Assignment 2

Vuong Ho s0592666 | Emily Nguyen s0599147 | Mandy Nguyen s0579116

Problem 1 - Mandy

Problem 1.1

$$\begin{aligned}
\text{Transmission Delay } L_2 &= \frac{L}{R} && (1)\\
&= \frac{800000\ \text{Bit}}{1000000\ \text{Bit/s}} && (2)\\
&= \frac{8}{10} = 0.8\ \text{s} = 800\ \text{ms} && (3)\\
&= 800\ \text{ms}\ \cdot 2 = \boxed{1600\ \text{ms}} && (4)
\end{aligned}$$

$$\begin{aligned}
\text{Transmission Delay } L_1 &= \frac{L}{R} && (5)\\
&= \frac{400000\ \text{Bit}}{1000000\ \text{Bit/s}} && (6)\\
&= \frac{4}{10} = 0.4\ \text{s} = 400\ \text{ms} && (7)\\
&= 400\ \text{ms}\ \cdot 2 = \boxed{800\ \text{ms}} && (8)
\end{aligned}$$

$$\begin{aligned}
\text{RTT}\ &= 2 \cdot (10\ \text{ms}\ + 20\ \text{ms}\ + 5\ \text{ms}\ + 10\ \text{ms}) && (9)\\
&= \boxed{90\ \text{ms}} && (10)
\end{aligned}$$

$$\begin{aligned}
T_1 &= \text{RTT} + \text{RTT} + D_{trans} && (11)\\
&= 90\ \text{ms}\ + 90\ \text{ms}\ + 800\ \text{ms} && (12)\\
&= \boxed{980\ \text{ms}} && (13)
\end{aligned}$$

$$
\begin{aligned}
T_2 &= \mathrm{RTT} + \mathrm{RTT} + D_{trans} && (14)\\
&= 90\ \mathrm{ms}\ + 90\ \mathrm{ms}\ + 1600\ \mathrm{ms} && (15)\\
&= \boxed{1780\ \mathrm{ms}} && (16)
\end{aligned}
$$

$$
\begin{aligned}
T_{ges} &= T_1 + T_2 + D_{trans} && (17)\\
&= 980\ \mathrm{ms}\ + 1780\ \mathrm{ms} && (18)\\
&= 2760\ \mathrm{ms}\ = \boxed{2.76\ \mathrm{s}} && (19)
\end{aligned}
$$

Problem 1.2

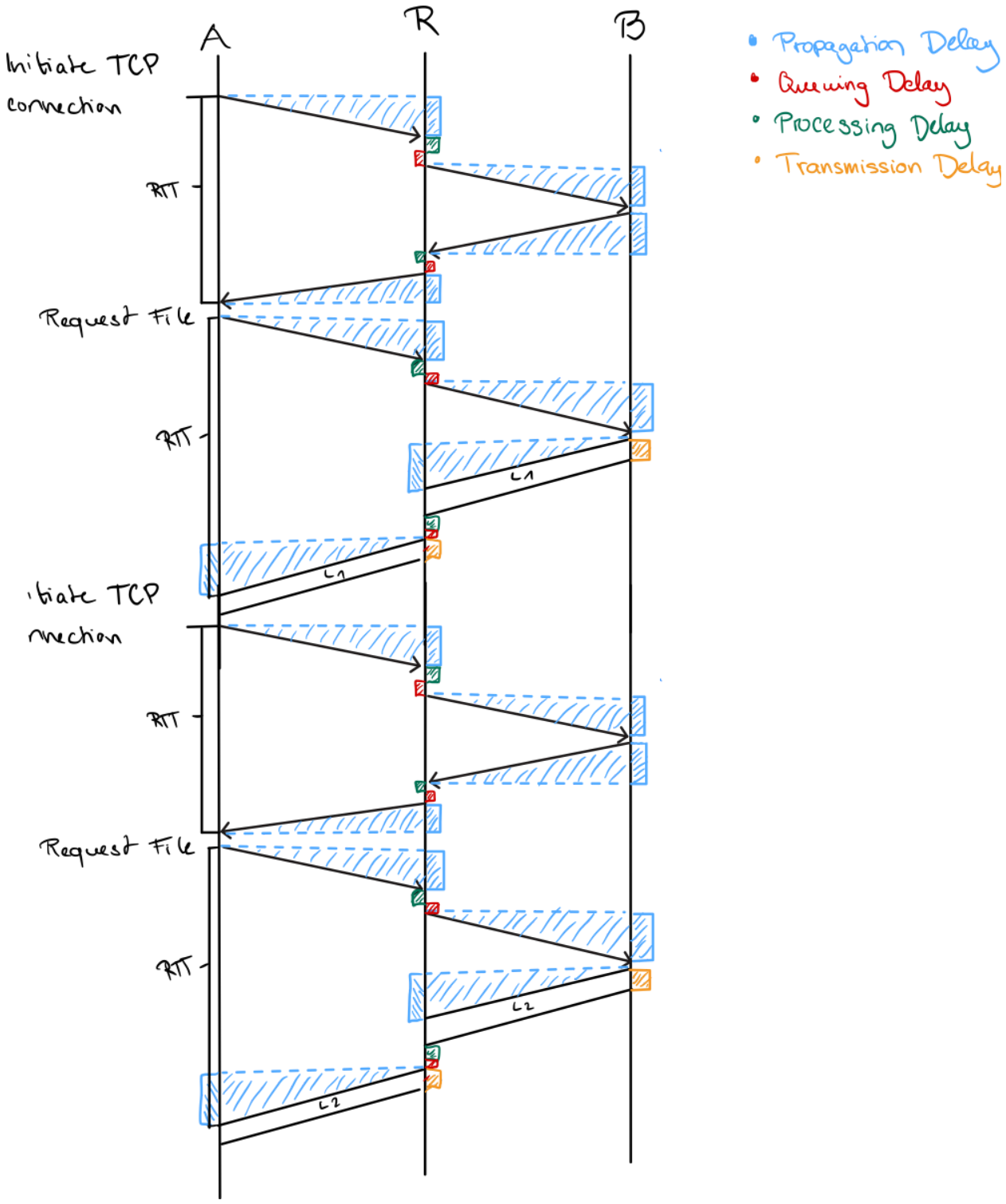
$$
\begin{aligned}
T_1 &= \mathrm{RTT} + \mathrm{RTT} + D_{trans} && (20)\\
&= 90\ \mathrm{ms}\ + 90\ \mathrm{ms}\ + 800\ \mathrm{ms} && (21)\\
&= \boxed{980\ \mathrm{ms}} && (22)
\end{aligned}
$$

$$
\begin{aligned}
T_2 &= \mathrm{RTT} + D_{trans} && (23)\\
&= 90\ \mathrm{ms}\ + 1600\ \mathrm{ms} && (24)\\
&= \boxed{1690\ \mathrm{ms}} && (25)
\end{aligned}
$$

