**计算机网络入门实验实验报告**

**姓名：陈鹤影 学号：PB21061287 日期：2022.9.11**

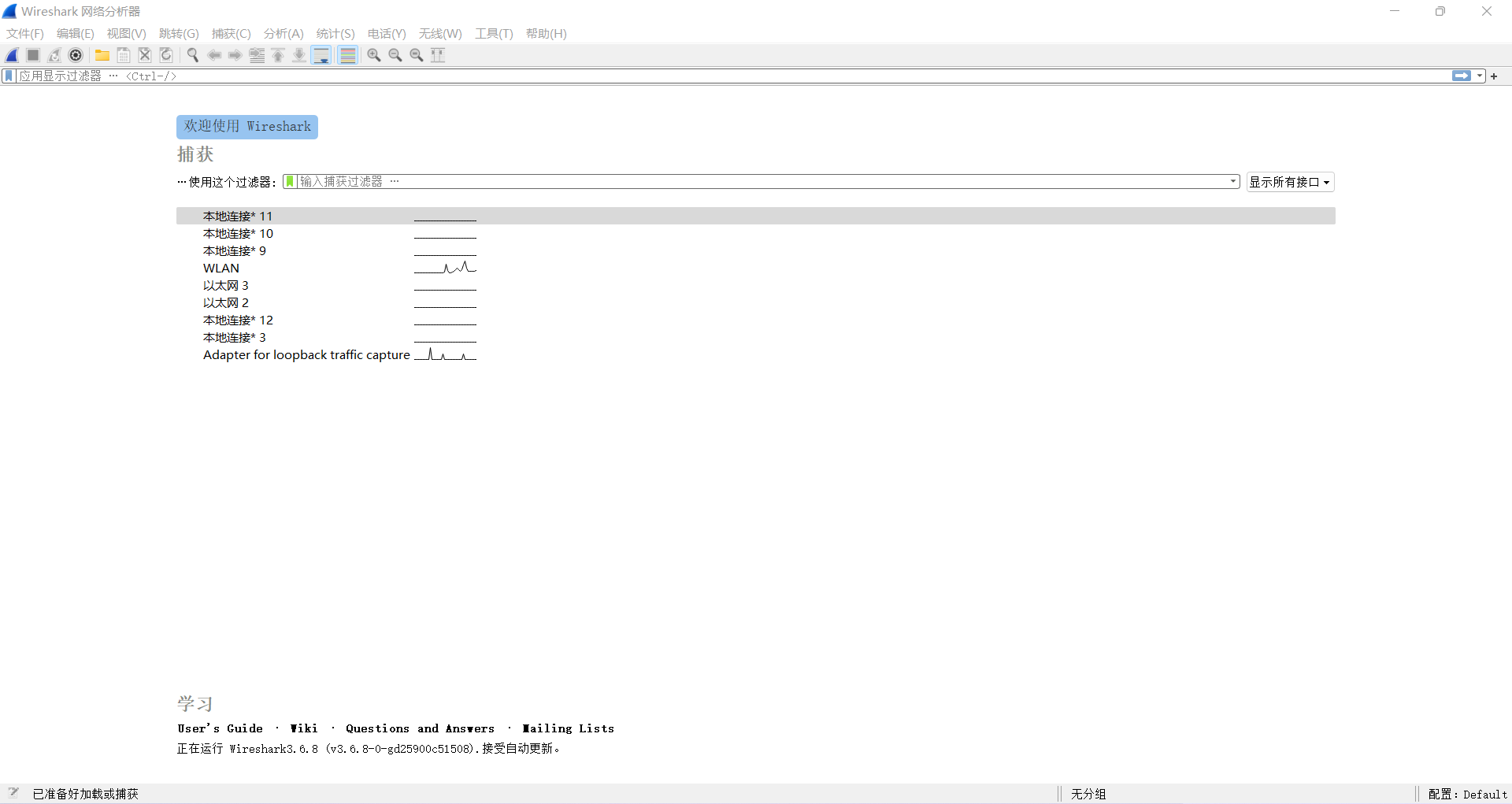
1. **实验流程：**

**Step 1：Getting Wireshark**

登入网页<http://www.wireshark.org/>，根据计算机系统选择并下载Windows-64安装包。

