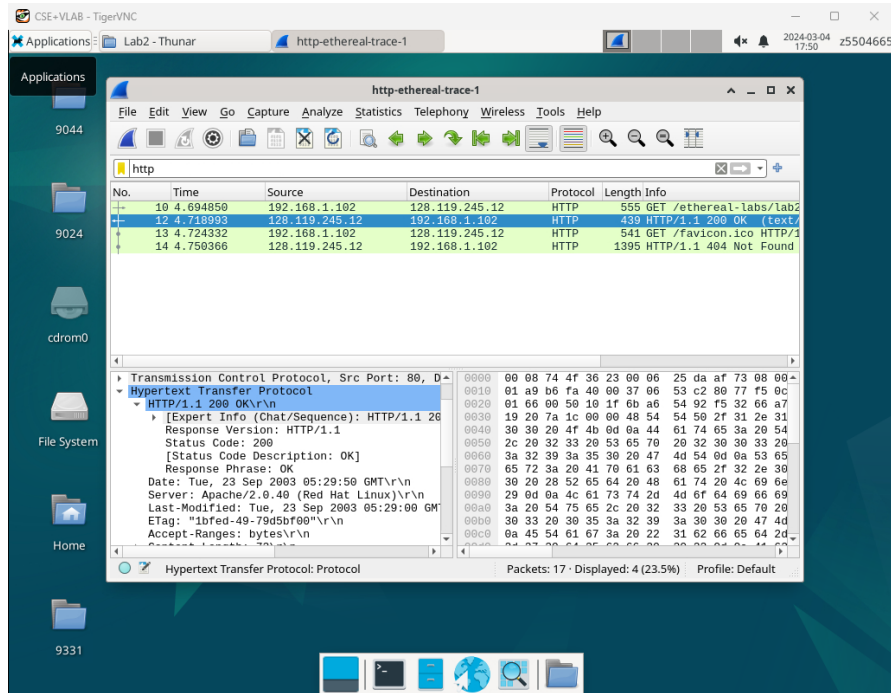


Lab2

Exercise 3

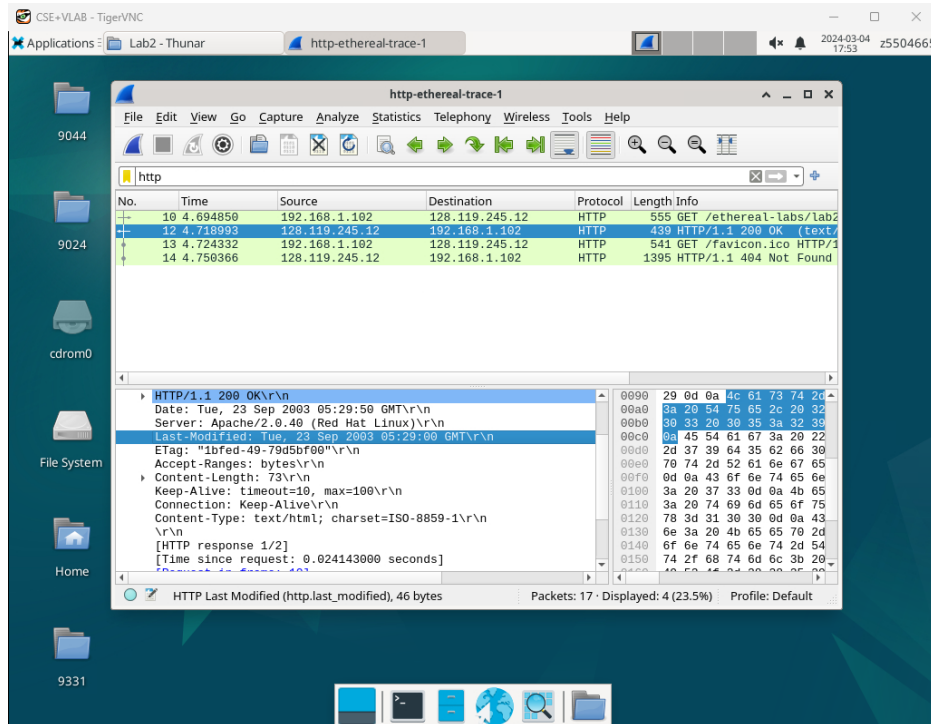
Q1:



state code:200

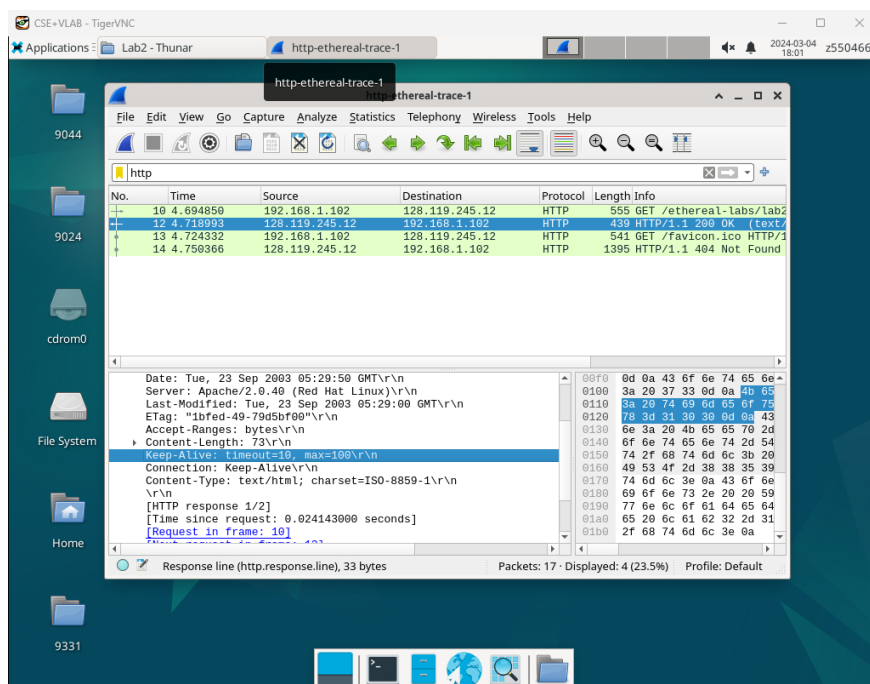
phrase:OK

Q2:



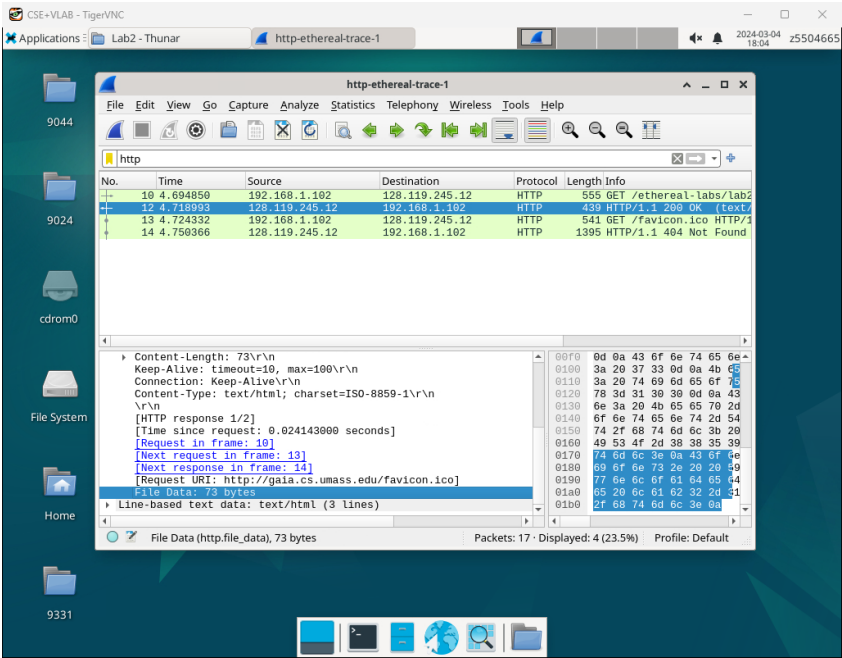
- 1) Last-modified time is Tue, 23 Sep 2003 05:29:00 GMT.
- 2) Yes, it does.
- 3) Date is the time when package was transported, and the last-modified is the last time when the package was changed.