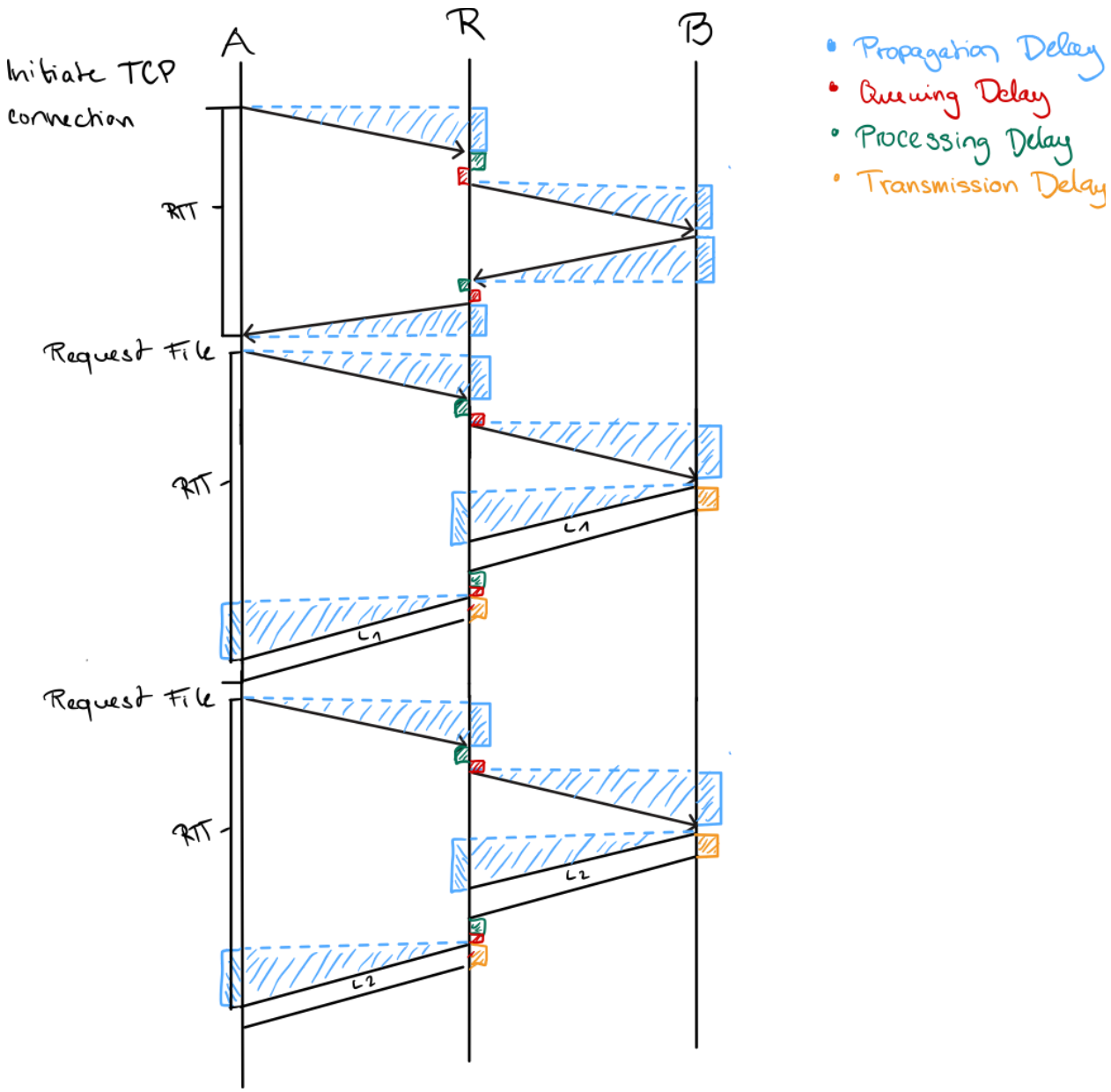
$$
\begin{aligned}
T_{ges} &= T_1 + T_2 && (26)\\
&= 980\ \mathrm{ms}\ + 1690\ \mathrm{ms} && (27)\\
&= \boxed{2670\ \mathrm{ms}} && (28)
\end{aligned}
$$

Problem 1.3

Non-Persistent connection



Persistent connection



Problem 1.4

In a non-persistent connection a new TCP connection is established for each packet which is why it takes longer for the client to receive the packets. There will be multiple handshakes but it divides the packets more clearly. While in a persistent connection it only needs to establish 1 TCP connection where multiple HTTP request can go through. The connection closes when it is actively told to do so or when it times out. That's why a persistent connection is faster than a non-persistent connection.

Problem 2 Emily

The following network-based application is an online collaborative canvas. Within the canvas there is a base model of a human, which needs to be dressed up. Users can customize the base model by drawing, uploading images, dragging elements and using various other tools through a mobile interface. The purpose of this application is to let users work in teams to create unique game character designs.
It does not require real-time responsiveness. Some delay is acceptable, however it should be kept minimal. For instance, when users make changes the other users should be able to see it right away.
Average throughput will definitely suffice. Uploaded images will be transmitted in small data packets and the overall data load will remain manageable.
This application needs highly reliable delivery of data. The reason for that is the synchronization of the changes. As a result there will be complete artworks with no losses of images or drawings.
Of course strong security is to be expected. Only authorized users who have a valid passkey should be able to edit the canvas project. In order to prevent vandalism and provide protection of the project.

A Client-Server architecture is the most suitable option for this network-based application. In this case there is a centralized server, which manages all user actions and their data. This can come in handy for a smooth experience.

The server will have a great overview of the users and can ensure security. For instance only users who have been granted permission with a passkey, are able to join a certain project and make changes to it. Additionally it can keep a track of who accessed the project and what time they have made the last change.

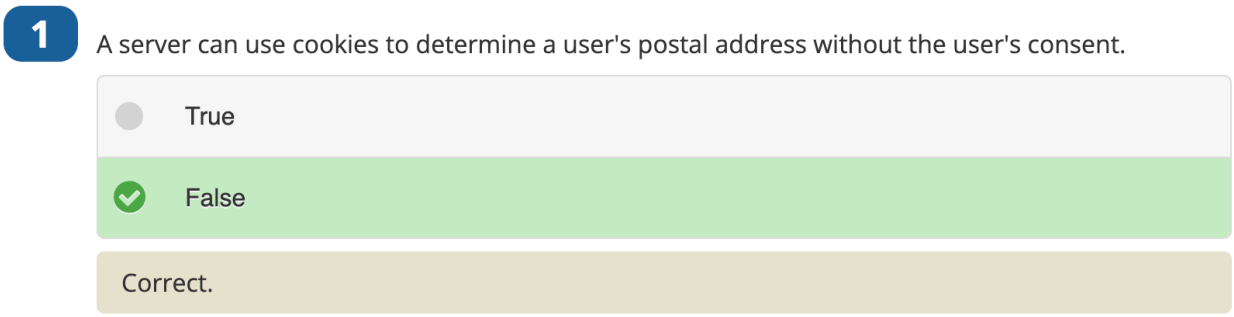
On the other hand a downside of Client-Server architecture is that it can’t handle overload. Therefore if too many clients send requests it can cause the server to slow down or in the worst case even shut down. Regardless of this disadvantage, the chosen architecture still provides enhanced overall performance, high reliability and control than the P2P architecture.