**Step 2：Running Wireshark**

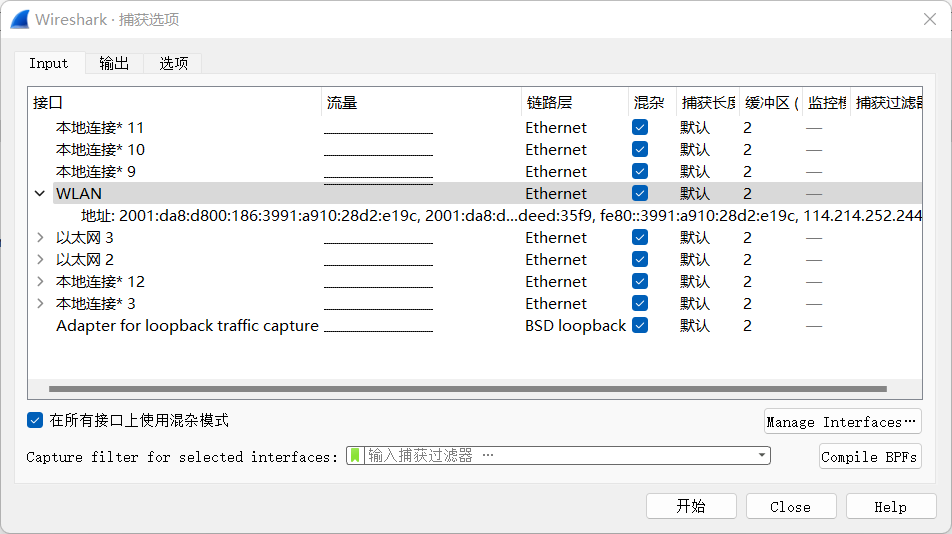
进入Wireshark界面，如下：



**（图1-1 Wireshark登入界面）**

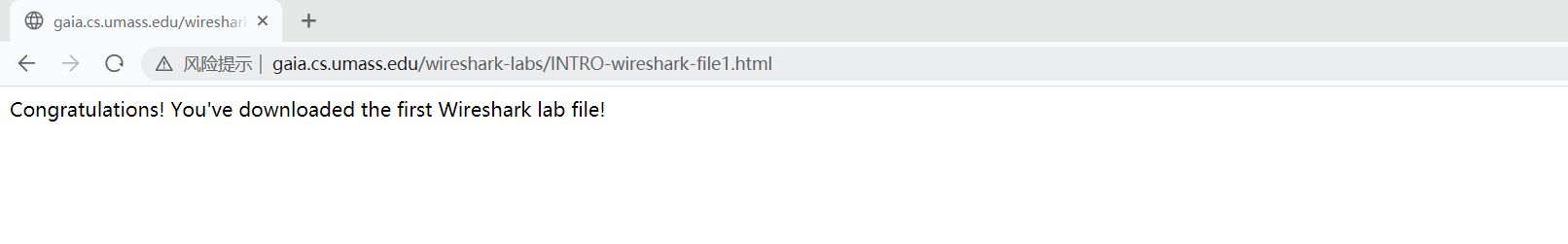
打开任意浏览器（实验中选择华为浏览器[Microsoft Edge会显示404 not found]）。

