**中国科学技术大学计算机学院**

**《计算机网络》实验报告**



实验题目：lab3\_WireShark Labs(TCP)

学生姓名：胡毅翔

学生学号：PB18000290

专业：计算机科学与技术

指导老师：张信明

完成日期：2020年11月15日

计算机实验教学中心制

2019年09月

## **实验目的**

1.进一步掌握Wireshark网络分析工具。

2.捕获观察并分析TCP报文段结构。

3.回答本次实验指导书中的问题。

## **实验原理**

本次实验使用WireShark工具。其中，用来观察执行协议实体之间交换的报文的基本工具称为分组嗅探器(packet sniffer)。分组嗅探器被动地拷贝(嗅探)由计算机发送和接收的报文；它也能显示出这些被捕获报文的各个协议字段的内容。分组嗅探器从不发送报文，同时接收到的报文也不会显式地发送到分组嗅探器。它接受的是发送/接收的报文的复制。

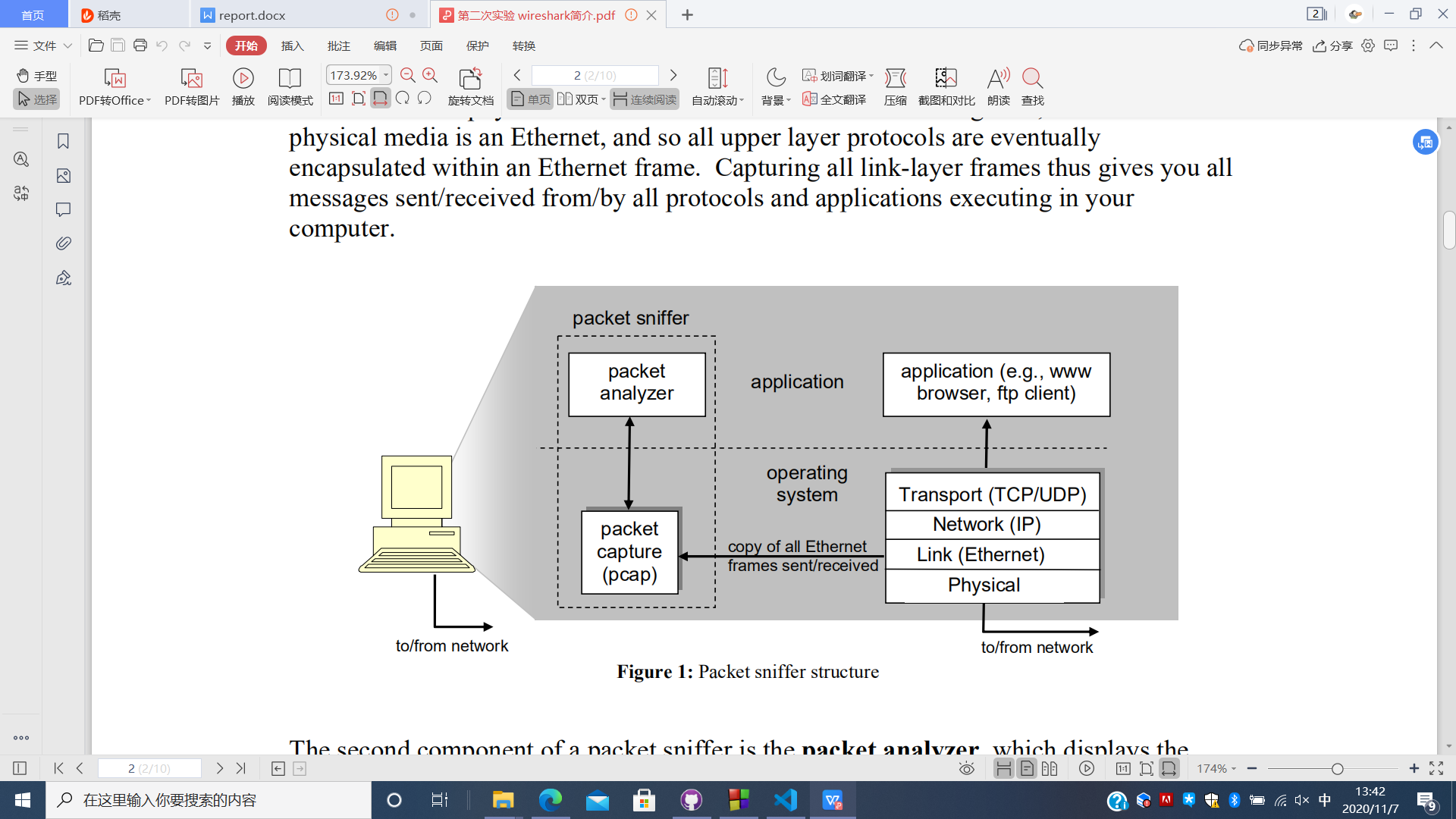


图 1

分组嗅探器的结构如图1所示。在图1的右边是运行在计算机上的协议和应用。分组嗅探器(图中画虚线框部分)是计算机中的附加软件(区别于上述协议和应用)，它包含两个部分。分组捕获库获取每一个链路层接收/发送的帧。第二部分是分组分析器，其中显示了协议所有字段的内容。为了实现这一目的，分组分析器必须理解所有协议所交换的信息的结构。比如，我们对图1 中HTTP协议的各个字段信息感兴趣。分组分析器理解以太网帧的格式，所以可以从以太网帧中区分出IP数据报。同时，它还理解IP数据报格式，所以它能从IP数据报分离出TCP报文段。最后，它还理解TCP报文段格式，从中分离出HTTP报文。又因它理解HTTP协议，所以能在实现WireShark中显示HTTP协议各字段信息的功能。

