# Department of Computing

# EE353: Computer Networks

# Class: BESE

**CLO 3:** **Investigate and analyze the behavior of network traffic**

**Lab 2:** **Wireshark – HTTP (Hypertext Transfer Protocol)**

# Date: 20-09-2023

# Time: 10:00 to 1:00 and 2:00 to 5:00

# Instructor: Dr. Humma Gafoor

**Lab Engineer: Syed Muhammad Ali Musa**

**Name:** Aimen Munawar

**CMS id:** 415867

**Class:** BESE-13-A

# Lab 2: Wireshark – HTTP (Hypertext Transfer Protocol)

**Introduction and Objective**

In this lab, we’ll explore several aspects of the HTTP protocol: the basic GET/response interaction, and HTTP message formats.

**Tools/Software Requirements**

Wireshark

**Description**

* Read carefully before starting the lab.
* These exercises are to be done individually.
* You are supposed to provide the answers to the in-line questions in this document and upload the completed document to your course’s LMS site.
* **For all questions, you must not only answer the question, but also supply all necessary information regarding how you arrived at the answer (e.g., use screenshots/ accompanying text, etc.) Use red font color to distinguish your replies from the rest of the text.**
* Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.

**Background:**

The world’s web browsers, servers and related web applications all talk to each other through HTTP, the Hypertext Transfer Protocol. Before proceeding to the experiments, it is recommended that you read introductions to some general terms used in this lab, to avoid any confusion.

1. **What is a web page?**

A Web page (also called a document) consists of objects. An object is a simple file -- such as a HTML file, a JPEG image, a GIF image, a Java applet, an audio clip, etc. -- that is addressable by a single URL. Most Web pages consist of a base HTML file and several referenced objects. For example, if a Web page contains HTML text and five JPEG images, then the Web page has six objects: the base HTML file plus the five images. The base HTML file references the other objects in the page with the objects' URLs. Each URL has two components: the host name of the server that houses the object and the object's path name. For example, the URL www.someSchool.edu/someDepartment/picture.gif has www.someSchool.edu for a host name and /someDepartment/picture.gif for a path name.

1. **What is a web browser?**

A browser is a user agent for the Web; it displays to the user the requested Web page and provides numerous navigational and configuration features. Web browsers also implement the client side of HTTP. Thus, in the context of the Web, we will interchangeably use the words "browser" and "client". Popular Web browsers include Google Chrome, Netscape Communicator, Apple Safari and Microsoft Explorer.

1. **What is a web server?**

A Web server hosts Web objects, each addressable by a URL. Web servers also implement the server side of HTTP. Popular Web servers include Apache, Microsoft Internet Information Server, and the Netscape Enterprise Server.

1. **Introduction to HTTP:**

The Hypertext Transfer Protocol (HTTP), the Web's application-layer protocol, is at the heart of the Web. HTTP is implemented in two programs: a client program and server program. The client program and server programs, executing on different end systems, talk to each other by exchanging HTTP messages. HTTP defines the structure of these messages and how the client and server exchange the messages.  HTTP defines how Web clients (i.e., browsers) request Web pages from servers (i.e., Web servers) and how servers transfer Web pages to clients. When a user requests a Web page (e.g., clicks on a hyperlink), the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response messages that contain the objects.