Problem 3 Vuong

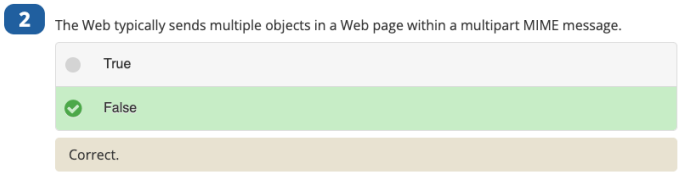
1. We are designing a real-time multiplayer game. Players move together in a shared
   world and each client sends continuous updates about movement and position to the
   game server. The server then forwards these updates to all other players. The interaction must feel fast and responsive so players can react immediately.
2. Reliability: Our application does not require fully reliable delivery. Losing a few position packets does not break the game because new updates arrive every few
   milliseconds.
   Latency: The game needs very low latency. Any noticeable delay affects the gameplay and makes movement feel unresponsive. This is especially important if it's
   a competitive game where latency matters most.
   Throughput: The throughput demand is low. The packets are small and sent frequently, so the application does not need high bandwidth.
3. TCP or UDP?
   We choose UDP as the transport protocol because our application focuses on real-time
   responsiveness. In a multiplayer game, the players need to see movements and actions
   from others as quickly as possible. UDP supports this because it sends packets immediately
   without waiting for acknowledgements or retransmissions. This keeps the delay very low,
   which is essential for gameplay that depends on fast reactions.
   Another reason for choosing UDP is that losing a few packets does not harm the game.
   Since the clients constantly send new position updates, the information is quickly replaced.
   TCP would try to retransmit older packets, which would create delays and

result in outdated
game states being delivered. This would negatively affect the user experience. However, UDP also has a drawback. it doesn't guarantee delivery, order, or error checking.
This means we must handle missing or out-of-order updates ourselves inside the game
logic. Still, for our use case, the benefits of low latency clearly outweigh the disadvantages.

Problem 4 Mandy



Cookies only save the data (like postal address) when the users put them in themselves.



The Web uses HTTP to send multiple objects separately, so multiple separate HTTP requests.

3 It is possible for a mail client to send e-mail messages over HTTP rather than SMTP.

- ✓ True
- False

Correct.

Mail providers like Gmail or Outlook.com use HTTP/HTTPS to send mails. The user opens the site through their browser and sends a message via HTTP to the corresponding web server. The server then forwards the message via SMTP.

5 With SMTP, it is possible to send multiple mail messages over the same TCP connection.

- ✓ True
- False

Correct.

With SMTP, it is possible to send multiple mails over the same TCP connection as long as the client starts a new MAIL FROM sequence.

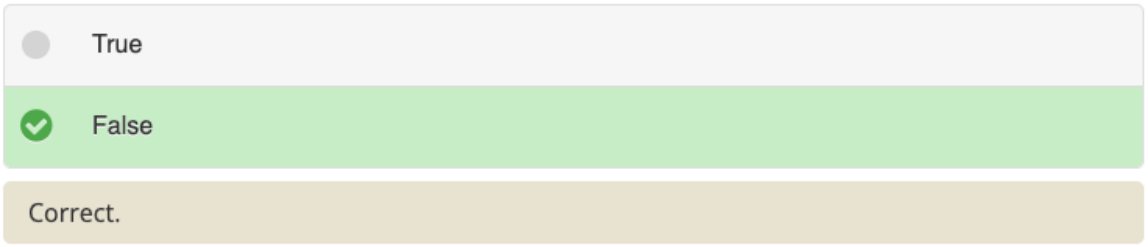
6 DNS lookups often involve a combination of recursive and iterative queries.

- ✓ True
- False

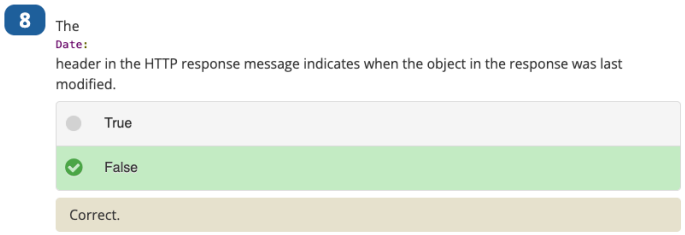
Correct.

The client speaks to the DNS server which is recursive and the DNS server speaks to multiple other servers to find out the IP which is iterative.

7 With non-persistent connections between browser and origin server, it is possible for a single TCP segment to carry two distinct HTTP request messages.

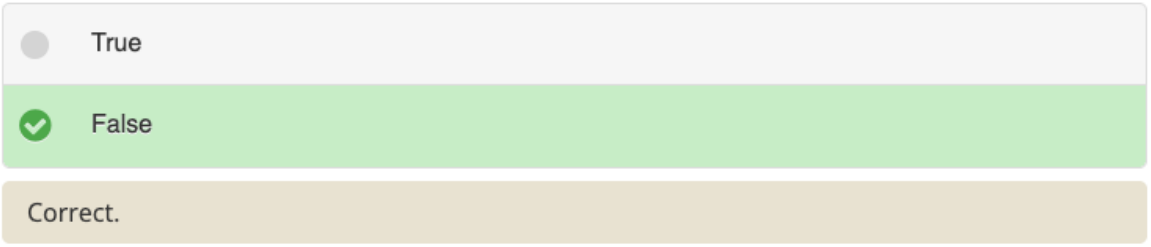


In a non-persistent connection every HTTP request gets its own connection.Í



The date in the header shows the date and time when the server send the response. There is another header that says when the object was last modified: “Last-Modified:”.

9 In BitTorrent, a peer’s instantaneous download rate can never exceed its instantaneous upload rate.



In BitTorrent a peer’s download rate can exceed its upload rate because it receives from multiple other peers but it can’t send to many others.

11 If an HTTP request message uses the
`Accept-language: fr`
header, and the server only has an English version of the object, then the server will return the
`404 Document Not Found`
error message.

- ○ True
- ✓ False

Correct.

The header only prefers the language given and doesn't force it. If it is available the language will be displayed and if not the default language of the website will be displayed.

