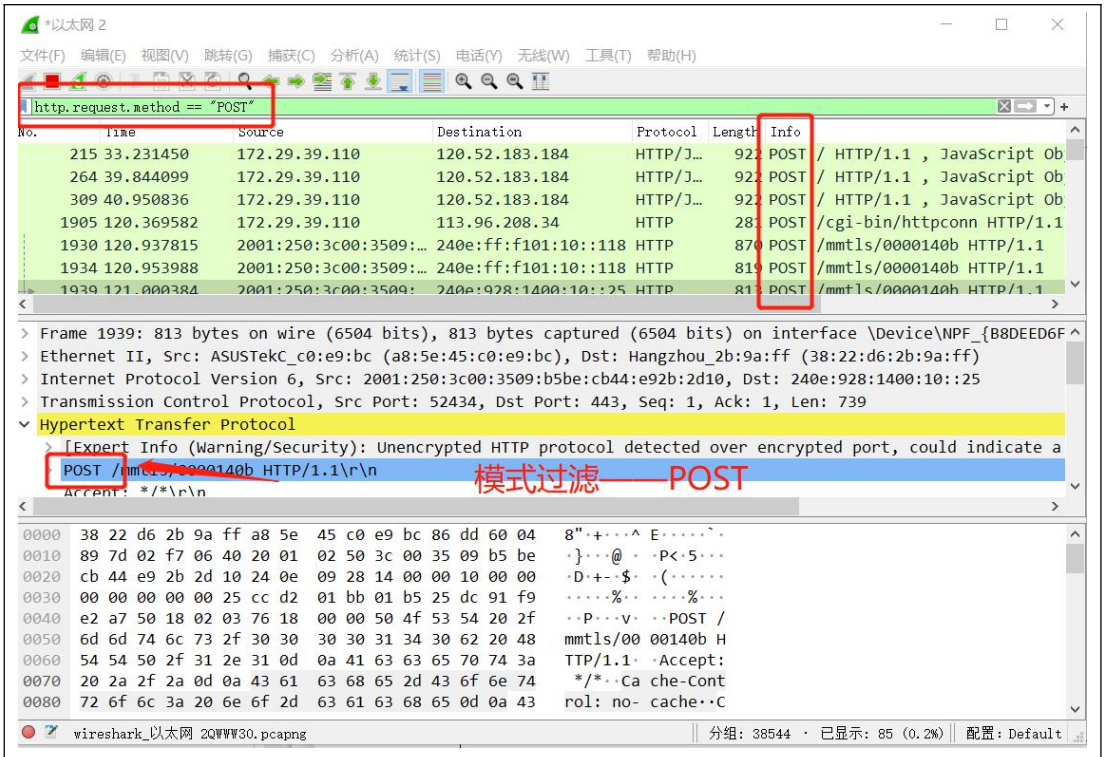


图 1.2.4 模式过滤（POST）

端口过滤以 tcp.port == 80 为例，含义为过滤 TCP 端口号是 80 的分组，如图 1.2.5、1.2.6 所示。这里只要端口为 80 就会筛选，包括了 Src 和 Dst。