**Steps for performing this lab:**

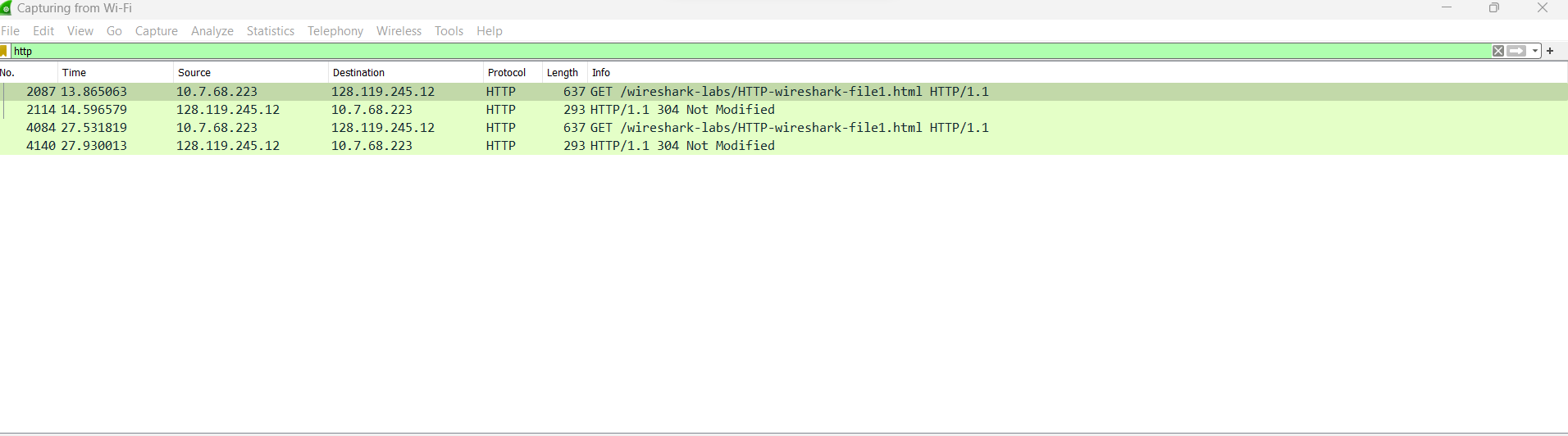
**For all the experiments we will use Wireshark packet analyzer.**

**Exercise 01:** The Basic HTTP GET/response interaction

**Aim of this exercise:** We will now learn about what packets are exchanged during a HTTP conversation---we will learn about the HTTP GET message that is sent from the HTTP client to the HTTP server and the HTTP message that is sent as response to this message.

Follow the steps below to complete this exercise and to provide answers to the questions below

* Start up your web browser.
* Start up the Wireshark packet sniffer (but don’t yet begin packet capture). Enter “http” (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We’re only interested in the HTTP protocol here, and don’t want to see the clutter of all captured packets).
* Begin Wireshark packet capture.
* Enter the following to your browser <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>

