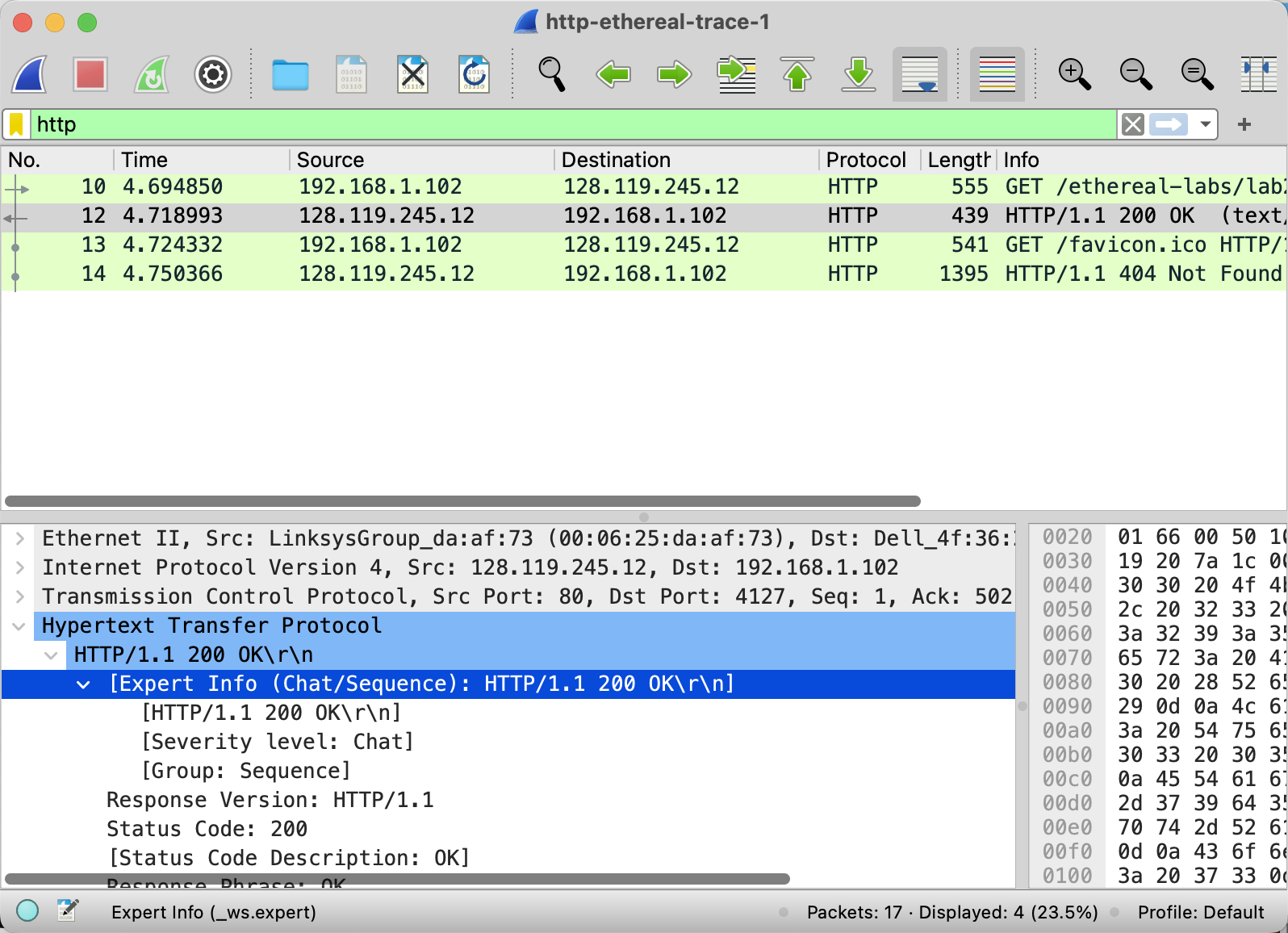
Lab 02

Exercise 3:



Status code: HTTP/1.1 **200**

Phrase: OK



Status code: HTTP/1.1 **404**

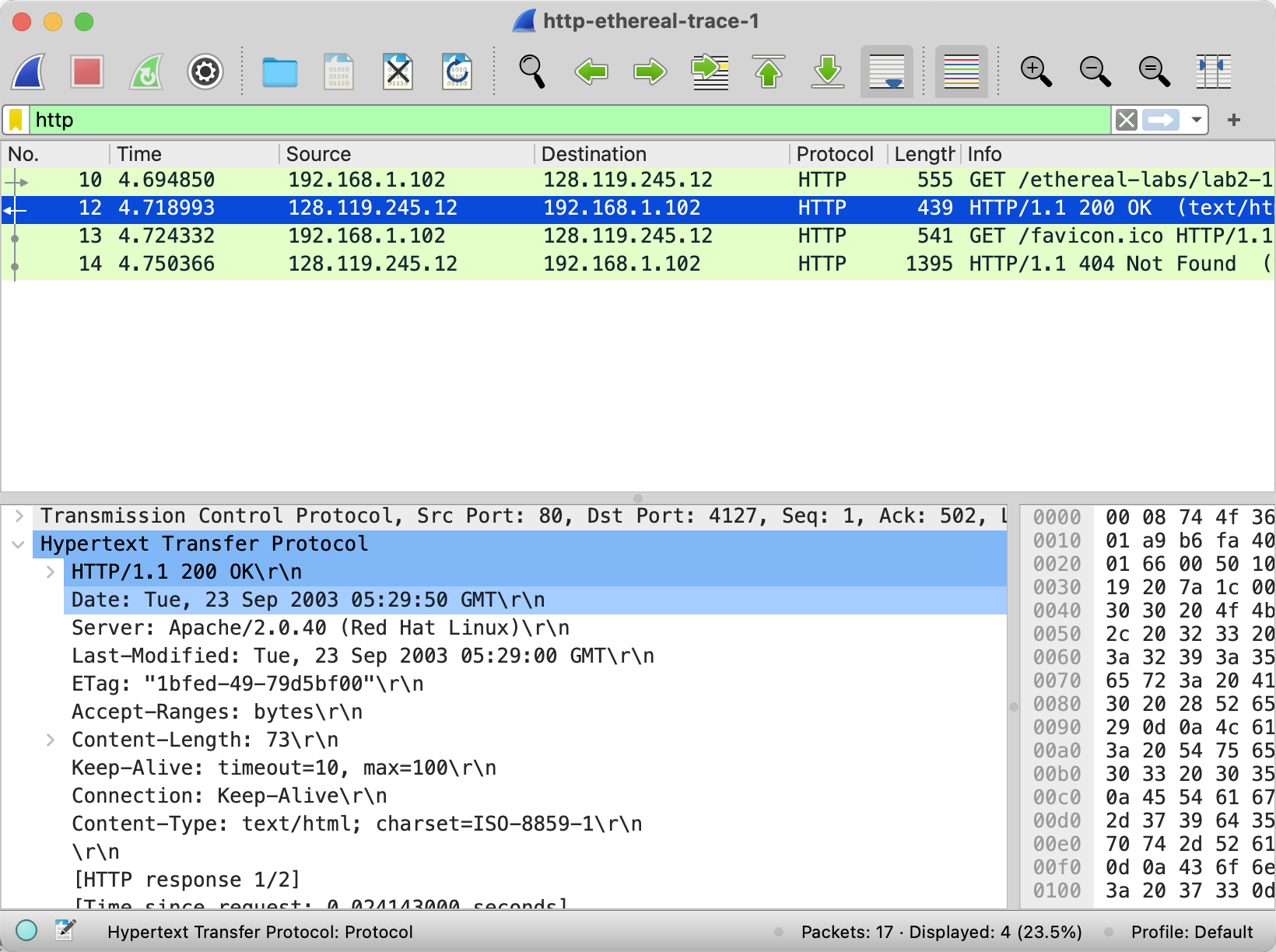
Phrase: not found



1. Question 2: When was the HTML file the browser retrieves last modified at the server? Does the response also contain a DATE header? How are these two fields different?