## **实验环境**

1.PC一台

2.Windows系统

3.WireShark网络分析工具(版本 3.2.7)

4.Edge浏览器(版本 86.0.622.56)

## **实验过程**

### **WireShark Lab: TCP**

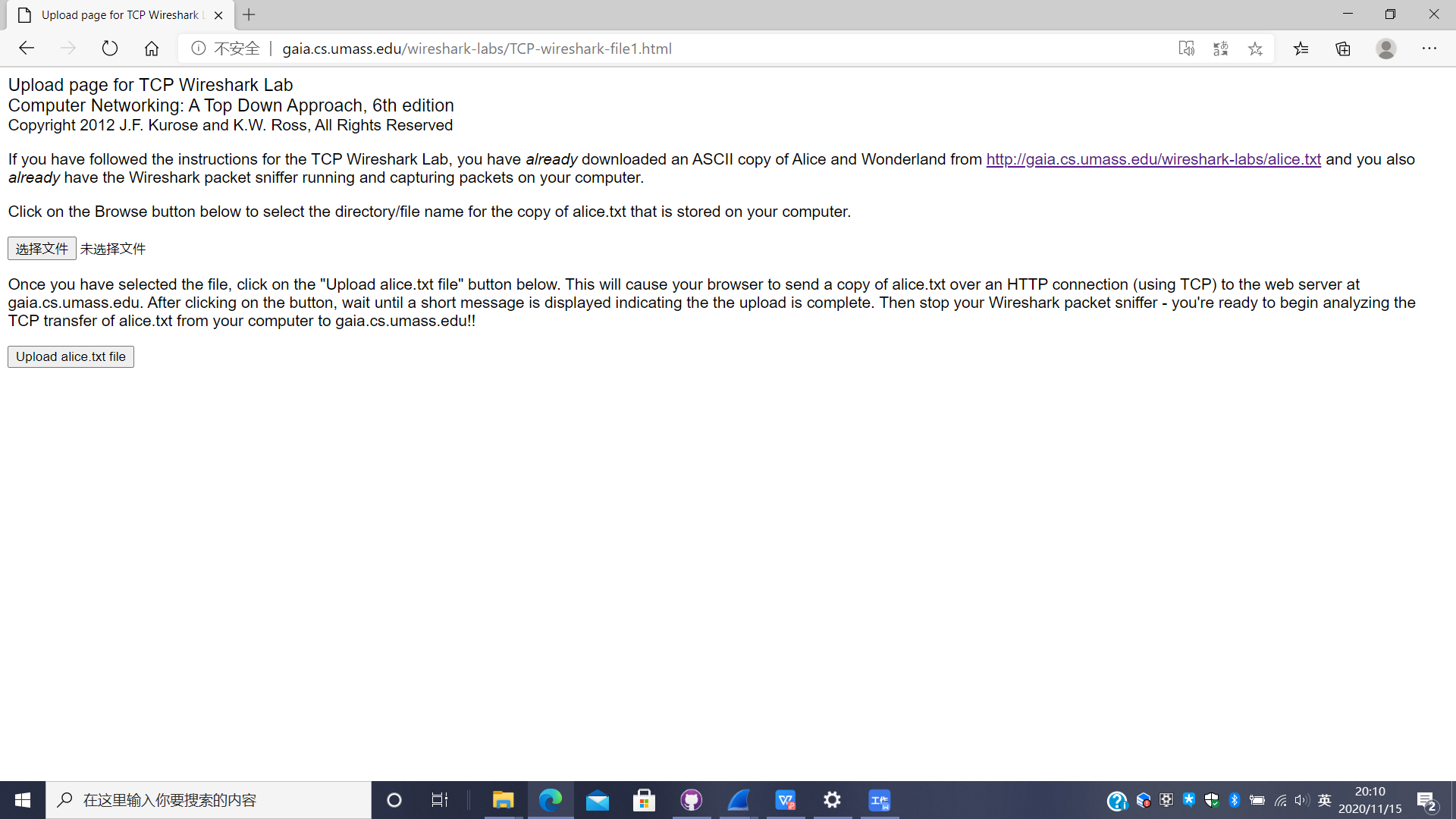
**Capturing a bulk TCP transfer from your computer to a remote server**

##### **实验步骤**

1.启动浏览器。前往<http://gaia.cs.umass.edu/wireshark-labs/alice.txt>，得到《爱丽丝梦游仙境》的ASCII码副本。保存到计算机上。

