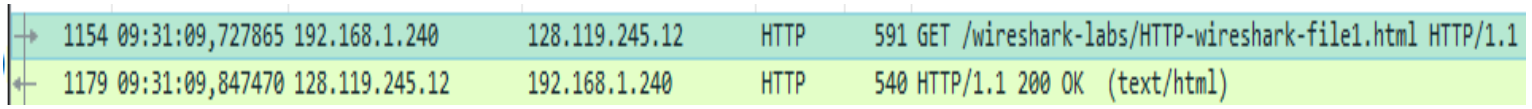
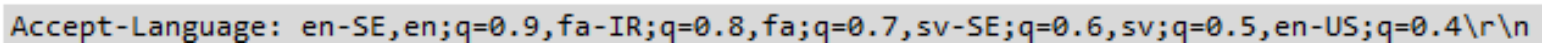
1. Both are using version HTTP/1.1.



2. Swedish, English, Farsi.

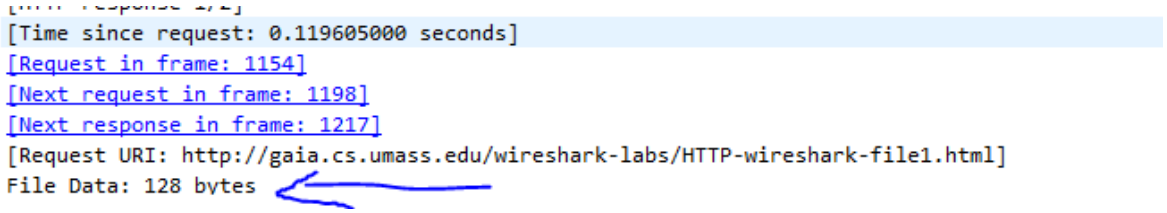


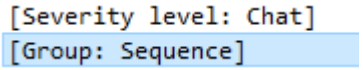
3. My computer : 192.168.1.240, server ip: 128.119.245.12 (look at picture in question 1).

4. Status code : 200. (look at picture in question 1).

5. 

6. File data : 128 bytes.



7. 

Task A: In the packet-list window we can see the url my browser requested together with HTTP-version for both my browser and the server, status code and response phrase, ip-address for my computer and for the server, the time i requested and the time i got an answer, and finally the content-type I requested.

---

8. NO.

9. Yes, we can see it in the line-based text data section.

```
Line-based text data: text/html (10 lines)
    \n
    <html>\n
    \n
    Congratulations again!  Now you've downloaded the file lab2-2.html. <br>\n
    This file's last modification date will not change.  <p>\n
    Thus  if you download this multiple times on your browser, a complete copy <br>\n
    will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
    field in your browser's HTTP GET request to the server.\n
    \n
    </html>\n
```

10. Yes, the last time the file was modified.

```
If-Modified-Since: Wed, 19 Jan 2022 06:59:01 GMT\r\n
```

11. 304 not modified. No, because it's not modified so there is no reason to send it again.

```
293 HTTP/1.1 304 Not Modified
```

Task B: The first time we asked for the file it sent the content as well, but the second time we asked for the same file it checked if the file had been modified since the last time we asked for it, and because it wasn't there was no reason to send it again.

---

12. 1 http request. packet number 3373.

```
3373 11:59:36,041208 192.168.1.240   128.119.245.12   HTTP   400 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
3413 11:59:36,170303 128.119.245.12  192.168.1.240    HTTP   535 HTTP/1.1 200 OK  (text/html)
```

13. Packet number 3413, status code: 200, response phrase: ok (look at the picture above).

14. 4 TCP segments.

```
[4 Reassembled TCP Segments (4861 bytes): #3407(1460), #3409(1460), #3411(1460), #3413(481)]
   [Frame: 3407, payload: 0-1459 (1460 bytes)]
   [Frame: 3409, payload: 1460-2919 (1460 bytes)]
   [Frame: 3411, payload: 2920-4379 (1460 bytes)]
   [Frame: 3413, payload: 4380-4860 (481 bytes)]
   [Segment count: 4]
   [Reassembled TCP length: 4861]
   [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a205765642c203139204a616e2032…]
```

15. No, TCP packets have only data from the file. (look at the picture above).

Task C : If a packet is too big to be sent as a whole it's going to be fragmented to several segments so it is easier to send them individually.

---

16. 3 st,

sida: 128.119.245.12

pearson.png : 128.119.245.12

BE_cover_small.png: 178.79.137.164

```
2358 12:18:52,547453 192.168.1.240   128.119.245.12   HTTP   400 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
2407 12:18:52,675951 128.119.245.12  192.168.1.240   HTTP  1355 HTTP/1.1 200 OK  (text/html)
2409 12:18:52,679418 192.168.1.240   128.119.245.12   HTTP   460 GET /pearson.png HTTP/1.1
2474 12:18:52,807872 128.119.245.12  192.168.1.240   HTTP   745 HTTP/1.1 200 OK  (PNG)
2517 12:18:52,881394 192.168.1.240   178.79.137.164   HTTP   467 GET /8E_cover_small.jpg HTTP/1.1
2524 12:18:52,917520 178.79.137.164  192.168.1.240   HTTP   225 HTTP/1.1 301 Moved Permanently
```

17. Seriell, because the second image was requested after the browser got the first image. (look at the picture above).

Task D : A browser might need to send many requests to receive all the information it needs to load the file. In this case it needed three, one for the main file and two for the images. If it sends two http requests at the same time then it requests them in parallel but if it waits until the first arrives and then sends the other request then it's requesting them in serial.

---

18. 401 unauthorized.

```
771 HTTP/1.1 401 Unauthorized  (text/html)
```

19.

```
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
    Credentials: wireshark-students:network
```

Task E : When we attempt to load a page that requires a login we won't get access at first only after we send the username and the password encoded in some string on the second attempt we get access to the page.

---

20. **Closed:** It means that the conversation between client and the server is over.

    Keep-alive: It means that the connection is not dropped after the server sends a response, if the client sends another request it will use the same connection.

Both of them are used after each other. When client and the server want the connection to be continued they use keepalive and once the conversation is over they must use Closed otherwise they connection will remain open and others cannot use the resources from server that is used to keep the connection alive.