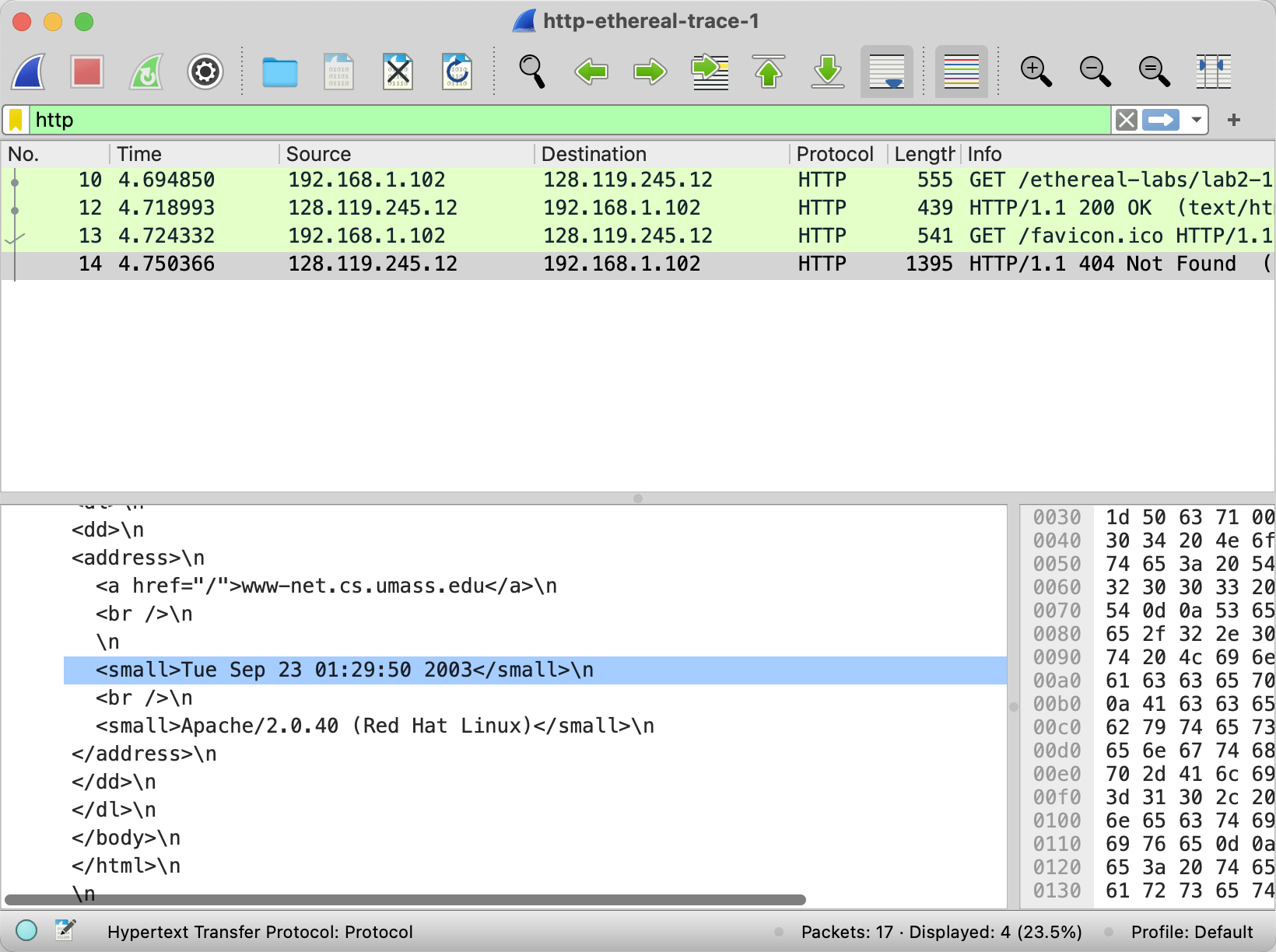
When server response to the client, last-Modified: Tue ,23 Sep 2003 05:29:50

Yes it is also contain the date header, last-Modified which mean last change time for resource, DATE header represent the time of server repones generation .