点击Capture下拉菜单选择接口（实验中选择WLAN）并开启Wireshark。



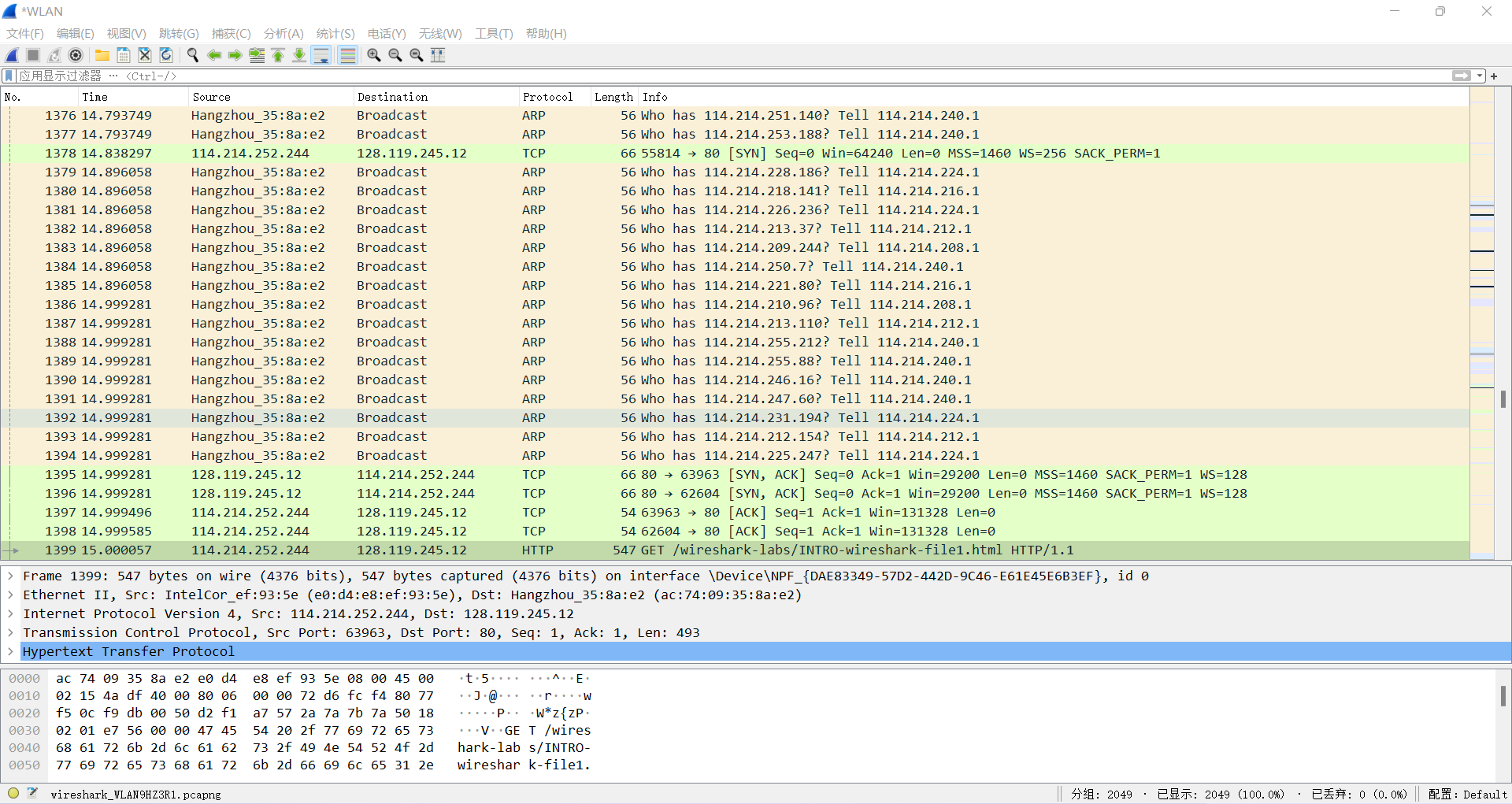
**（图1-2 Wireshark接口选择界面）**

在运行Wireshark的情况下，打开URL：<http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html> 。显示界面如下：



**（图1-3 URL显示界面）**

当浏览器成功显示上述信息后停止Wireshark抓包。Wireshark显示界面如下：



**（图1-4 Wireshark抓包显示界面）**

观察IP包结构，发现其包含：版本、头部长度、总长度、TTL、协议类型等字段。Packet Details Pane(数据包详细信息), 在数据包列表中选择指定数据包，在数据包详细信息中会显示数据包的所有详细信息内容。各行信息分别为：

