# 华东师范大学软件工程学院实验报告

|  |  |  |
| --- | --- | --- |
| **实验课程**：计算机网络实践 | **年级**：2023级 | **实验成绩**： |
| **实验名称**：Lab 1 Protocol Layer | **姓名**：陆尚辰 |  |
| **实验编号**： | **学号**：10235101545 | **实验日期**：2024.11.15 |
| **指导教师**：刘献忠 | **组号**： | **实验时间**： |

**一、实验目的**

1. 学会通过Wireshark获取各协议层的数据包

2. 掌握协议层数据包结构

3. 分析协议开销

**二、实验内容与实验步骤**

**（一）获取协议层数据包**

1.打开命令行，输入：wget http://www.baidu.com. 观察到 OK为止。

2.启动Wireshark，在菜单栏的捕获->选项中进行设置，选择已连接的以太网，设置捕获过滤器为tcp port 80，将混杂模式设为关闭,勾选 enable network name resolution.然后开始捕获.

3.点开命令行，重新输入:wget http://www.baidu.com

4.打开Wireshark， 停止捕获。

**（二）绘制数据包结构**

1.根据实验结构，分析Http GET协议包的内容。

2. 分析并且绘制协议包。

**（三）分析协议开销**

1.根据实验结构，分析Http应用协议额外开销。

2. 估计上面捕获的Http协议的额外开销。

**（四）分析解复用键**

观察下载的以太网和IP包头部信息回答下面问题：

1.以太网头部中哪一部分是解复用键并且告知它的下一个高层指的是IP，在这一包内哪一个值可以表示IP？

2.IP头部中哪一部分是解复用键并且告知它的下一一个高层指的是TCP，在这一包内哪一个值可以表示TCP？

**（五）问题讨论**

1.Look at a short TCP packet that carries no higher-layer data. To what entity is this packet destined? After all, if it carries no higher-layer data then it does not seem very useful to a higher layer protocol such as HTTP!

2.In a classic layered model, one message from a higher layer has a header appended by the lower layer and becomes one new message. But this is not always the case. Above, we saw a trace in which the web response (one HTTP message comprised of an HTTP header and an HTTP payload) was converted into multiple lower layer messages (being multiple TCP packets). Imagine that you have drawn the packet structure (as in step 2) for the first and last TCP packet carrying the web response. How will the drawings differ?

3.In the classic layered model described above, lower layers append headers to the messages passed down from higher layers. How will this model change if a lower layer adds encryption?

4.In the classic layered model described above, lower layers append headers to the messages passed down from higher layers. How will this model change if a lower layer adds compression?