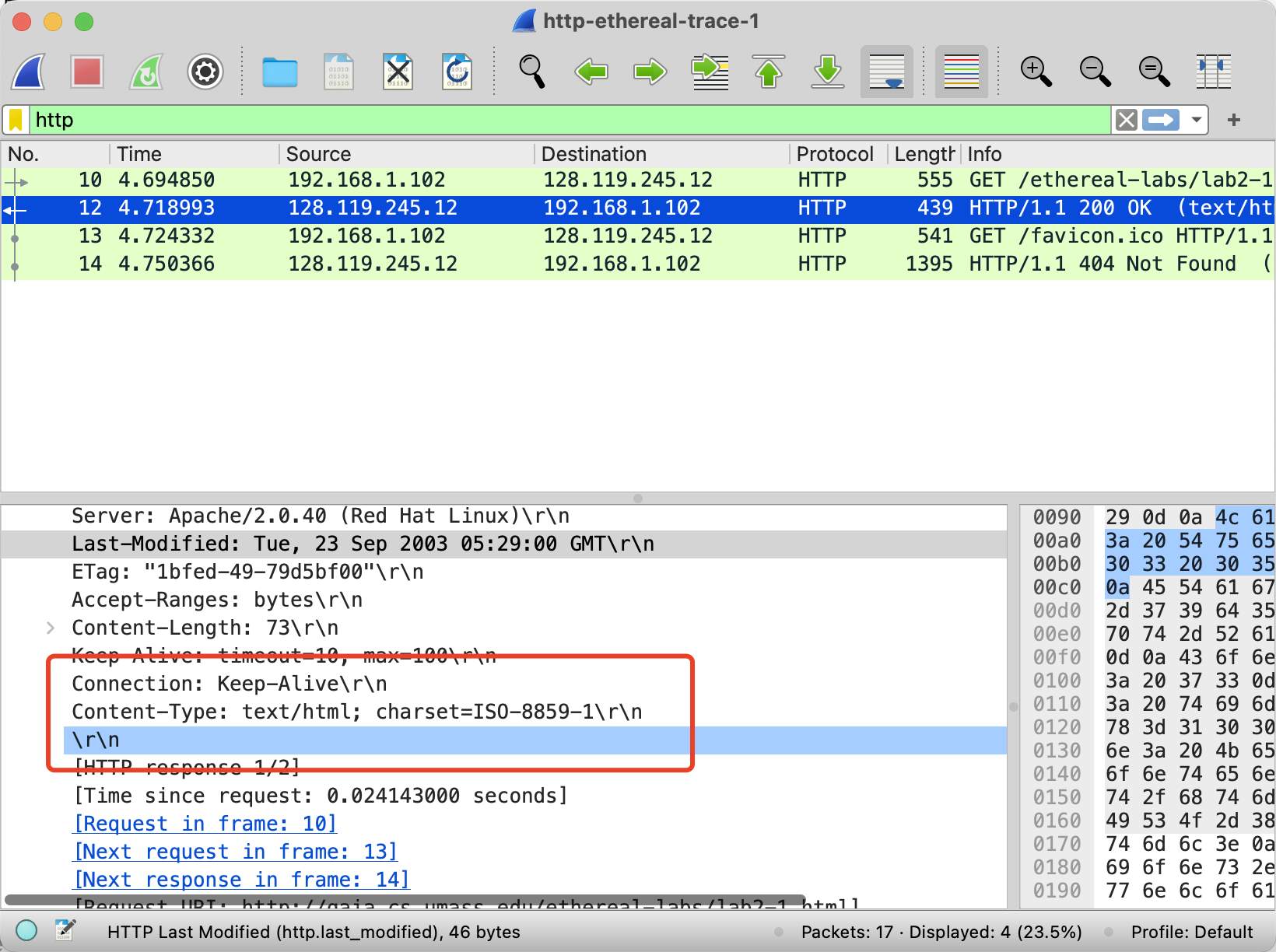


A screenshot of a computer

Description automatically generated



Connection shows that “keep -Alive \r\n”, and the protocol is Http/1.1 which means persistent.