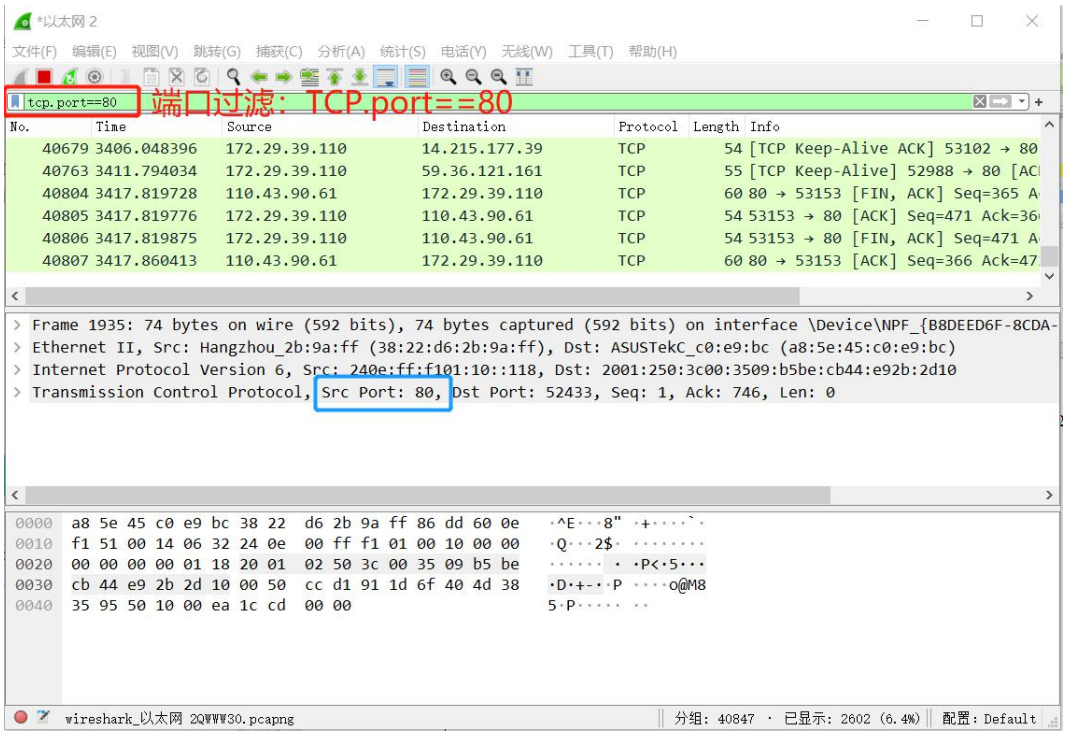


图 1.2.5 端口过滤（tcp.port==80，该图为 Src）

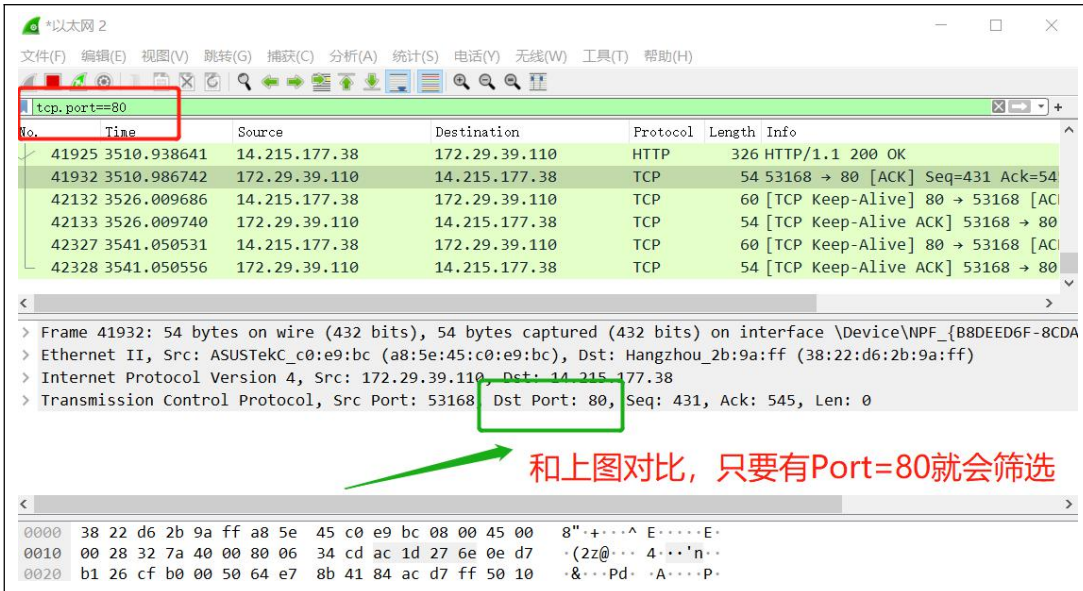


图 1.2.6 端口过滤（tcp.port==80，该图为 Dst）

2 抓包与分析 HTTP 协议

2.1 开启 Wireshark 抓包

开启 Wireshark 抓包，在过滤器中输入 http，即过滤 http 协议的分组；

接下来打开浏览器，**输入一个网址，以 SZUOJ：172.31.221.67 为例（该网站为 http 协议，注意 http 与 https 不同，当前绝大多数网址已经都在使用 https）**；Wireshark 返回的结果如图 2.1.1 所示，可以看到其分组列表栏中出现了 HTTP 协议分组。

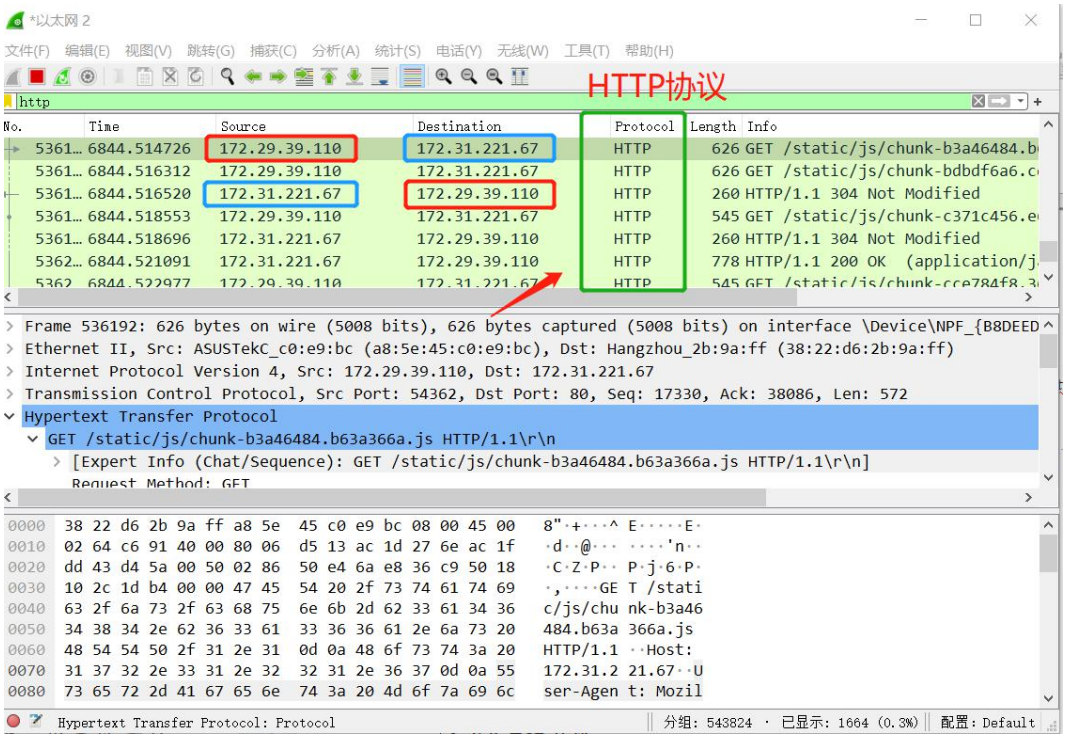


图 2.1.1 观察到 Wireshark 分组列表栏中出现 HTTP 协议分组

2.2 分析 HTTP 协议

分析哪些分组是前一步浏览网页发生可以根据 IP 地址来判断。当 IP 中包含 172.31.221.67 时，该分组就是由于前一步打开 OJ 网页而产生；单击任意分组，分组详情栏显示其具体信息。