**三、实验环境**

1.Wireshark v2.0.2

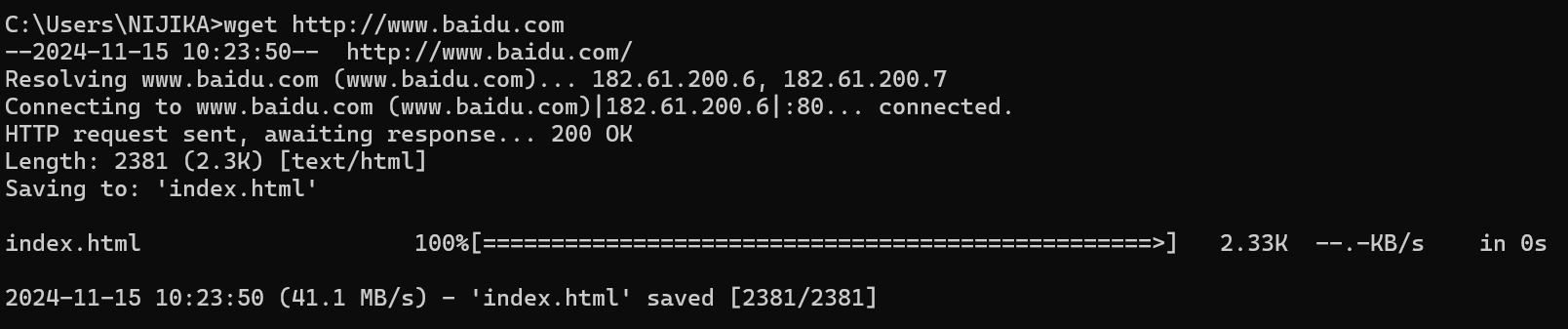
2.Windows操作系统

3.wget工具

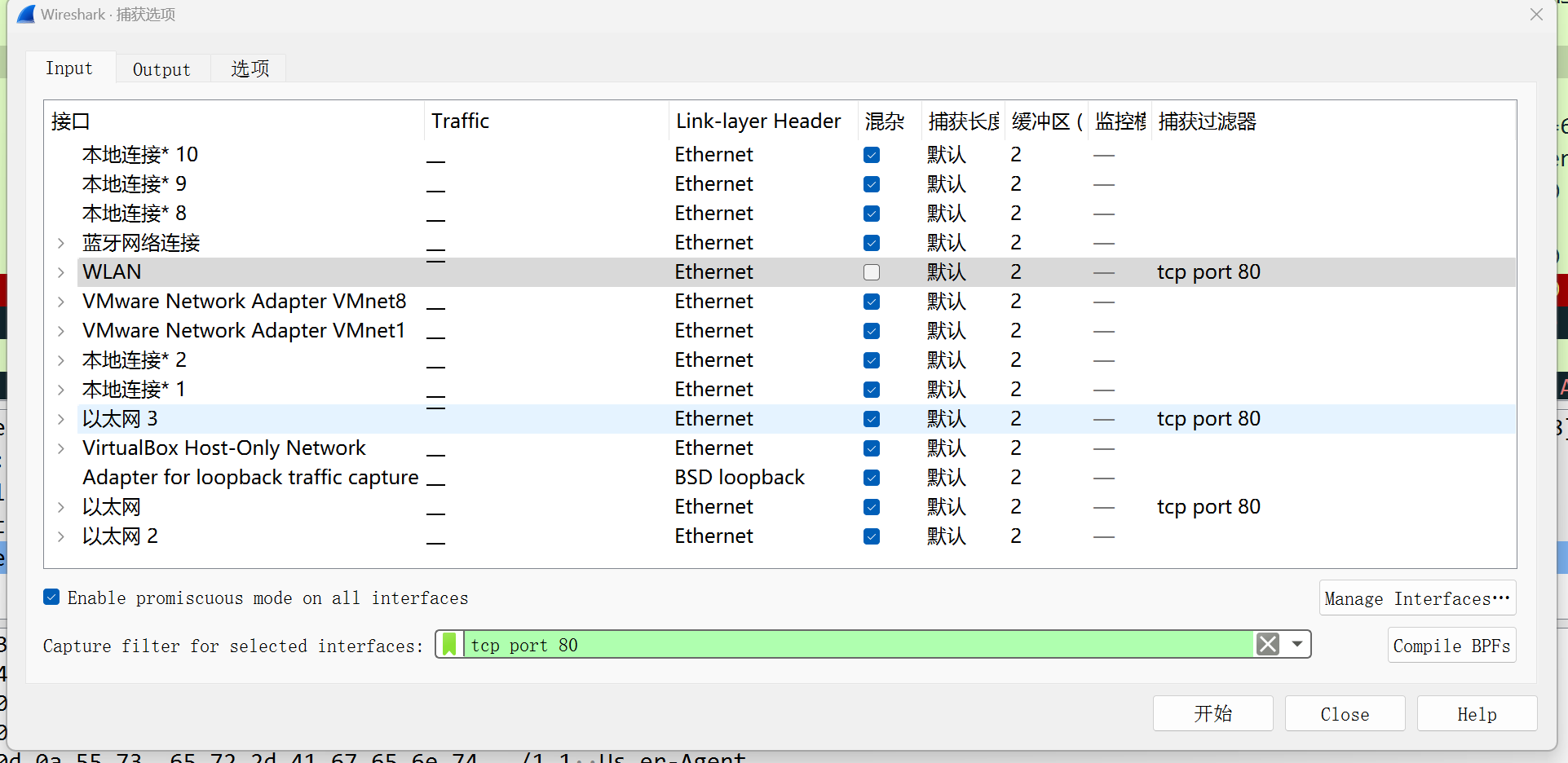
**四、实验过程与分析**

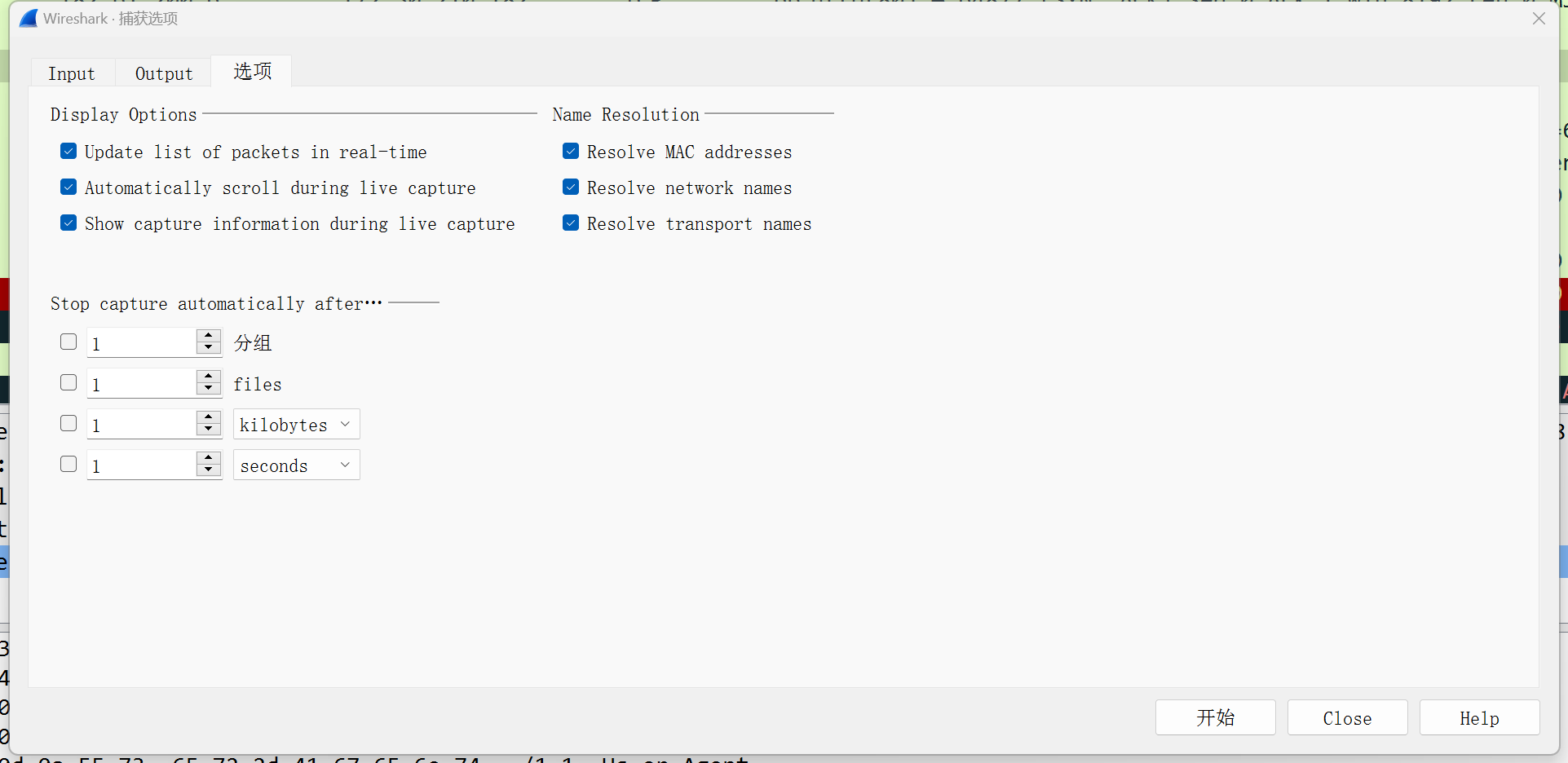