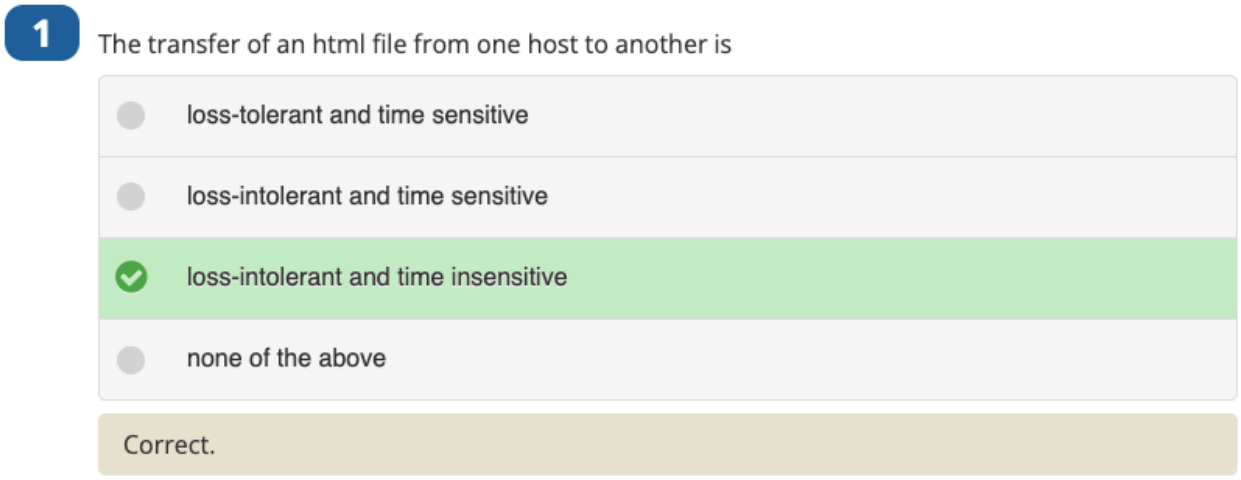
12 With POP3 client, user folder information is kept on the mail server.

- ○ True
- ✓ False

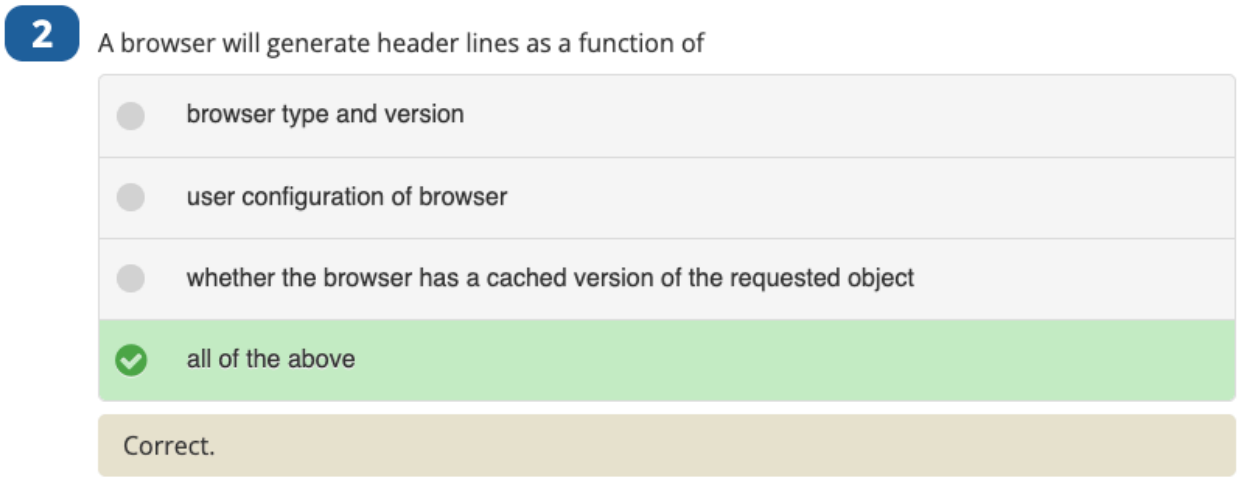
Correct.

The mail server doesn't store the folder information. It only knows the Inbox and temporarily stores them until the client accesses them. Then they are deleted from the server.

Problem 5 Emily,Vuong



HTTP uses the TCP protocol to prevent any data loss. If anything was missing the HTML wouldn't load properly. That's also why it is time insensitive since TCP makes sure that everything arrives where it should.



The header differs from browser to browser since one browser could support extra headers which the other one can't. The user can also influence the header by setting a specific language or other customizations. If there is a cached version the header also checks if the object is up to date.

3 Suppose a client sends an HTTP request message with the `If-modified-since:` header. Suppose the object in a server has not changed since the last time a client retrieved the object. Then the server will send a response message with the status code:

- `200 OK`
- `404 Not Found`
- ✓ `304 Not Modified`
- none of the above

Correct.

The server sees that the object hasn't changed and tells the client to use the cached version.

5 SMTP is used to

- ✓ to transfer messages from one mail server to another
- to transfer messages from mail server to a user agent
- to define the format of message headers
- all of the above

Correct.

SMTP transfers messages from the source mail server to the destination mail server.

7 Local DNS name servers

- never cache resource records
- ✓ cache resource records, but discard them after a period of time that is on the order of a few days
- cache resource records and never discard them
- obtain resource records from Web caches

Correct.

DNS servers discard cached information after a period of time (often set to two days).

9 Consider the minimum time for P2P file distribution, as discussed in Section 2.6. Suppose the upload rate of the server is10 Mbps, the upload rates of all peers is 1 Mbps, and the download rates of all peers is 10 Mbps. Further suppose that there are 100 peers and the file is 100 million bits. The minimum distribution time is

- approximately 1 second
- approximately 10 seconds
- approximately 1000 seconds
- ✓ approximately 100 seconds

Correct.

Formel:

$$D_{\text{P2P}} = \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}$$

Gegeben:

us (upload rate server): 10Mbps →10000000 = 10 × 10^6 bps

ui (upload rate of all peers): 1 Mbps →1000000 = 1 × 10^6 bps

N = peers = 100

F = 100000000 = 100 × 10^6 bits

dmin = 10 Mbps = 10 × 10^6 bps

F\us = 100 × 10^6 bits / 10 × 10^6 bps = 10 sekunden

F/dmin = 100 × 10^6 bits /10 × 10^6 bps = 10 sekunden

NF\ us +

$$\sum_{i=1}^{N} u_i$$

= 100 × 100 × 10^6/ 10 × 10^6 + 100 × 1 × 10^6= 90,90909091

gerundet: 91 sekunden

DP2P = max {10,10,90,909}

11 For the Web application, the user agent is

- the user's keyboard
- the user's monitor
- the user's mouse
- ✓ none of the above

Correct.

It is a software agent. For instance web browsers (such as Opera GX or Bing) and email clients.

12 Internet telephony typically runs over

- ✓ UDP
- TCP
- HTTP
- none of the above

Correct.

UDP is connectionless and generally more suited for real-time performance. It transfers data quicker than TCP but is less reliable. Additionally there will be no retransmission delays.

