实验 2：配置 Web 服务器，分析 HTTP 交互过程

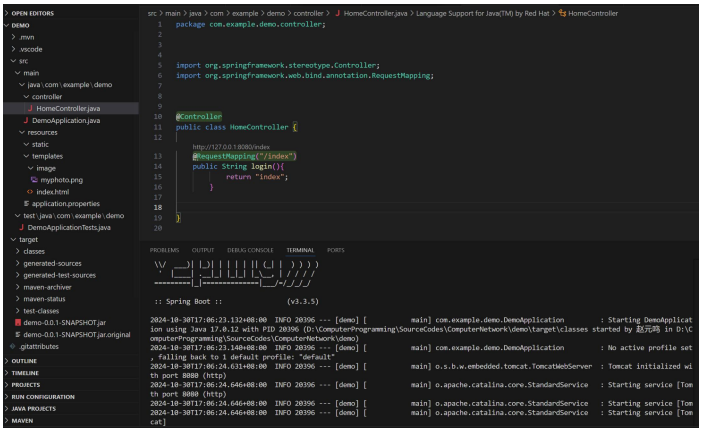
（赵元鸣 2211757 计算机科学与技术 0850 班）

一. 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）、自己的 LOGO、自我介绍的音频信息。
（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，使用 Wireshark 过滤器使其只显示 HTTP 协议。
（4）现场演示。
（5）提交 HTML 文档、Wireshark 捕获文件和实验报告，对 HTTP 交互过程进行详细说明。

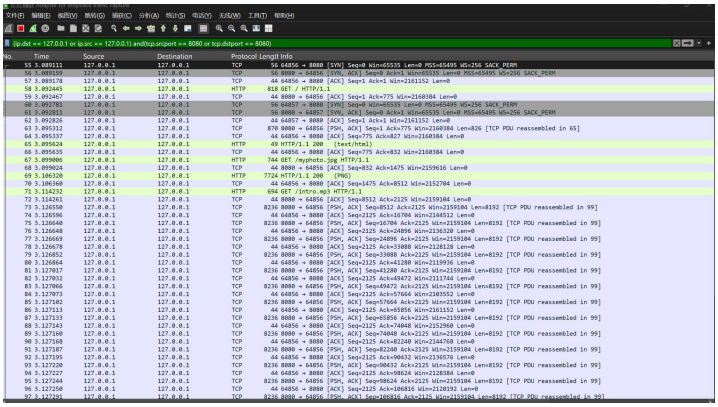
二. 实验流程

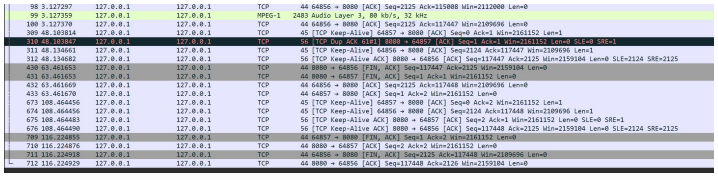
1. 利用 springboot，同时使用 maven 框架搭建 web 服务器：



2. 使用 wireshark 抓包：

选择 Adapter for loopback traffic capture 接口，设定设置过滤器(ip.dst == 127.0.0.1 or ip.src == 127.0.0.1) and(tcp.srcport == 8080 or tcp.dstport == 8080)，得到抓包结果如下。

