



南開大學
Nankai University

计算机学院
计算机网络实验报告

配置 Web 服务器 + 编写简单页面 +
分析交互过程

姓名：杨馨仪

学号：2011440

专业：计算机科学与技术

2022 年 10 月 28 日

目录

1 实验要求	2
2 编写 web 页面	2
3 配置 web 服务器	3
4 程序测试	4
5 Wireshark 的使用	5
6 交互过程分析	6
6.1 三次握手	6
6.2 http 请求	6
6.3 四次挥手	8
7 实验反思	9

1 实验要求

(1) 搭建 Web 服务器（自由选择系统），并制作简单的 Web 页面，包含简单文本信息（至少包含专业、学号、姓名）和自己的 LOGO。

(2) 通过浏览器获取自己编写的 Web 页面，使用 Wireshark 捕获浏览器与 Web 服务器的交互过程，并进行简单的分析说明。

2 编写 web 页面

在简单学习过后，我编写了如下 html 文件。里面的主要内容包括我的一些基本信息：姓名，学号，专业，学校等等。此外，还放置了我的个人 logo，并且更换为了更具观赏性的背景图片。其代码如下所示：

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Cynthia-Young's Home Page</title>
8     <style>
9     body
10    {
11        background-image: url(/6.jpg);
12        background-size: cover;
13    }
14 </style>
15 <style>
16     div.transbox
17     {
18         width: 1050px;
19         height: 320px;
20         margin: 300px 330px;
21         background-color: #ffffff;
22         border: 1px solid #FFFFFF;
23         opacity: 0.8;
24     }
25 </style>
26 <style>
27     h1 {text-align: center;}
28     h2.ex1 {margin-left: 5cm;}
29     h3.ex1 {margin-left: 5cm;}
30 </style>
31 <style>
32     img.ex1 {margin-left: 19cm; position: absolute; top: 10; left: 150;}
33 </style>
34
35 </head>
```