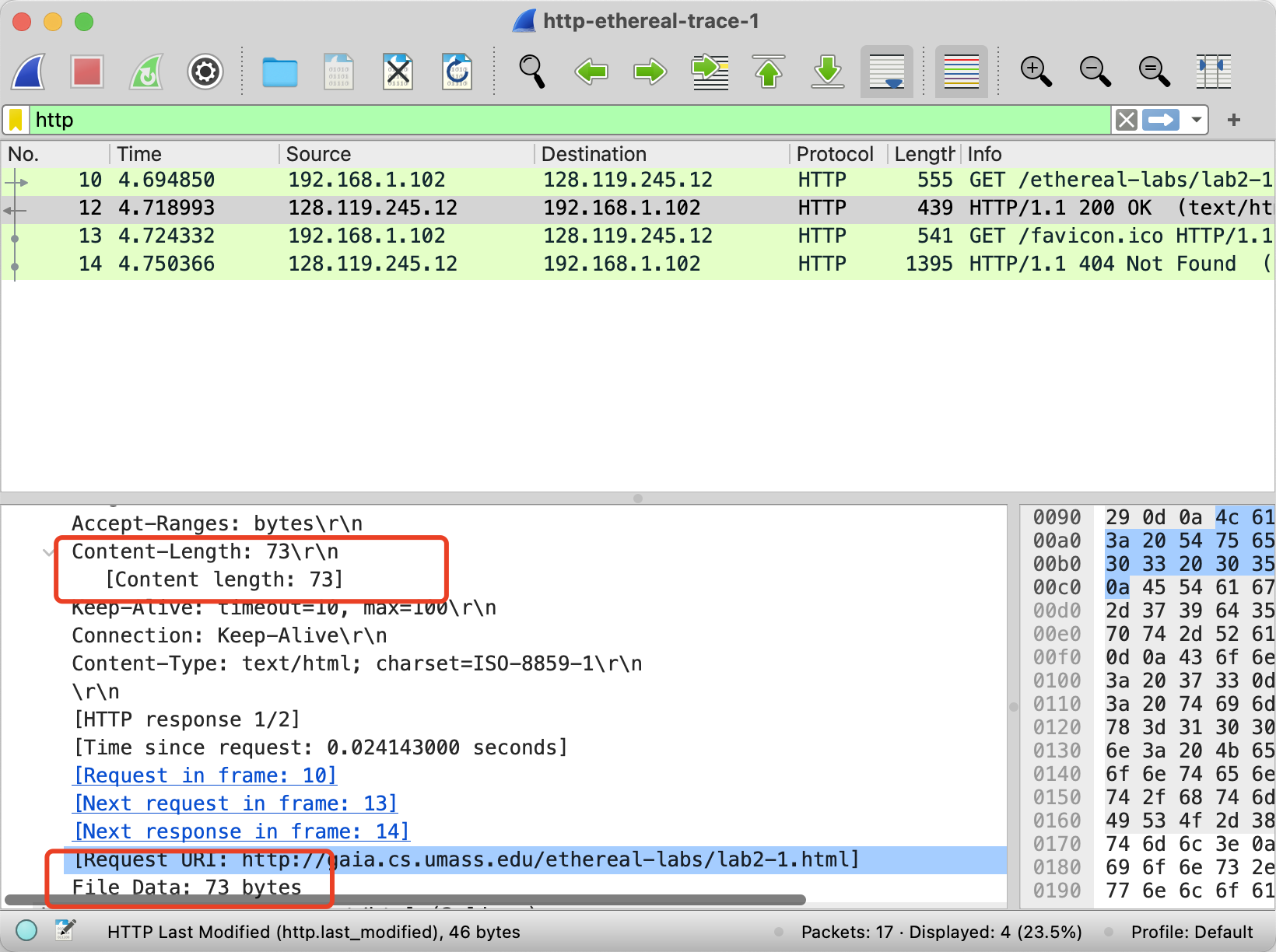


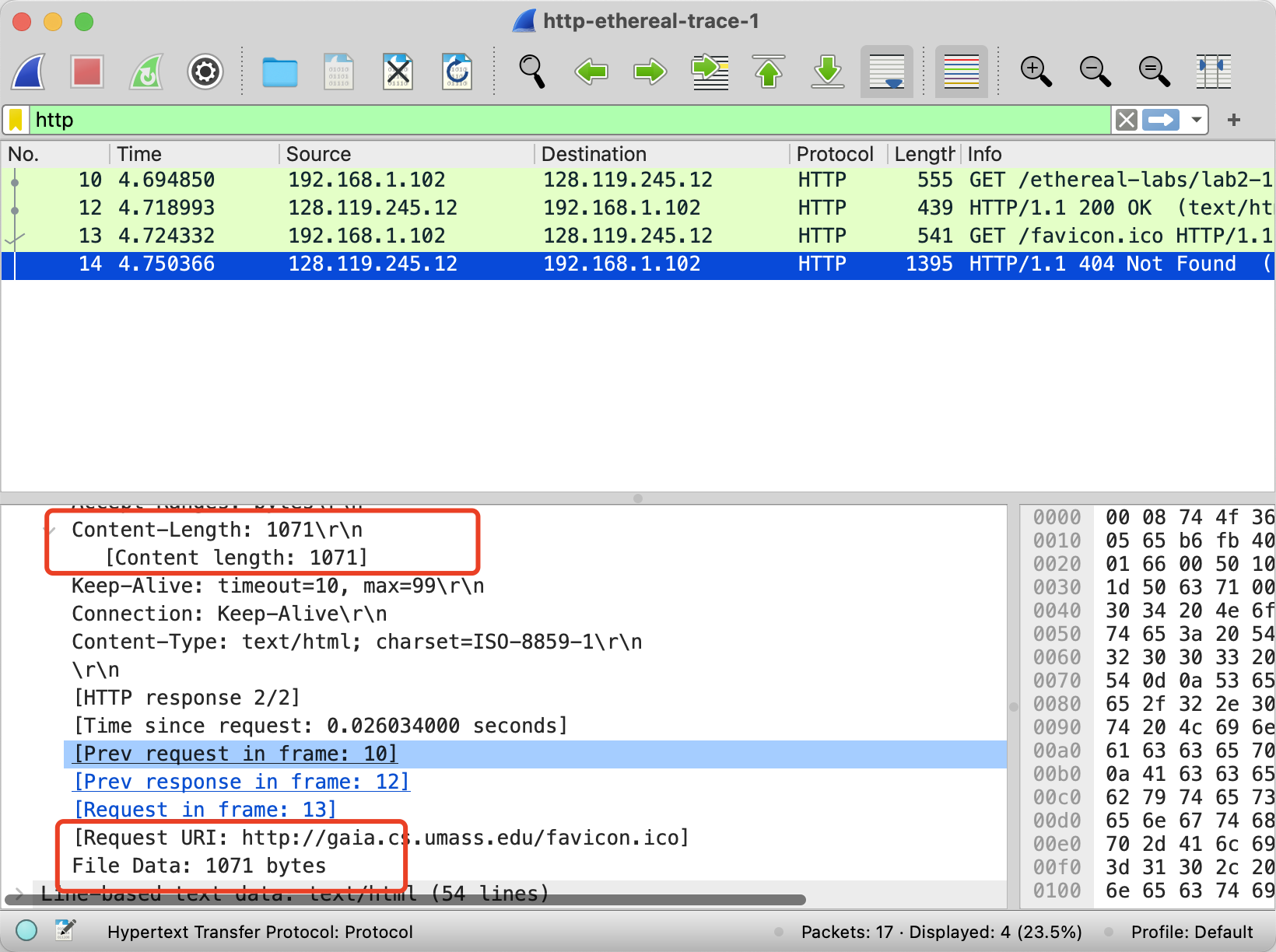


(4)

For the first response, content is 73 bytes.

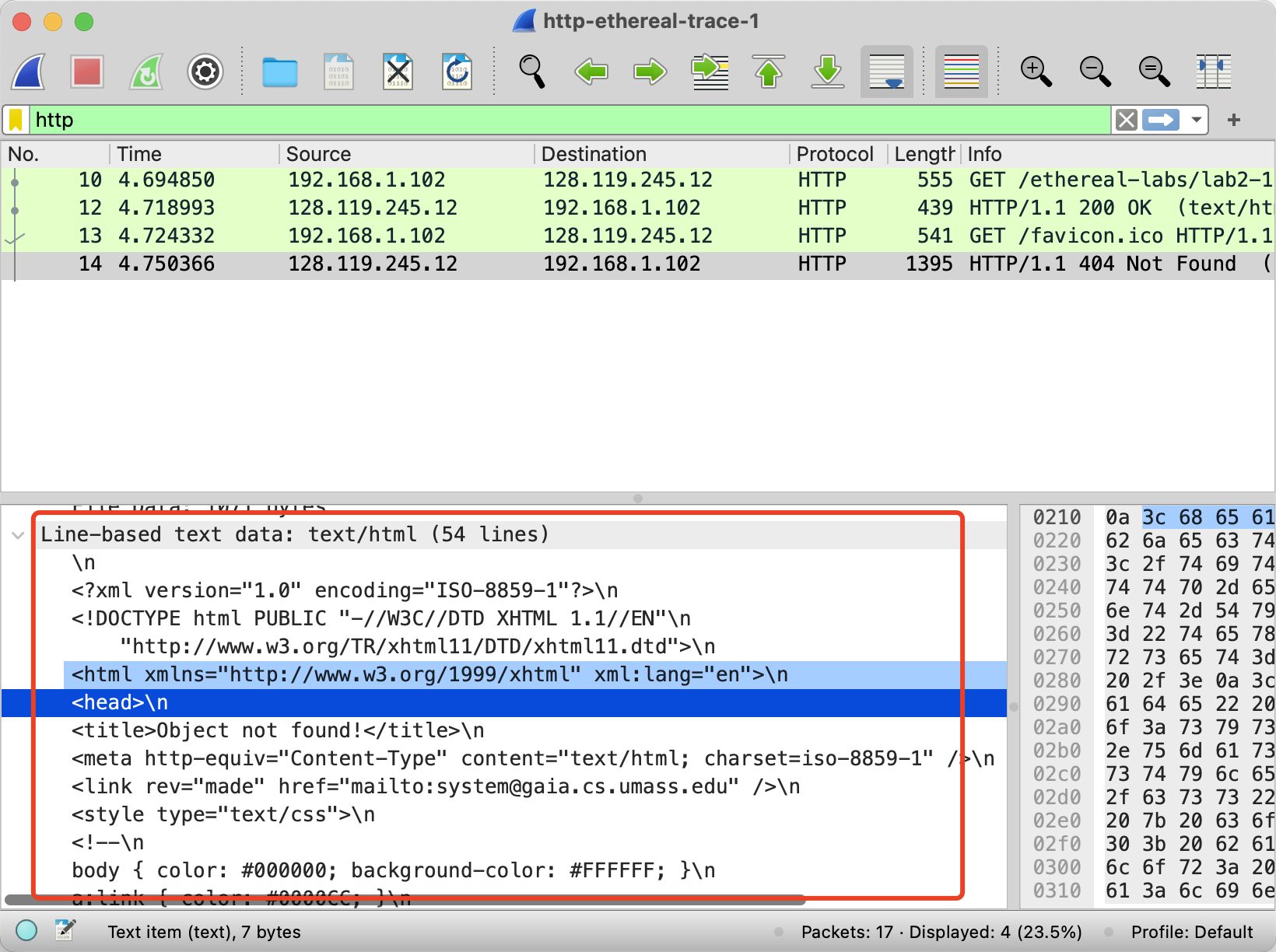
For the second response , content-length is 1071 bytes.





(5)

That is the html interface .