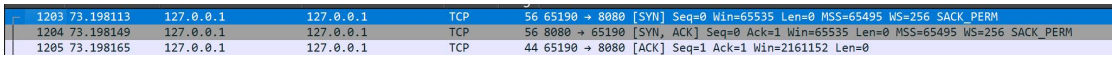




三．实验分析

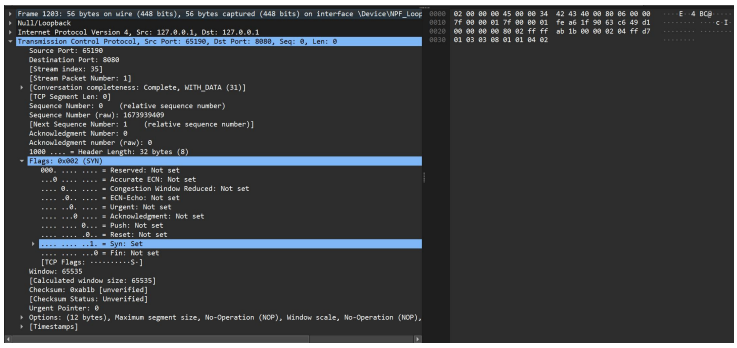
1. 下面对三次握手过程进行分析

建立连接时，客户端发送 SYN 到服务器，并进入 SYN_SENT 状态，等待服务器确认。

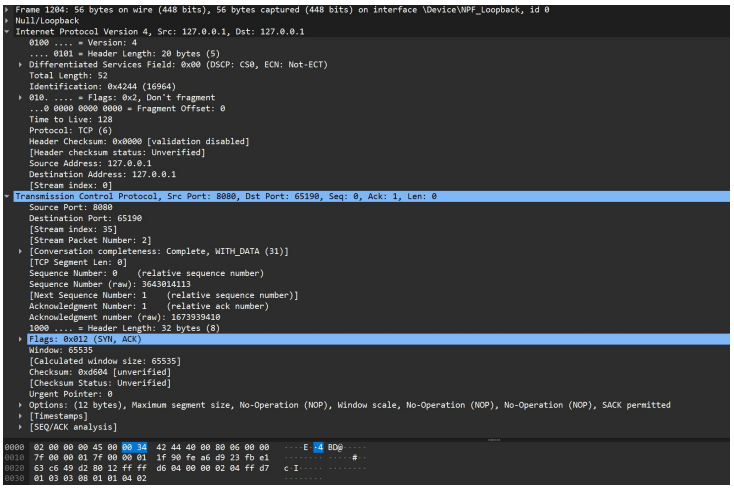


TCP 客户端向服务器端发送连接请求报文段，即客户端发送 SYN 到服务器，并进入 SYN_SENT 状态，等待服务器端确认第一次握手成功说明客户端的数据可以被服务端收到，说明客户端的发功能可用，说明服务端的收功能可用。变量的值如下：

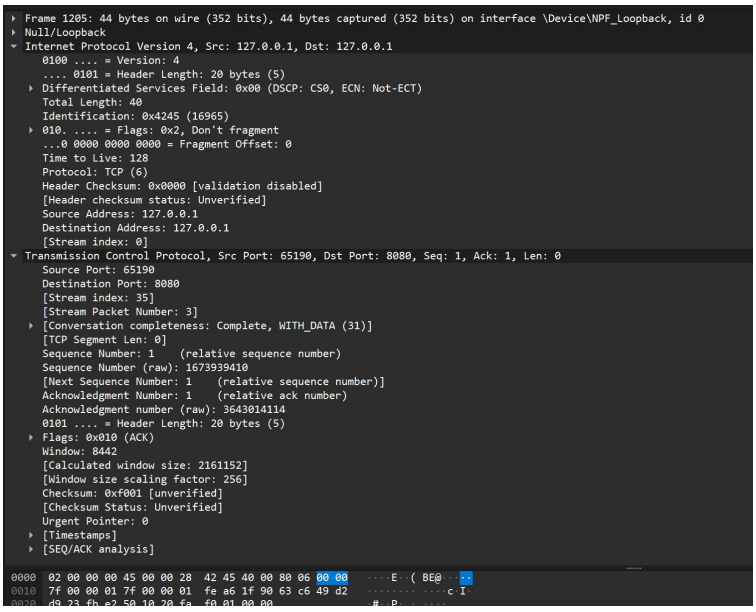
- SYN ：标志位，表示请求建立连接
- Seq = 0 ：初始建立连接值为 0，数据包的相对序列号从 0 开始，表示当前还没有发送数据
- Ack =0：初始建立连接值为 0，已经收到包的数量，表示当前没有接收到数据
- WIN = 65535 指的是 Window size: 65535
- WS = 256 指的是 windows scale 窗口扩张，声明 shift count，作为 2 的指数，再乘以 TCP 定义的接收窗口，得到 TCP 窗口
- MSS = 65495 指的是 Maximum segment size: 65495byte ，最长报文段，TCP 包所能携带的最大数据量，不包含 TCP 头和 Option。



