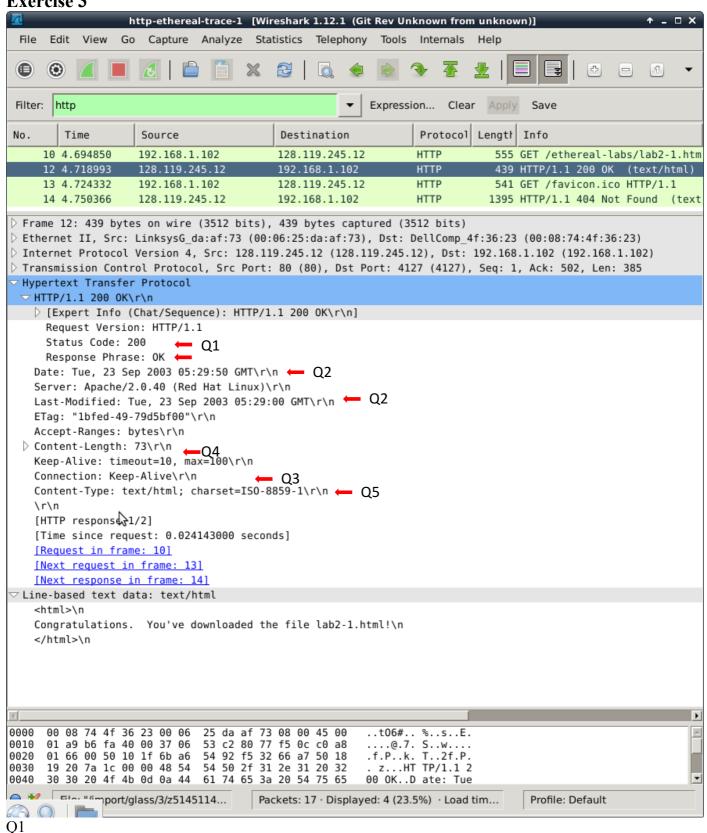# COMP9331 Lab2

Xiaodan Wang
Z5145114

Exercise 3

Q1
status: 200
phrase: OK

Q2
last modified: Tue, 23 Sep 2003 05:29:00 GMT

date header: Tue, 23 Sep 2003 05:29:50 GMT
The different of two fields is that last-modified shows the latest time of html modified in server as well as date header is the time of response sent. They may be same or date header is later than last-modified.
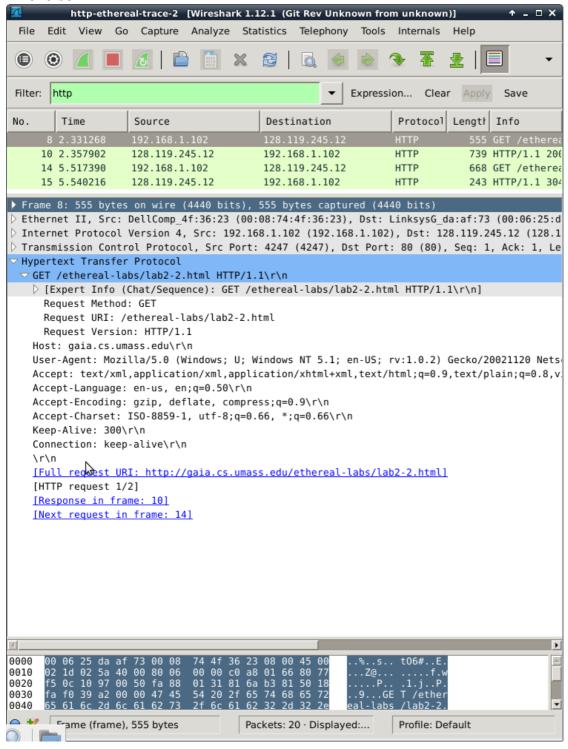
Q3
The connection between the browser and the server is persistent.
The connection is kept alive, means that the server keep connection open after sending response.
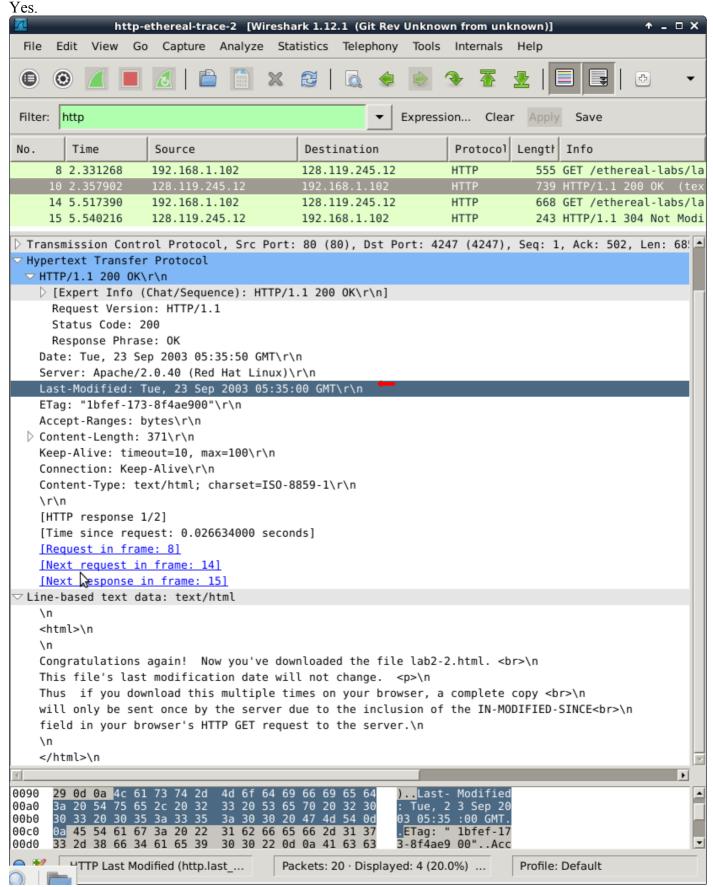
Q4
73 bytes