**（1）Frame: 物理层的数据帧概况**

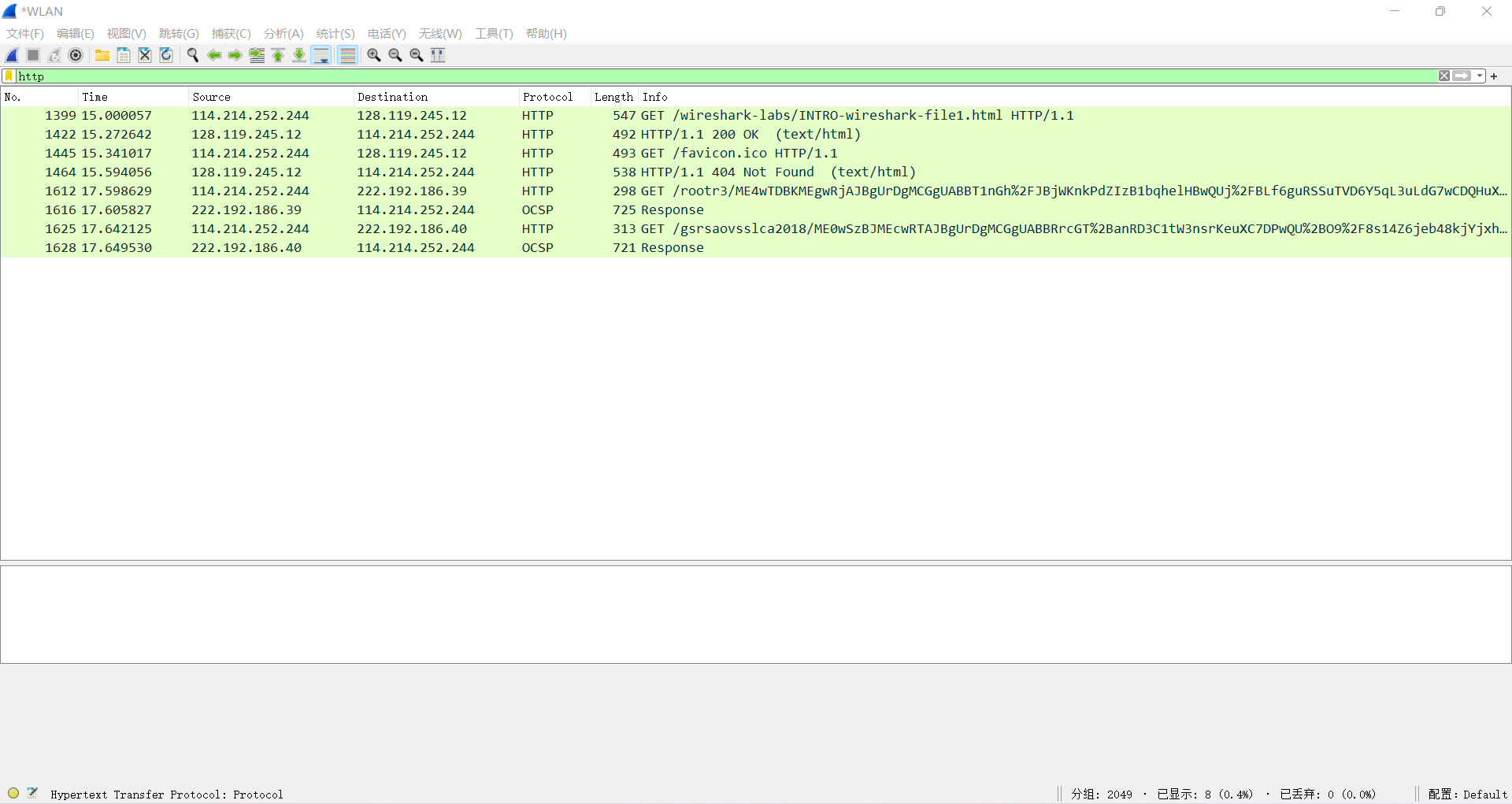
**（2）Ethernet II: 数据链路层以太网帧头部信息**

**（3）Internet Protocol Version 4: 互联网层IP包头部信息**