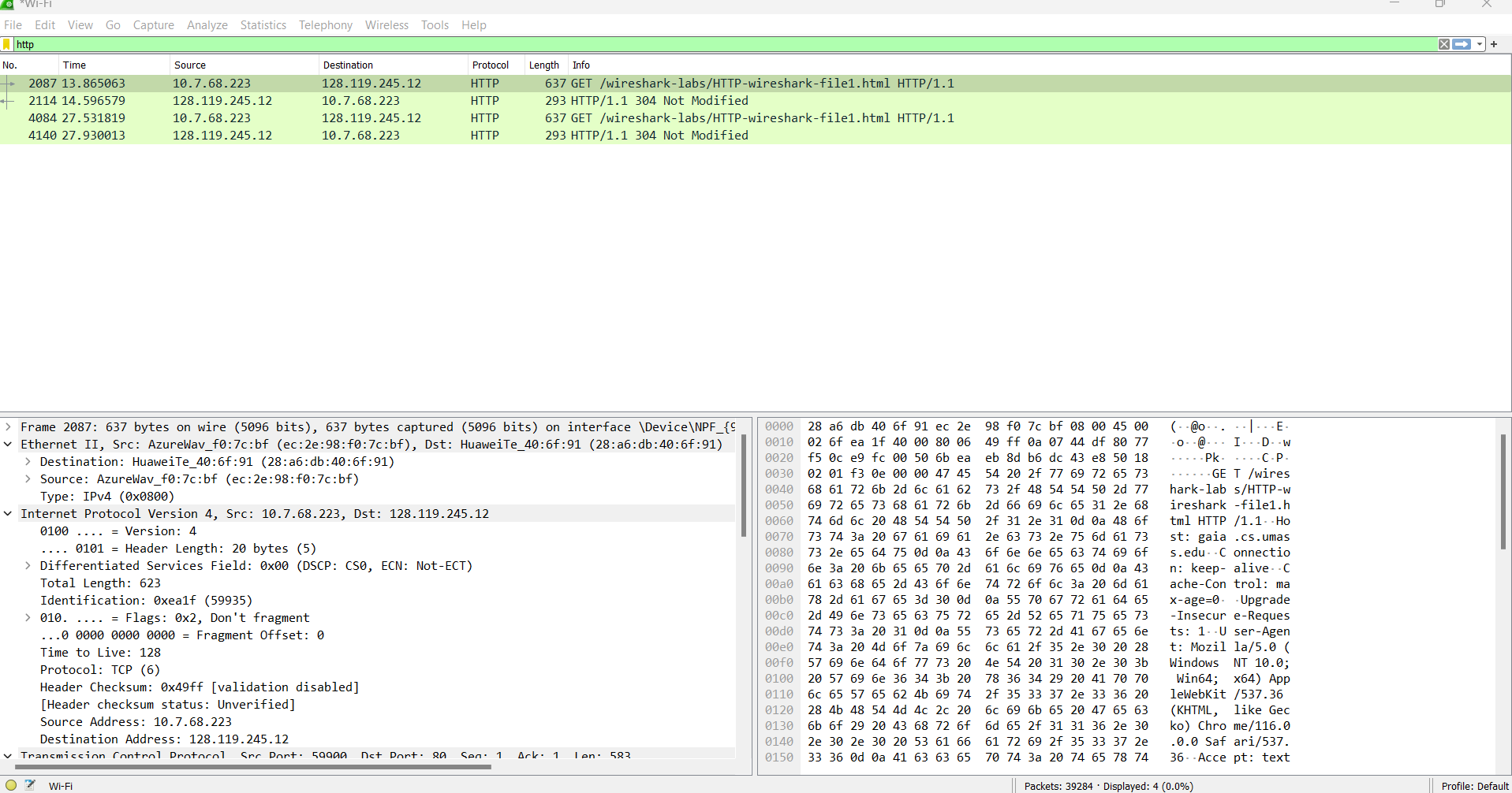


* Your browser should display the very simple, one-line HTML file.
* Stop Wireshark packet capture.

A screenshot of a computer

Description automatically generated

The example in Figure 1 shows in the packet-listing window that two HTTP messages were captured: the GET message (from your browser to the gaia.cs.umass.edu web server) and the response message from the server to your browser. The packet-contents window shows details of the selected message (in this case the HTTP GET message, which is highlighted in the packet- listing window). Recall that since the HTTP message was carried inside a TCP segment, which was carried inside an IP datagram, which was carried within an Ethernet frame, Wireshark displays the Frame, Ethernet, IP, and TCP packet information as well.

****

**Figure 1:** Wireshark display after http://gaia.cs.umass.edu/wireshark-labs/ HTTP-wireshark-file1.html has been retrieved by your browser

By looking at the information in the HTTP GET and response messages that you have captured, answer the following questions:

* 1. **Which version of HTTP is the browser running 1.0 or 1.1? Which HTTP version is the server running?**

Both browser and server are running the 1.1 version.

****

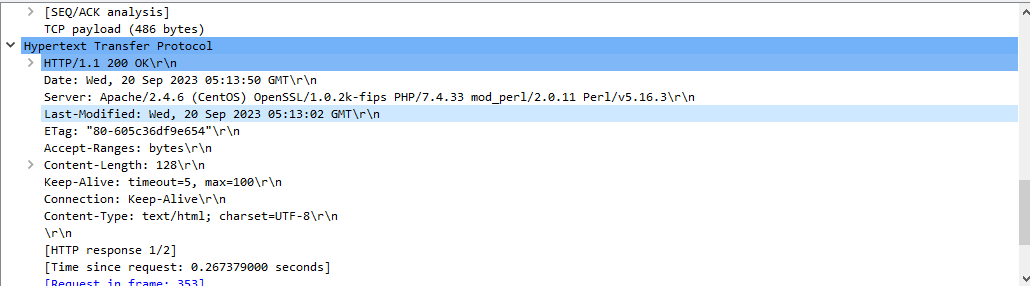
* 1. **What is the status code returned from the server to your browser?**

The status code returned from the server to browser is 200.