```

36 <body>
37   <div class="background">
38     <div class="transbox">
39       <h1><font color=lightcoral>Cynthia-Young's Home</h1>
40       
41       <h2 class="ex1">姓名：杨馨仪</h2>
42       <h2 class="ex1">学号：2011440</h2>
43       <h2 class="ex1">专业：计算机科学与技术</h2>
44       <h2 class="ex1">学校：南开大学</h2>
45     </div>
46   </div>
47 </body>
48 </html>

```

3 配置 web 服务器

PhpStudy 是一个 PHP 调试环境的程序集成包。该程序包集成最新的 Apache+PHP+MySQL+phpMyAdmin+ZendOptimizer，一次性安装，无须配置即可使用，是非常方便、好用的 PHP 调试环境。在此次实验中使用 PhpStudy 来配置 web 服务器。

在官网下载成功后，双击可执行文件即可进入 PhpStudy 主页面。首先要打开服务，先点击启动 WNMP，然后再点击启动套件中的 Apache。

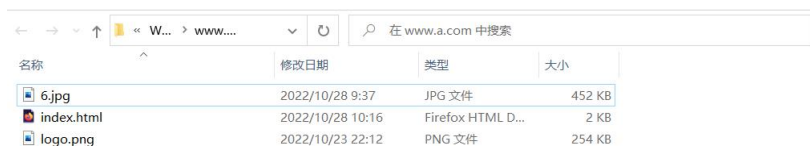


点击右侧目录中的“网站”，然后点击“创建网站”，在弹出的窗口中，有基本配置，高级配置等选项，一般情况下我们选择“基本配置”。



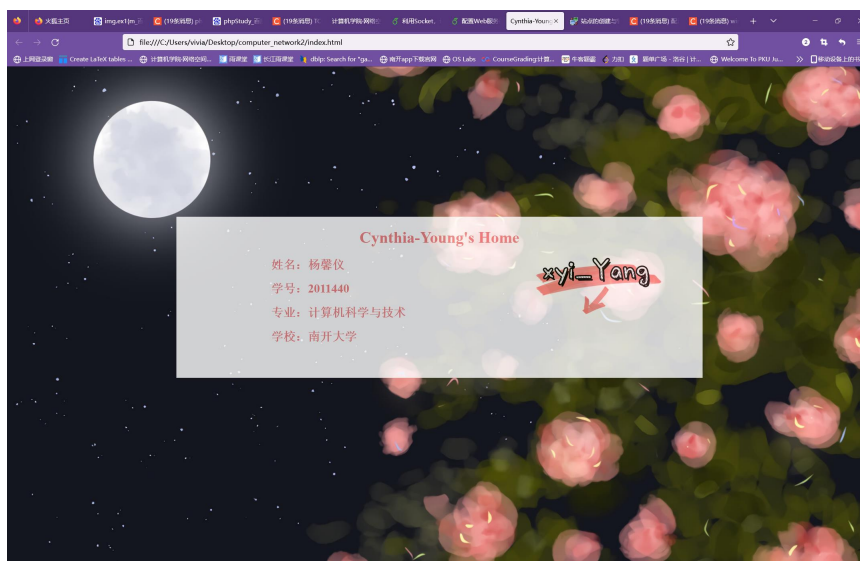
本次实验，我将域名命名为 www.demo.com，端口选择为 http 8080（没有选择 80 端口是因为 80 端口被 IIS 占用）。创建环境勾选“同步 hosts”，程序类型为 PHP。

打开刚刚创建的网站的根目录，将里面的文件删除，加入自己写的 html 文件，如下图所示：



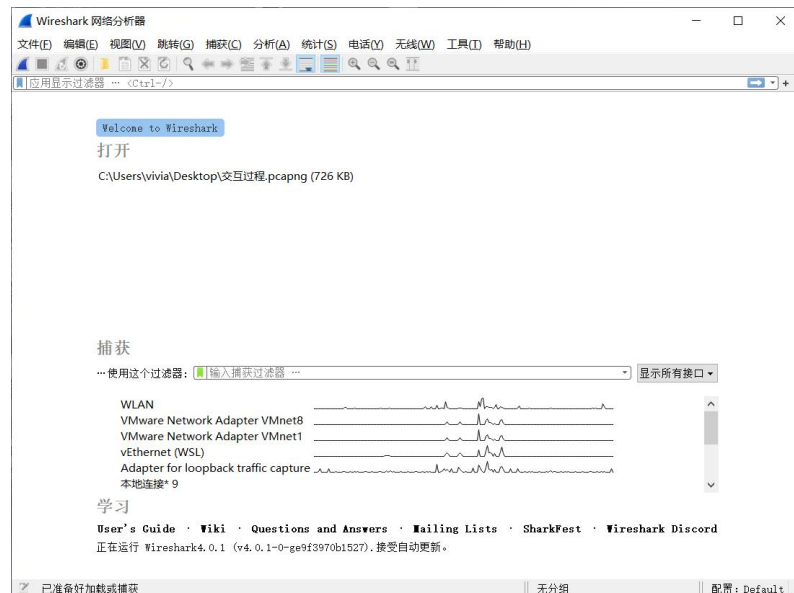
4 程序测试

在 PhpStudy 中点击打开网站，即可看到我编写的 html 网页，如下图所示：



5 Wireshark 的使用

Wireshark（前称 Ethereal）是一个网络封包分析软件。Npcap 基于 WinPcap 库，可以抓取通过 127.0.0.1 本地环回地址的包。从官网下载 Wireshark 来做进一步的服务器与浏览器的交互过程分析。下载并安装成功后，双击可执行程序进入 WireShark 主界面。



选择 Adapter for loopback traffic capture，这个能够捕捉到本地发出，本地接收的报文。同时选择过滤 port 8080，因为我本地服务器代码指定的是 8080 端口。

然后点击左上角的“鲨鱼图标”，开始捕获分组。我们通过搭建好的 web 服务器打开我们编写的 html 文件，然后再将浏览器关闭，获取整个交互过程如下图所示：

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.1	TCP	56	5097 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
127.0.0.1	127.0.0.1	TCP	56	8080 → 5097 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
127.0.0.1	127.0.0.1	HTTP	437	GET / HTTP/1.1
127.0.0.1	127.0.0.1	TCP	44	8080 → 5097 [ACK] Seq=1 Ack=394 Win=2619648 Len=0