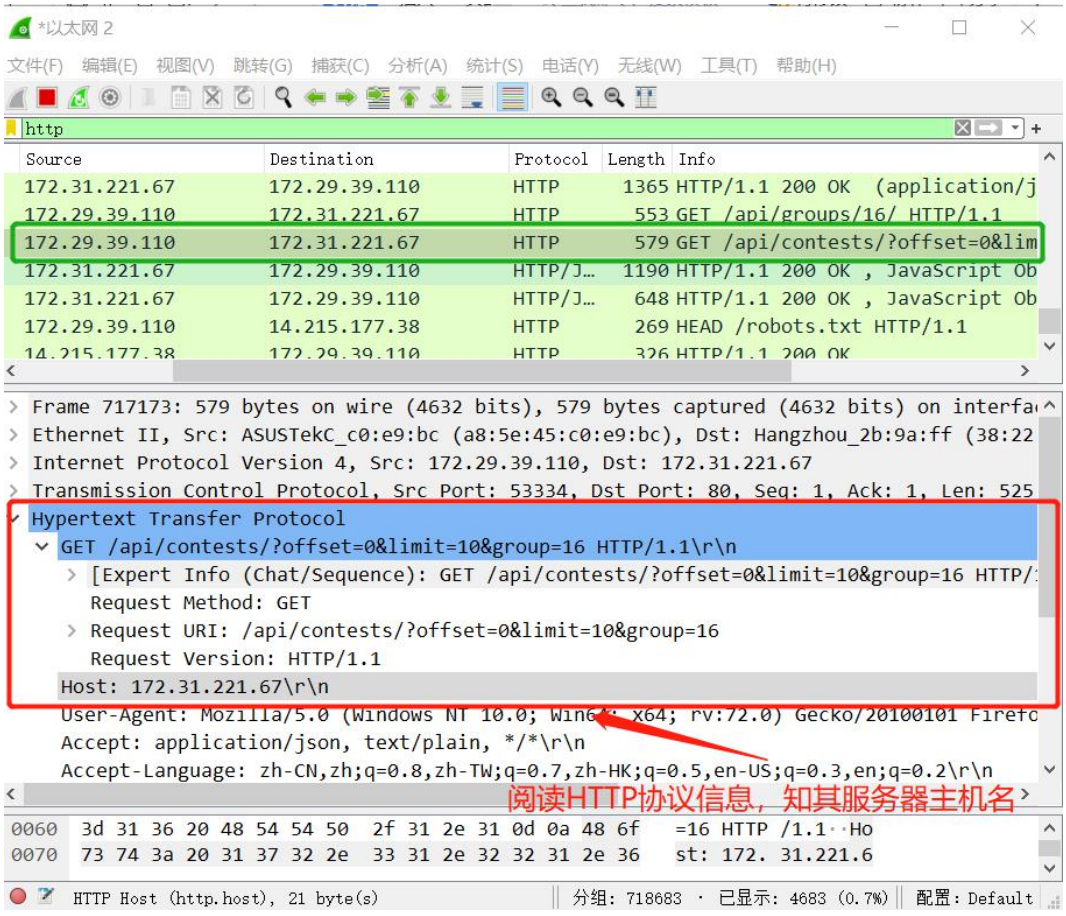


图 2.2.1 阅读 HTTP 协议信息，知其服务器主机名

对于所得的分组，获知其此次通信的源 IP 地址和目的 IP 地址。可以看到源 IP 地址为 172.29.39.110（即自己的电脑），目的 IP 地址为 172.31.221.67。

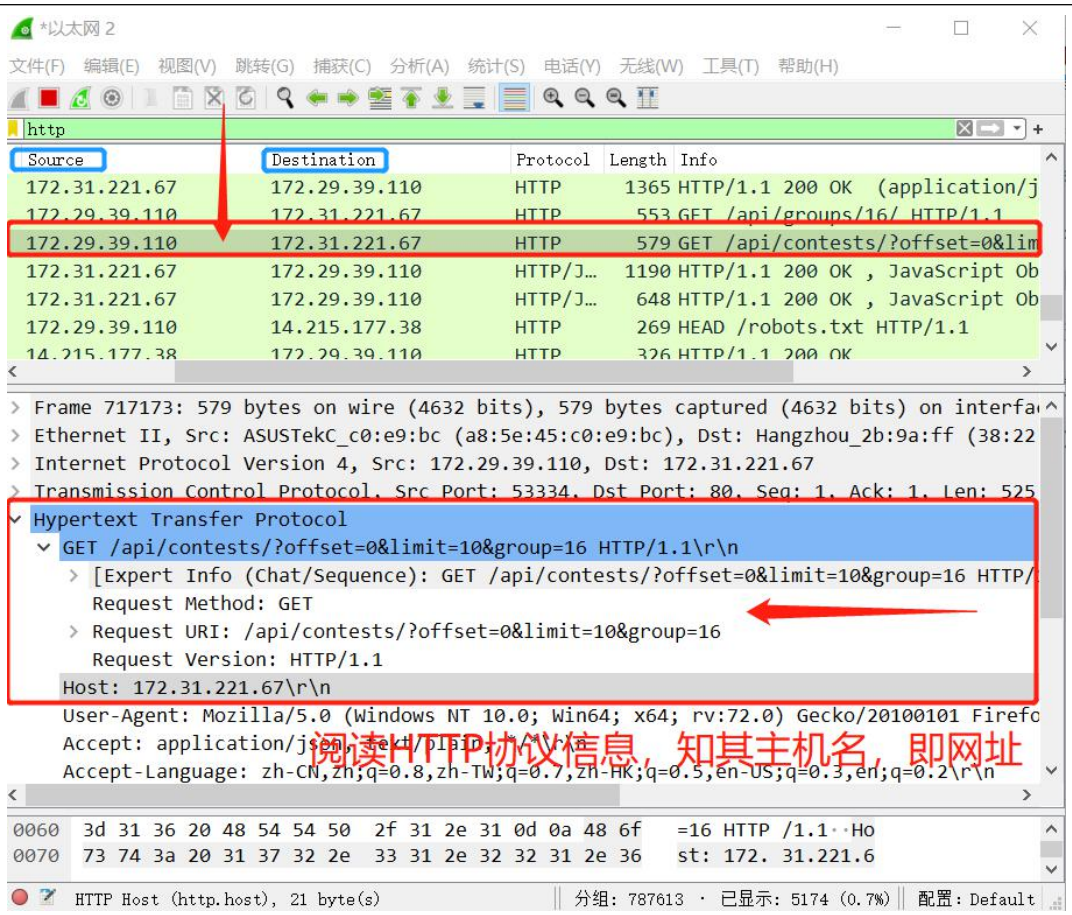
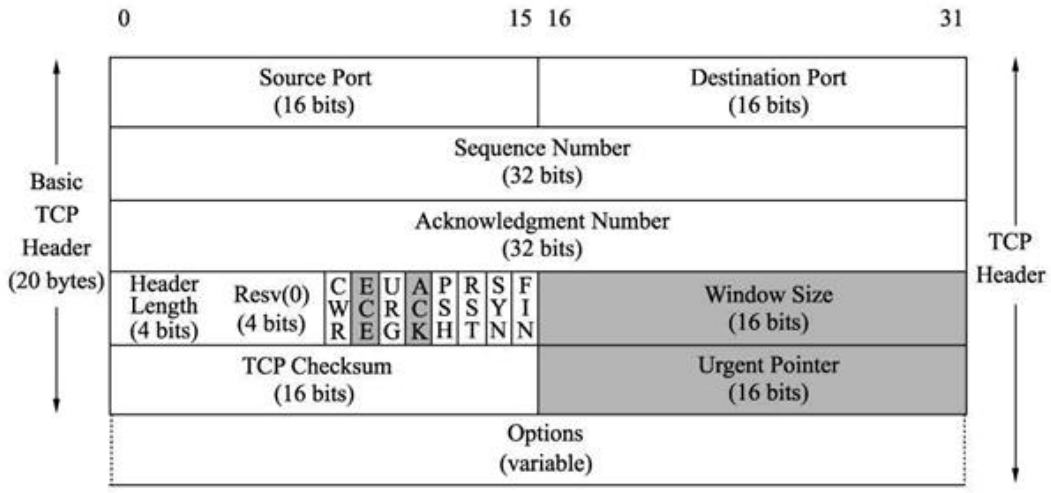