第二次握手：服务器端收到数据包后由标志位 SYN=1 得知客户端请求建立连接，然后便发送确认报文段（SYN+ACK 信令）到客户端，接着服务器进入 SYN_RECV 状态，握手成功说明服务端的数据可以被客户端收到，即说明服务端的发功能，客户端的收功能可用。同时客户端知道自己的数据已经正确到达服务端，自己的发功能正常。但是服务端自己不知道数据是否被接收 。Seq = 0 表示初始建立值为 0，表示当前还没有发送数据 ；Ack = 1 表示当前端成功接收的数据位数，因为包含 SYN 或 FIN 标志位尽管客户端没有发送任何有效数据，确认号仍加 1。



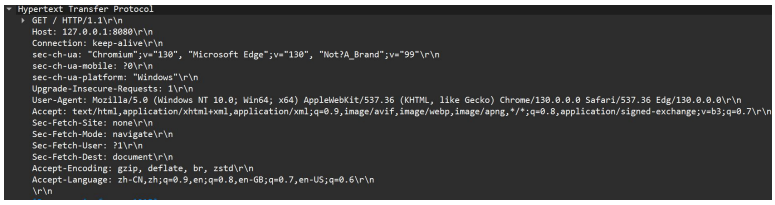
第三次握手：客户端收到服务器端确认后，检查 ack 是否为 x+1，ACK 是否为 1，如果正确则再发送确认报文段给服务器，客户端进入 ESTABLISHED 状态，服务器收到后也进入 ESTABLISHED 状态。



2. http 协议，发送和接受数据：

（1）请求：GET / HTTP/1.1\r\n 称为请求行，由三个部分组成：请求方法、URI、HTTP 协议版本，他们之间用空格分隔。该部分位于数据首行。

基本格式为：该部分的请求方法字段（GET）给出了请求类型，URI 给出请求的资源位置），最后 HTTP 协议版本给出 HTTP 的版本号。接下来的部分称为请求头部，其紧跟着请求行，该部分主要是用于描述请求正文，主要是用于说明请求源、连接类型、以及一些 Cookie 信息等。下面时点开 http 的 get 中显示的协议的具体内容：



然后除了最常用的 get 和 post 请求方式，还有如下的请求方式：

·GET：用于从服务器获取资源，也是最常见的请求方式。GET 请求将请求的参数附加在 URL 的末尾，发送给服务器。

·POST：用于向服务器提交数据，一般用于发送表单数据。POST 请求将请求的参数放在请求的主体中，而不是 URL 中。

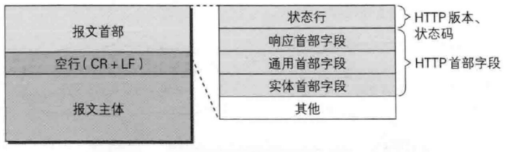
·PUT：用于向服务器上传文件或更新资源。PUT 请求会将请求的数据存储在服务器上指定的位置。