Q3:



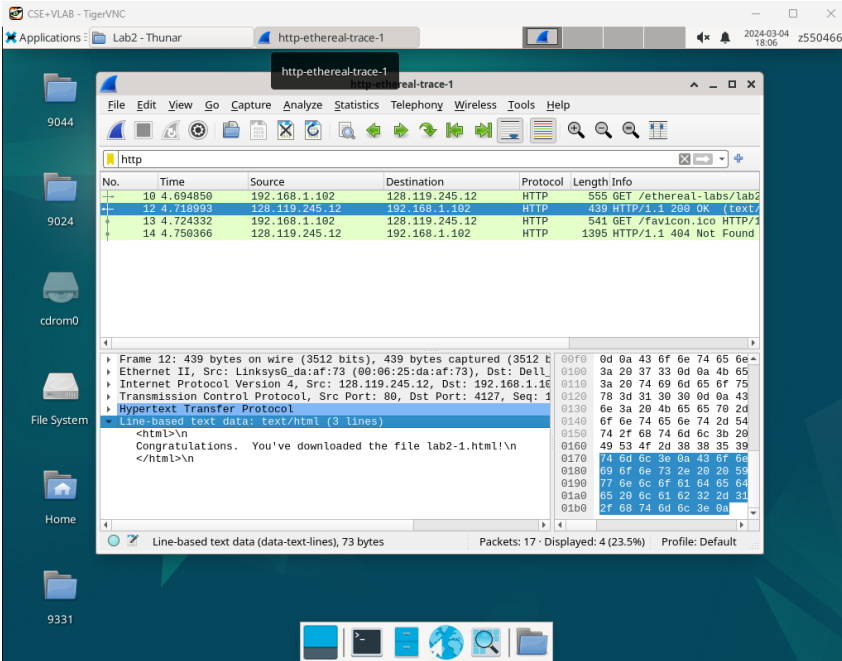
It is persistent, because we can see that there is a Keep-Alive in connection line, which means this connection is persistent.

Q4:



73 bytes

Q5:



Data:

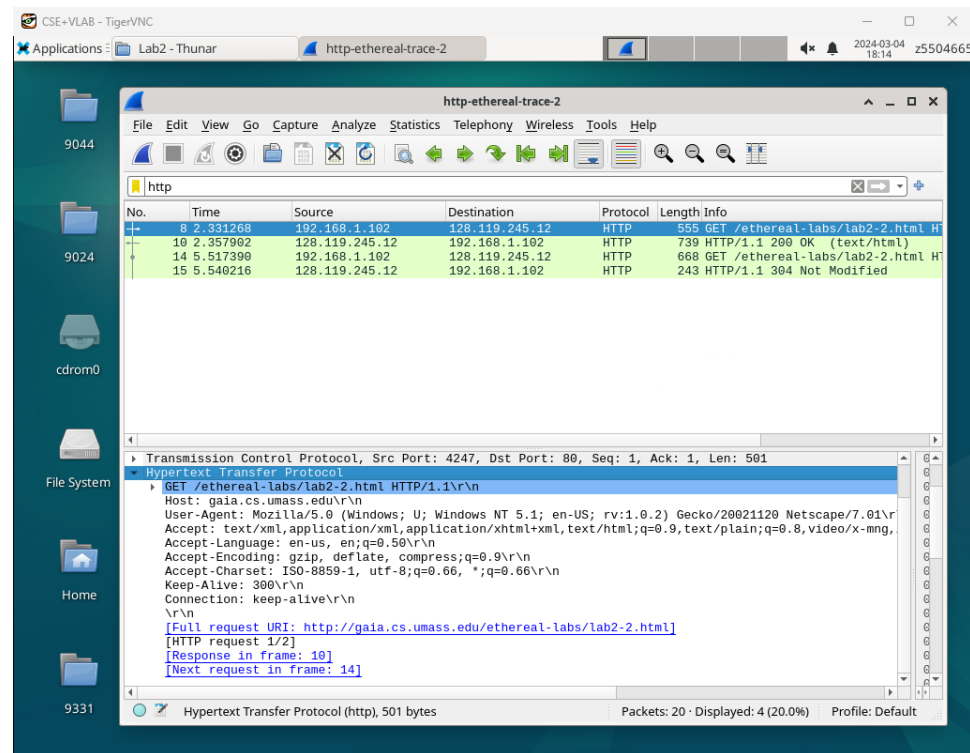
<html>\n

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

<html>\n

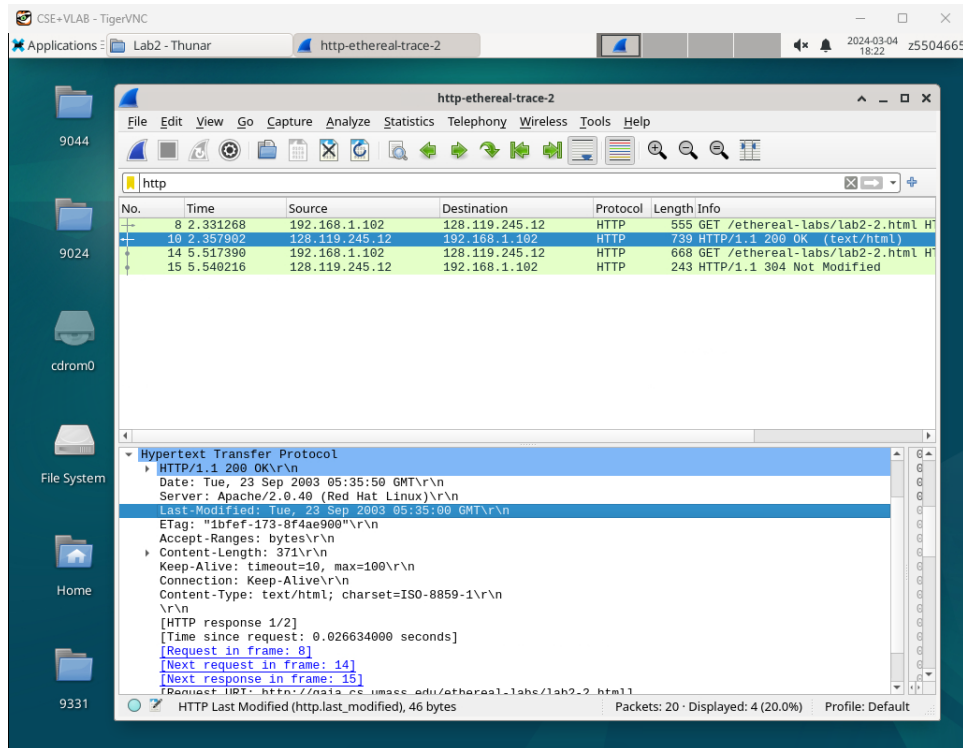
Exercise 4

Q1:



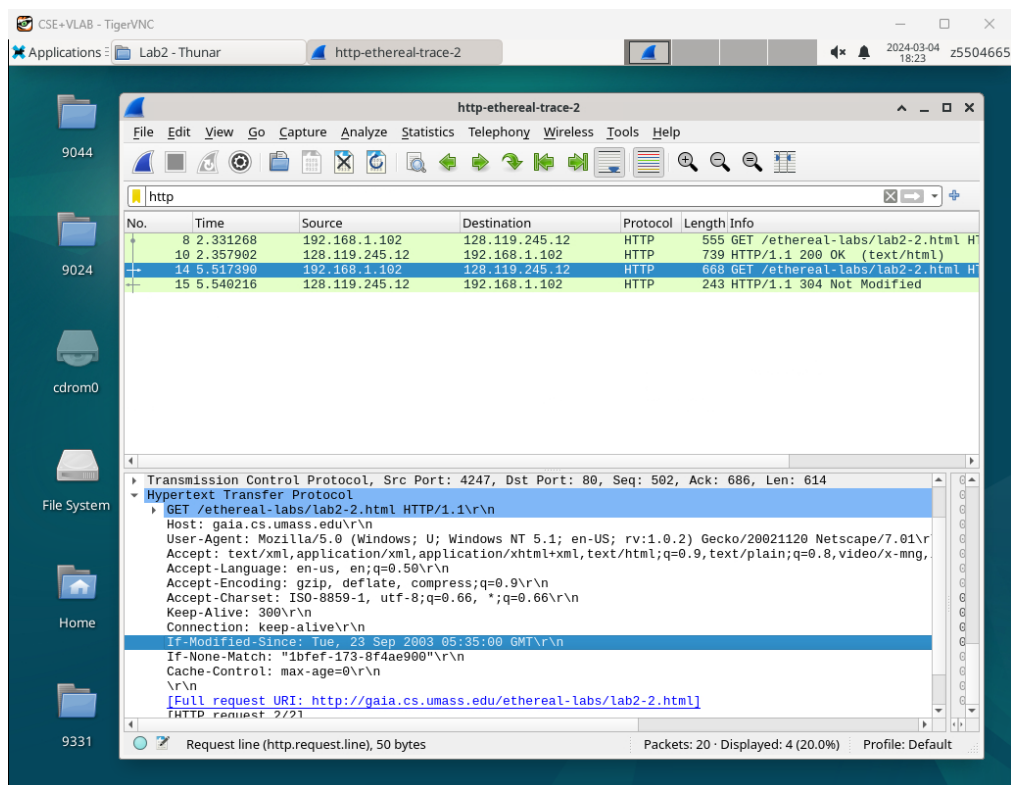
There is not "IF-MODIFIED-SINCE" line in the HTTP GET.

Q2:



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

Q3:



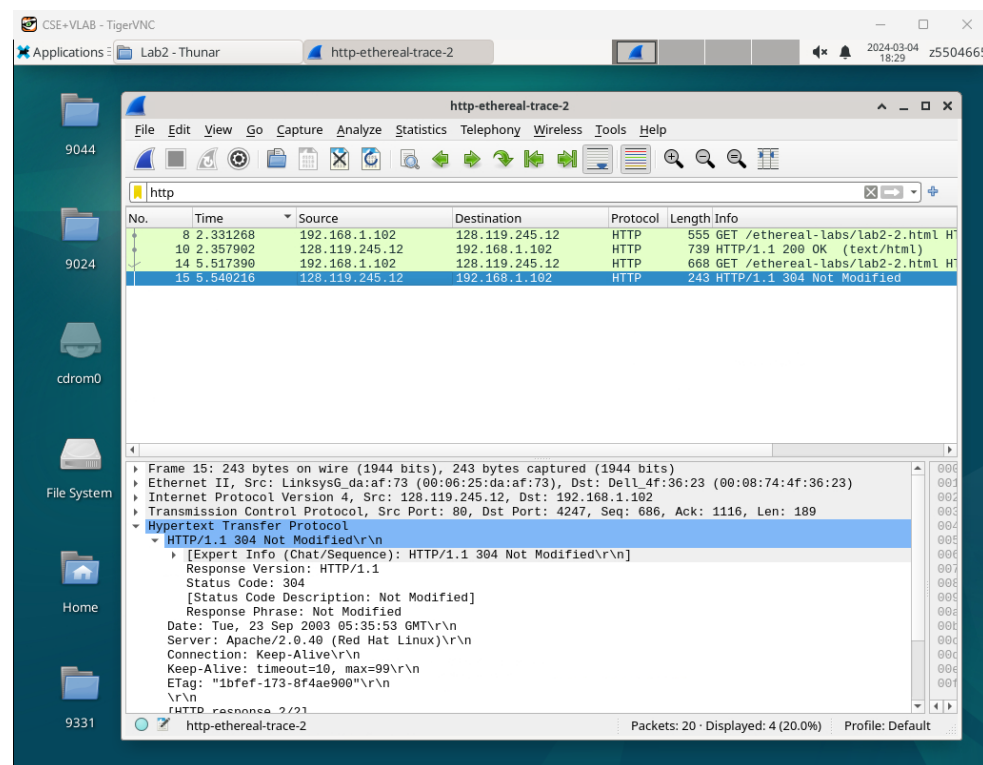
1) Yes, I see these two lines.