图 2.2.2 获知通信的源 IP 地址和目的 IP 地址

3 分析 TCP 协议



3.1 分析 TCP 协议信息

对 Wireshark 中的分组信息，分析其 TCP 协议信息，如图 3.1.1 所示，图中给出其源端口号与目的端口号、序列号、确认号、报头长度、标志位、窗口大小、校验和与最后的数据标注，解释了 TCP 协议信息中每个部分的含义。

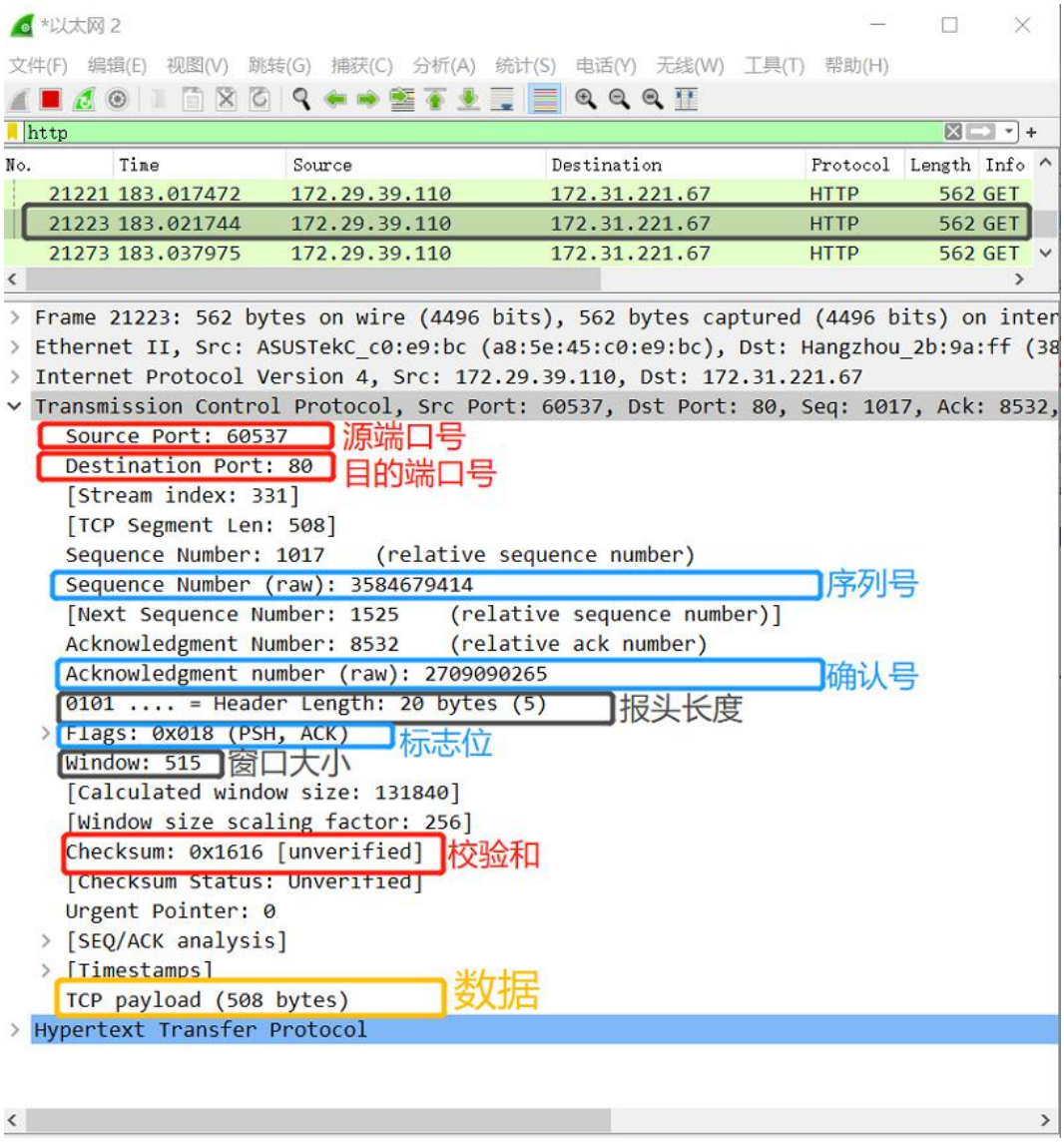


图 3.1.1 分析 HTTP 协议传输中的 TCP 协议信息

3.2 追踪分组的 TCP 流

对图 3.1.1 的分组，追踪其 TCP 流。点击右键，从下拉菜单中选择 TCP 流，如图 3.2.1 所示。在选择后上方的筛选条件将变为 tcp.stream eq 331；