****

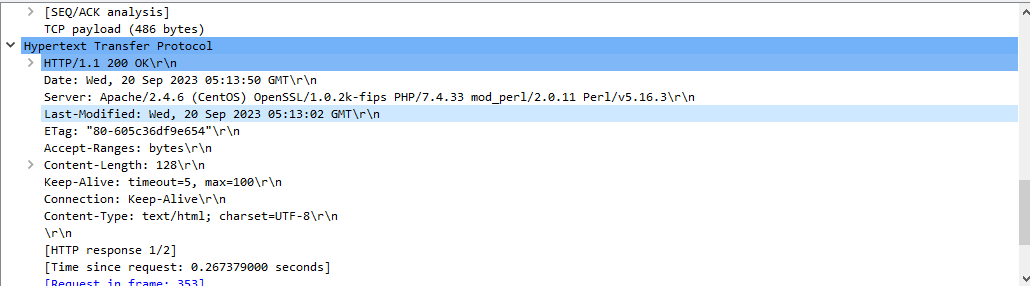
* 1. **When the HTML file that you are retrieving was last modified at the server?**

The HTML file that I am retrieving was last modified on Wed, 20 sept 2023 05:13:02 at the server.

****

* 1. **How many bytes of content are being returned to your browser?**

128 Bytes

****

**Exercise 02:** The HTTP CONDITIONAL GET/response interaction

**Aim of this exercise:** We will now learn about a variant of the HTTP GET request message that we’ve seen earlier. We will note how the HTTP CONDITIONAL GET request and the reply to such a request differs from a simple HTTP GET request. Before performing the steps below, make sure your browser’s cache is empty. (To do this under Firefox, select Tools->Clear Recent History and check the Cache box, or for Internet Explorer, select Tools->Internet Options->Delete File; these actions will remove cached files from your browser’s cache.)

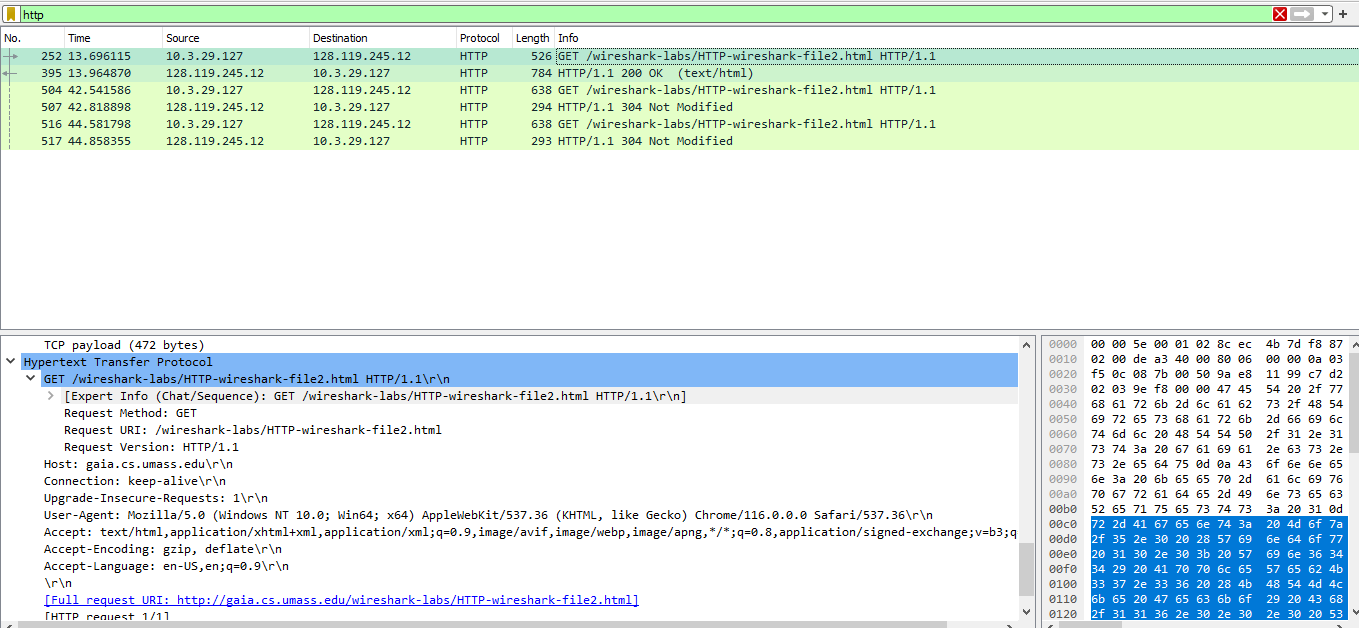
The following indicate the steps for this experiment:

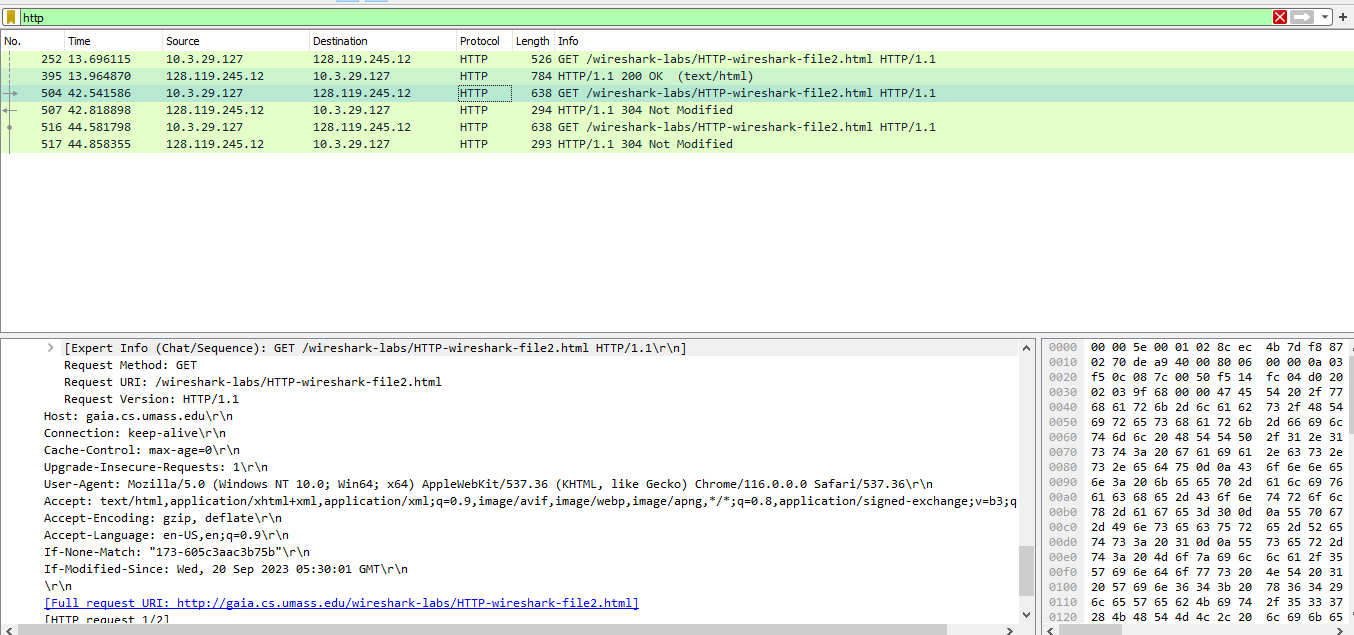
* Start up your web browser, and make sure your browser’s cache is cleared, as discussed above.
* Start up the Wireshark packet sniffer
* Enter the following URL into your browser <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>
* Your browser should display a very simple five-line HTML file.
* Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
* Stop Wireshark packet capture, and enter “http” in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window.
* Filter out all the non-HTTP packets and focus on the HTTP header information in the packet-header detail window.
* By looking at the information in the HTTP GET and response messages, answer the following questions:
  1. **Inspect the contents of the first and 2nd HTTP GET requests from the browser to the server. Do you see “IF-MODIFIED-SINCE” and “IF-NONE-MATCH” lines in these HTTP GET message? Why?**

No, I don’t see “IF-MODIFIED-SINCE” and “IF-NONE-MATCH” lines in the first HTTP GET message, but it is present in the second HTTP GET message.