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:

CSE+VLAB - TigerVNC

Applications: Lab2 - Thunar http-ethereal-trace-2

2024-03-04 18:39 z5504665

http-ethereal-trace-2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
8	2.331268	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-2.html H
10	2.357902	128.119.245.12	192.168.1.102	HTTP	739	HTTP/1.1 200 OK (text/html)
14	5.517390	192.168.1.102	128.119.245.12	HTTP	668	GET /ethereal-labs/lab2-2.html H
15	5.540216	128.119.245.12	192.168.1.102	HTTP	243	HTTP/1.1 304 Not Modified

Frame 15: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits)

Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Dell_4f:36:23 (00:08:74:4f:36:23)

Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102

Transmission Control Protocol, Src Port: 80, Dst Port: 4247, Seq: 686, Ack: 1116, Len: 189

Hypertext Transfer Protocol

HTTP/1.1 304 Not Modified\r\n

Date: Tue, 23 Sep 2003 05:35:53 GMT\r\n

Server: Apache/2.0.40 (Red Hat Linux)\r\n

Connection: Keep-Alive\r\n

Keep-Alive: timeout=10, max=99\r\n

ETag: "1bfef-173-8f4ae900"\r\n

\r\n

[HTTP response 2/2]

[Time since request: 0.022826000 seconds]

[Prev request in frame: 8]

[Prev response in frame: 10]

[Request in frame: 14]

Request URI: http://gaia.cs.umass.edu/ethereal-labs/lab2-2.html

Response line (http.response.line), 28 bytes

Packets: 20 · Displayed: 4 (20.0%) Profile: Default

CSE+VLAB - TigerVNC

Applications: Lab2 - Thunar http-ethereal-trace-2

2024-03-04 18:39 z5504665

http-ethereal-trace-2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
8	2.331268	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-2.html H
10	2.357902	128.119.245.12	192.168.1.102	HTTP	739	HTTP/1.1 200 OK (text/html)
14	5.517390	192.168.1.102	128.119.245.12	HTTP	668	GET /ethereal-labs/lab2-2.html H
15	5.540216	128.119.245.12	192.168.1.102	HTTP	243	HTTP/1.1 304 Not Modified

Frame 10: 739 bytes on wire (5912 bits), 739 bytes captured (5912 bits)

Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Dell_4f:36:23 (00:08:74:4f:36:23)

Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.102

Transmission Control Protocol, Src Port: 80, Dst Port: 4247, Seq: 1, Ack: 502, Len: 685

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Date: Tue, 23 Sep 2003 05:35:50 GMT\r\n

Server: Apache/2.0.40 (Red Hat Linux)\r\n

Last-Modified: Tue, 23 Sep 2003 05:35:00 GMT\r\n

ETag: "1bfef-173-8f4ae900"\r\n

Accept-Ranges: bytes\r\n

Content-Length: 371\r\n

Keep-Alive: timeout=10, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: text/html; charset=ISO-8859-1\r\n

\r\n

[HTTP response 1/2]

[Time since request: 0.026634000 seconds]

Response line (http.response.line), 28 bytes

Packets: 20 · Displayed: 4 (20.0%) Profile: Default

- 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:

The screenshot shows a CSE+VLAB TigerVNC session. In the foreground, a terminal window displays the output of a Python script named `PingClient.py`. The script sends 20 ping requests to `127.0.0.1` with sequence numbers ranging from 56662 to 56681. The output shows various RTT values, with some timeouts. The final summary line is: `minimum = 18 ms , maximum = 196 ms, average = 106.0 ms`.

In the background, a web browser window shows the results of the ping test. It displays a list of ping requests and their corresponding replies, including the sequence number, IP address, and timestamp. The browser window also shows the task instructions for the lab exercise.

Your Task: Implementing Ping Client

You should write the `PingClient.py` such that it sends 20 ping requests to the server. Each message contains a sequence number starting from a random number between