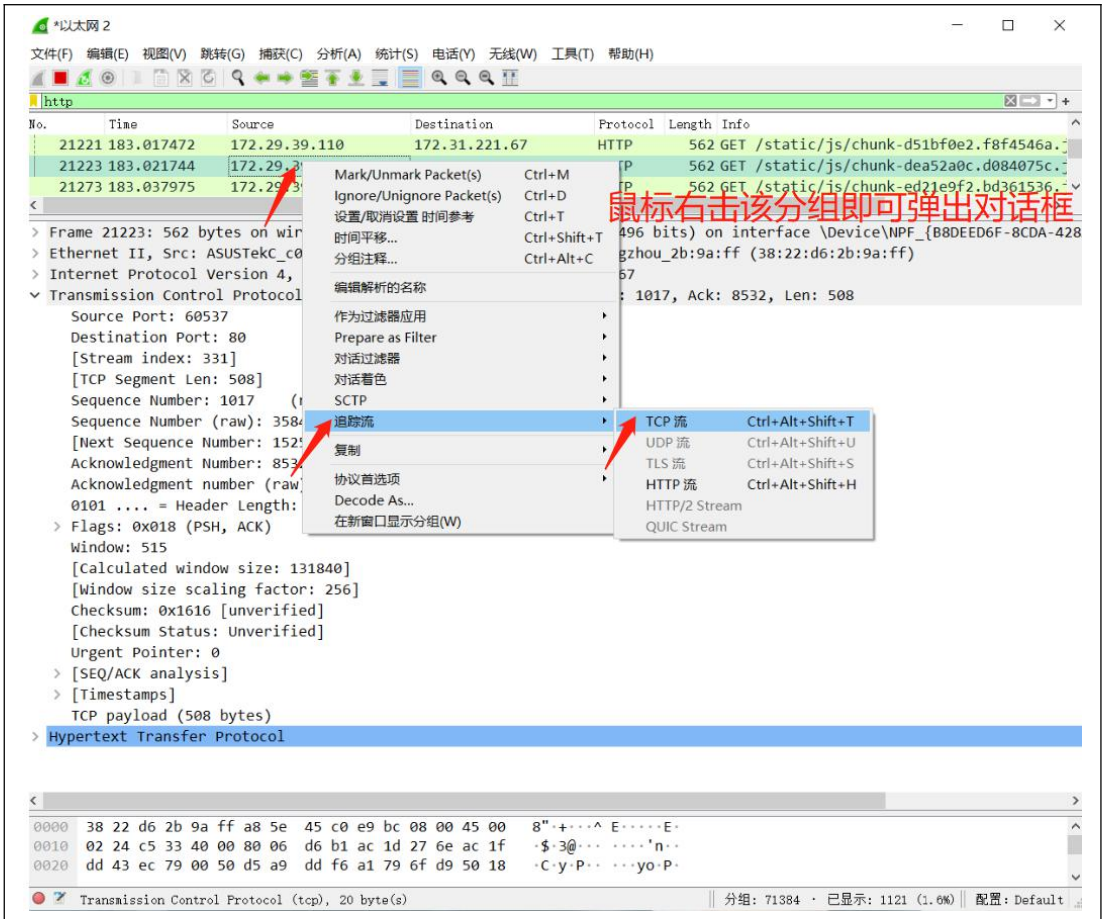


图 3.2.1 右击分组追踪 TCP 流

接下来就可以去寻找 TCP 建立连接的分组。根据原理：1）TCP 连接建立应该在 HTTP GET 请求之前完成；2）**TCP 建立连接时会设置标志位 SYN**（即第一次握手：客户端将同步号 SYN 置为 1，随机产生一个值 seq=x，将该数据包发送给服务端，客户端进入 SYN_SENT 状态，等待服务端确认）。因此筛选后按时间顺序，滑动到最上方即可找到建立连接的分组，如图 3.2.2 所示。

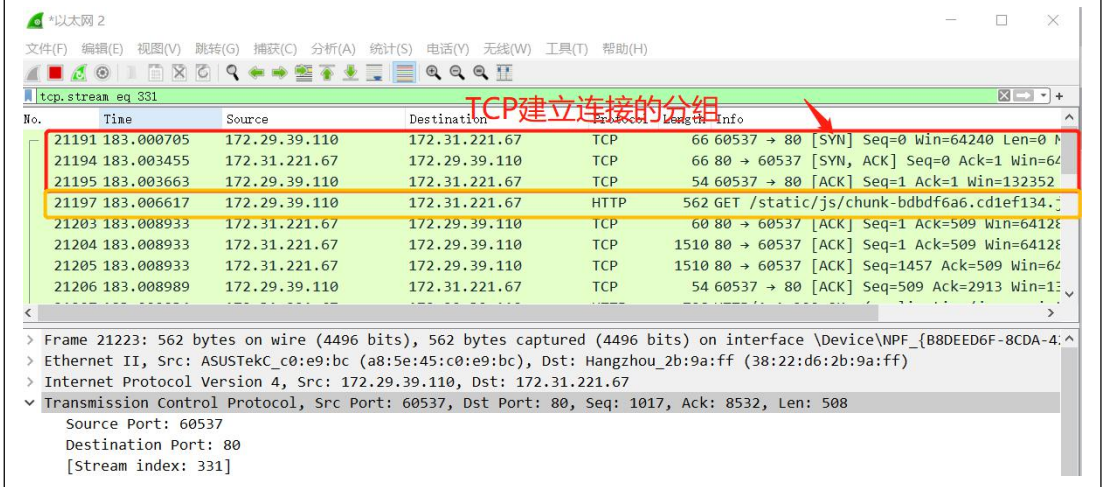


图 3.2.2 找到 TCP 建立连接的分组

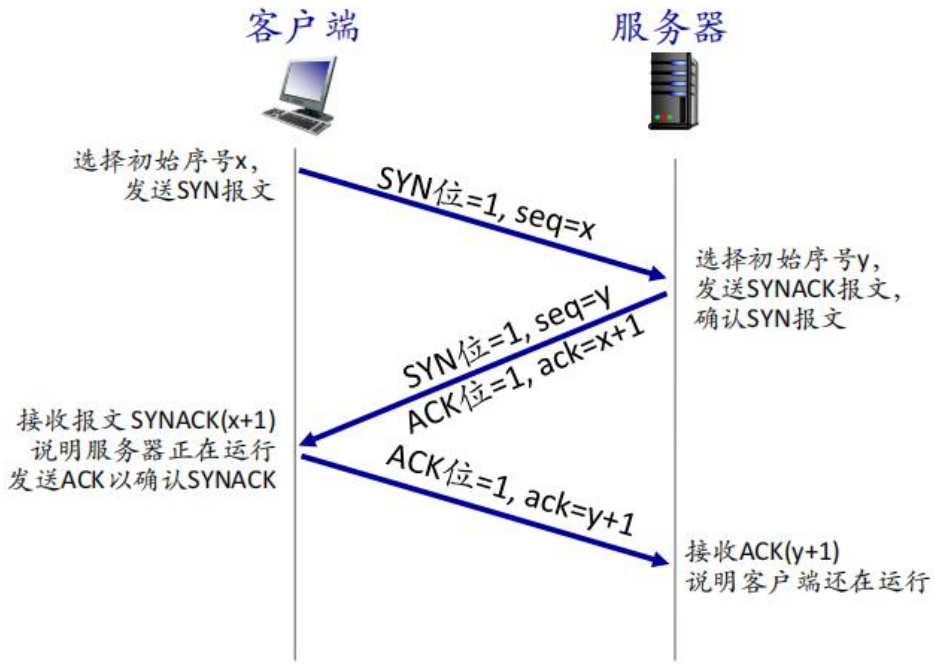
4 分析 TCP 三次握手

TCP 三次握手建立连接和数据传输的过程：

第一次握手：客户端将同步号 SYN 置为 1，选择初始序号 seq=x，将该数据报发送给服务器，客户端进入 SYN_SENT 状态，等待服务器确认；

第二次握手：服务器收到数据报后，由 SYN=1 知道客户端请求建立连接，将标志位 SYN 和 ACK 都置为 1，ack 设为 x+1，并选择初始序号 y，将该数据报发送给客户端以确认连接请求，服务器进入 SYN_RCVD 状态；