2.前往<http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>。

3.进入到如下界面。



4.点击“选择文件”，选择刚刚保存的alice.txt。

5.启动WireShark分组嗅探器。

6.上传alice.txt。

7.得到上传成功的信息后，停止捕获分组。

**A first look at the captured trace**

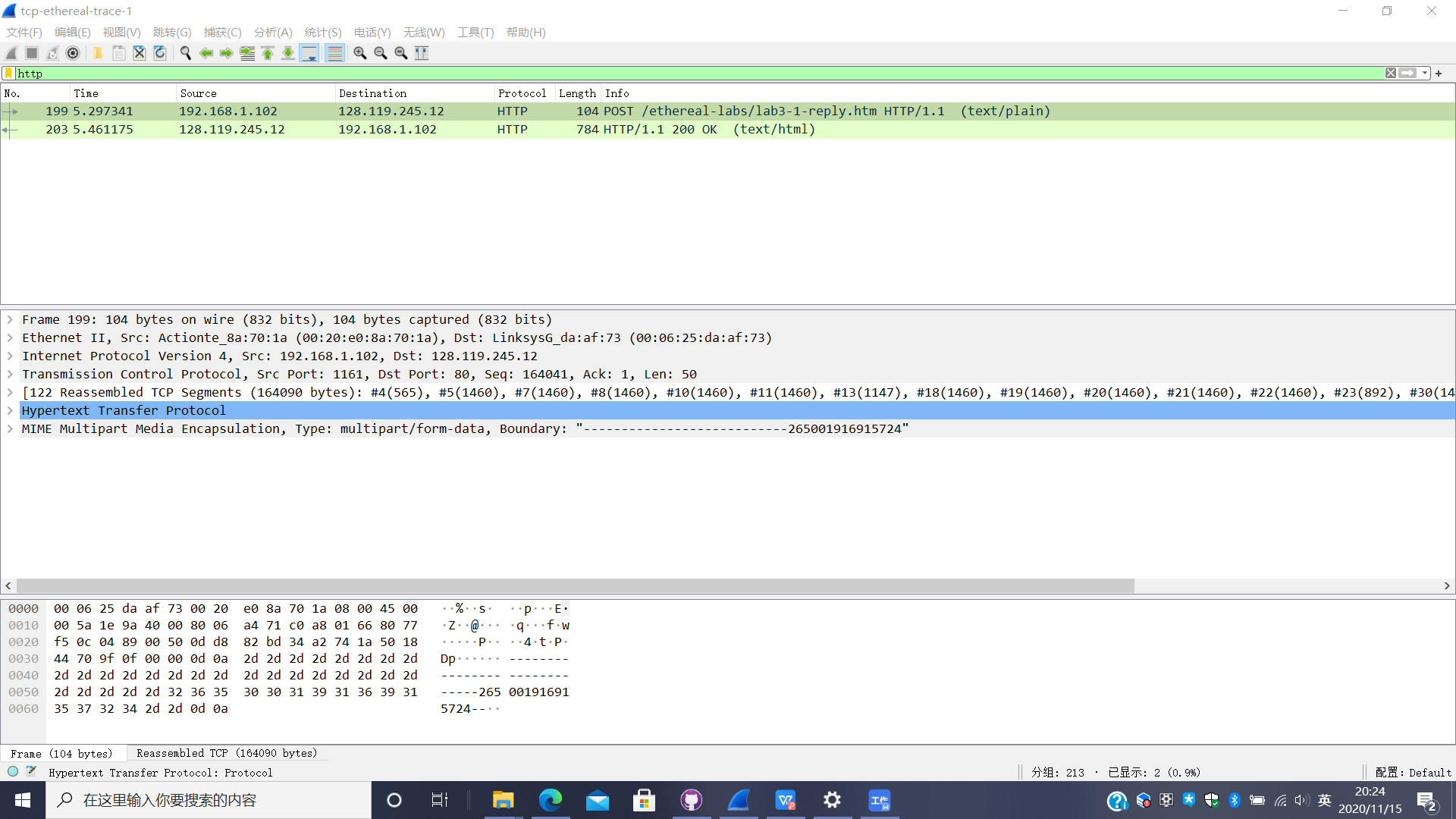
##### **实验步骤**

1.在显示过滤器中输入”tcp”，观察捕获到的数据。

2.根据实验指导书要求，下载wireshark-traces.zip，观察tcp-ethereal-trace-1，并回答对应问题。

##### **问题**

Q1:What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu? To answer this question, it’s probably easiest to select an HTTP message and explore the details of the TCP packet used to carry this HTTP message, using the “details of the selected packet header window” (refer to Figure 2 in the “Getting Started with Wireshark” Lab if you’re uncertain about the Wireshark windows).