Problem 6.1

Mandy s0579116

ARP, MDNS, TCP, DNS, TLSv1.3, NTP, IPv4, TLSv1.2, IGMPv2, SSDP, ICMP, HTTP, UDP, ICMPv6

Emily s0599147

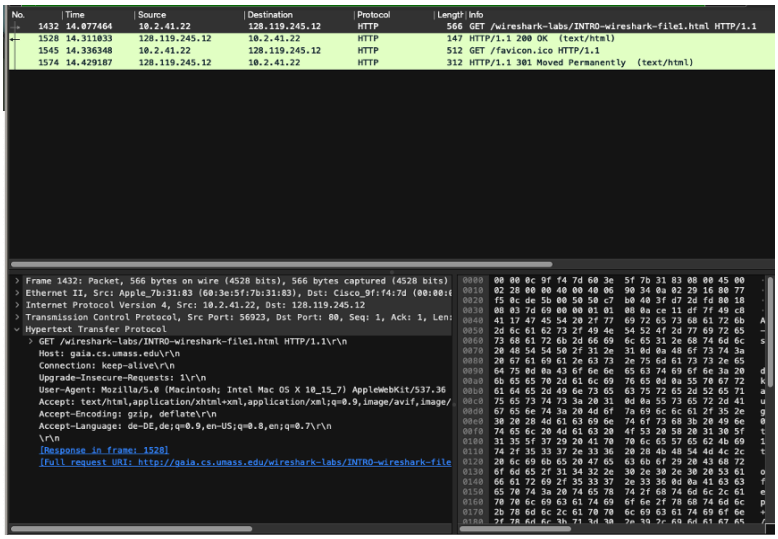
TCP, DNS, MDNS, UDP, QUIC, TLSv1.2, TLSv1.3, SSL, HTTP, SSL, ICMPv6, IPv6, ARP, NBNS, LLMNR

