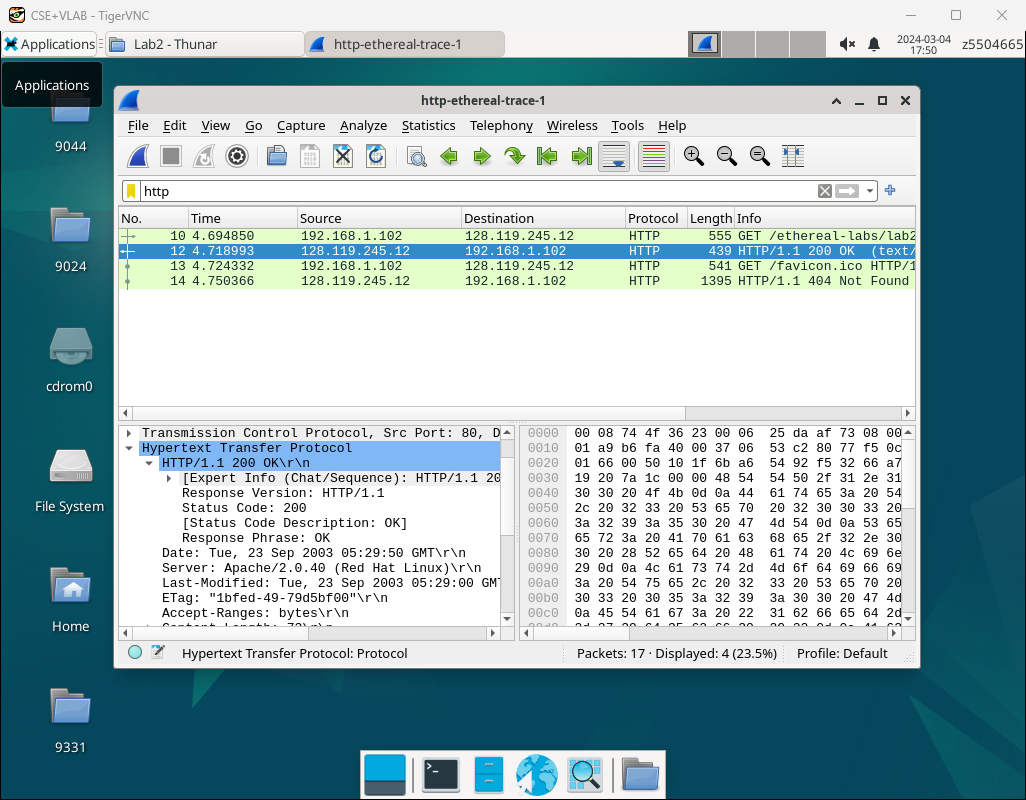
**Lab2**

Exercise 3

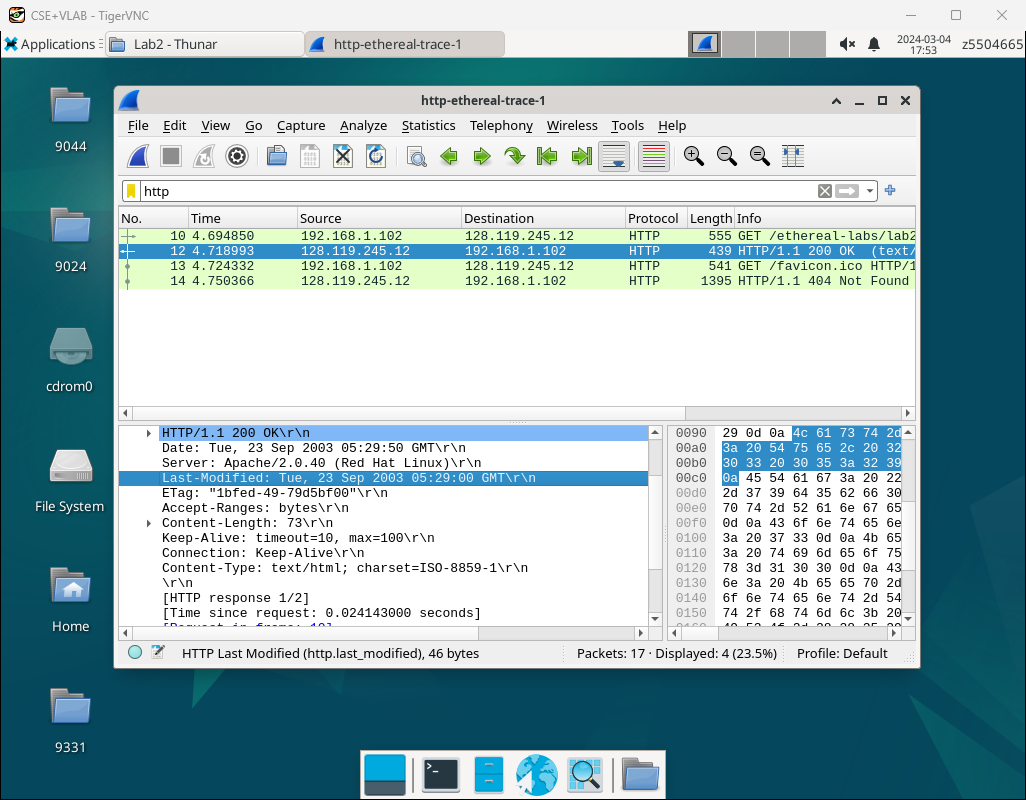