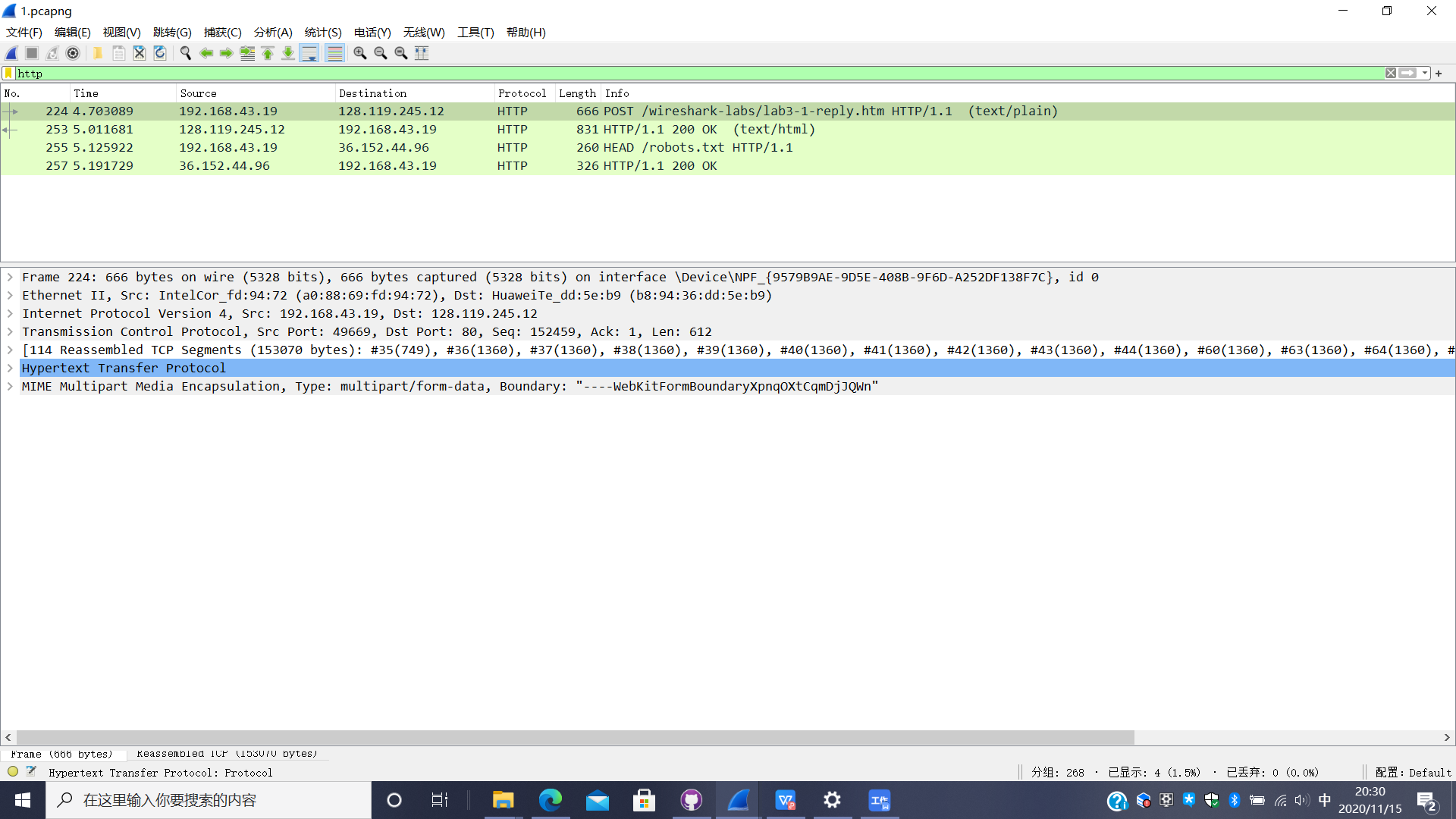
A1:客户端的IP地址是192.168.1.102，TCP端口号是1161。

Q2:What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?

A2:gaia.cs.umass.edu的IP地址是128.119.245.12。本次链接使用的端口号是80。

Q3:What is the IP address and TCP port number used by your client computer (source) to transfer the file to gaia.cs.umass.edu?