In the first HTTP GET there was no cached copy in the browser cache, therefore the HTML file was retrieved from the origin server.

In the second HTTP GET the browser realizes that the file has already been downloaded thus the browser sends the If-Modified-Since header back to the server because the If-Modified-Since header indicates the time for which a browser first downloaded a resource from the server .Here the If-None-Match header is used to update the local cache for resources obtained using HTTP GET .The server then evaluated it's copy and the browser's cached copy and determined that there have been no changes since the last download, resulting in a Not Modified status.





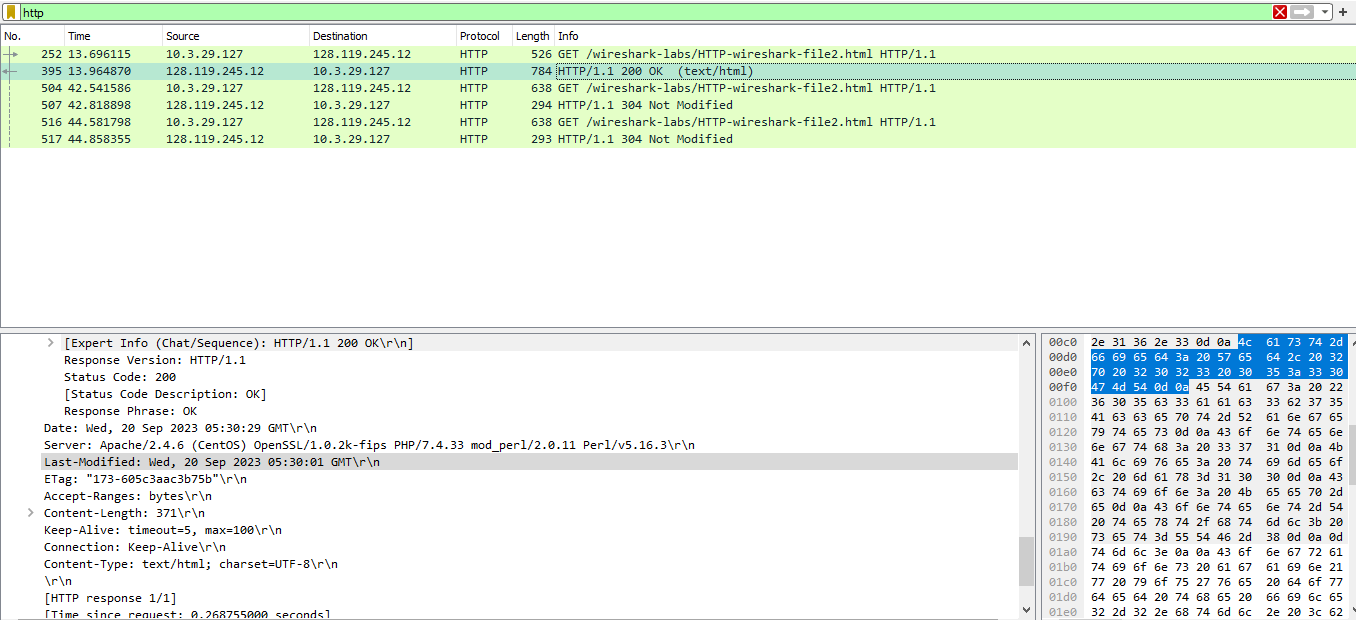
* 1. **What is the difference in first and second response received? What is the last modified time in the first response message?**

The difference between the first and second response received is that in the first response received the last modified message shown is wed, 20 Sep 2023 05:30:01.

And in the second response the last modified message is not shown.