Q1:



state code:200

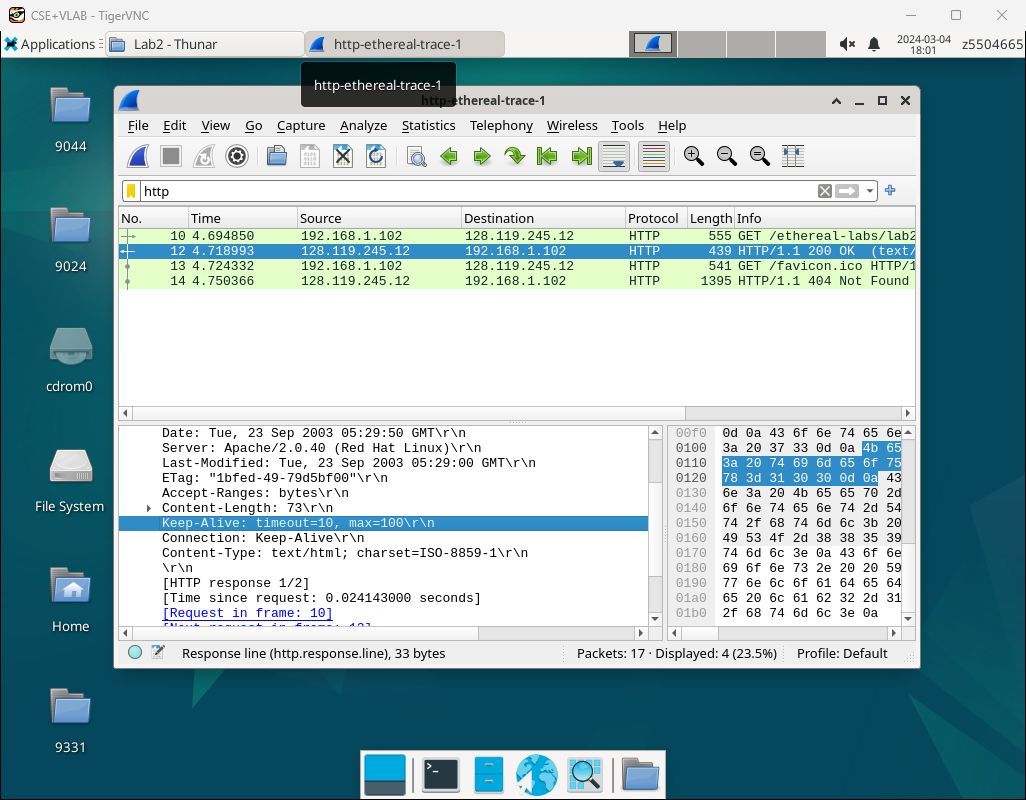
phrase:OK

Q2:



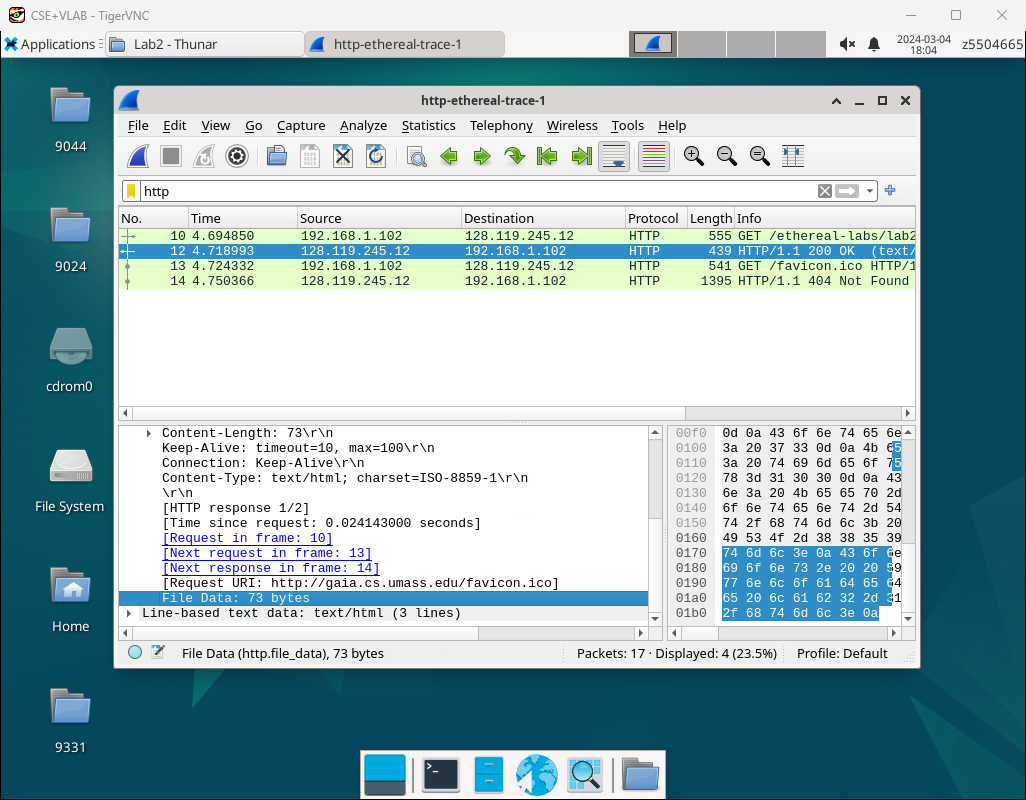
1. Last-modified time is Tue, 23 Sep 2003 05:29:00 GMT.
2. Yes,it does.
3. Date is the time when package was transported, and the last-modified is the last time when the package was changed.

Q3:



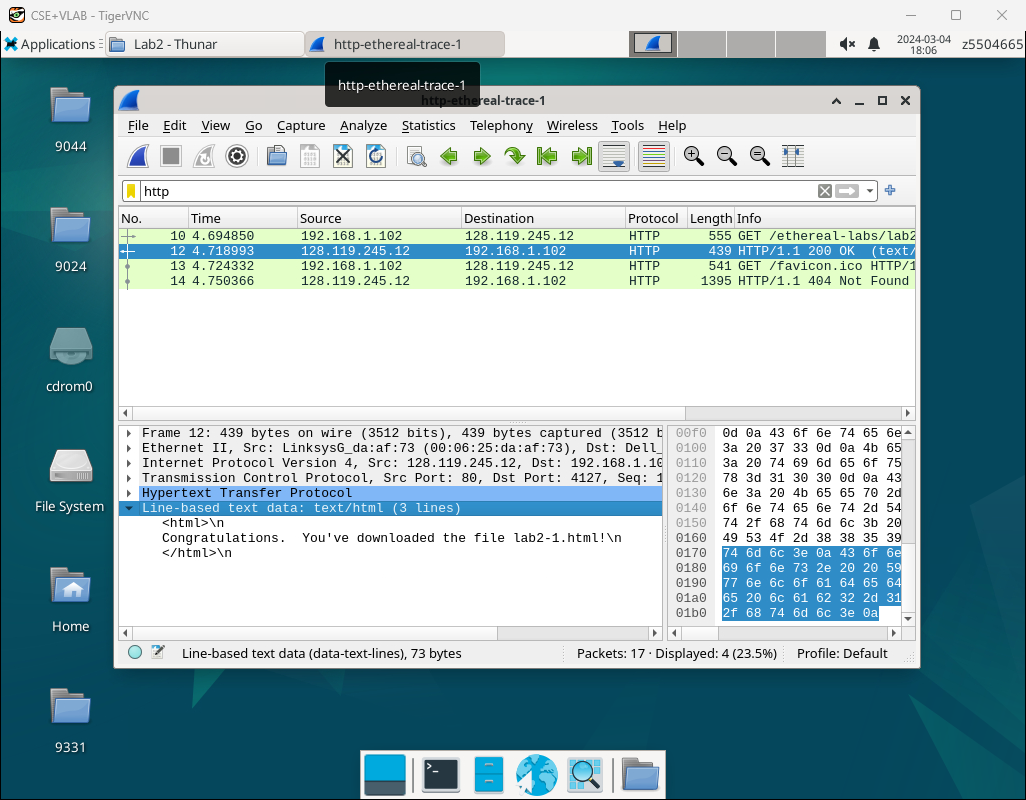
It is persistent, because we can see that there is a Keep-Alive in connection line,which means this connection is persistent.

Q4:



73 bytes

Q5：



Data:

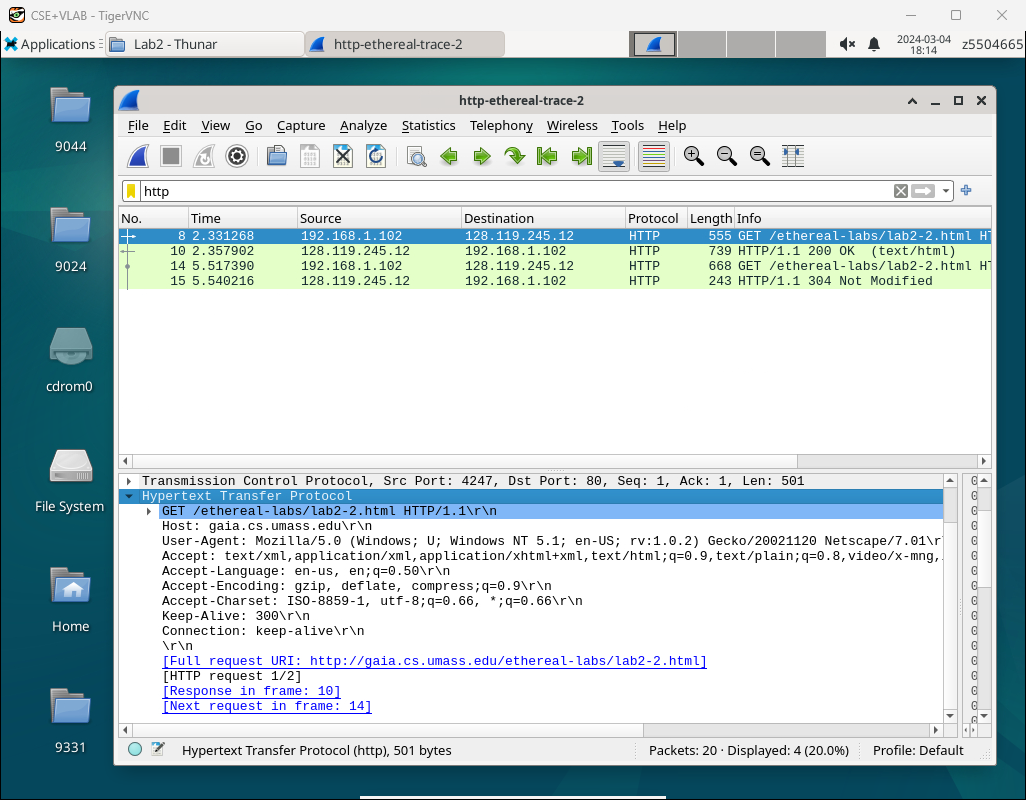
<html>\n

Congratulations. You've downloaded the file lab2-1.html!\n

<html>\n

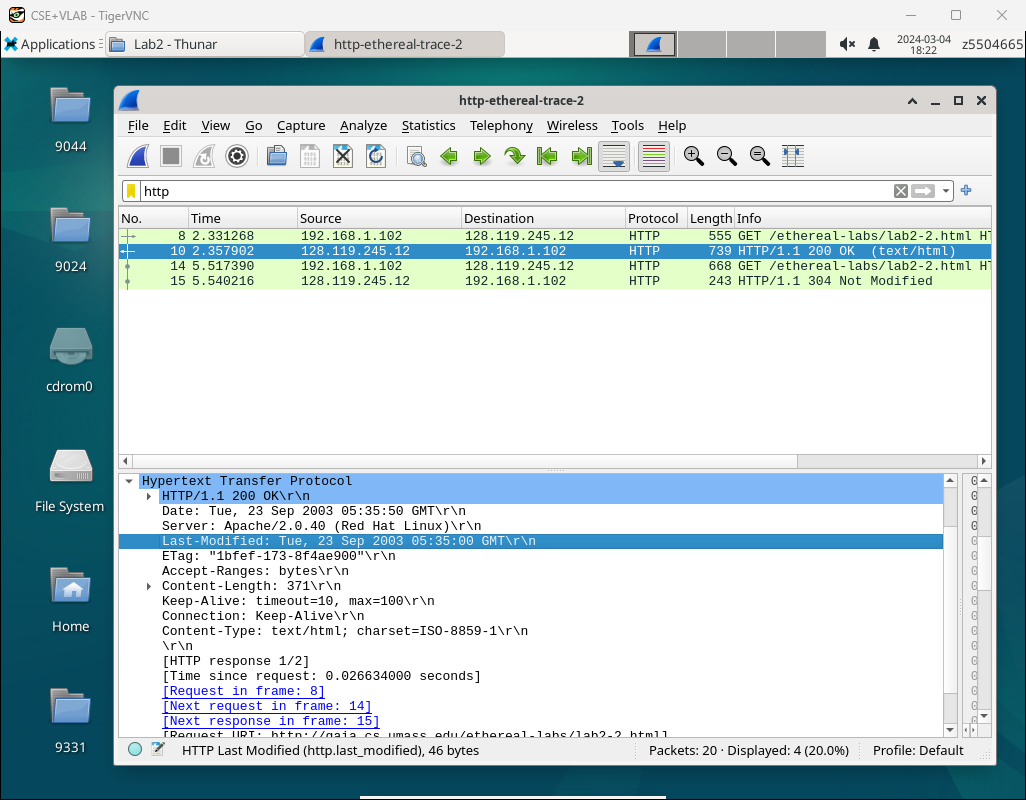
Exercise 4

Q1：



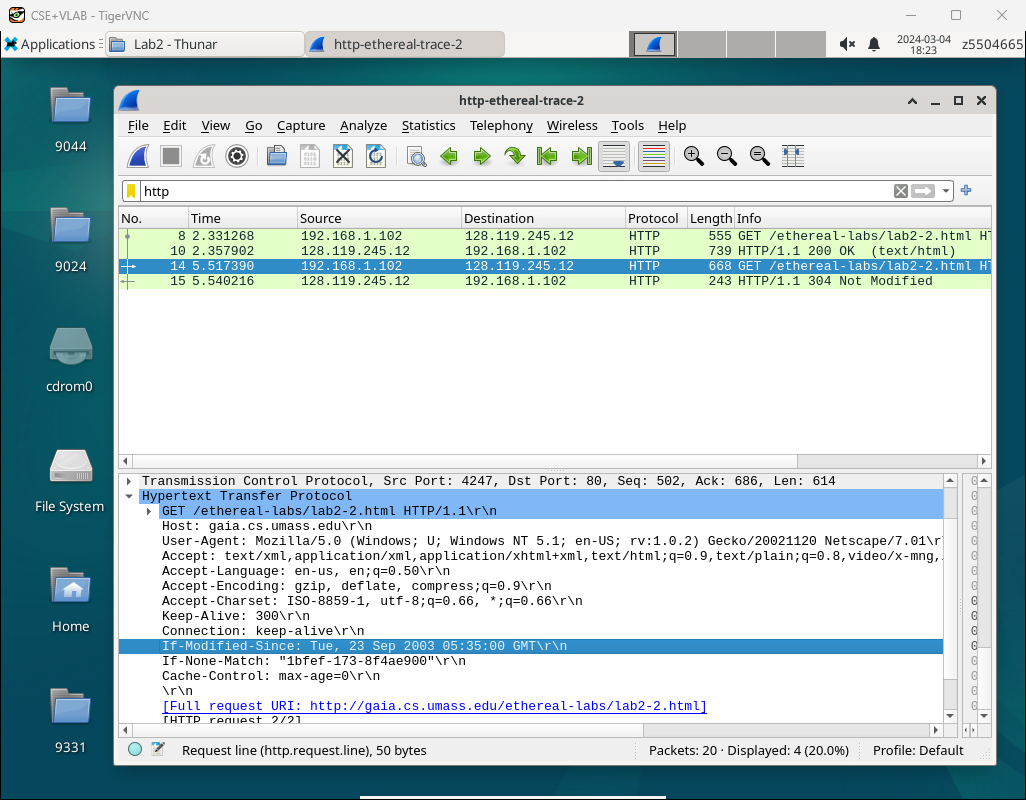