A3:我的计算机的IP地址是192.168.43.19，使用的TCP端口号是49669。



**TCP Basics**

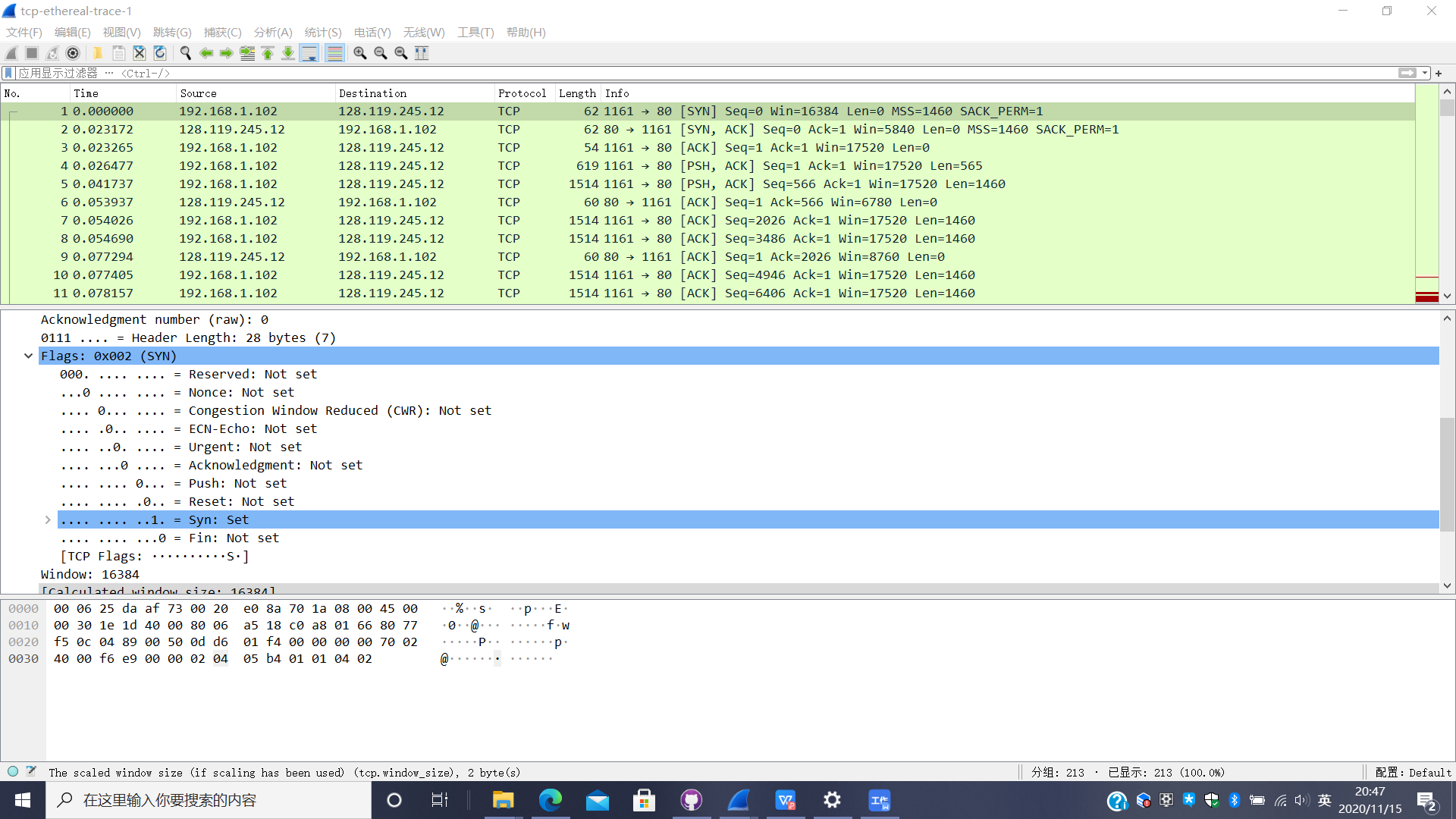
##### **实验步骤**

1.在分析->启动的协议中，反选HTTP，作进一步分析。

##### **问题**

Q4: What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is it in the segment that identifies the segment as a SYN segment?

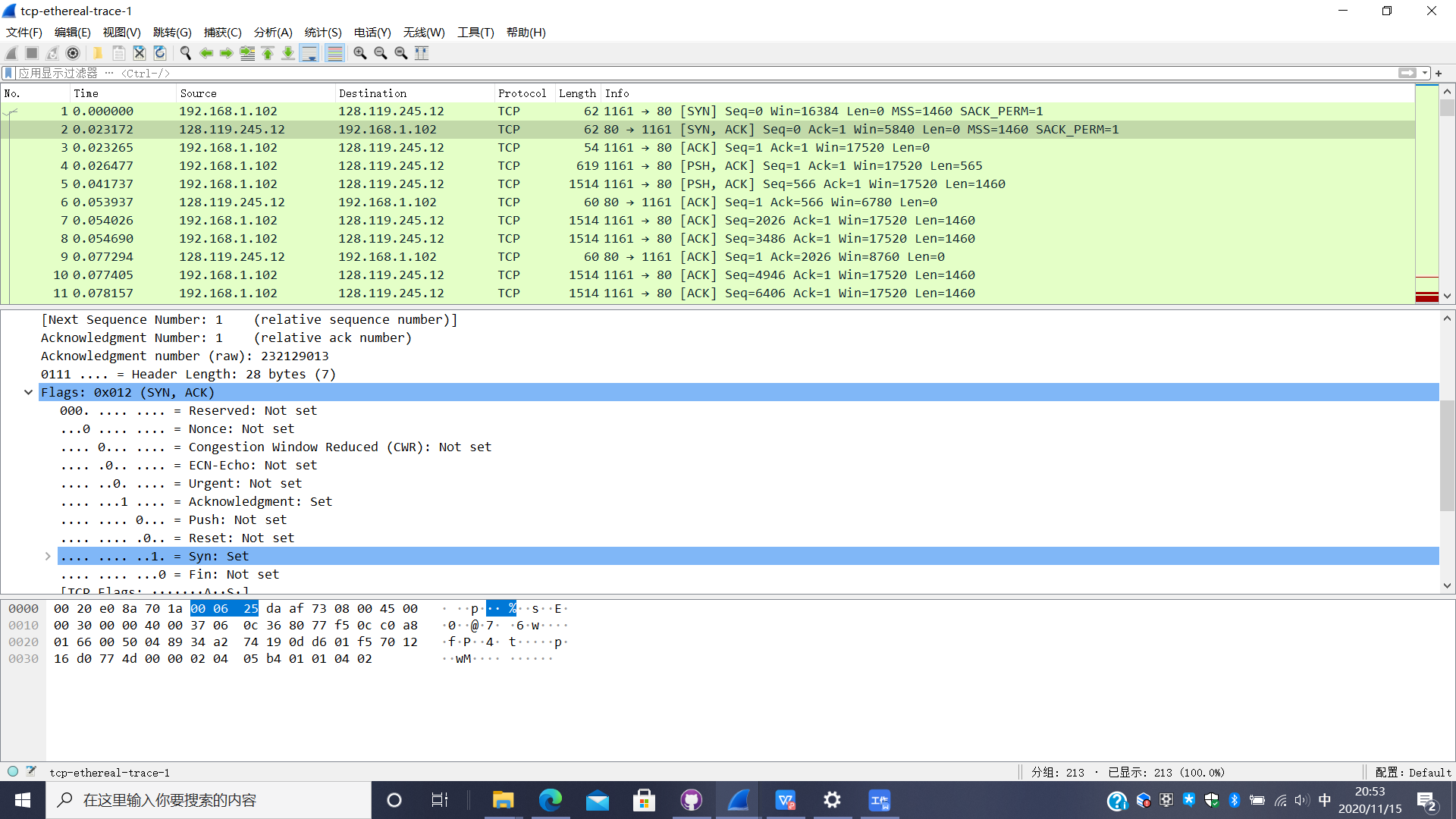
A4:客户端的SYN报文段的序列号是0。从图中可见，在Flags中SYN被置为1，表示该报文段是SYN报文段。



Q5:What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the ACKnowledgement field in the SYNACK segment? How did gaia.cs.umass.edu

determine that value? What is it in the segment that identifies the segment as a SYNACK segment?