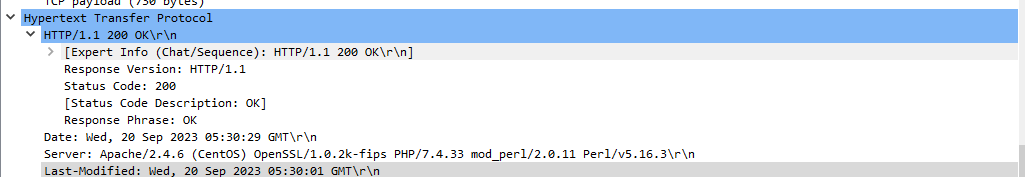
As, in the first HTTP message there was no cached copy in the browser cache, therefore the HTML file was retrieved from the origin server.

In the second HTTP message the browser realizes that the file has already been downloaded thus the browser sends the If-Modified-Since header back to the server. The server then evaluated its copy and the browser's cached copy and determined that there have been no changes since the last download, resulting in a Not Modified status.

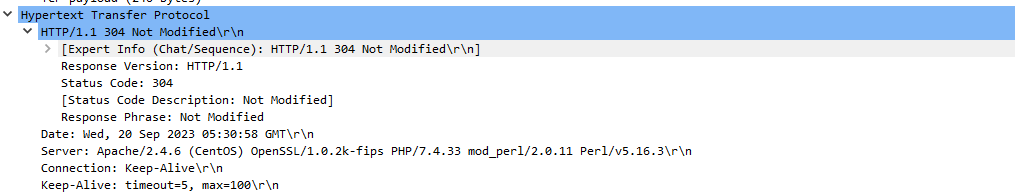
****

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

The status and the phrase returned from the server for first HTML GET is HTTP/1.1 200 OK.



The status and the phrase returned from the server for Second HTML GET is HTTP/1.1 304 Not Modified.

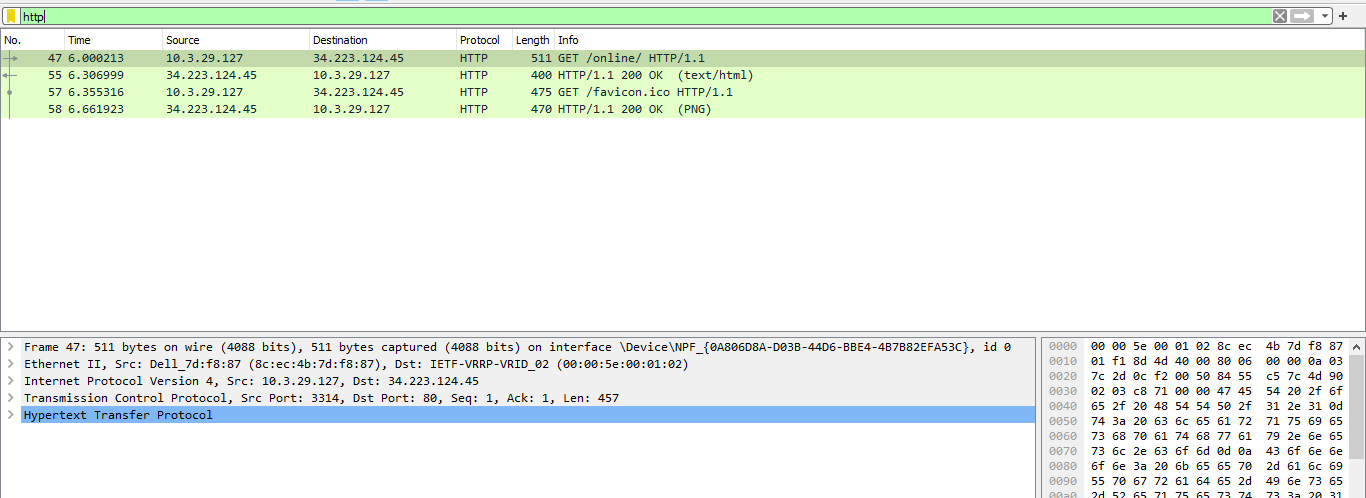


The server did not return the file's contents since the browser just retrieved them from cache. If the file had changed since it was last viewed, it would have provided its contents; instead, it instructed browser to get the previous version of the file from its cached memory.