Vuong s0592666

ICMPv6, TCP, MDNS, QUIC, TLSv1.2, TLSv1.3, DNS, HTTP

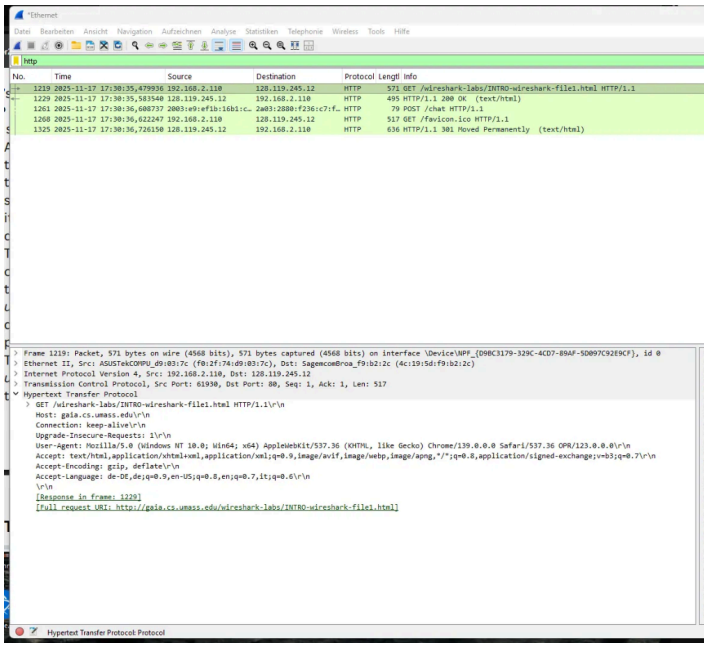
Problem 6.2

Mandy s0579116



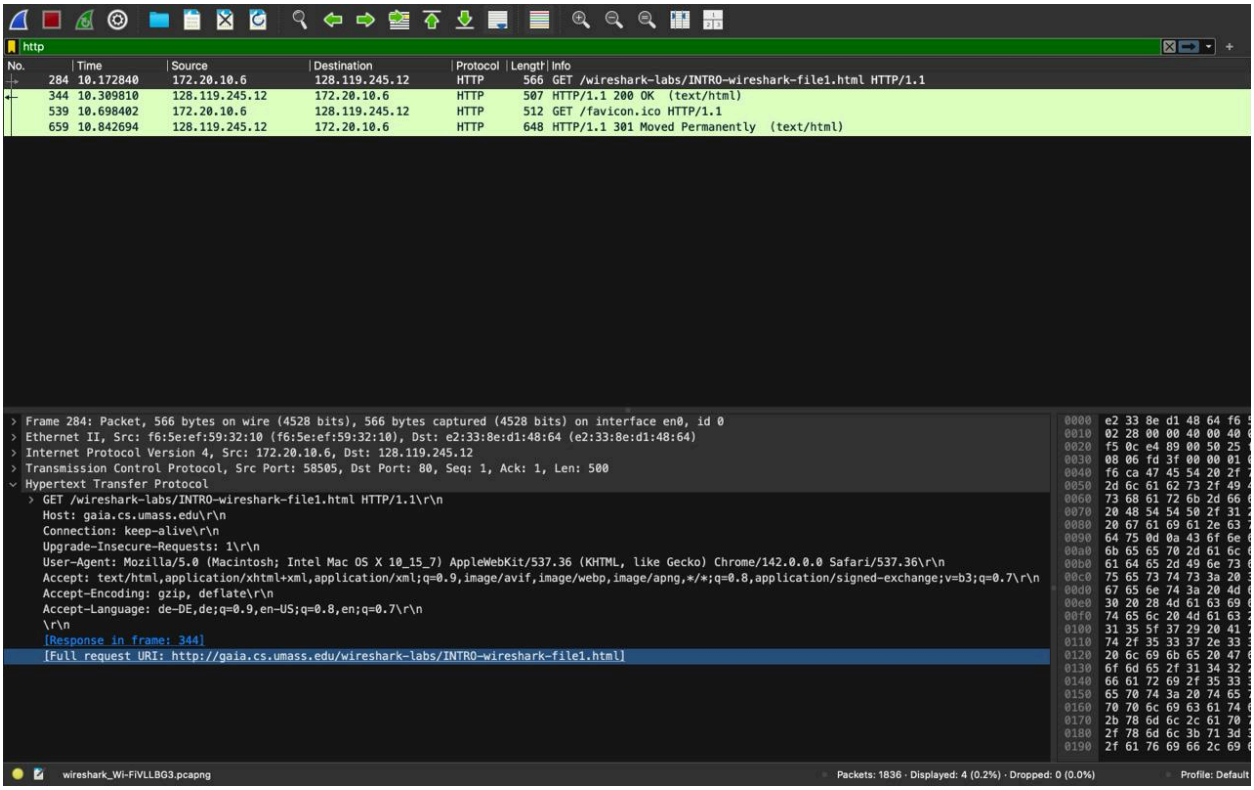
233,57 ms

Emily s0599147



103,604 ms

Vuong s0592666



136,97ms

Problem 6.3

Mandy s0579116

www-net.cs.umass.edu = 128.119.245.12 | myIP = 10.2.41.22

Emily s0599147

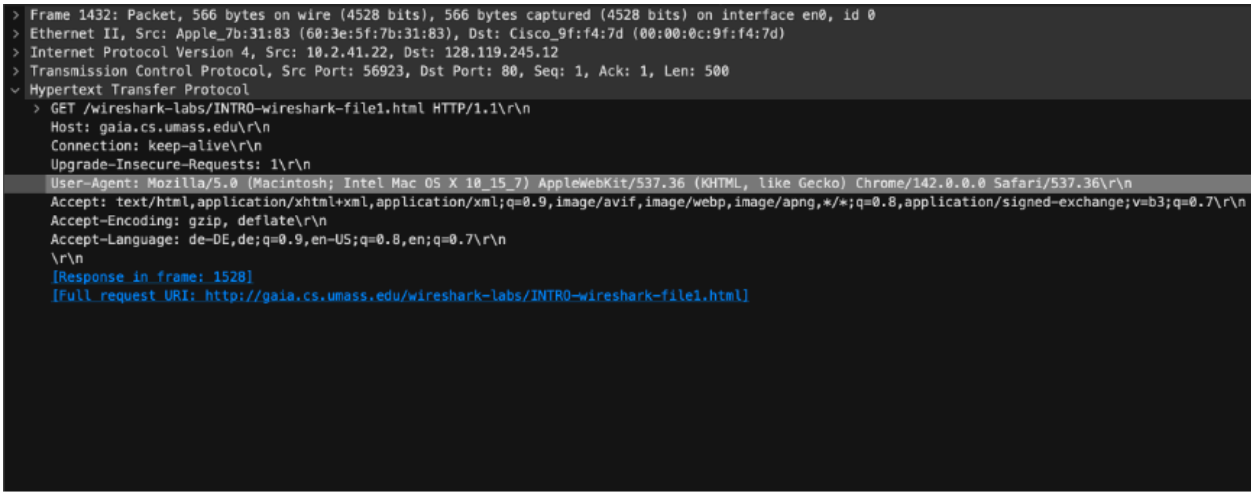
myIP = 192.168.2.110 | www-net.cs.umass.edu = 128.119.245.12

Vuong Ho s0592666

source(ich): 172.20.10.6
destination(server): 128.119.245.12

Problem 6.4

Mandy s0579116



Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36

Emily s0599147

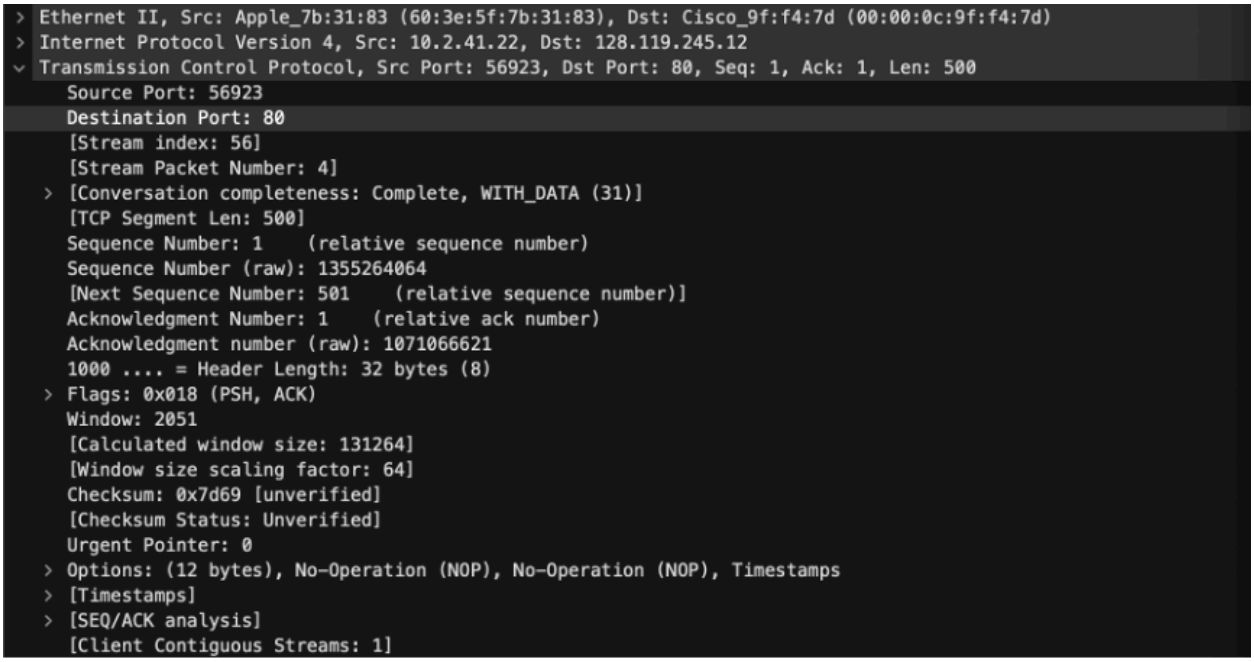
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36 OPR/123.0.0.0\r\n

Vuong Ho s0592666

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36\r\n

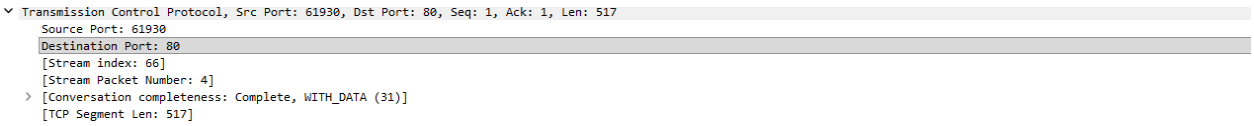
Problem 6.5

Mandy s0579116



Destination Port: 80

Emily s0599147



Destination Port: 80

Vuong Ho s0592666

Destination Port: 80

Problem 6.6

Mandy s0579116

/Users/mandy.nguyen/Documents/Uni/Netzwerke/Assignments/assignment2netzwerke.pcapng 2198 total packets, 1 shown

```
No.     Time           Source                Destination           Protocol Length Info
   1528 14.311033      128.119.245.12        10.2.41.22            HTTP     147    HTTP/1.1 200
OK  (text/html)
Frame 1528: Packet, 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface
en0, id 0
Ethernet II, Src: Cisco_9f:f4:7d (00:00:0c:9f:f4:7d), Dst: Apple_7b:31:83 (60:3e:5f:7b:31:83)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.2.41.22
Transmission Control Protocol, Src Port: 80, Dst Port: 56923, Seq: 361, Ack: 501, Len: 81
    Source Port: 80
    Destination Port: 56923
    [Stream index: 56]
    [Stream Packet Number: 8]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 81]
    Sequence Number: 361    (relative sequence number)
    Sequence Number (raw): 1071066981
    [Next Sequence Number: 442    (relative sequence number)]
    Acknowledgment Number: 501    (relative ack number)
    Acknowledgment number (raw): 1355264564
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x018 (PSH, ACK)
    Window: 4096
    [Calculated window size: 262144]
    [Window size scaling factor: 64]
    Checksum: 0xf423 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (81 bytes)
    TCP segment data (81 bytes)
[2 Reassembled TCP Segments (441 bytes): #1527(360), #1528(81)]
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Mon, 17 Nov 2025 16:03:57 GMT\r\n
    Server: Apache/2.4.62 (AlmaLinux) OpenSSL/3.2.2 mod_fcgid/2.3.9 mod_perl/2.0.12 Perl/
v5.32.1\r\n
    Last-Modified: Tue, 28 Oct 2025 05:59:01 GMT\r\n
    ETag: "51-64231b6715777"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 81\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [Request in frame: 1432]
    [Time since request: 233.569000 milliseconds]
    [Request URI: /wireshark-labs/INTRO-wireshark-file1.html]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    File Data: 81 bytes
Line-based text data: text/html (3 lines)
```