A5:服务器的SYNACK报文段的序列号是0。该报文段中ACKnowledgment域的值为1，该值为1是因为：已收到序列号为0的报文段，期望收到序列号为1的报文段。Flags中SYN，ACK置为1，表明这是SYNACK报文段。



Q6:What is the IP address of your computer? Of the gaia.cs.umass.edu server?What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you’ll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a “POST” within its DATA field.

A6:gaia.cs.umass.edu服务器的IP地址是128.119.245.12。我的计算机的IP地址是192.128.43.19。

Q7:Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value (see page 249 in text) after the receipt of each ACK? Assume that the value of the EstimatedRTT is equal to the measured RTT for the first segment, and then is computed using the EstimatedRTT equation on page 249 for all subsequent segments.

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the “listing of captured packets” window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics->TCP Stream Graph- >Round Trip Time Graph.

A7:根据响应行”HTTP/1.1 200 OK\r\n”,可知返回的状态码是200。

Q8: What is the length of each of the first six TCP segments?

A8:根据首部行”Last-Modified: Sat, 07 Nov 2020 06:59:01 GMT\r\n”，可知最后一次修改的时间是格林威治时间2020年11月7日06:59:01。

Q9:What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

A9:根据首部行”Content-Length: 128\r\n”，可知返回的内容程度是128字节。

Q10: Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

A10:通过观察，请求报文packeting-list window中有首部行：Host,Connection,Upgrade-Insecure-Request,User\_Agent, Accept,Accept-Encoding,Accept-Lauguage，与packet content window中所示相同。 通过观察，响应报文packeting-list window中有首部行：Date,Server,Last-Modified、ETag、Accept-Ranges、Content-Length、Keep-Alive,Connection, Content-Type，与packet content window中所示相同。

Q11:How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (see Table 3.2 on page 257 in the text).

A11:

Q12:What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

A12:

#### **The HTTP CONDITIONAL GET/response interaction**

##### **实验步骤**

1.启动Edge浏览器，清除缓存。

2.启动WireShark分组嗅探器。

3.进入<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>。

4.刷新页面。

5.停止WireShark捕获分组

6.在显示过滤器上输入http。

##### **问题**

Q1:Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

A1:观察第一个GET请求报文的首部行，并没有看到“IF-MODIFIED-SINCE”。

Q2:Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

A2:是的。在实体体中有”HTTP-wireshark-file2.html”这一文件的内容。

Q3:Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

A3:在第二个GET请求报文中有首部行”If-Modified-Since: Sat, 07 Nov 2020 06:59:01 GMT”。

Q4:What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

A4:对应第二个GET请求报文的响应报文的响应行为”HTTP/1.1 304 Not Modified\r\n”，状态码为304，状态信息Not Modified。在该响应报文中没有返回所请求的html文件的内容，因为服务器中的该文件相较于上次请求未发生修改，说明本地缓存的该文件于所请求的内容相同，那么服务器只需告诉客户端未发生修改这一信息，便可实现其功能，这有利于减少网络负载。

#### **Retrieving Long Documents**

##### **实验步骤**

1.启动Edge浏览器，清除缓存。

2.启动WireShark分组嗅探器。

3.进入<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>。

4.刷新页面。

5.停止WireShark捕获分组。

6.在显示过滤器上输入http。

##### **问题**

Q1:How many HTTP GET request messages were sent by your browser?