第三次握手：客户端收到确认后，检查 ACK 是否为 1 以及 ack 是否为 x+1，如果正确则将 ACK 设为 1，ack 设为 y+1，将该数据报发送给服务器，服务器检查如果 ACK=1，ack=y+1 则建立连接成功，客户端和服务器都进入 ESTABLISHED 状态，完成三次握手，开始数据传输。



三次握手的示意图

4.1 第一次握手

TCP 连接第一次握手，客户端发送请求信息，请求建立 TCP 连接，发送一个序列号（SYN）到服务器。如图 4.1.1 所示，SYN 为 1，序号为 358467897。

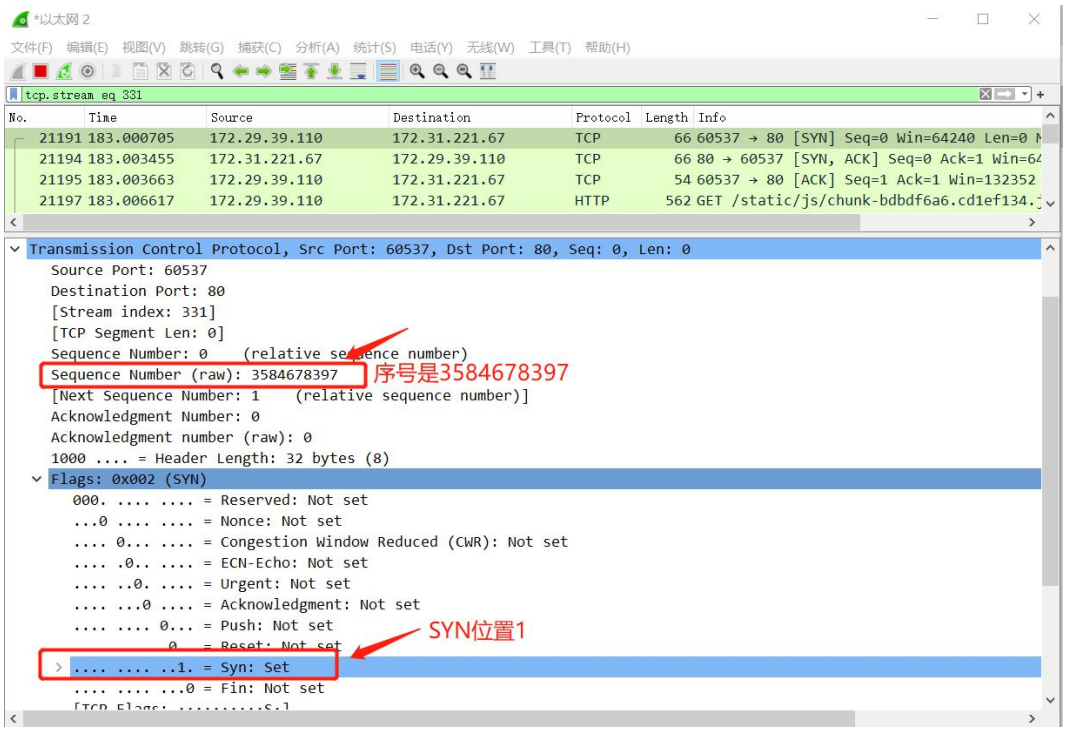


图 4.1.1 分析 TCP 三次握手，第一次握手（SYN）

4.2 第二次握手

服务器收到信息后，返回一个 ACK=1 确认收到客户端信息，其确认号应该为 x+1。如图 4.2.1 所示，和图 4.1.1 相比，ACK 从 not set 变为 set，确认号为图 4.1.1 中序号+1。

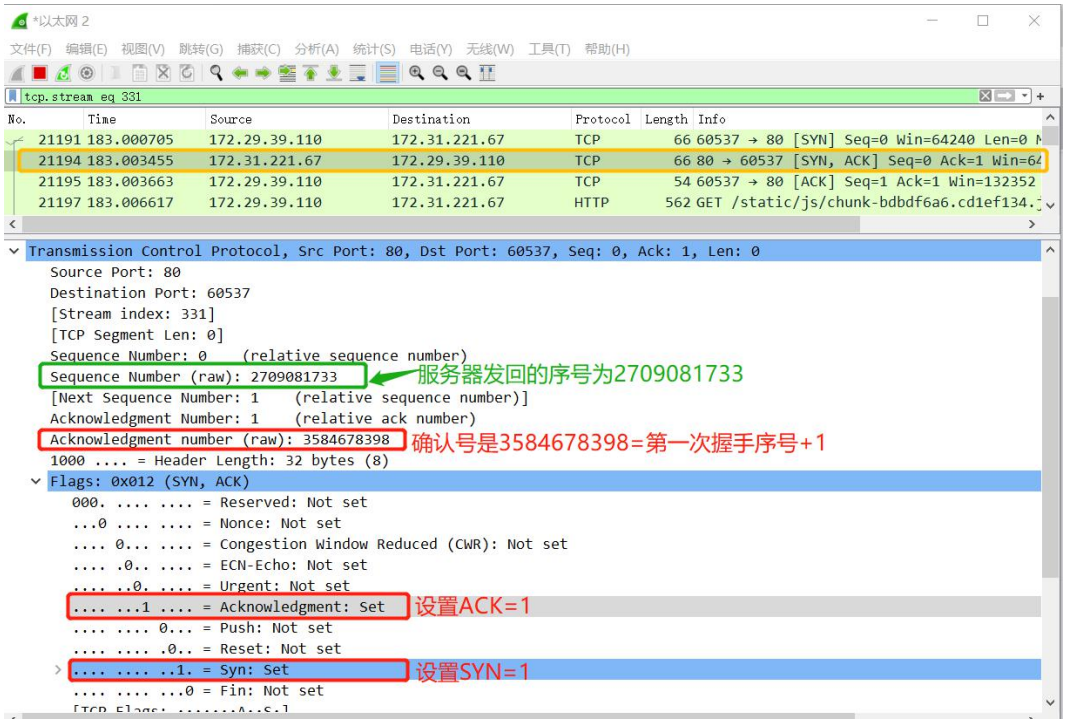


图 4.2.1 分析 TCP 三次握手，第二次握手（SYNACK）

4.3 第三次握手

第三次握手时客户端向服务器发送一个 ACK，并发送确认号 y+1，说明客户端确认和服务器建立 TCP 连接。如图 4.3.1 所示，客户端（端口：60537）向服务器（端口：80）发送的 ACK=1，确认号 2709081734 为图 4.2.1 的序号+1，第三次握手成功后即可建立连接，下面就是 HTTP 的 GET 方法了。