**（4）Transmission Control Protocol: 传输层T的数据段头部信息，此处是TCP**

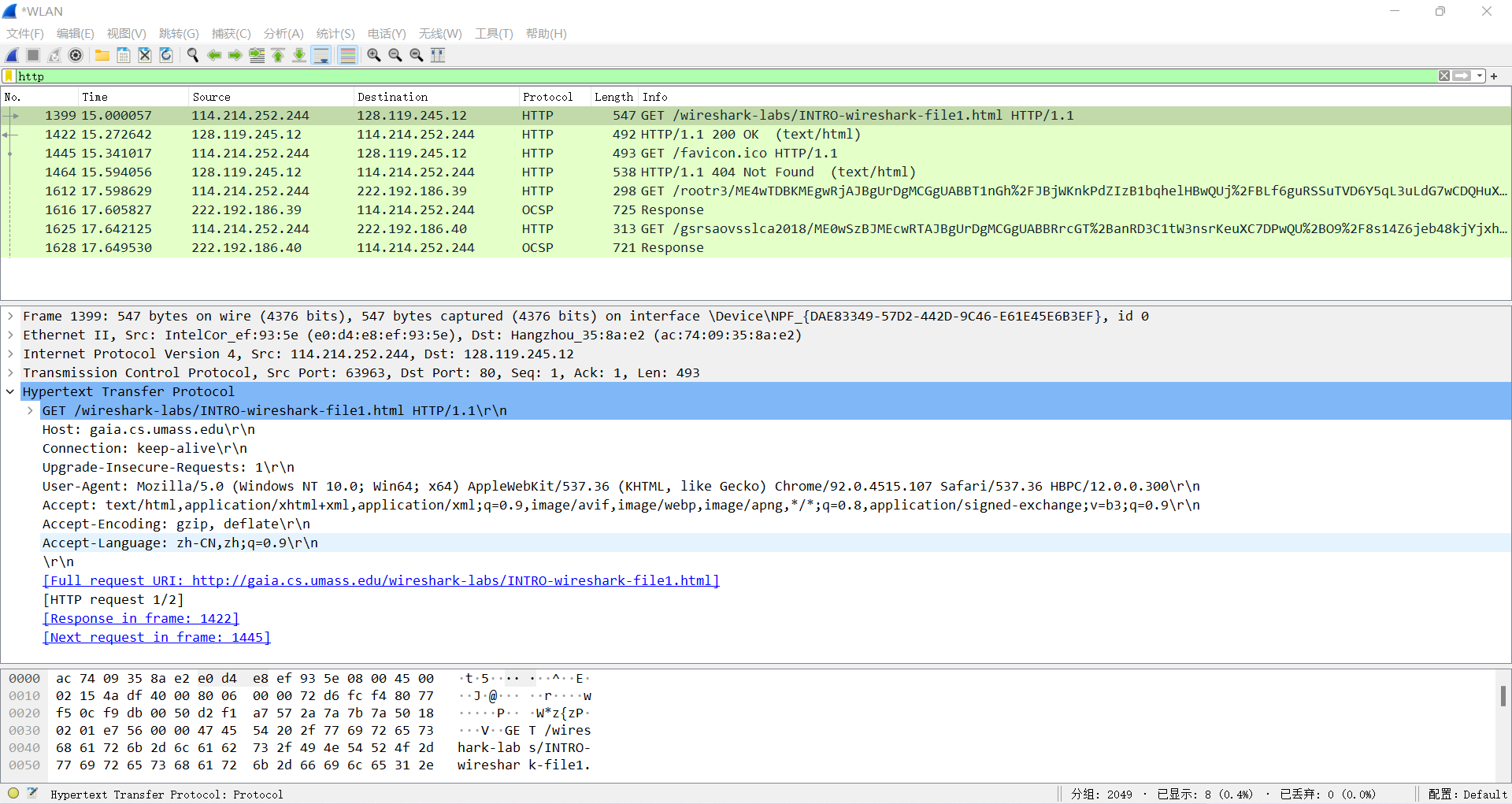
**（5）Hypertext Transfer Protocol: 应用层的信息，此处是HTTP协议**