127.0.0.1	127.0.0.1	HTTP	1765	HTTP/1.1 200 OK (text/html)
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [ACK] Seq=394 Ack=1722 Win=2617856 Len=0
127.0.0.1	127.0.0.1	HTTP	388	GET /logo.png HTTP/1.1
127.0.0.1	127.0.0.1	TCP	44	8080 → 5097 [ACK] Seq=1722 Ack=738 Win=2619392 Len=0
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=1722 Ack=738 Win=2619392 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=67217 Ack=738 Win=2619392 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=132712 Ack=738 Win=2619392 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	HTTP	63524	HTTP/1.1 200 OK (PNG)
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [ACK] Seq=738 Ack=261687 Win=2490624 Len=0
127.0.0.1	127.0.0.1	TCP	44	[TCP Window Update] 5097 → 8080 [ACK] Seq=738 Ack=261687 Win=2556160 Len=0
127.0.0.1	127.0.0.1	HTTP	385	GET /6.jpg HTTP/1.1
127.0.0.1	127.0.0.1	TCP	44	8080 → 5097 [ACK] Seq=261687 Ack=1079 Win=2618880 Len=0
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=261687 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=327182 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=392677 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=458172 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=523667 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=589162 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	TCP	65539	8080 → 5097 [ACK] Seq=654657 Ack=1079 Win=2618880 Len=65495 [TCP segment of a reassembled PDU]
127.0.0.1	127.0.0.1	HTTP	4530	HTTP/1.1 200 OK (JPEG JFIF image)
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [ACK] Seq=1079 Ack=724638 Win=2287616 Len=0
127.0.0.1	127.0.0.1	TCP	44	[TCP Window Update] 5097 → 8080 [ACK] Seq=1079 Ack=724638 Win=2353152 Len=0