**2.1 获取协议层数据包：**

1.打开命令行win+R，输入：wget http://www.baidu.com. 出现以下界面：

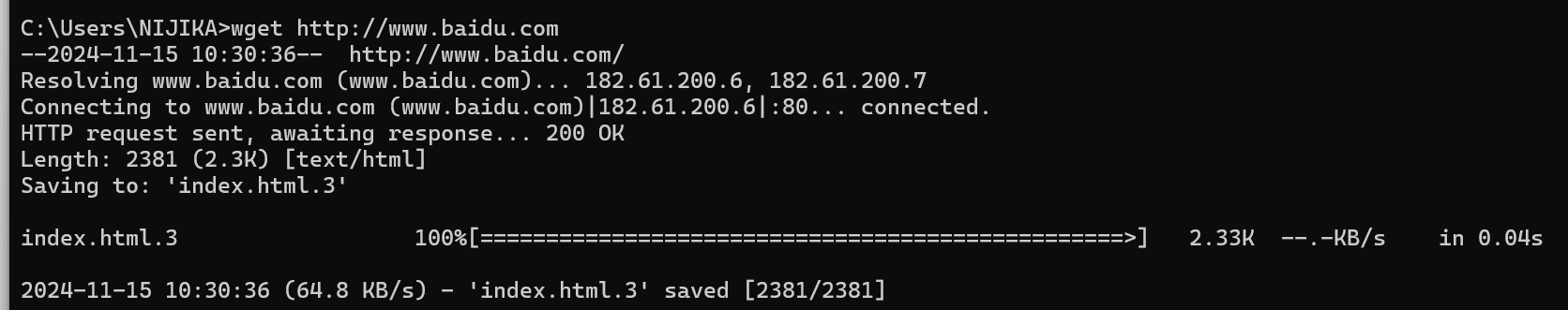


2.启动Wireshark，在菜单栏的捕获->选项中进行设置，选择WLAN，设置捕获过滤器为tcp port 80，将混杂模式设为关闭,勾选resolve network name然后开始捕获.