在分组过滤器中选择http（小写）协议，显示如下：



**（图1-5 Wireshark抓包http协议显示界面）**

找到GET包内容如下：



**（图1-6 GET包内容显示）**

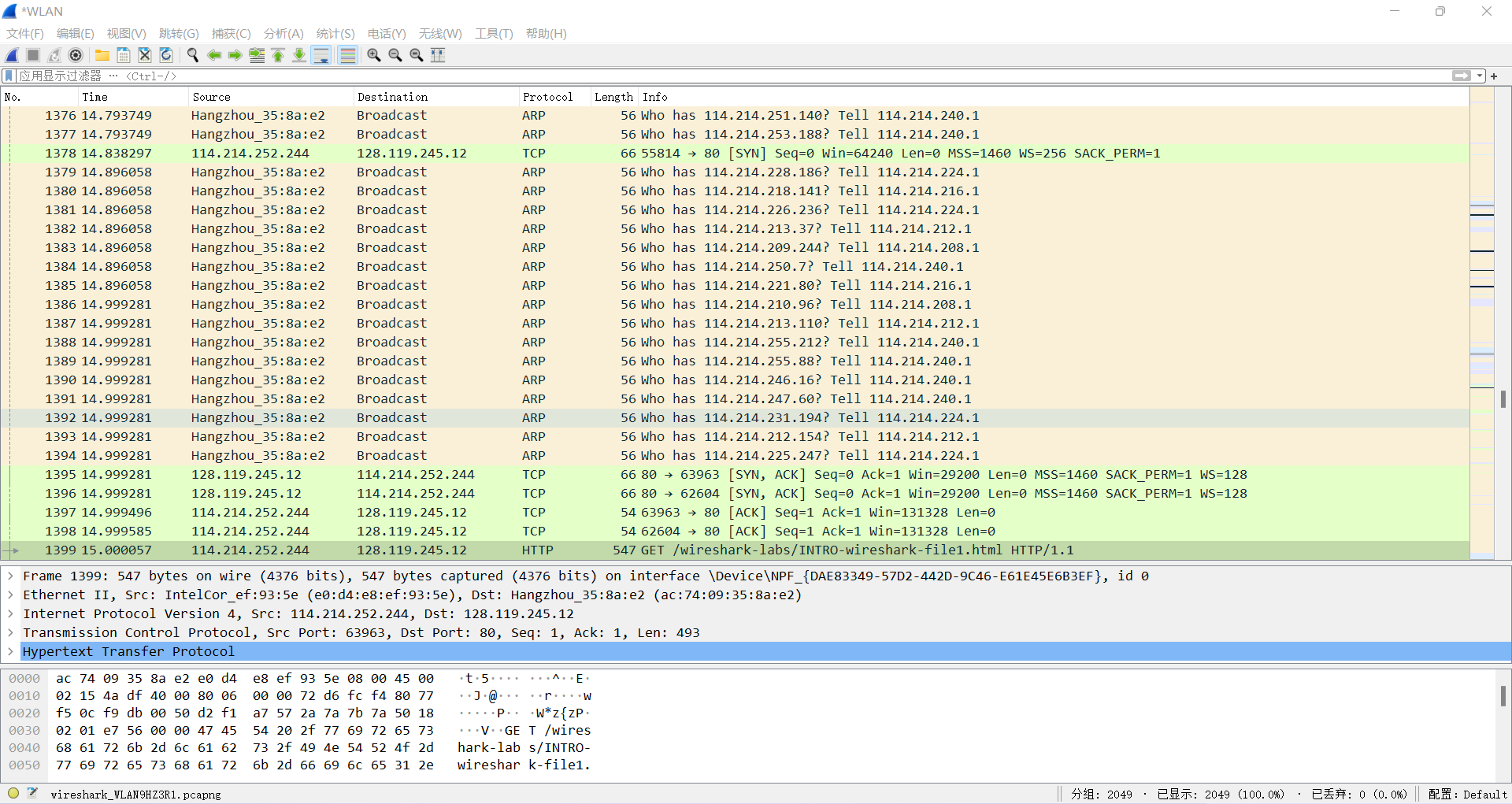
退出Wireshark。

1. **实验结果：**

成功打开[gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html](http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html) 网页，并获得**Congratulations! You've downloaded the first Wireshark lab file!**的内容显示。Wireshark成功捕相应的GET和OK报文。

1. **Hand in：**
2. **List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.**

**Ans：**TCP、ARP、HTTP。



**（图2-1 Wireshark抓包显示的几种不同协议）**

1. **How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-isting window is the amount of time, in seconds, since Wireshark tracing began.To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)**

**Ans：**根据总时间显示：发送GET报文时间为自抓包启动后15.000057s，接受到OK报文时间为自抓包启动后15.272642s。上述过程一共用时t = 15.272642s -15.000057s = 0.272585s。将时间显示格式改用为时间和日期后GET报文和OK报文对应时间分别为2022-09-11 20:56:49.404237及2022-09-11 20:56:49.676822。时间差为0.272585s。

1. **What is the** **Internet address of the gaia.cs.umass.edu (also known as www-net.cs.umass.edu)? What is the** **Internet address of your computer?**

**Ans：**Internet address of the gaia.cs.umass.edu：128.119.245.12。

Internet address of your computer：114.214.252.244。

1. **Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select Print from the Wireshark File command menu, and select the “Selected Packet Only” and “Print as displayed” radial buttons, and then click OK.**

**Ans：**报文导出内容如下：（附.pcapng和.prn文件）

**No. Time Source Destination Protocol Length Info**

**1399 2022-09-11 20:56:49.404237 114.214.252.244 128.119.245.12 HTTP 547 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1**