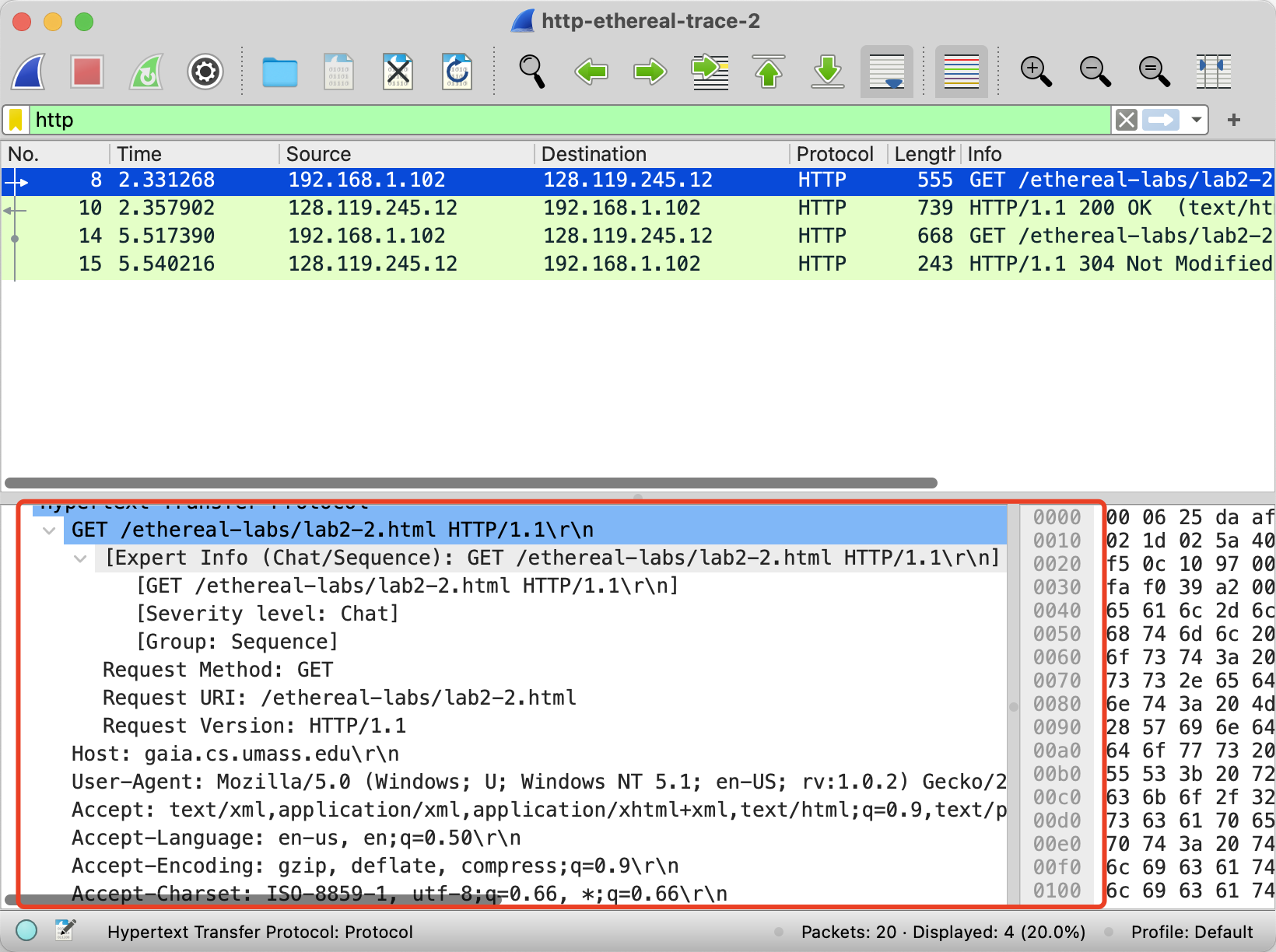




**Exercise 4:**

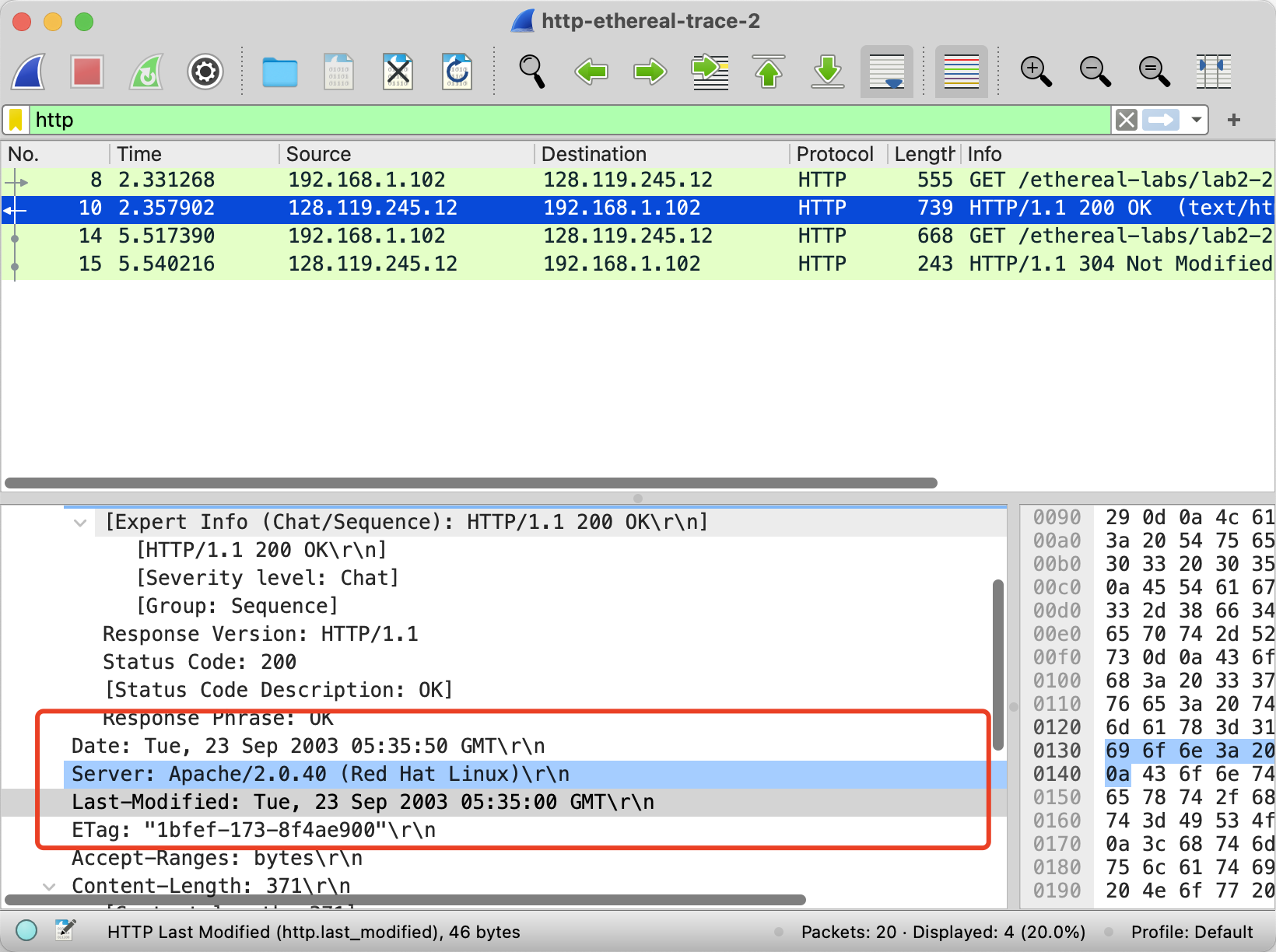
(1)