There is not “IF-MODIFIED-SINCE” line in the HTTP GET.

Q2：



Yes it does,and the time is Tue, 23 Sep 2003 05:35:00 GMT.

Q3：



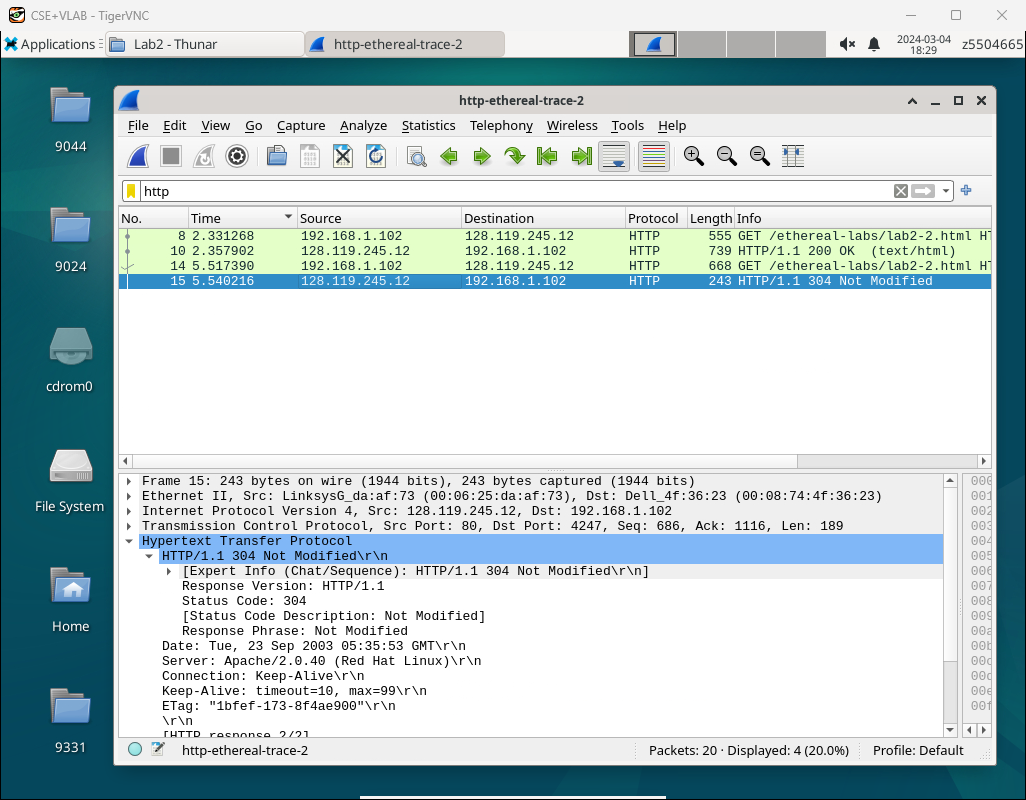
1)Yes,I see these two line.