127.0.0.1	127.0.0.1	TCP	44	[TCP Window Update] 5097 → 8080 [ACK] Seq=1079 Ack=724638 Win=2619648 Len=0
127.0.0.1	127.0.0.1	HTTP	391	GET /favicon.ico HTTP/1.1
127.0.0.1	127.0.0.1	TCP	44	8080 → 5097 [ACK] Seq=724638 Ack=1426 Win=2618624 Len=0
127.0.0.1	127.0.0.1	HTTP	639	HTTP/1.1 404 Not Found (text/html)
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [ACK] Seq=1426 Ack=725233 Win=2619136 Len=0
127.0.0.1	127.0.0.1	TCP	44	8080 → 5097 [FIN, ACK] Seq=725233 Ack=1426 Win=2618624 Len=0
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [ACK] Seq=1426 Ack=725234 Win=2619136 Len=0
127.0.0.1	127.0.0.1	TCP	44	5097 → 8080 [FIN, ACK] Seq=1426 Ack=725234 Win=2619136 Len=0
127.0.0.1	127.0.0.1	TCP	44	8080 → 5097 [ACK] Seq=725234 Ack=1427 Win=2618624 Len=0

6 交互过程分析

6.1 三次握手

127.0.0.1	127.0.0.1	TCP	56 5097 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
127.0.0.1	127.0.0.1	TCP	56 8080 → 5097 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
127.0.0.1	127.0.0.1	TCP	44 5097 → 8080 [ACK] Seq=1 Ack=1 Win=2619648 Len=0

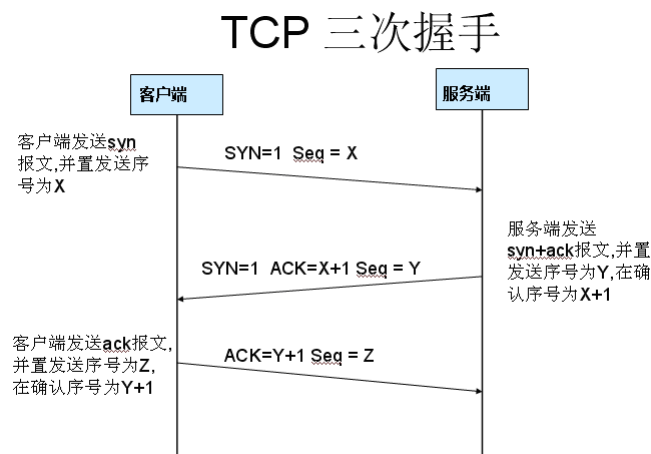
TCP 三次握手连接建立过程:

Step1: Client 将标志位 SYN 置为 1, 随机产生一个值 seq=J, 并将该数据包发送给 Server, Client 进入 SYN_SENT 状态, 等待 Server 确认, 这是第一次握手。

Step2: Server 收到数据包后由标志位 SYN=1 知道 Client 请求建立连接, Server 将标志位 SYN 和 ACK 都置为 1, ack=J+1, 随机产生一个值 seq=K, 并将该数据包发送给 Client 以确认连接请求, Server 进入 SYN_RCVD 状态, 这是第二次握手。

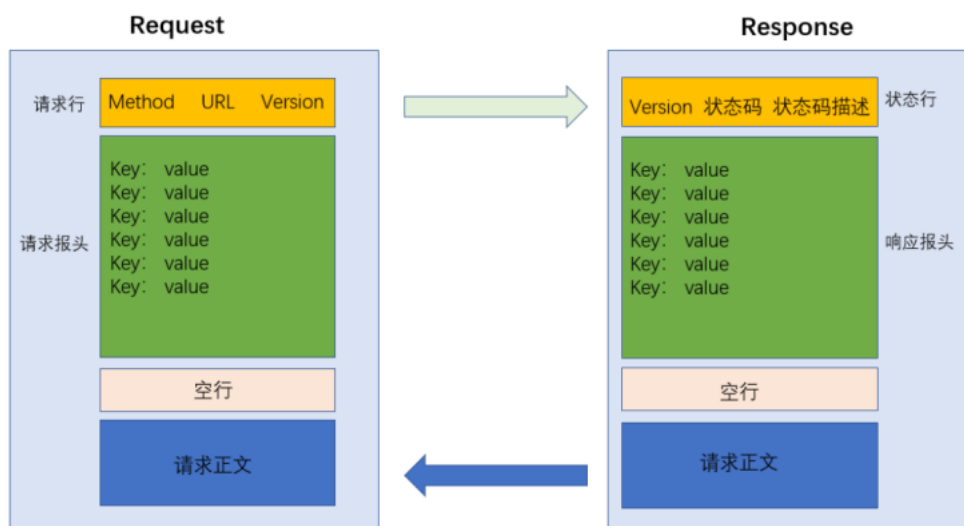
Step3: Client 收到确认后, 检查 ack 是否为 J+1, ACK 是否为 1, 如果正确则将标志位 ACK 置为 1, ack=K+1, 并将该数据包发送给 Server, Server 检查 ack 是否为 K+1, ACK 是否为 1, 如果正确则连接建立成功, Client 和 Server 进入 ESTABLISHED 状态, 完成三次握手, 随后 Client 与 Server 之间可以开始传输数据了。

TCP 连接建立, 开始通讯。



6.2 http 请求

HTTP 协议格式如下图所示:



请求报文：