**Frame 1399: 547 bytes on wire (4376 bits), 547 bytes captured (4376 bits) on interface \Device\NPF\_{DAE83349-57D2-442D-9C46-E61E45E6B3EF}, id 0**

**Ethernet II, Src: IntelCor\_ef:93:5e (e0:d4:e8:ef:93:5e), Dst: Hangzhou\_35:8a:e2 (ac:74:09:35:8a:e2)**

**Internet Protocol Version 4, Src: 114.214.252.244, Dst: 128.119.245.12**

**Transmission Control Protocol, Src Port: 63963, Dst Port: 80, Seq: 1, Ack: 1, Len: 493**

**Hypertext Transfer Protocol**

**GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n**

**Host: gaia.cs.umass.edu\r\n**

**Connection: keep-alive\r\n**

**Upgrade-Insecure-Requests: 1\r\n**

**User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36 HBPC/12.0.0.300\r\n**

**Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n**

**Accept-Encoding: gzip, deflate\r\n**

**Accept-Language: zh-CN,zh;q=0.9\r\n**

**\r\n**

**[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]**

**[HTTP request 1/2]**

**[Response in frame: 1422]**

**[Next request in frame: 1445]**

**No. Time Source Destination Protocol Length Info**

**1422 2022-09-11 20:56:49.676822 128.119.245.12 114.214.252.244 HTTP 492 HTTP/1.1 200 OK (text/html)**

**Frame 1422: 492 bytes on wire (3936 bits), 492 bytes captured (3936 bits) on interface \Device\NPF\_{DAE83349-57D2-442D-9C46-E61E45E6B3EF}, id 0**

**Ethernet II, Src: Hangzhou\_35:8a:e2 (ac:74:09:35:8a:e2), Dst: IntelCor\_ef:93:5e (e0:d4:e8:ef:93:5e)**

**Internet Protocol Version 4, Src: 128.119.245.12, Dst: 114.214.252.244**

**Transmission Control Protocol, Src Port: 80, Dst Port: 63963, Seq: 1, Ack: 494, Len: 438**

**Hypertext Transfer Protocol**

**HTTP/1.1 200 OK\r\n**

**Date: Sun, 11 Sep 2022 12:56:50 GMT\r\n**

**Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.30 mod\_perl/2.0.11 Perl/v5.16.3\r\n**

**Last-Modified: Sun, 11 Sep 2022 05:59:01 GMT\r\n**

**ETag: "51-5e86079cbdf11"\r\n**

**Accept-Ranges: bytes\r\n**

**Content-Length: 81\r\n**

**Keep-Alive: timeout=5, max=100\r\n**

**Connection: Keep-Alive\r\n**

**Content-Type: text/html; charset=UTF-8\r\n**

**\r\n**

**[HTTP response 1/2]**

**[Time since request: 0.272585000 seconds]**

**[Request in frame: 1399]**

**[Next request in frame: 1445]**

**[Next response in frame: 1464]**

**[Request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]**

**File Data: 81 bytes**

**Line-based text data: text/html (3 lines)**

1. **补充实验：**

作为补充，我了解了wireshark内容过滤规则，将其中部分记录如下：

**wireshark过滤器表达式的规则**

**1、抓包过滤器语法**

（1）协议过滤：直接在抓包过滤框中输入相应的协议名，如：tcp、http（应注意使用小写）。

（2）IP过滤：使用host+IP地址的格式，若需要查询特定发送源或接收端对应分组，可在host前加上src、dst进行查询。

（3）端口过滤：使用port+端口号的格式，若需要查询特定发送端口或接收端口对应分组，可在port前加上src、dst进行查询。

（4）过滤器语法支持逻辑运算符：与（&&）、或（||）、非（！）

**2、显示过滤器语法和实例**

（1）支持比较操作符：等于（==）、不等于（！=）、大于（>）、小于（<）、大于等于（>=）、小于等于（<=）。

（2）协议过滤：直接在Filter框中输入协议名。

（3）IP过滤：使用ip.src == IP地址、ip.dst== IP地址或ip.addr == IP地址的格式。

（4）端口过滤:使用tcp.srcport == 端口号、tcp.dstport == 端口号或tcp.port == 端口号的格式。

（5） Http模式过滤：如http.request.method"GET"（只显示HTTP GET方法的）。

（6）逻辑运算符为 and/or/not