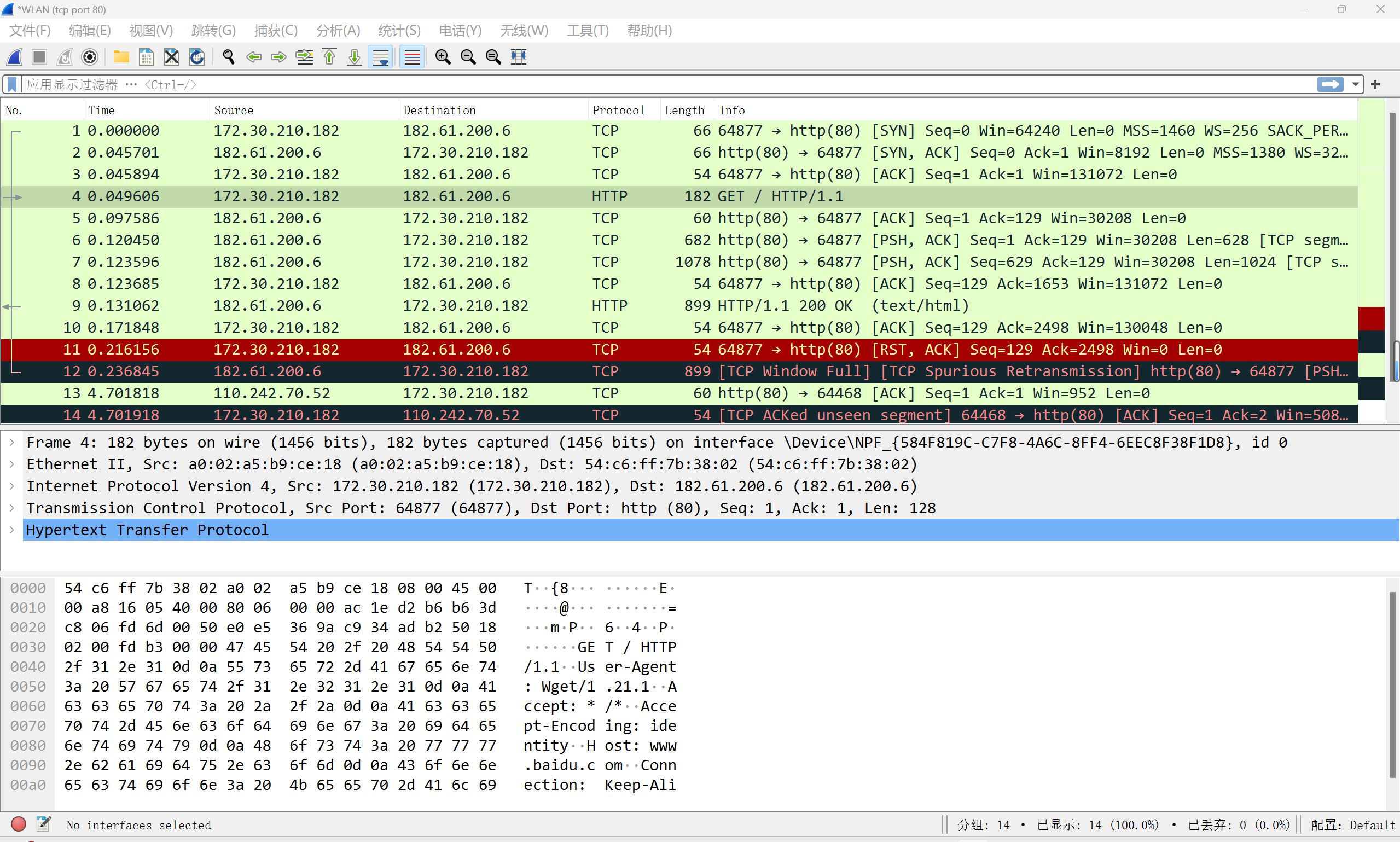




3.点开命令行win+R，重新输入:wget <http://www.baidu.com>



4.打开Wireshark，停止捕获。我们需要找到Http GET协议包



这样，我们成功获取了HttpGET的协议包。

**2.2 绘制数据包结构：**

1.根据实验结构，分析Http GET协议包的内容。

（1）Frame

Frame展示了包的总体信息。



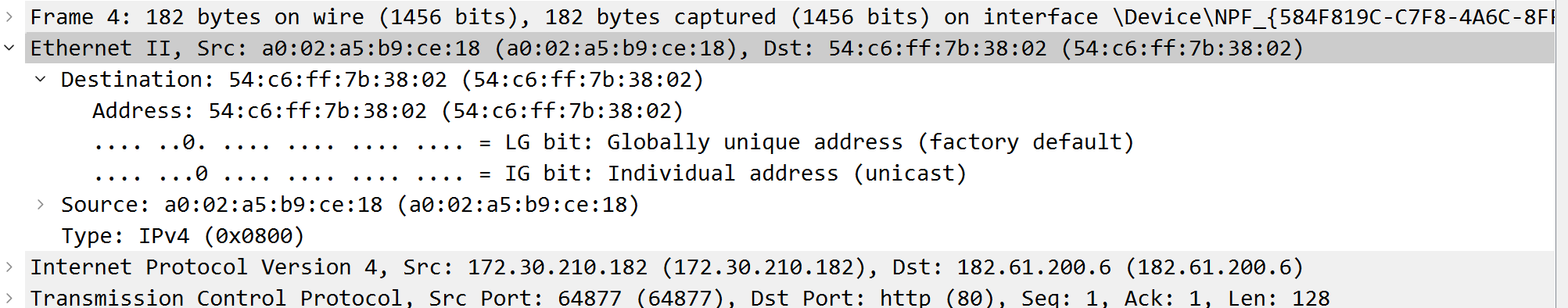