2)And the information they contained are:

If-Modified-Since: Tue, 23 Sep 2003 05:35:00 GMT

If-None-Match: "1bfef-173-8f4ae900"

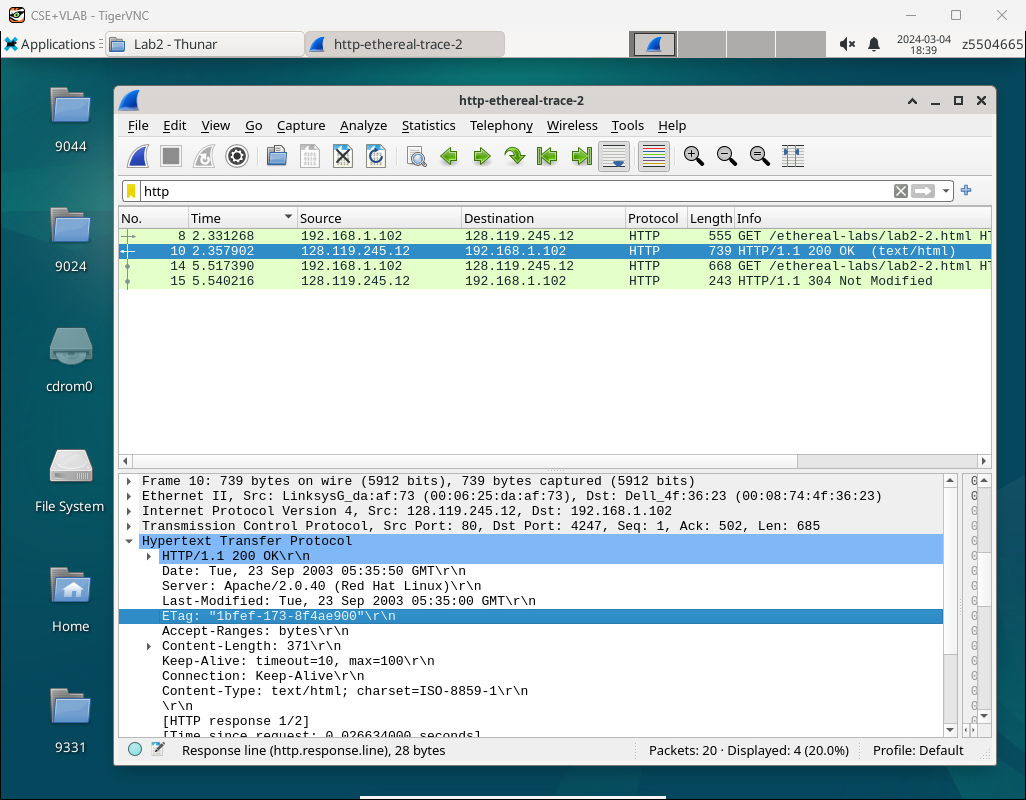
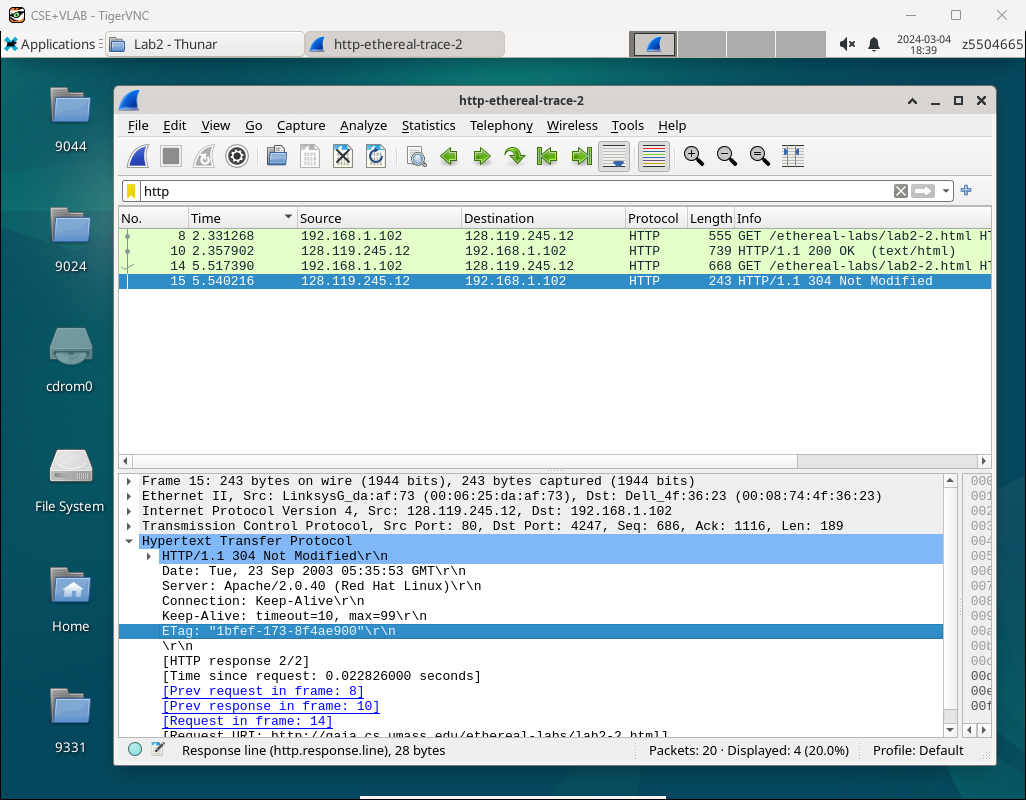
Q4：