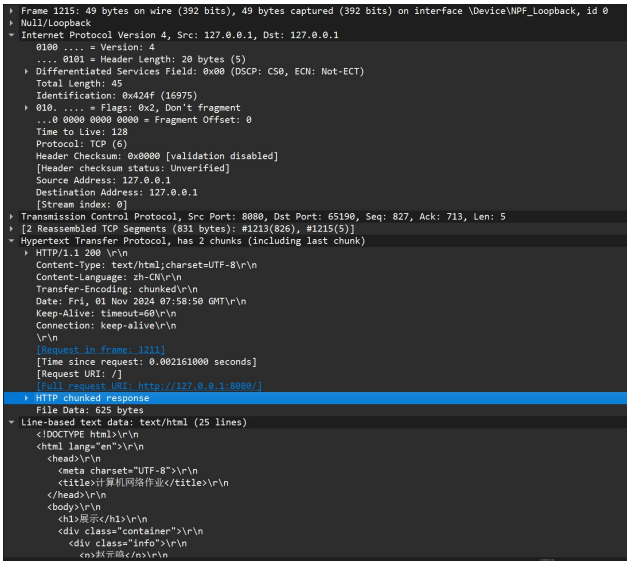
·DELETE：用于删除服务器上的资源。

·HEAD：用于获取服务器对资源的头部信息，而不获取实际的资源内容。

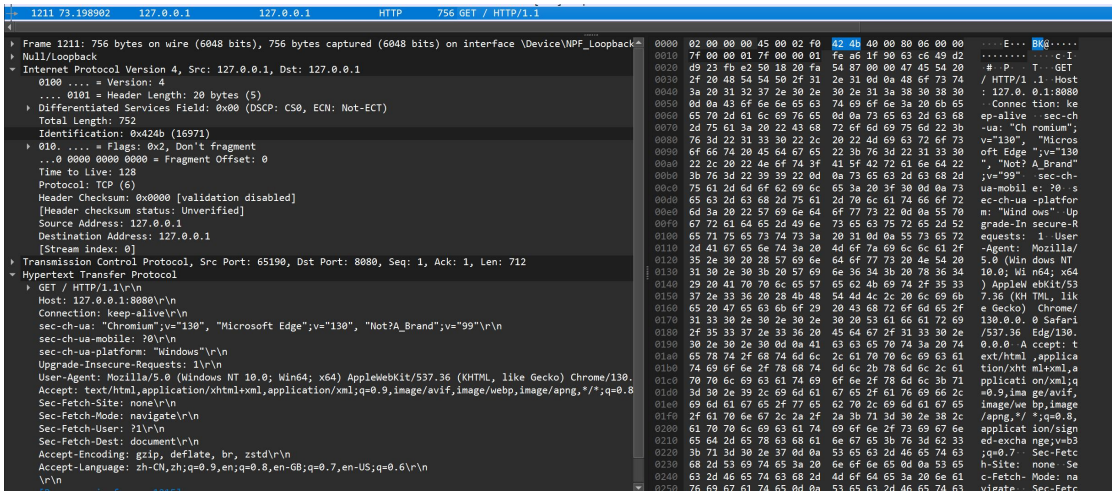
·OPTIONS：用于获取服务器支持的请求方法。

（2）响应： 响应报文由状态行、响应头部、空行和响应体 4 个部分构成。首先是状态行，主要给出响应 HTTP 协议的版本号、响应返回状态码、响应描述；响应头部主要是返回一些服务器的基本信息，以及一些 Cookie 值等；响应体为所请求的数据。