包括了捕获时间：Nov 15,2024 10:30:36

Frame number：4 Length: 182bytes



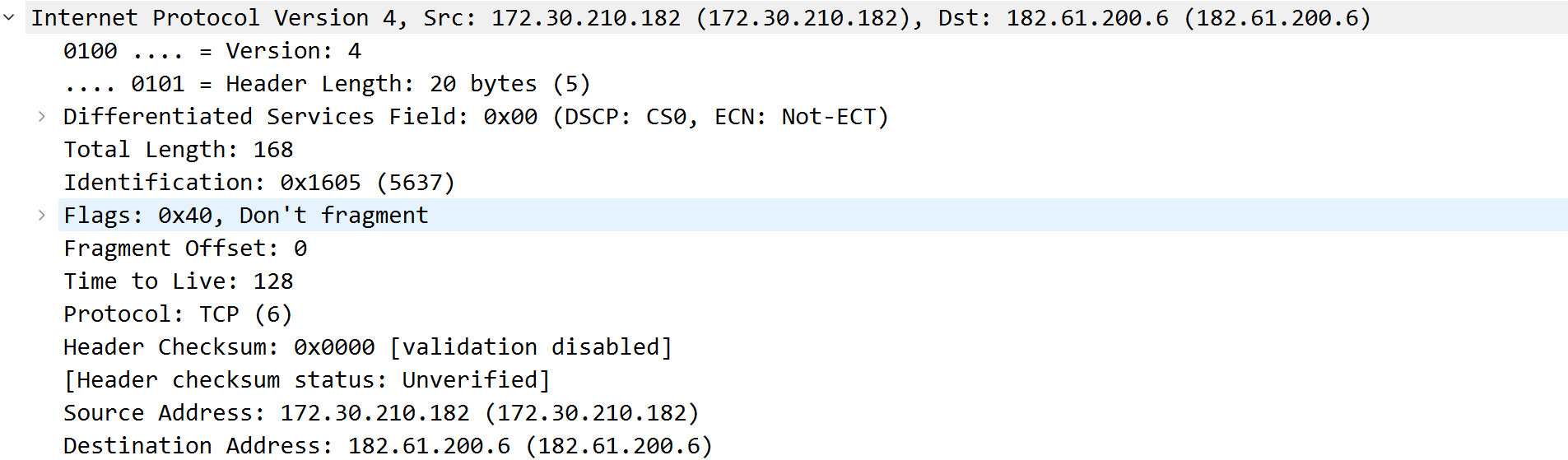
还显示了包里的协议内容：包含Ethernet、IP、TCP、HTTP协议，且按照从低到高的顺序放在协议栈中。

(2) Ethernet协议:



14byte

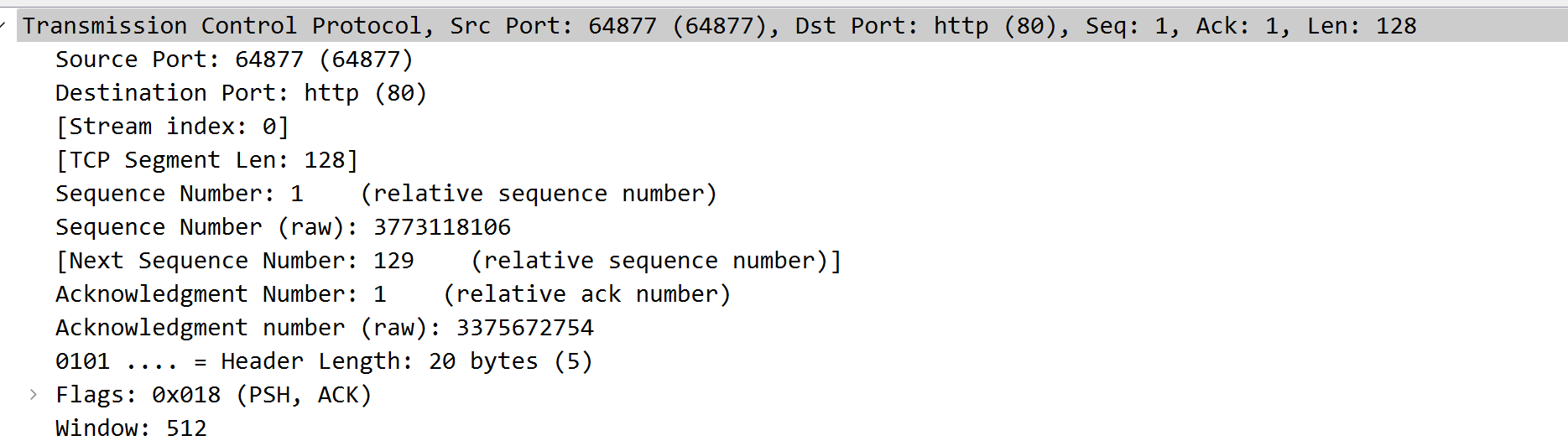
(3) IP协议:

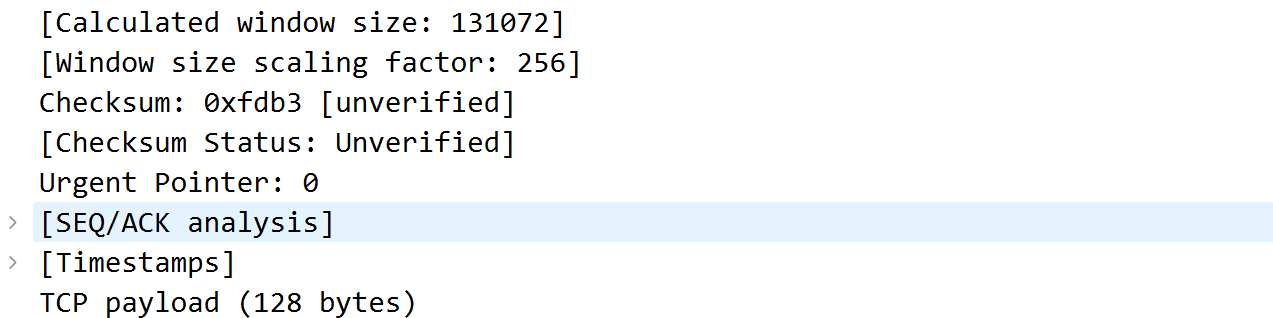


Total length是168

Header Length是20Byte

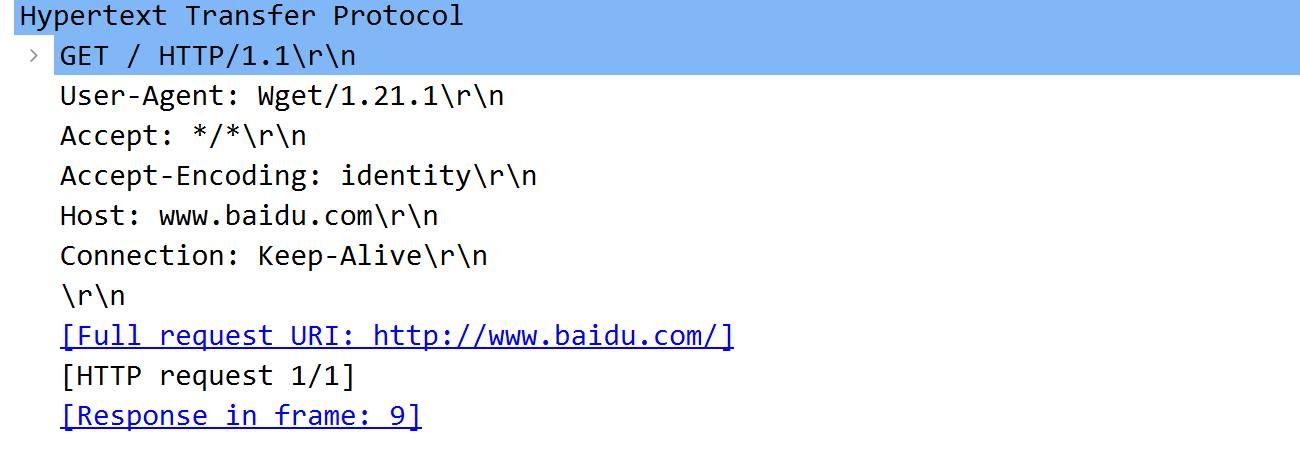
(4) TCP协议:





TCP length为20byte

(5) HTTP协议:



HTTP长度为128byte

2.分析并且绘制协议包。

综合上述信息，可以得到一下结果：

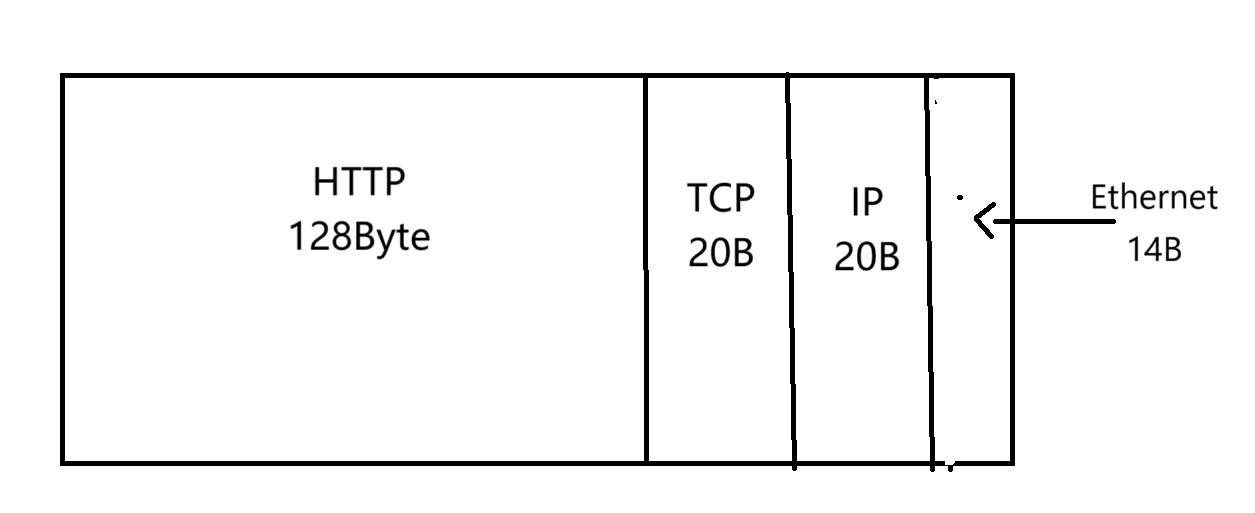
HTTP：128byte

TCP：20byte

IP：20byte

Ethernet：14byte

可以绘制出协议包的结构图：



**2.3分析协议开销**

1.根据实验结构，分析Http应用协议额外开销。

HTTP（超文本传输协议）应用协议额外开销是指在 HTTP 通信过程中，除了实际传输的有效内容（如网页的文本、图像、视频等数据）之外，因协议自身的控制信息、格式信息等所占用的额外数据量。请求行和状态行开销，消息头开销，空行开销，协议版本和兼容性相关开销都是额外开销的组成部分。

HTTP数据（包含头部和信息）的开销都是有用的。而TCP，IP，Ethernet的头部是额外的开销。

2. 估计上面捕获的Http协议的额外开销。

TCP：20/(128+20）=13.51%

IP:20/168=11.90%

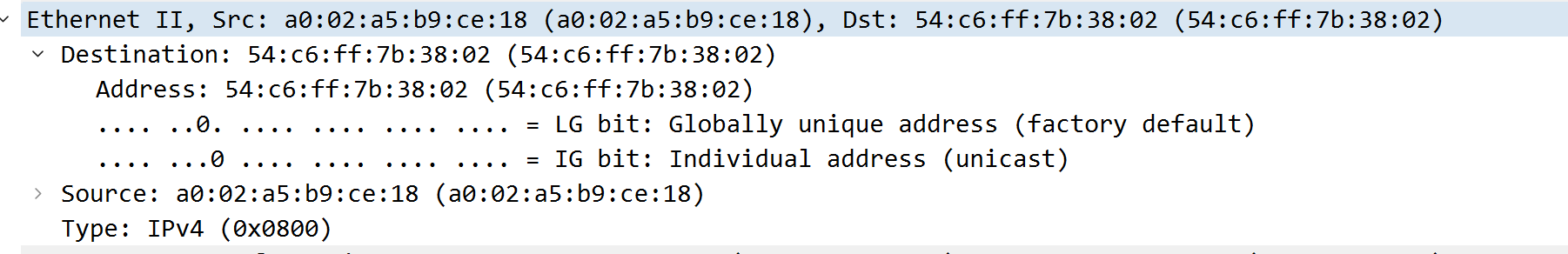
Ethernet:14/182=7.69%

**2.4 分析解复用键：**

不同的协议有其特定的标识符。在数据链路层、网络层、传输层等各个层次都有相应的协议来处理数据。例如，在以太网中，帧头中的类型字段可以指示网络层协议IP 数据包中，协议字段可以指示传输层协议。

观察下载的以太网和IP包头部信息回答下面问题：

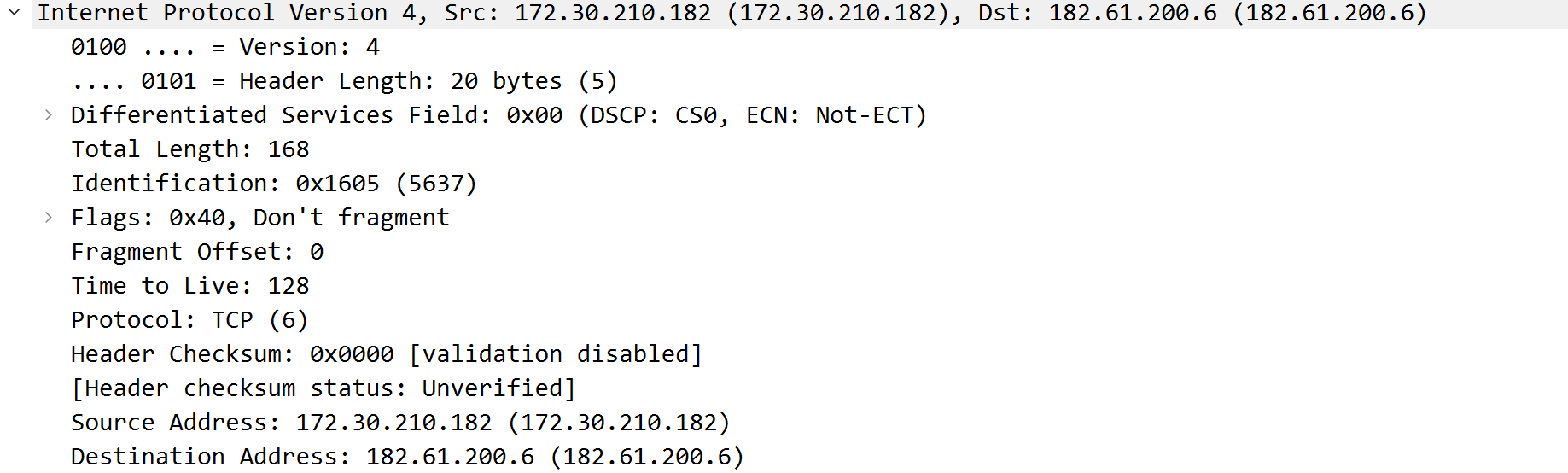
1.以太网头部中哪一部分是解复用键并且告知它的下一个高层指的是IP，在这一包内哪一个值可以表示IP？