Q5
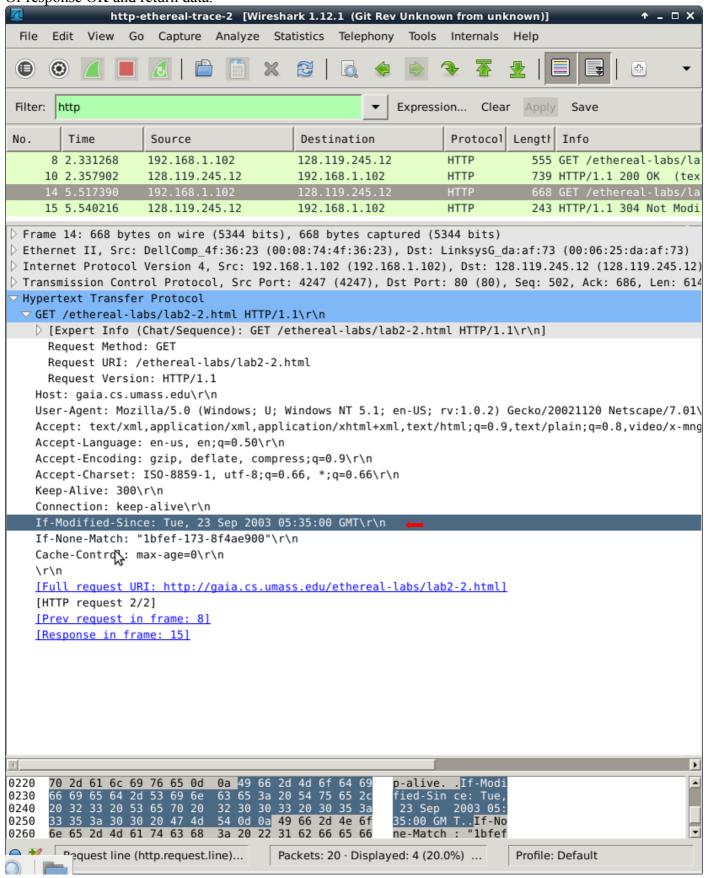text/html

Exercise 4

Q1
No.

Q2
Yes.

last modified: Tue, 23 Sep 2003 05:35:00 GMT

Q3

Yes. It is a conditional GET which the response is depends on the request data.