```

1 GET / HTTP/1.1
2 Host: www.a.com:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101
  Firefox/106.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1

```

请求行：

1. 请求方法：GET 方法

GET 请求的特点：

- 首行里面的第一个部分就是 GET
- URL 里面的 query string 可以为空，也可以不为空
- GET 请求的 header 有若干个键值对结构
- GET 请求的 body 一般是空的

2. url 格式：由于在本实验中是同一台主机，于是 URL 就被省略了。

3. http 版本：本实验中是 HTTP/1.1

请求报头：header 的整体格式是键值对结构，每个键值对占一行，键和值之间使用冒号 + 空格进行分割

key	value
Host	表示服务器主机的地址和端口
Content-Length	表示 body 的数据长度, 长度单位是字节
Content-Type	表示 body 的数据格式
User-Agent	表示浏览器或者操作系统的属性
Referer	表示这个页面是从哪个页面跳转过来的
Cookie	是浏览器提供的一种让程序员在本地存储数据的能力

请求体: GET 请求, 不包含请求体。

响应报文:

```

1 HTTP/1.1 200 OK
2 Date: Fri, 28 Oct 2022 09:25:50 GMT
3 Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02
4 Last-Modified: Fri, 28 Oct 2022 02:16:16 GMT
5 ETag: "568-5ec0ed7b9075c"
6 Accept-Ranges: bytes
7 Content-Length: 1384
8 Keep-Alive: timeout=5, max=100
9 Connection: Keep-Alive
10 Content-Type: text/html
11
12 //响应体为编写的html文件, 在此不再贴出

```

响应行: HTTP 版本 + HTTP 状态码 + 原因描述

常见状态码	说明
200 OK	这是一个最常见的状态码, 表示访问成功。抓包抓到的大部分结果都是 200
404 Not Found	没有找到资源。URL 标识的资源不存在, 那么就会出现 404
403 Forbidden	表示访问被拒绝。有的页面通常需要用用户具有一定的权限才能访问(登录后才能访问)。如果用户没有登陆直接访问, 就容易见到 403
405 Method Not Allowed	我们学习了 HTTP 中所支持的方法, 有 GET, POST, PUT, DELETE 等。但是对方的服务器不一定都支持所有的方法(或者不允许用户使用一些其他的方法)。
500 Internal Server Error	服务器出现内部错误。一般是服务器的代码执行过程中遇到了一些特殊情况(服务器异常崩溃)会产生这个状态码, 一般很少见
504 Gateway Timeout	当服务器负载比较大的时候, 服务器处理单条请求的时候消耗的时间就会很长, 就可能会导致出现超时的情况
302 Move temporarily	临时重定向。在登陆页面中经常会见到 302。用于实现登陆成功后自动跳转到主页
301 Moved Permanently	永久重定向。当浏览器收到这种响应时, 后续的请求都会被自动改成新的地址。301 也是通过 Location 字段来表示要重定向到的新地址

响应报头: 响应报头的基本格式和请求报头的格式基本一致。

响应体: 就是编写的 html 文件

除此之外, 由于页面中有两张图片, 所以客户端还向服务器发送过两次请求, 并成功得到了响应, 在此不再赘述。

6.3 四次挥手

四次挥手即终止 TCP 连接, 就是指断开一个 TCP 连接时, 需要客户端和服务端总共发送 4 个包以确认连接的断开。在 socket 编程中, 这一过程由客户端或服务端任一方执行 close 来触发。

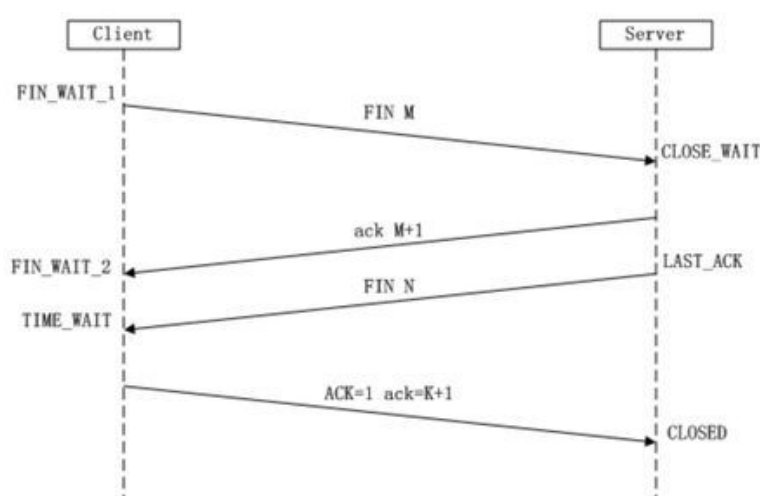
127.0.0.1	127.0.0.1	TCP	44 8080 → 5097 [FIN, ACK] Seq=725233 Ack=1426 Win=2618624 Len=0
127.0.0.1	127.0.0.1	TCP	44 5097 → 8080 [ACK] Seq=1426 Ack=725234 Win=2619136 Len=0
127.0.0.1	127.0.0.1	TCP	44 5097 → 8080 [FIN, ACK] Seq=1426 Ack=725234 Win=2619136 Len=0
127.0.0.1	127.0.0.1	TCP	44 8080 → 5097 [ACK] Seq=725234 Ack=1427 Win=2618624 Len=0

Step1: Client 发送一个 FIN，用来关闭 Client 到 Server 的数据传送，Client 进入 FIN_WAIT_1 状态。

Step2: Server 收到 FIN 后，发送一个 ACK 给 Client，确认序号为收到序号 +1（与 SYN 相同，一个 FIN 占用一个序号），Server 进入 CLOSE_WAIT 状态。

Step3: Server 发送一个 FIN，用来关闭 Server 到 Client 的数据传送，Server 进入 LAST_ACK 状态。

Step4: Client 收到 FIN 后，Client 进入 TIME_WAIT 状态，接着发送一个 ACK 给 Server，确认序号为收到序号 +1，Server 进入 CLOSED 状态，完成四次挥手。



7 实验反思

在本次实验中，经过反思我认为有以下几点可以进一步改进：

- 搭建服务器可以进阶尝试使用 socket 编程
- 在编写 html 语言时，对其了解程度有限，应当可以在网页设计上提高其功能性与美观性
- 使用 wireshark 进行抓包时，由于并不熟悉该工具，对于配置捕获选项以及使用过滤器上出现了很多问题，导致花费了很长时间。
- 在第二次打开网站捕获分组时的响应报文总是 304 Not Modified。后来发现是因为网页有缓存机制。因此，每一次重新创建网站就可以避免这一点，得以成功捕获到了 200 OK 的响应报文。
- 最开始纠结于为什么四次挥手中的第一次是由浏览器发出的，后来发现客户端和服务端都可以主动发起挥手动作，实验中我关闭了网页，所以由客户端先向服务器发送终止连接的请求。