IPv4(0x0800)是解复用键，下一个高层是IP

其中0x0800表示IP

2.IP头部中哪一部分是解复用键并且告知它的下一一个高层指的是TCP，在这一包内哪一个值可以表示TCP？



Protocol：TCP(6)是解复用键，下一个高层是TCP

其中 6 表示TCP

**五、实验结果总结**

**问题讨论：**

1. **查看一个不携带高层数据的简短 TCP 数据包。这个数据包要发往哪个实体呢？毕竟，如果它不携带高层数据，那么对于像 HTTP 这样的高层协议来说，它似乎没什么用处！**

**ANS**：即使 TCP 数据包不携带高层数据（如 HTTP 数据），它仍然可能用于多种目的。这个数据包可能是用于 TCP 协议自身的控制信息传输。例如，它可能是一个 ACK（确认）数据包，目的是确认之前收到的数据段，其目的地是发送端对应的 TCP 实体，目的是控制连接的状态。

**2. 在经典的分层模型中，来自高层的一条消息会被低层附加一个头部，从而变成一条新消息。但情况并非总是如此。在上面，我们看到了一个追踪示例，其中网络响应（一条由 HTTP 头部和 HTTP 有效载荷组成的 HTTP 消息）被转换成了多个低层消息（即多个 TCP 数据包）。设想一下，你已经为携带网络响应的第一个和最后一个 TCP 数据包绘制了数据包结构（如步骤 2 中那样）。这两种情况画出的结构图会有何不同？**

对于携带网络响应的第一个 TCP 数据包，在其 TCP 头部之后，会包含 HTTP 消息的起始部分。通常会是 HTTP 头部的一部分或者全部。如果 HTTP 消息头比较长，可能第一个 TCP 数据包几乎全部是 HTTP 头部内容。

而最后一个 TCP 数据包主要包含 HTTP 消息的结尾部分。会包含HTTP有效载荷数据，但没有HTTP头

**3. 在上述描述的经典分层模型中，低层会给从高层传下来的消息附加头部。如果低层添加了加密功能，这个模型会发生怎样的变化？**

在加密的情况下，低层协议层将重写其主体中的有效载荷以及附加其头。从高层的角度来看，加密过程是透明的。HTTP协议仍然按照正常的方式生成消息并传递给下层。但是，在下层进行加密后，传输过程中的中间设备将无法直接查看和处理 HTTP消息内容，因为它们看到的是加密后的数据包。当数据包在网络上时，上层协议的内部结构将不再可见；当接收的对等协议对消息进行解密时，它将对接收方可见。

**4. 在上述描述的经典分层模型中，低层会给从高层传下来的消息附加头部。如果低层添加了压缩功能，这个模型会发生怎样的变化？**

当低层添加压缩功能时，通常会在附加头部之后进行压缩。对于上层协议，压缩过程也是透明的。HTTP 协议还是按照正常方式生成消息。但是，压缩后的数据包大小会发生变化，有效载荷将比原始有效载荷短，这可能会影响到一些基于数据包大小进行操作的机制。在接收端，相应的低层需要先对压缩的数据进行解压，需要有相应的错误处理机制，增加了复杂性。

**实验总结：**

本次实验我成功地完成了wget和wireshark的安装，并学会了一些基础的操作，掌握了一些基础的抓包方法，成功获取了各协议的数据包，通过具体的实践分析，我也对数据包的有了更加深入的了解，对协议开销的分析和计算也有了更加深刻的认识。同时，通过对思考题的回答，我也有了特殊情况下数据包的认知，对模型的变换有了更多的理解。

实验过程中，由于对本次实验的内容在实验前认识不深，对解复用键这一概念不够理解，通过实验也对这个概念有了更加具体的认知，很有利于之后理论的学习！

**六、附录**