A1:如下图4可见，发送了3个GET请求报文。

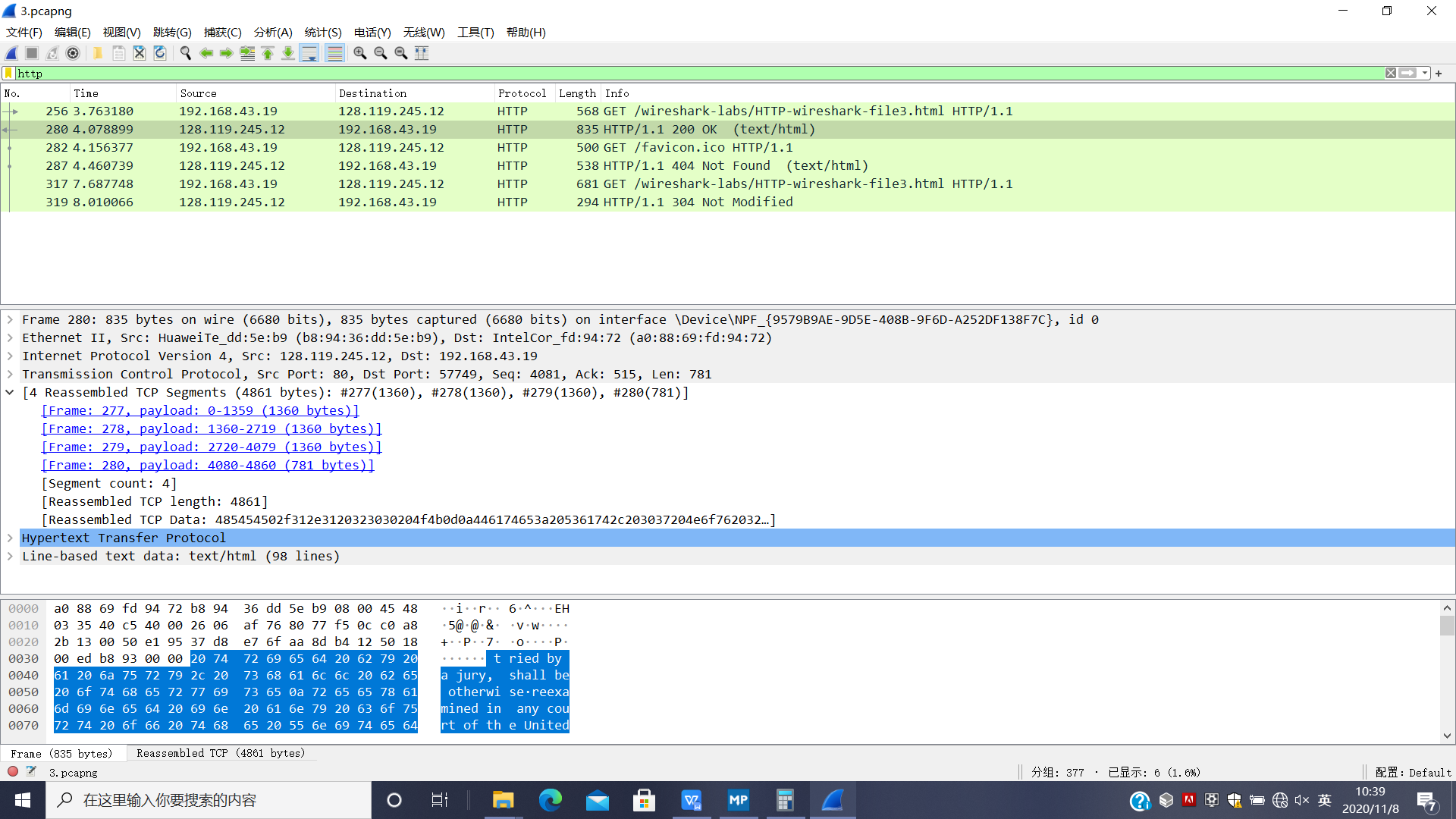


图 4

Q2:How many data-containing TCP segments were needed to carry the single HTTP response?

A2:如图4中TCP详细信息可见，这一HTTP响应报文分为了4个TCP报文段。

Q3:What is the status code and phrase associated with the response to the HTTP GET request?

A3:第一个GET请求报文(请求html文件)对应的响应报文的首部行为”HTTP/1.1 200 OK\r\n”。第二个GET请求报文(请求网站图标)对应的响应报文的首部行为”HTTP/1.1 404 Not Found\r\n”。第三个GET请求报文(再次请求同一html文件)对应的响应报文的首部行为”HTTP/1.1 304 Not Modified\r\n”。

Q4:Are there any HTTP status lines in the transmitted data associated with a TCP induced “Continuation”?

A4:根据封包详细信息，并没有在HTTP中发现和TCP引起的持续的相关信息。

#### **HTML Documents with Embedded Objects**

##### **实验步骤**

1.启动Edge浏览器，清除缓存。

2.启动WireShark分组嗅探器。

3.进入<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>。

4.停止WireShark捕获分组。

5.在显示过滤器上输入http。

##### **问题**

Q1:How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?

A1:如图5，浏览器发送了3个GET请求报文。第一个是发往gaia.cs.umass.edu;第二个是发往gaia.cs.umass.edu;第三个是发往manic.cs.umass.edu。