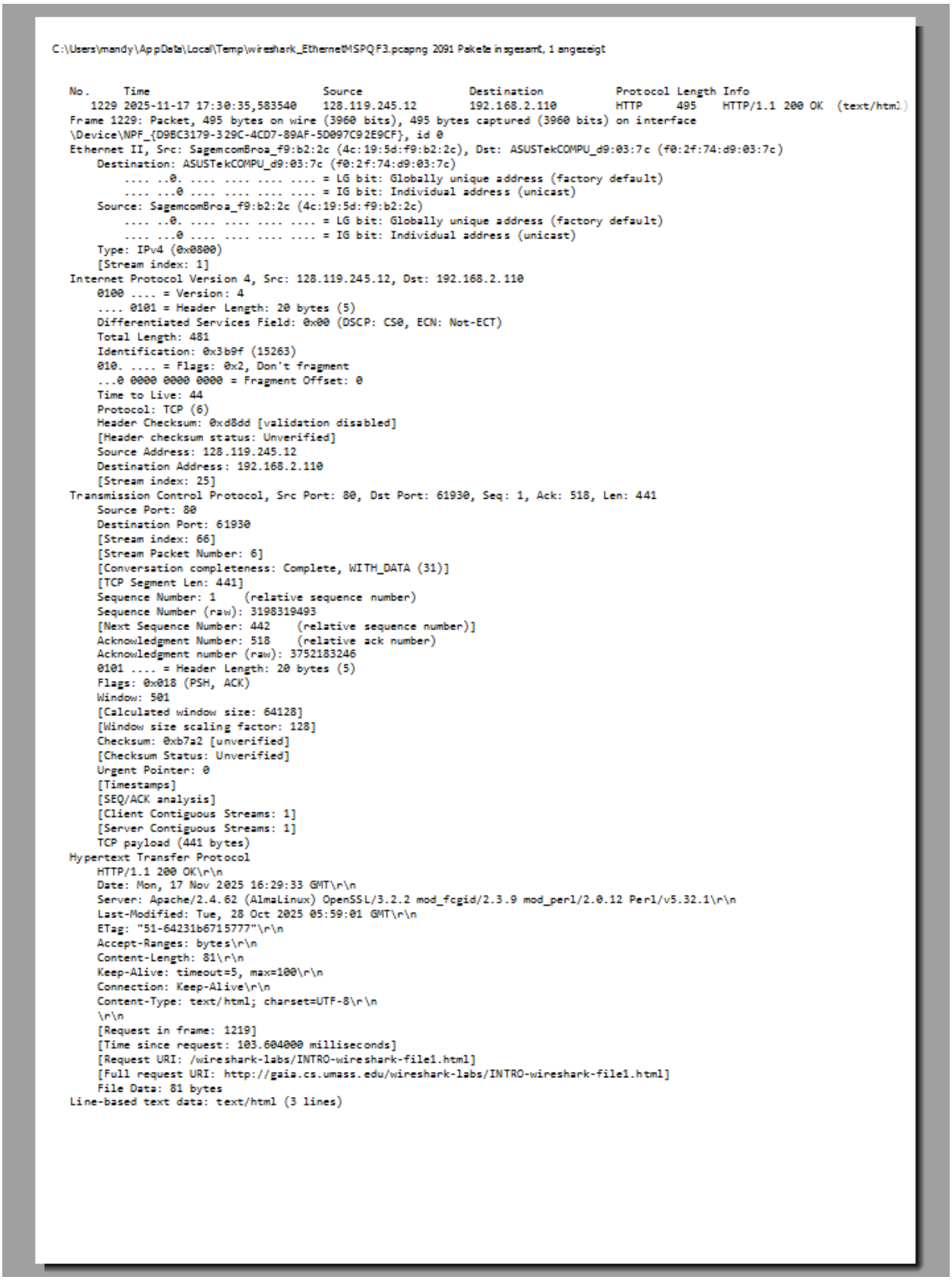
/Users/mandy.nguyen/Documents/Uni/Netzwerke/Assignments/assignment2netzwerke.pcapng 2198 total packets, 1 shown

```
No.     Time           Source                Destination           Protocol Length Info
   1432 14.077464      10.2.41.22            128.119.245.12        HTTP     566    GET /
wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
Frame 1432: Packet, 566 bytes on wire (4528 bits), 566 bytes captured (4528 bits) on interface
en0, id 0
Ethernet II, Src: Apple_7b:31:83 (60:3e:5f:7b:31:83), Dst: Cisco_9f:f4:7d (00:00:0c:9f:f4:7d)
Internet Protocol Version 4, Src: 10.2.41.22, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 56923, Dst Port: 80, Seq: 1, Ack: 1, Len: 500
    Source Port: 56923
    Destination Port: 80
    [Stream index: 56]
    [Stream Packet Number: 4]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 500]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1355264064
    [Next Sequence Number: 501    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1071066621
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x018 (PSH, ACK)
    Window: 2051
    [Calculated window size: 131264]
    [Window size scaling factor: 64]
    Checksum: 0x7d69 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (500 bytes)
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/142.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    \r\n
    [Response in frame: 1528]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
```