For the first http get , there is not information about “IF-MODIFIED-SINCE” line

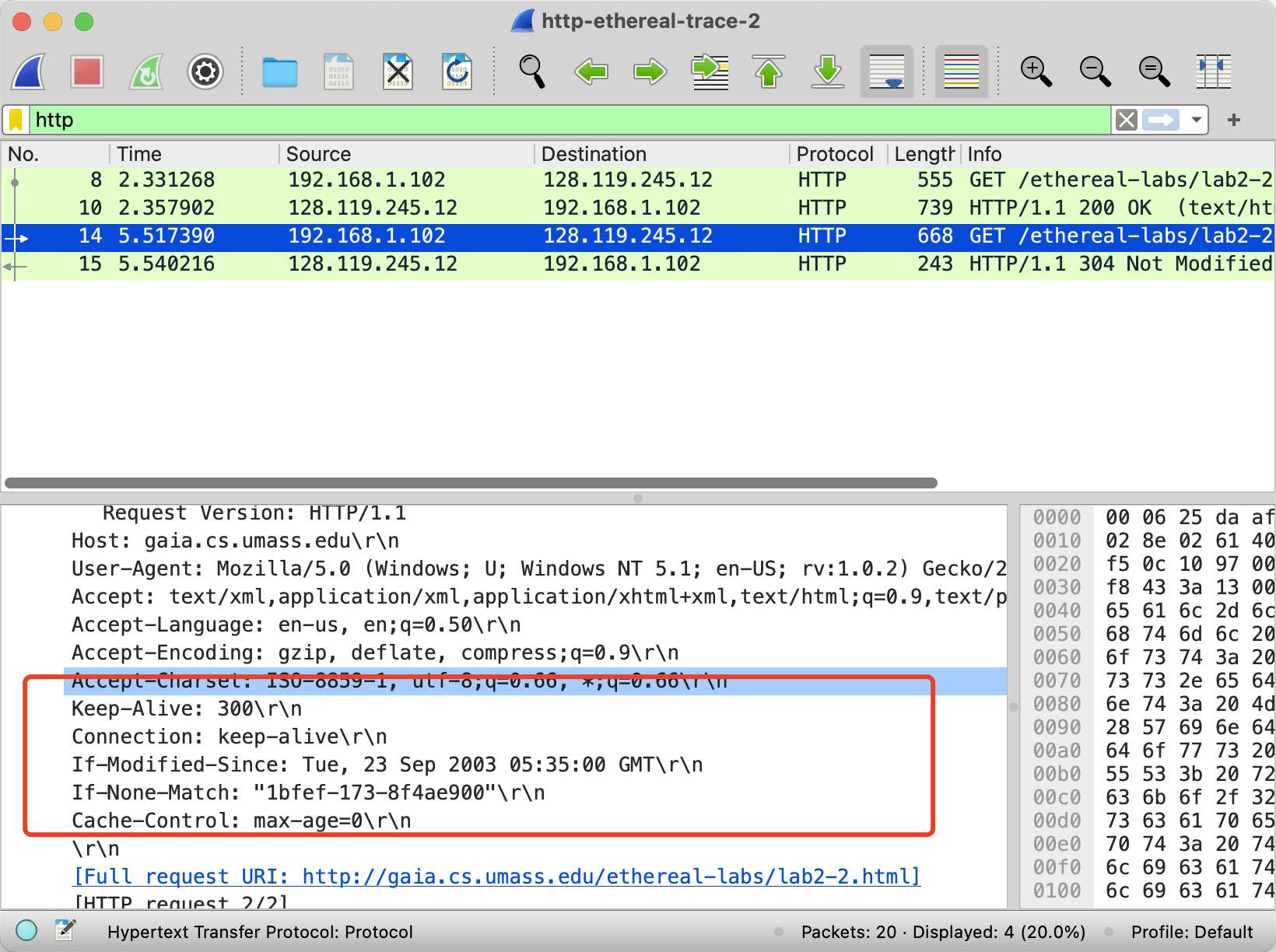


(2)

Yes it is showing following:



(3)



If-modified-Since: Specifies only resources created after 2003.09.23.

If-None-Match: Compare the eTag values of the client and server, and if they match, the server returns a 304-status code. If there is no match, the server returns the entire resource.

(4)