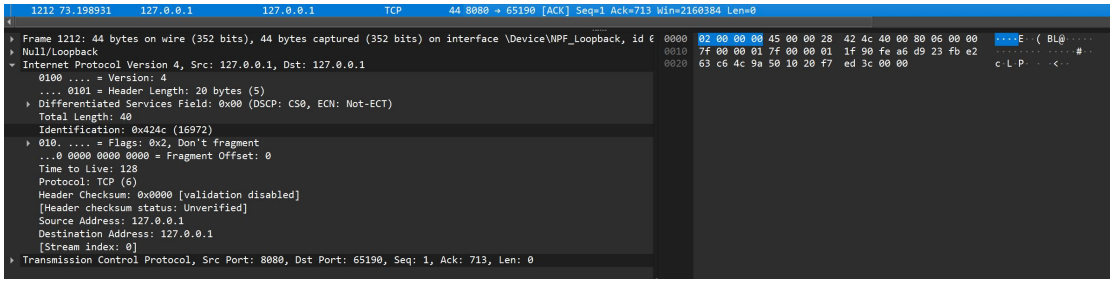




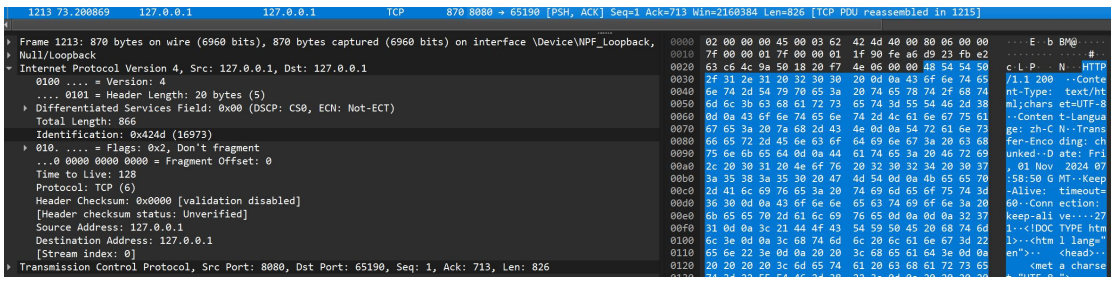
（3）在本次实验数据中，可以看到在三次握手建立连接与四次挥手关闭连接中间有 3 次 HTTP 请求与响应。以传递 text/html 那一轮为例，首先客户端向服务器端发送请求



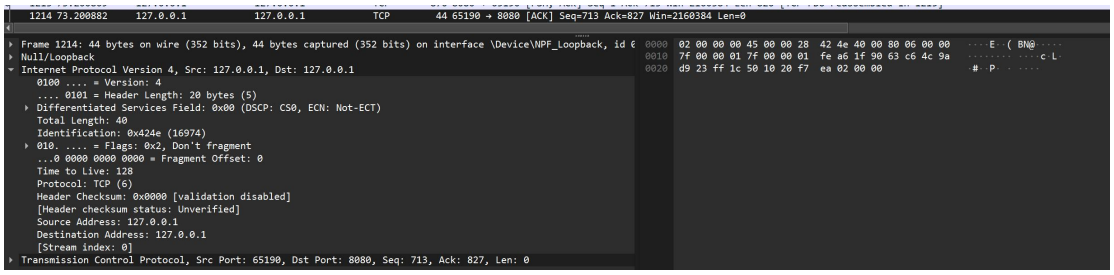
然后服务器端回复 ACK 表示收到请求



之后服务器端向客户端发送响应报文

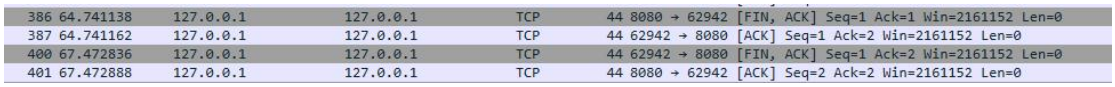


最后客户端回复 ACK 表示收到请求



3. 下面对四次挥手进行分析：

第一次挥手，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。第二次挥手时，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。第三次挥手时，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。第四次挥手时，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。



四. 总结

在本次实验中，我们搭建了一个简单的 Web 服务器，并制作了包含个人信息和音频自我介绍的 Web 页面。通过使用 Wireshark，我们捕获了浏览器与 Web 服务器之间的 HTTP 交互过程，并对其中的三次握手和四次挥手过程进行了详细分析。

通过实验，我们深入了解了 TCP 连接的建立和关闭机制，掌握了 HTTP 协议的基本请求和响应结构，学会了如何使用 Wireshark 过滤和分析网络数据包。这些实践不仅增强了我们对网络通信原理的理解，还提高了我们在实际应用中分析和解决问题的能力。