If the object not modified before date, response 304 not modified.

Or response OK and return data.

Q4

status code: 304

phrase: not modified

The server didn't return the contents of file.
As the request date is later than If-Modified-Since date, the HTTP response is 304 not modified.

Q5
Etag: "1bfef-173-8f4ae900"
As the If-Modified-Since conditional GET above, the request with date 23/09/2003 05:35:53 went to branch If-None-Match, and then Etag becomes that.
Since the 1st response message was received, the value hasn't changed.

Exercise 5

```
[appledeMacBook-Pro-2:lab2 apple$ java PingServer 8888
Received from 127.0.0.1: PING 0 1533823635861
   Reply sent.
Received from 127.0.0.1: PING 1 1533823635995
   Reply sent.
Received from 127.0.0.1: PING 2 1533823636147
   Reply sent.
Received from 127.0.0.1: PING 3 1533823636217
   Reply sent.
Received from 127.0.0.1: PING 4 1533823636364
   Reply sent.
Received from 127.0.0.1: PING 5 1533823636372
   Reply not sent.
Received from 127.0.0.1: PING 6 1533823637377
   Reply sent.
Received from 127.0.0.1: PING 7 1533823637528
   Reply sent.
Received from 127.0.0.1: PING 8 1533823637692
   Reply sent.
Received from 127.0.0.1: PING 9 1533823637888
   Reply sent.
```

```
lab2    lab3
[appledeMacBook-Pro-2:9331 apple$ cd lab2
[appledeMacBook-Pro-2:lab2 apple$ ls
PingClient.py        TCPClient.py          http-ethereal-trace-2
PingServer.class     TCPServer.py          lab2pdf
PingServer.java      http-ethereal-trace-1  pic
[appledeMacBook-Pro-2:lab2 apple$ python PingClient.py 127.0.0.1 8888
pint to 127.0.0.1, seq = 0, rtt = 134 ms
pint to 127.0.0.1, seq = 1, rtt = 152 ms
pint to 127.0.0.1, seq = 2, rtt = 70 ms
pint to 127.0.0.1, seq = 3, rtt = 147 ms
pint to 127.0.0.1, seq = 4, rtt = 8 ms
pint to 127.0.0.1, seq = 5, time out
pint to 127.0.0.1, seq = 6, rtt = 151 ms
pint to 127.0.0.1, seq = 7, rtt = 164 ms
pint to 127.0.0.1, seq = 8, rtt = 196 ms
pint to 127.0.0.1, seq = 9, rtt = 55 ms
min rtt: 8
max rtt: 196
avg rtt: 119
loss rate: 0.1
appledeMacBook-Pro-2:lab2 apple$
```

Left part is PingServer and right part is PingClient.

Below is the codes for PingClient in Python3.

```python
import time
from socket import *
from numpy import mean

# host and port
serverName = str(sys.argv[1])
serverPort = str(sys.argv[2])
# establish client socket
clientSocket = socket(AF_INET, SOCK_DGRAM)
# timeout 1 second wait
clientSocket.settimeout(1)

# min, max, avg
rtt = []

# 10 ping requests
for sequence in range(10):
    # massage sent: PING sequence_number timestamp
    timestamp = int(round(time.time() * 1000))
    massage1 = 'PING'+' '+str(sequence)+' '+str(timestamp)
    clientSocket.sendto(massage1.encode(),(serverName, int(serverPort)))

    # massage recieved: PING sequence_number timestamp CRLF
    try:
        # massage recieved
        recieved = clientSocket.recv(20)
        timerecv = int(round(time.time() * 1000))
        massage2 = recieved.decode()
        a  = massage2.split()
        print(f'pint to {serverName}, seq = {a[1]}, rtt = {timerecv-int(a[2])} m
        rtt.append(timerecv-int(a[2]))

    # time out
    except:
        print(f'pint to {serverName}, seq = {sequence}, time out')
        continue

if rtt:
    print(f'min rtt: {min(rtt)}\nmax rtt: {max(rtt)}\navg rtt: {int(mean(rtt))}'
else:
    print('min rtt: None\nmax rtt: None\navg rtt: None')
print(f'loss rate: {(10 - len(rtt))/10}')

clientSocket.close()
```