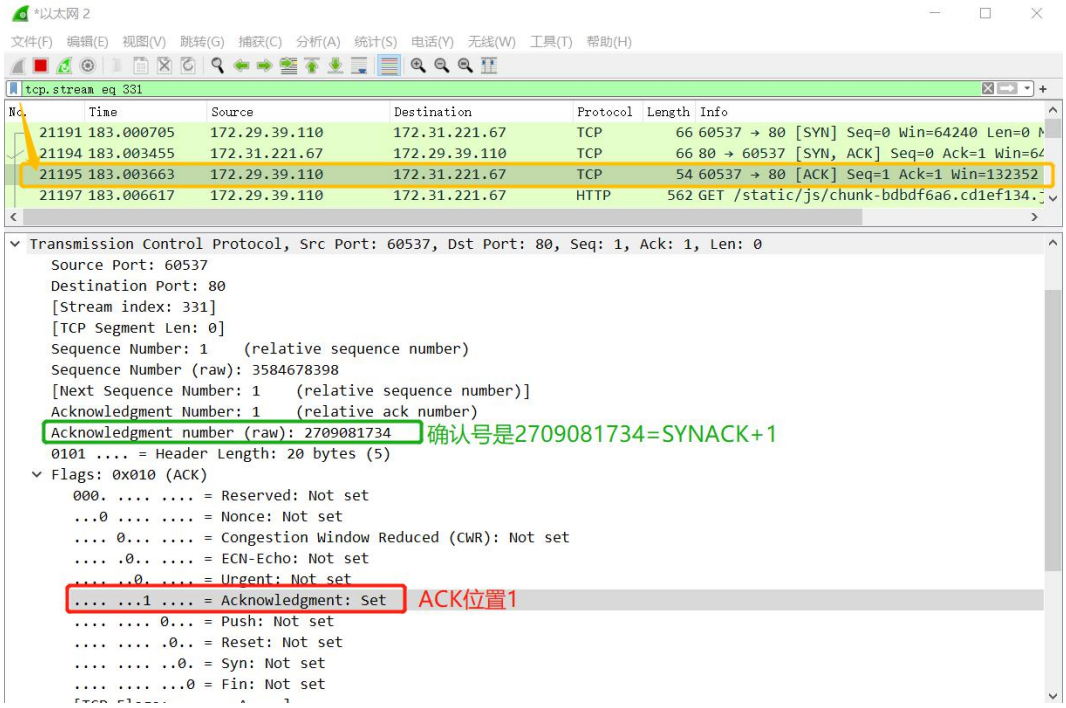
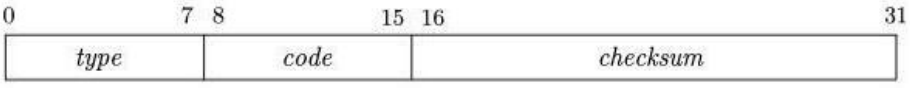


图 4.3.1 分析 TCP 三次握手，第三次握手（ACK）

5 分析 ICMP 协议

ICMP 协议（Internet Control Message Protocol）是一个网络层协议，一个新搭建好的网络，往往需要先进行一个简单的测试，来验证网络是否畅通；但是 IP 协议并不提供可靠传输，如果丢包其不能通知传输层是否丢包以及其原因。这时就需要一种协议来完成这样的功能——ICMP 协议。



ICMP 报文格式

在 Wireshark 过滤栏里输入 icmp（这是 ping 指令使用的协议），然后在 Powershell 里使用 ping 指令，得到的结果如图 5.1 所示。可以看到 ping 后 Wireshark 分组列表栏出现了被 ping 网址（百度）的 IP，协议为 ICMP，内容为 ping 指令的请求与回答。当没有丢失时，request 与 reply 均为默认的 4 组。

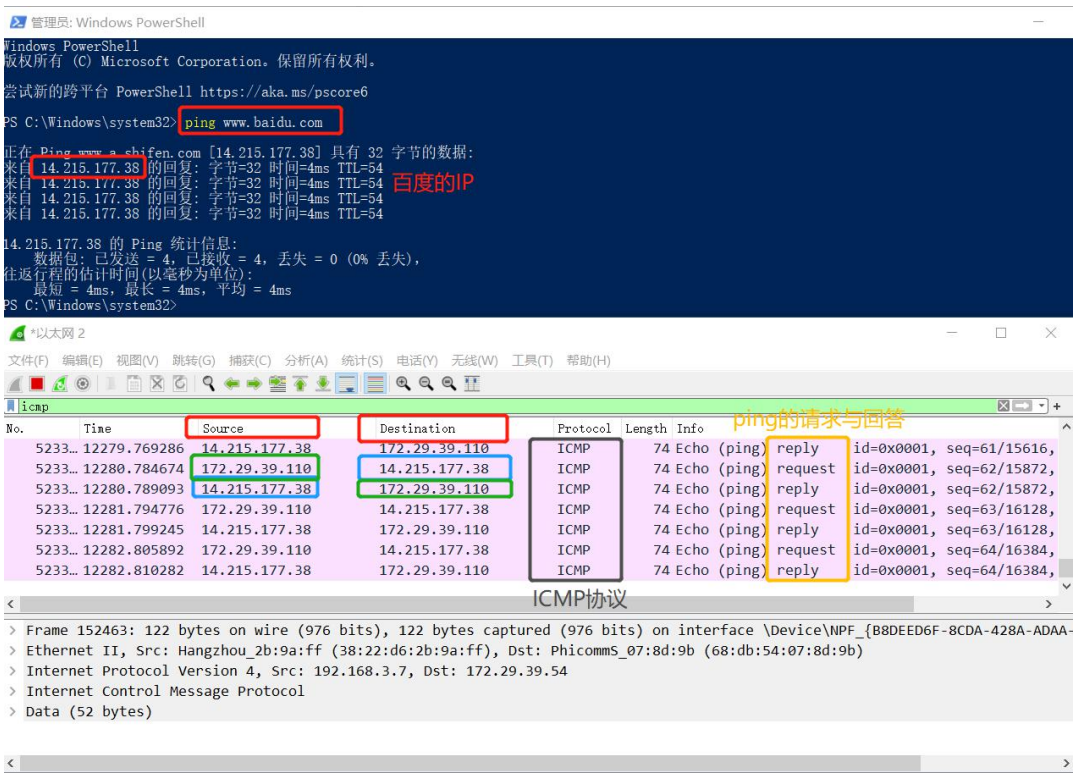


图 5.1 对比 Wireshark 分组列表栏与 ping 指令显示结果

到此实验部分结束。

实验结果：

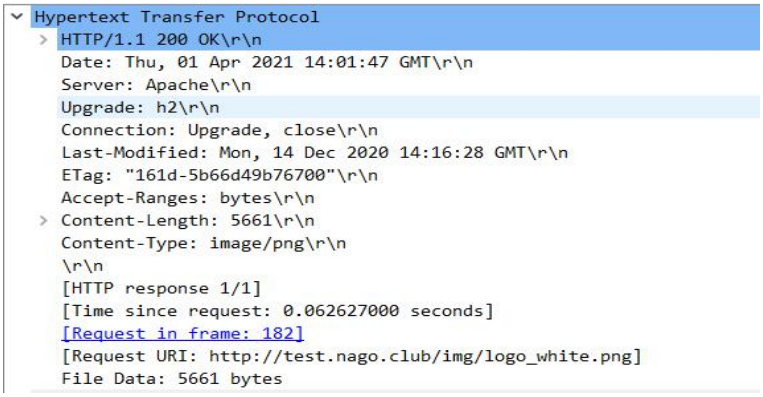
实验报告中第一部分为安装学习 Wireshark 软件，第二部分为抓包与分析 HTTP 协议，第三部分为分析 TCP 协议，第四部分为分析 TCP 三次握手，第五部分为分析 ICMP 协议。实验要求内容全部完成，所有实验截图已经全部位于实验过程中。

对于实验内容的补充分析：

1）在前两部分设置过滤器时，过滤方式有很多，其中模式过滤介绍较少，实际上 HTTP 请求方法有很多，HTTP1.0 定义了三种请求方法：GET，POST 和 HEAD 方法；HTTP1.1 新增了：OPTIONS、PUT、PATCH、DELETE、TRACE 和 CONNECT，具体应用如下表：

序号	方法	描述
1	GET	请求指定的页面信息，并返回实体主体。
2	HEAD	类似于 GET 请求，只不过返回的响应中没有具体的内容，用于获取报头。
3	POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST 请求可能会导致新的资源的建立和/或已有资源的修改。
4	PUT	从客户端向服务器传送的数据取代指定的文档的内容。
5	DELETE	请求服务器删除指定的页面。
6	CONNECT	HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。
7	OPTIONS	允许客户端查看服务器的性能。
8	TRACE	回显服务器收到的请求，主要用于测试或诊断。
9	PATCH	是对 PUT 方法的补充，用来对已知资源进行局部更新。

2）对于 HTTP 协议，上面实验内容介绍了如何查看主机名、网址、IP 地址等，还有一些其他的可分析内容，下面是一个 HTTP 响应报文：



其响应行为第一行，其中有 HTTP 版本（HTTP/1.1）、状态码（200）以及消息“OK”。常见的状态码如下表：

序号	状态码	描述
1	200	客户端请求成功，是最常见的状态。
2	302	重定向。
3	404	请求资源不存在，是最常见的状态。
4	400	客户端请求有语法错误，不能被服务器所理解。
5	401	请求未经授权。
6	403	服务器收到请求，但是拒绝提供服务。
7	500	服务器内部错误，是最常见的状态。
8	503	服务器当前不能处理客户端的请求。

响应头：第二行以后，如 Date 代表系统的 GMT 时间，Server 代表服务器名字，Content-Length 代表表示内容长度等等。

响应正文：是服务器向客户端发送的 HTML 数据。

3）对于 TCP 的三次握手，这里再分析其原因，即为什么不是两次或四次握手。TCP 三次握手的目的就是要建立可靠的通信信道；通信简单来说就是数据的发送和接收；三次握手即为了确认自己与对方的发送和接收是正常的。

发送端（Client）和接收端（Server）总共发起的 3 次请求就叫做 3 次握手。

第一次握手：Client 什么都无法确认，虽然自己发送了 SYN 数据，但是只要没收到 Server 端的 SYN/ACK 数据，就都无法确认自己的发送是否正常。此时 Client 状态如下：

Client（1）	己方发送正常	己方接收正常	对方发送正常	对方接收正常
第一次握手				
第二次握手				
第三次握手				

Server 确认时，自己的接收正常，对方的发送正常；此时 Server 状态如下：

Server（1）	己方发送正常	己方接收正常	对方发送正常	对方接收正常
第一次握手		√	√	
第二次握手				
第三次握手				

第二次握手：Client 确认：自己发送正常，接收正常；对方发送正常、接收正常。**Client 确认状态完成。**此时 Client 状态如下：

Client（2）	己方发送正常	己方接收正常	对方发送正常	对方接收正常
第一次握手				
第二次握手	√	√	√	√
第三次握手				

Server 依然只能确认对方发送正常与自己接收正常。此时 Server 状态如下：

Server（2）	己方发送正常	己方接收正常	对方发送正常	对方接收正常
第一次握手		√	√	
第二次握手				
第三次握手				

第三次握手：Client 确认状态在第二次握手时已经完成。

Client（3）	己方发送正常	己方接收正常	对方发送正常	对方接收正常
第一次握手				
第二次握手	√	√	√	√
第三次握手	√	√	√	√

Server 在第三次握手可以确认下来自己的发送正常以及对方的接收正常，**Server 确认状态完成。**此时 Server 状态如下：

Server（2）	己方发送正常	己方接收正常	对方发送正常	对方接收正常
第一次握手		√	√	
第二次握手		√	√	
第三次握手	√	√	√	√

可以看到，一共只需要三次握手就可完全确认连接，不需要更多的次数；如果只有两次握手，服务器无法确认自己到客户端的发送是否正常，因此还需要第三次握手。

第二次握手 Server 传回 SYN，为了告诉 Client，自己接收到的信息确实就是 Client 发出的信号，因此才会是 x+1。

实验小结：

通过本次实验，我安装了 Wireshark 软件，学习使用 Wireshark 软件抓包，分析了 HTTP 协议、TCP 协议、TCP 三次握手以及 ICMP 协议，理解了 TCP/IP 协议分层模型、了解 TCP 报文格式并理解了 TCP 三次握手。实验让我对网络协议有了更深的理解与认识。

本次实验遇到了两个困难，一个是 HTTP 协议网站目前已经比较罕见，主流网页全部使用 HTTPS 协议加密过了，因此一般只能寻找自己搭建的网站，如学校的 OJ；另一个困难是 TCP 三次握手过程的理解，我查阅了很多资料，虽然感觉资料上的各种表述可能并不完全相同，不同的网站、不同的人可能都有自己的理解，我参考了很多讲解说明后，将自己的理解尽量都展现在报告中。实验结果中关于原理的分析，我也用表格的方式表达了自己的理解感觉也会比较直观。总之，本次实验使我受益匪浅。

指导教师批阅意见：

成绩评定：

指导教师签字：

年　　月　　日

备注：