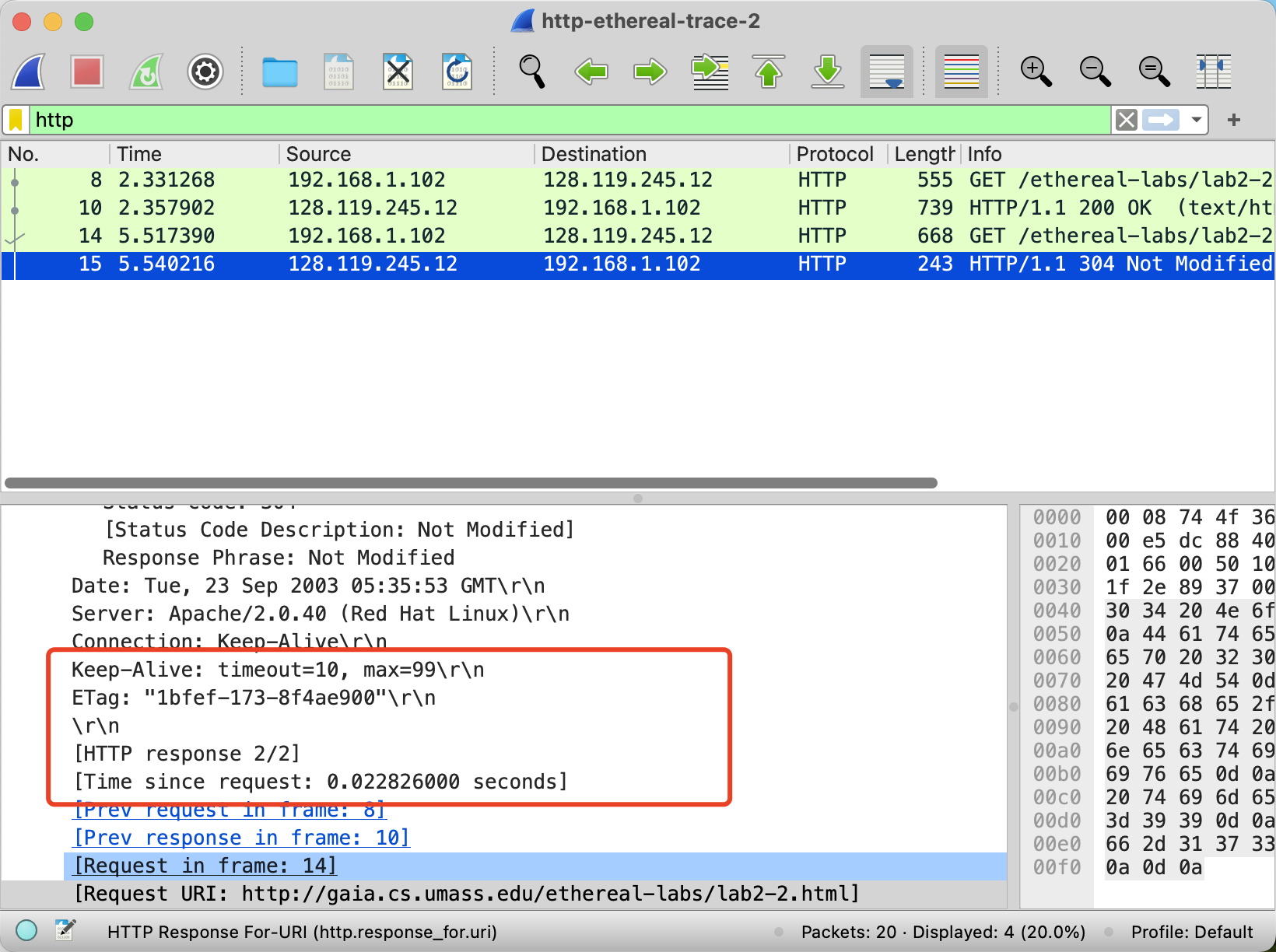


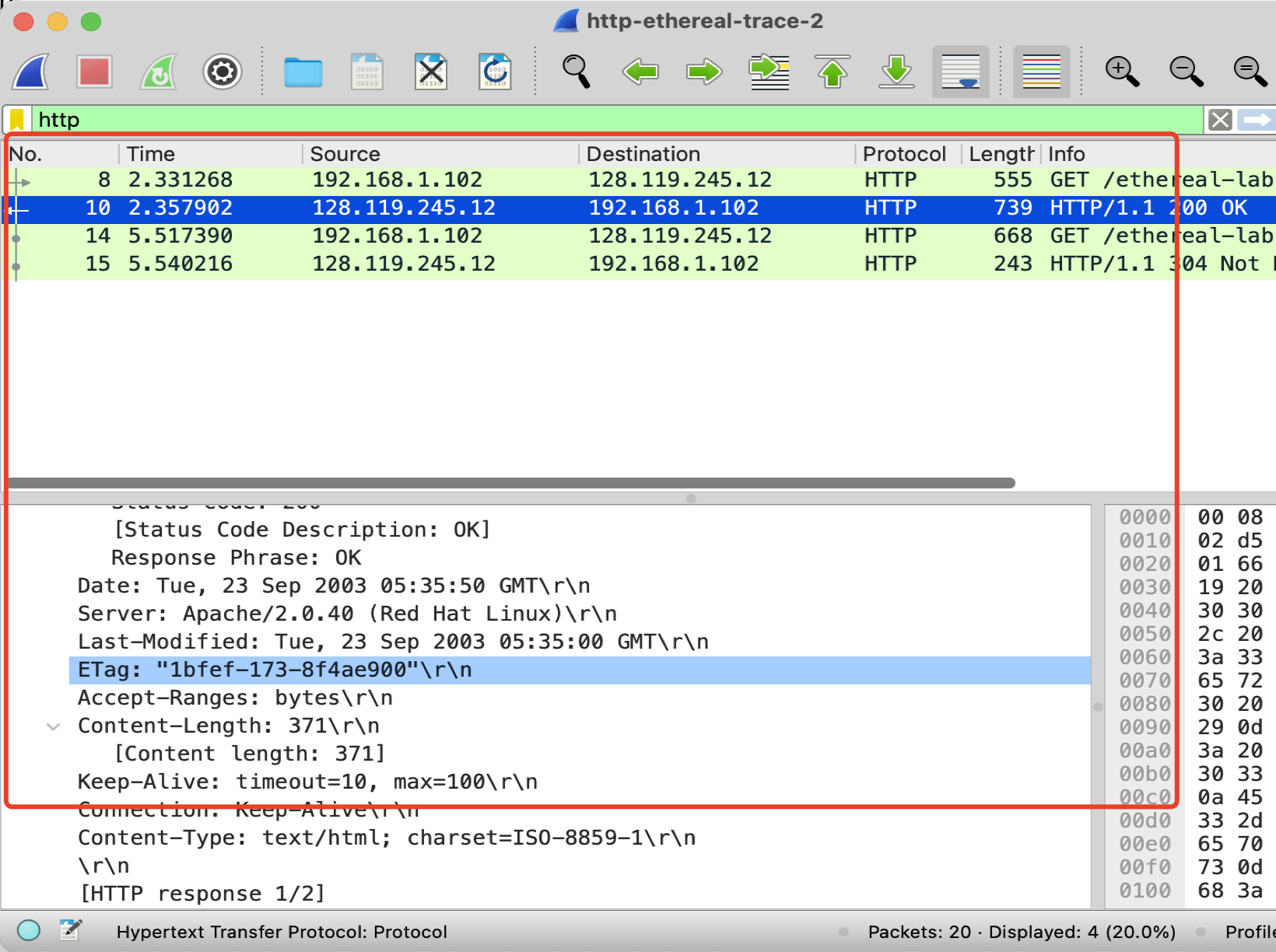
No, it didn’t. Because the server judge the content is the latest one ,it won’t resent the content again.

(5)

After the first response, eTag keep in client , next GET request for comparing with server.

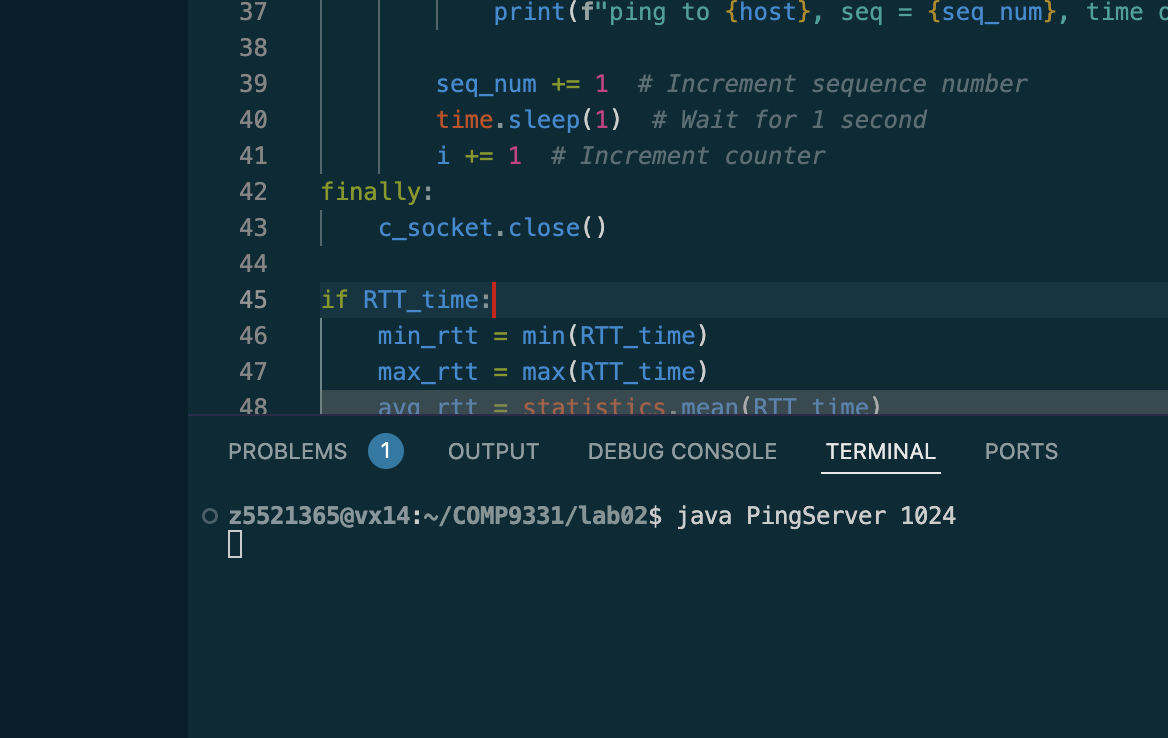
The Compare with two eTag, they have same. So the last response will reckon that the content is the latest. (no need to update) Not modified which mean eTag is same.