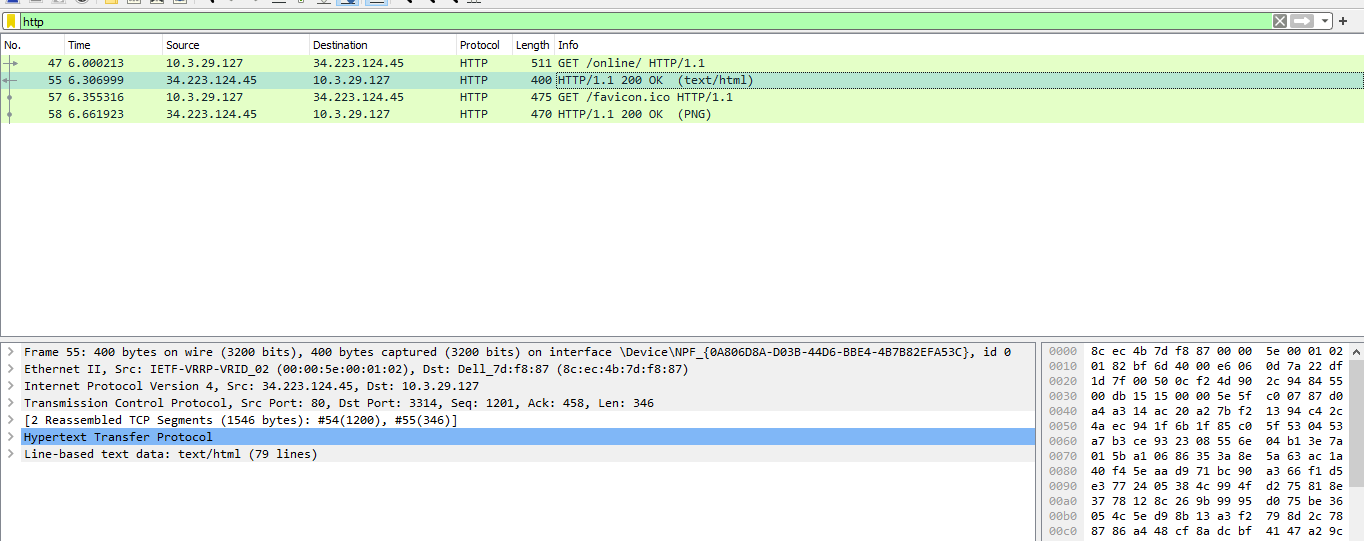
* 1. **Empty your browser cache again and open the webpage** https://seecs.nust.edu.pk/ **and capture the GET and OK response messages. How many total objects does the server return?**

**Web page used:** <http://clearquietfreshpathway.neverssl.com/online/>

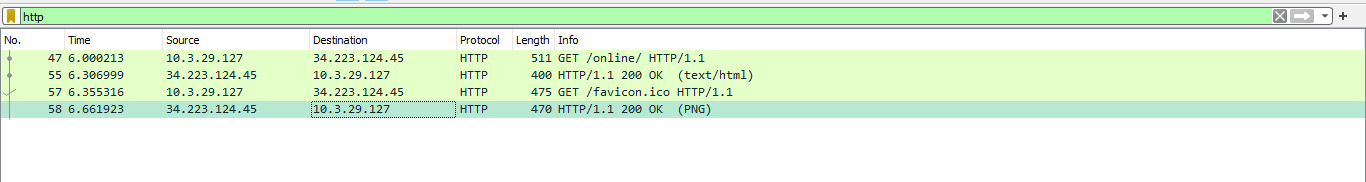
**GET response:**

****

**OK response:**

****

The server returned two object .One object is a text/html object and the other object is an image (PNG).



* 1. **What is the page load time (PLT) for the interaction in 2.4?**

The page load time for the interaction is 0.66171 s.