Emily **s0599147**

C:\Users\mandy\AppData\Local\Temp\wireshark_EthernetMSPQF3.pcapng 2091 Pakete insgesamt, 1 angezeigt

```
No.     Time                           Source                Destination           Protocol Length Info
   1219 2025-11-17 17:30:35,479936     192.168.2.110         128.119.245.12        HTTP     571    GET /wireshark-labs/INTRO-
wireshark-file1.html HTTP/1.1
Frame 1219: Packet, 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface
\Device\NPF_{D9BC3179-329C-4CD7-89AF-5D097C92E9CF}, id 0
Ethernet II, Src: ASUSTekCOMPU_d9:03:7c (f0:2f:74:d9:03:7c), Dst: SagemcomBroa_f9:b2:2c (4c:19:5d:f9:b2:2c)
    Destination: SagemcomBroa_f9:b2:2c (4c:19:5d:f9:b2:2c)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: ASUSTekCOMPU_d9:03:7c (f0:2f:74:d9:03:7c)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
    [Stream index: 1]
Internet Protocol Version 4, Src: 192.168.2.110, Dst: 128.119.245.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 557
    Identification: 0x8456 (33878)
    010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.2.110
    Destination Address: 128.119.245.12
    [Stream index: 25]
Transmission Control Protocol, Src Port: 61930, Dst Port: 80, Seq: 1, Ack: 1, Len: 517
    Source Port: 61930
    Destination Port: 80
    [Stream index: 66]
    [Stream Packet Number: 4]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 517]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 3752182729
    [Next Sequence Number: 518    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3198319493
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
    Checksum: 0x3aba [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (517 bytes)
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/
537.36 OPR/123.0.0.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7,it;q=0.6\r\n
    \r\n
    [Response in frame: 1229]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
```



```
C:\Users\mandy\AppData\Local\Temp\wireshark_EthernetMSPQF3.pcapng 2091 Pakete insgesamt, 1 angezeigt

No.     Time                           Source                Destination           Protocol Length Info
   1229 2025-11-17 17:30:35,583540    128.119.245.12        192.168.2.110         HTTP     495    HTTP/1.1 200 OK  (text/html)
Frame 1229: Packet, 495 bytes on wire (3960 bits), 495 bytes captured (3960 bits) on interface
\Device\NPF_{D9BC3179-329C-4CD7-89AF-5D097C92E9CF}, id 0
Ethernet II, Src: SagemcomBroa_f9:b2:2c (4c:19:5d:f9:b2:2c), Dst: ASUSTekCOMPU_d9:03:7c (f0:2f:74:d9:03:7c)
    Destination: ASUSTekCOMPU_d9:03:7c (f0:2f:74:d9:03:7c)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: SagemcomBroa_f9:b2:2c (4c:19:5d:f9:b2:2c)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
    [Stream index: 1]
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.2.110
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 481
    Identification: 0x3b9f (15263)
    010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 44
    Protocol: TCP (6)
    Header Checksum: 0xd8dd [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 128.119.245.12
    Destination Address: 192.168.2.110
    [Stream index: 25]
Transmission Control Protocol, Src Port: 80, Dst Port: 61930, Seq: 1, Ack: 518, Len: 441
    Source Port: 80
    Destination Port: 61930
    [Stream index: 66]
    [Stream Packet Number: 6]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 441]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 3198319493
    [Next Sequence Number: 442    (relative sequence number)]
    Acknowledgment Number: 518    (relative ack number)
    Acknowledgment number (raw): 3752183246
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x018 (PSH, ACK)
    Window: 501
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0xb7a2 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (441 bytes)
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Mon, 17 Nov 2025 16:29:33 GMT\r\n
    Server: Apache/2.4.62 (AlmaLinux) OpenSSL/3.2.2 mod_fcgid/2.3.9 mod_perl/2.0.12 Perl/v5.32.1\r\n
    Last-Modified: Tue, 28 Oct 2025 05:59:01 GMT\r\n
    ETag: "51-64231b6715777"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 81\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [Request in frame: 1219]
    [Time since request: 103.604000 milliseconds]
    [Request URI: /wireshark-labs/INTRO-wireshark-file1.html]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    File Data: 81 bytes
Line-based text data: text/html (3 lines)
```

Vuong Ho s0592666

/var/folders/nl/vq9qj8n16c1dcb8fdr31ssbh0000gn/T/wireshark_Wi-FiVLLBG3.pcapng 1836 total packets, 2 shown

```
No.     Time           Source                Destination           Protocol Length Info
    284 10.172840      172.20.10.6           128.119.245.12        HTTP     566    GET /
wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
Frame 284: Packet, 566 bytes on wire (4528 bits), 566 bytes captured (4528 bits) on interface
en0, id 0
Ethernet II, Src: f6:5e:ef:59:32:10 (f6:5e:ef:59:32:10), Dst: e2:33:8e:d1:48:64
(e2:33:8e:d1:48:64)
Internet Protocol Version 4, Src: 172.20.10.6, Dst: 128.119.245.12
Transmission Control Protocol, Src Port: 58505, Dst Port: 80, Seq: 1, Ack: 1, Len: 500
    Source Port: 58505
    Destination Port: 80
    [Stream index: 11]
    [Stream Packet Number: 4]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 500]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 637059181
    [Next Sequence Number: 501    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2308460095
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x018 (PSH, ACK)
    Window: 2054
    [Calculated window size: 131456]
    [Window size scaling factor: 64]
    Checksum: 0xfd3f [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (500 bytes)
Hypertext Transfer Protocol
    GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n
        Request Method: GET
        Request URI: /wireshark-labs/INTRO-wireshark-file1.html
        Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/142.0.0.0 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    \r\n
    [Response in frame: 344]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
No.     Time           Source                Destination           Protocol Length Info
    344 10.309810      128.119.245.12        172.20.10.6           HTTP     507    HTTP/1.1 200
OK  (text/html)
Frame 344: Packet, 507 bytes on wire (4056 bits), 507 bytes captured (4056 bits) on interface
en0, id 0
Ethernet II, Src: e2:33:8e:d1:48:64 (e2:33:8e:d1:48:64), Dst: f6:5e:ef:59:32:10
(f6:5e:ef:59:32:10)
Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.20.10.6
Transmission Control Protocol, Src Port: 80, Dst Port: 58505, Seq: 1, Ack: 501, Len: 441
    Source Port: 80
    Destination Port: 58505
    [Stream index: 11]
    [Stream Packet Number: 6]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 441]
```

/var/folders/nl/vq9qj8n16c1dcb8fdr31ssbh0000gn/T/wireshark_Wi-FiVLLBG3.pcapng 1836 total packets, 2 shown

```
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 2308460095
    [Next Sequence Number: 442    (relative sequence number)]
    Acknowledgment Number: 501    (relative ack number)
    Acknowledgment number (raw): 637059681
    1000 .... = Header Length: 32 bytes (8)
    Flags: 0x018 (PSH, ACK)
    Window: 506
    [Calculated window size: 64768]
    [Window size scaling factor: 128]
    Checksum: 0xddc1 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
    Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    [Timestamps]
    [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
    TCP payload (441 bytes)
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
        Response Version: HTTP/1.1
        Status Code: 200
        [Status Code Description: OK]
        Response Phrase: OK
    Date: Mon, 17 Nov 2025 20:11:49 GMT\r\n
    Server: Apache/2.4.62 (AlmaLinux) OpenSSL/3.2.2 mod_fcgid/2.3.9 mod_perl/2.0.12 Perl/
v5.32.1\r\n
    Last-Modified: Tue, 28 Oct 2025 05:59:01 GMT\r\n
    ETag: "51-64231b6715777"\r\n
    Accept-Ranges: bytes\r\n
    Content-Length: 81\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    [Request in frame: 284]
    [Time since request: 136.970000 milliseconds]
    [Request URI: /wireshark-labs/INTRO-wireshark-file1.html]
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]
    File Data: 81 bytes
Line-based text data: text/html (3 lines)
```