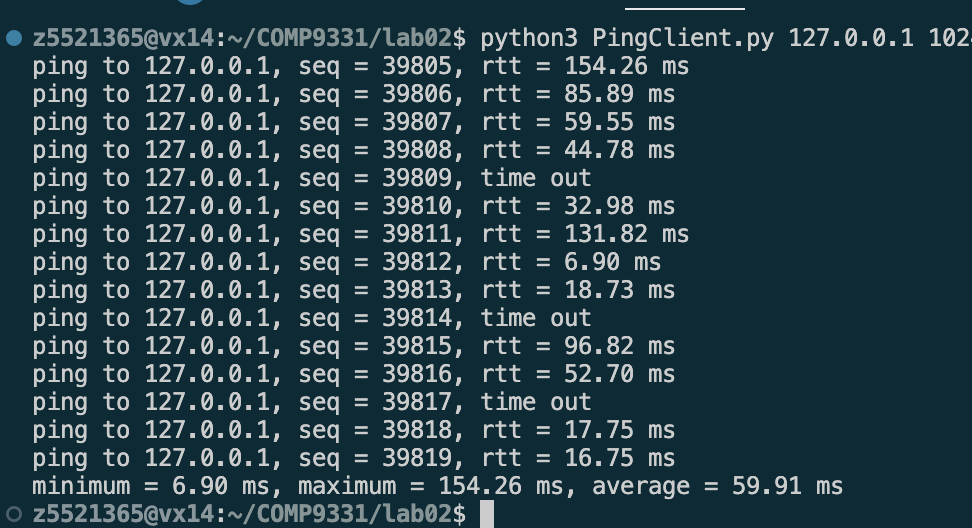


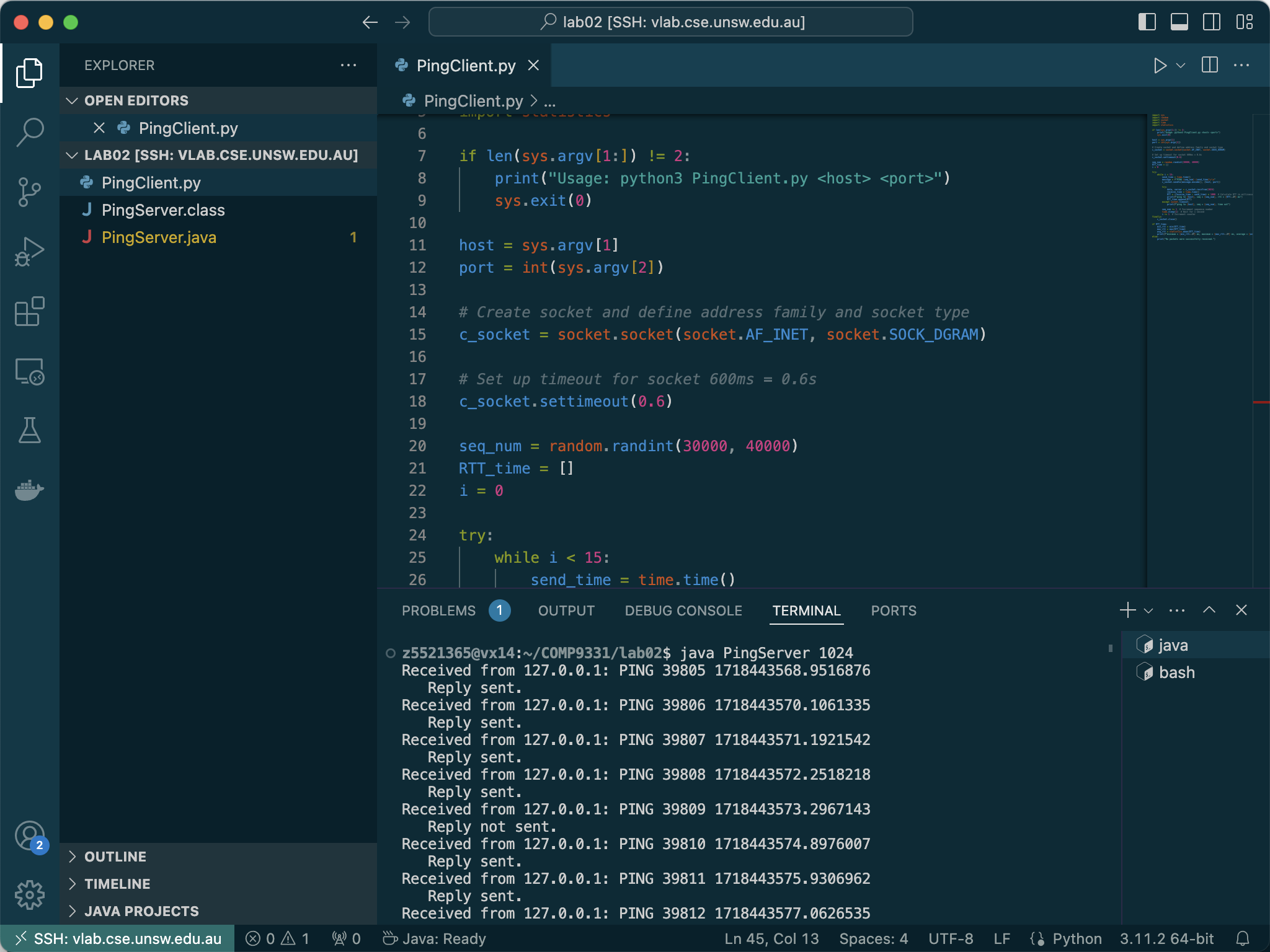


**Exercise 5:**

**Open the port**

****

****

****

**Python code for Ping Client**

import sys

import random

import socket

import time

import statistics

if len(sys.argv[1:]) != 2:

print("Usage: python3 PingClient.py <host> <port>")

sys.exit(0)

host = sys.argv[1]

port = int(sys.argv[2])

*# Create socket and define address family and socket type*

c\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

*# Set up timeout for socket 600ms = 0.6s*

c\_socket.settimeout(0.6)

seq\_num = random.randint(30000, 40000)

RTT\_time = []

i = 0

try:

while i < 15:

send\_time = time.time()

message = **f**"PING {seq\_num} {send\_time}\r\n"

c\_socket.sendto(message.encode(), (host, port))

try:

data, server = c\_socket.recvfrom(1024)

receive\_time = time.time()

RTT = (receive\_time - send\_time) \* 1000 *# Calculate RTT in milliseconds*

print(**f**"ping to {host}, seq = {seq\_num}, rtt = {RTT**:.2f**} ms")

RTT\_time.append(RTT)

except socket.timeout:

print(**f**"ping to {host}, seq = {seq\_num}, time out")

seq\_num += 1 *# Increment sequence number*

time.sleep(1) *# Wait for 1 second*

i += 1 *# Increment counter*

finally:

c\_socket.close()

if RTT\_time:

min\_rtt = min(RTT\_time)

max\_rtt = max(RTT\_time)

avg\_rtt = statistics.mean(RTT\_time)

print(**f**"minimum = {min\_rtt**:.2f**} ms, maximum = {max\_rtt**:.2f**} ms, average = {avg\_rtt**:.2f**} ms")

else:

print("No packets were successfully received.")