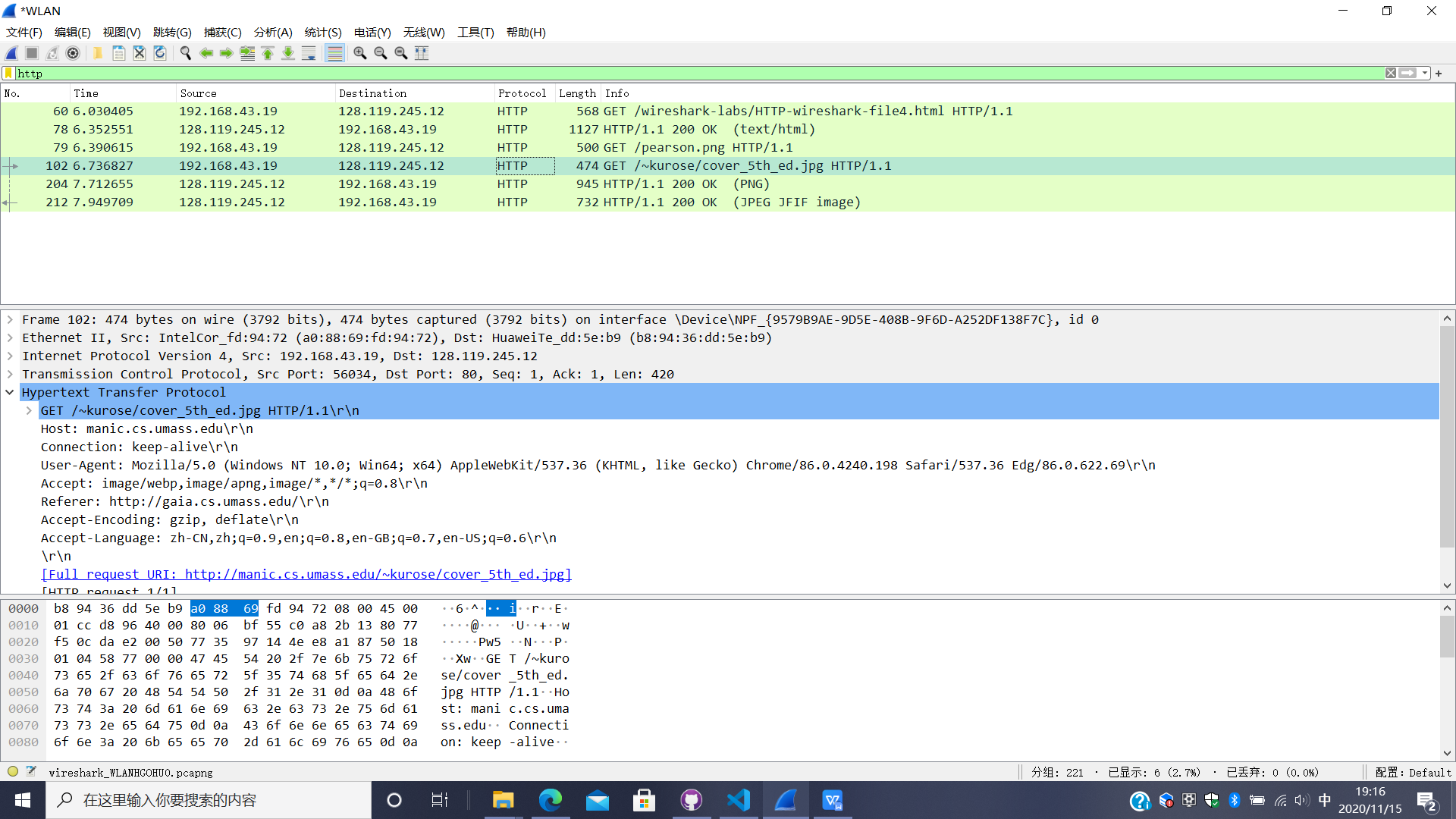


图 5

Q2:Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

A2:根据图5，是并行下载。因为在发送请求pearson.png文件，接收对应响应报文之前，就发送了cover\_5th\_ed.jpg文件的请求报文。

#### **HTTP Authentication**

##### **实验步骤**

1.清除缓存，关闭浏览器，启动Edge浏览器。

2.启动WireShark分组嗅探器。

3.进入<http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html>。

4.停止WireShark捕获分组。

5.在显示过滤器上输入http。

##### **问题**

Q1:What is the server’s response (status code and phrase) in response to the initial HTTP GET message from your browser?

A1:第一个GET请求报文对应的响应报文”HTTP/1.1 401 Unauthorized\r\n” 。

Q2:When your browser’s sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

A2:在第二次发送的GET请求报文中，新增首部行”Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcm s=\r\n”。

## **结果分析**

#### **HTTP中GET和POST请求方式的区别**

HTTP中GET方法请求指定的资源。使用GET 的请求应该只用于获取数据。

HTTP中POST方法发送数据给服务器。请求主体的类型由Content-Type首部指定。

一个 POST 请求通常是通过 [HTML 表单](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms)发送，并返回服务器的修改结果。在这种情况下, content type 是通过在 [<form>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form" \o "HTML <form> 元素表示文档中的一个区域，此区域包含交互控件，用于向 Web 服务器提交信息。) 元素中设置正确的 [enctype](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/form" \l "attr-enctype) 属性, 或是在 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input" \o "HTML <input> 元素用于为基于Web的表单创建交互式控件，以便接受来自用户的数据; 可以使用各种类型的输入数据和控件小部件，具体取决于设备和user agent。) 和 [<button>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/button" \o "HTML <button> 元素表示一个可点击的按钮，可以用在表单或文档其它需要使用简单标准按钮的地方。) 元素中设置 [formenctype](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input" \l "attr-formenctype) 属性来选择的:

* application/x-www-form-urlencoded: 数据被编码成以 '&' 分隔的键-值对, 同时以 '=' 分隔键和值. 非字母或数字的字符会被 [percent-encoding](https://developer.mozilla.org/zh-CN/docs/Glossary/percent-encoding): 这也就是为什么这种类型不支持二进制数据(应使用 multipart/form-data 代替).
* multipart/form-data
* text/plain

当 POST 请求是通过除 HTML 表单之外的方式发送时，例如使用 [XMLHttpRequest](https://developer.mozilla.org/zh-CN/docs/Web/API/XMLHttpRequest)，那么请求主体可以是任何类型。按HTTP 1.1规范中描述，POST为了以统一的方法来涵盖以下功能：

* 注释已有的资源
* 在公告板，新闻组，邮件列表或类似的文章组中发布消息;
* 通过注册新增用户;
* 向数据处理程序提供一批数据，例如提交一个表单;
* 通过追加操作，扩展数据库数据.

|  |  |  |
| --- | --- | --- |
|  | GET方法 | POST方法 |
| 请求是否有主体 | 否 | 是 |
| 成功的响应是否有主体 | 是 | 是 |
| 安全 | 是 | 否 |
| 幂等 | 是 | 否 |
| 可缓存 | 是 | Only if freshness information is included |
| HTML表单是否支持 | 是 | 是 |

表 1

#### **总结**

本次实验通过WireShark分组嗅探器捕获并分析HTTP报文，深入了解了HTTP请求报文、响应报文的结构，对报文中的内容有了深刻的理解，在回顾教材内容的同时又有所提升。各个小实验中体分别现了不同首部行的实际含义，深入浅出，为今后的理论学习和实验打下基础。

附：本次实验使用WireShark分组嗅探器获得的数据部分以截图形式体现于实验报告中，完整数据在[Github仓库](https://github.com/Alpha-Girl/Network_2020/tree/master/lab2/src)中。