1)state code: 300 phrase: Not Modified

2)No, it did not. Because the content the client second request has not modified after first time, so there is no need to return the file’s content, it is same as the first one the client gained.

Q5：



1. ETag: "1bfef-173-8f4ae900" By comparing the two etag information, you can determine whether the file content of the two requests has changed and decide whether to send the file content again
2. They are the same.

Exercise 5

Code:

import socket

import time

import datetime

import random

import sys

def main():

args = sys.argv

all\_rtt=[]

ip=args[1]

port=int(args[2])

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

udp\_socket.settimeout(0.6)

n=random.randint(50000, 60000)

for i in range(20):

date=datetime.datetime.now()

try:

sp=time.time()

text="PING "+ip+" "+str(n)+" "+date.strftime("%Y-%m-%d %H:%M:%S")

udp\_socket.sendto(text.encode('utf-8'), (ip, port))

recieve\_msg=udp\_socket.recvfrom(2048)

fp=time.time()

all\_rtt.append(int((fp-sp)\*1000))

print("ping to "+ip+", seq = "+str(n)+", rtt = "+str(int((fp-sp)\*1000))+" ms")

except:

print("ping to "+ip+", seq = "+str(n)+", "+"timeout")

n+=1

print("minimum = "+str(min(all\_rtt))+" ms , maximum = "+str(max(all\_rtt))+" ms, average = "+str(sum(all\_rtt) / len(all\_rtt))+" ms")

udp\_socket.close()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Test result:

