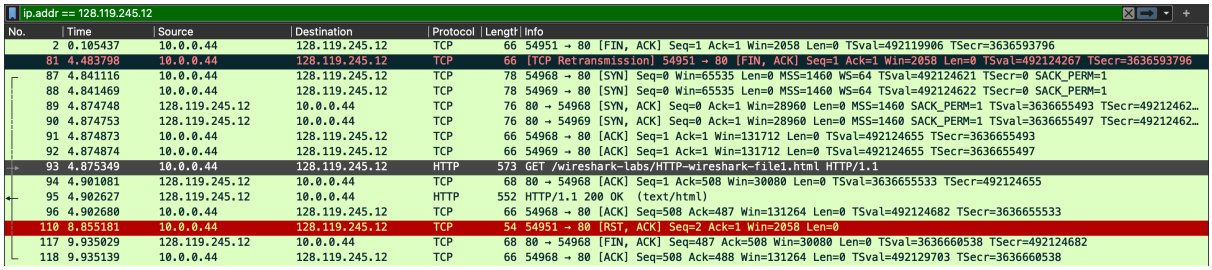
HW2 22000546 이예은

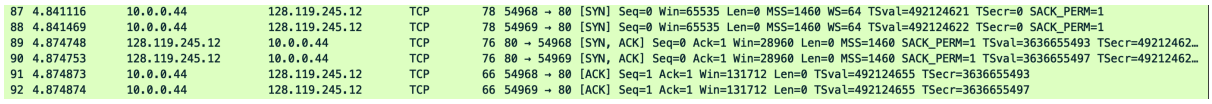
1.



No.	Time	Source	Destination	Protocol	Length	Info
2	0.105437	10.0.0.44	128.119.245.12	TCP	66	54951 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2058 Len=0 TSval=492119906 TSecr=3636593796
81	4.483798	10.0.0.44	128.119.245.12	TCP	66	[TCP Retransmission] 54951 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2058 Len=0 TSval=492124267 TSecr=3636593796
87	4.841116	10.0.0.44	128.119.245.12	TCP	78	54968 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=492124621 TSecr=0 SACK_PERM=1
88	4.841469	10.0.0.44	128.119.245.12	TCP	78	54969 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=492124622 TSecr=0 SACK_PERM=1
89	4.874748	128.119.245.12	10.0.0.44	TCP	76	80 → 54968 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3636655493 TSecr=49212462…
90	4.874753	128.119.245.12	10.0.0.44	TCP	76	80 → 54969 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3636655497 TSecr=49212462…
91	4.874873	10.0.0.44	128.119.245.12	TCP	66	54968 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=492124655 TSecr=3636655493
92	4.874874	10.0.0.44	128.119.245.12	TCP	66	54969 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=492124655 TSecr=3636655497
93	4.875349	10.0.0.44	128.119.245.12	HTTP	573	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
94	4.901081	128.119.245.12	10.0.0.44	TCP	68	80 → 54968 [ACK] Seq=1 Ack=508 Win=30080 Len=0 TSval=3636655533 TSecr=492124655
95	4.902627	128.119.245.12	10.0.0.44	HTTP	552	HTTP/1.1 200 OK (text/html)
96	4.902680	10.0.0.44	128.119.245.12	TCP	66	54968 → 80 [ACK] Seq=508 Ack=487 Win=131264 Len=0 TSval=492124682 TSecr=3636655533
110	8.855181	10.0.0.44	128.119.245.12	TCP	54	54951 → 80 [RST, ACK] Seq=2 Ack=1 Win=2058 Len=0
117	9.935029	128.119.245.12	10.0.0.44	TCP	68	80 → 54968 [FIN, ACK] Seq=487 Ack=508 Win=30080 Len=0 TSval=3636660538 TSecr=492124682
118	9.935139	10.0.0.44	128.119.245.12	TCP	66	54968 → 80 [ACK] Seq=508 Ack=488 Win=131264 Len=0 TSval=492129703 TSecr=3636660538

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html과 관련된 패킷만 보기 위해서는 display filtering을 이용하면 된다. Protocol이 http인 패킷들을 보면 해당 사이트의 IP 주소가 128.119.245.12라는 것을 알 수 있다. 이와 관련된 패킷을 보기 위하여 display filtering 부분에 ip.addr == 128.119.245.12를 적으면 위와 같은 결과를 확인할 수 있다.

2.

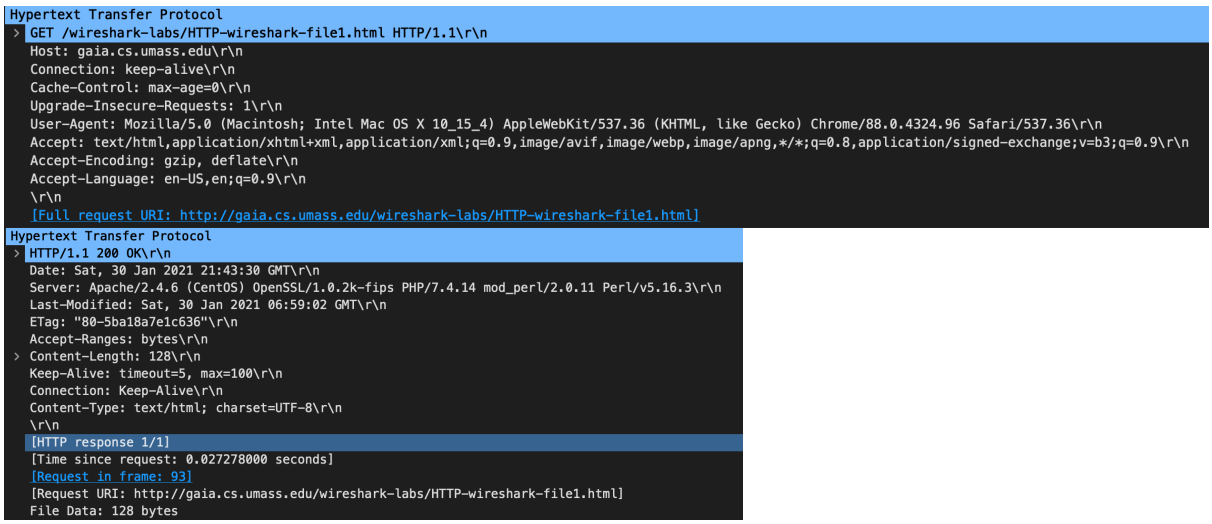


87	4.841116	10.0.0.44	128.119.245.12	TCP	78	54968 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=492124621 TSecr=0 SACK_PERM=1
88	4.841469	10.0.0.44	128.119.245.12	TCP	78	54969 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=492124622 TSecr=0 SACK_PERM=1
89	4.874748	128.119.245.12	10.0.0.44	TCP	76	80 → 54968 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3636655493 TSecr=49212462…
90	4.874753	128.119.245.12	10.0.0.44	TCP	76	80 → 54969 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=3636655497 TSecr=49212462…
91	4.874873	10.0.0.44	128.119.245.12	TCP	66	54968 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=492124655 TSecr=3636655493
92	4.874874	10.0.0.44	128.119.245.12	TCP	66	54969 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=492124655 TSecr=3636655497

87, 88, 89, 90, 91, 92 패킷들은 3-way handshaking 과정과 관련된 패킷들이다. 98, 88번 패킷은 클라이언트(10.0.0.44)로부터 서버(128.119.245.12)로 연결을 위한 first handshake인 SYN 패킷을 보냈다. 89, 90번 패킷은 서버의 ACK 응답과 SYN 연결을 클라이언트에게 보낸다. 이것이 second handshake이고, 91, 92번 패킷은 클라이언트가 서버로 ACK 패킷을 보내는 third handshake 과정이다. 87, 89, 91번 패킷은 source port가 54968번, 88, 90, 92번 패킷은 54969번인데 GET에 사용된 포트는 54968번이다. 따라서 GET을 위한 connection establishment에 사용된 패킷은 87, 89, 91번 패킷들이다.

```
> Transmission Control Protocol, Src Port: 54968, Dst Port: 80, Seq: 1, Ack: 1, Len: 507
v Hypertext Transfer Protocol
  > GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
```

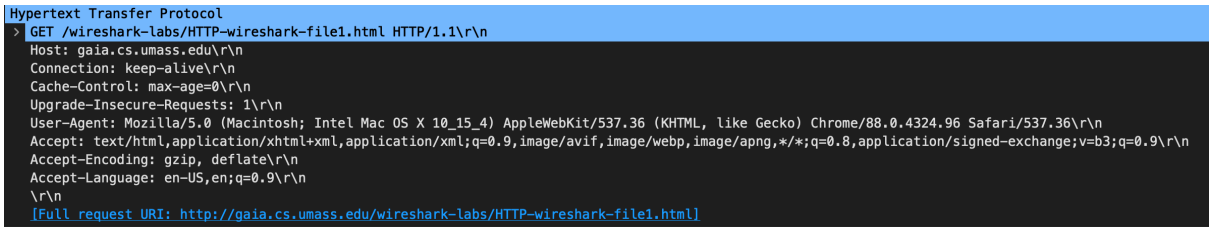
3.



```
Hypertext Transfer Protocol
> GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
  Host: gaia.cs.umass.edu\r\n
  Connection: keep-alive\r\n
  Cache-Control: max-age=0\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 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=0.9\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9\r\n
  \r\n
  [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
  Date: Sat, 30 Jan 2021 21:43:30 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3\r\n
  Last-Modified: Sat, 30 Jan 2021 06:59:02 GMT\r\n
  ETag: "80-5ba18a7e1c636"\r\n
  Accept-Ranges: bytes\r\n
> Content-Length: 128\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
  [HTTP response 1/1]
  [Time since request: 0.027278000 seconds]
  [Request in frame: 93]
  [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
  File Data: 128 bytes
```

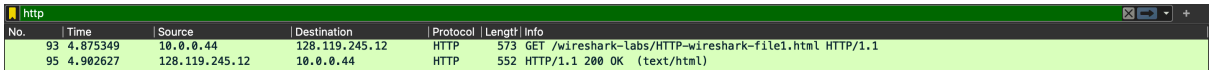
HTTP와 관련된 패킷에서 HTTP request message와 HTTP response message를 보면 클라이언트의 브라우저와 서버 모두 HTTP 1.1을 쓴다는 것을 알 수 있다.

4.



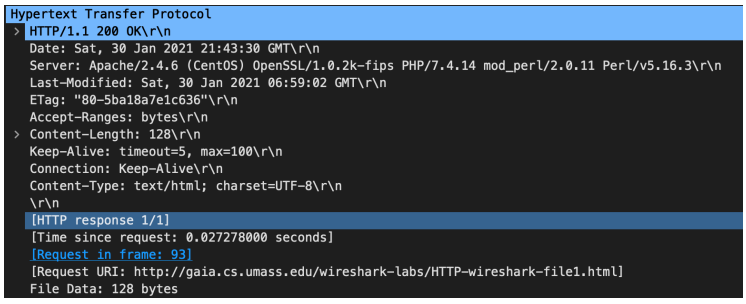
HTTP request message에서 Accept-Language를 보면 en-US, en으로 나와있으므로 영어(en-US, en)를 원하고 있음을 나타낸다.

5.



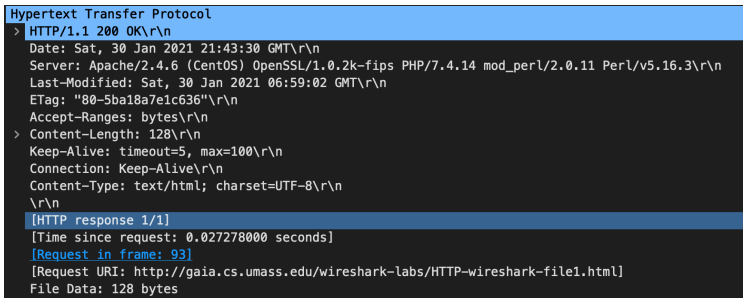
http로 필터링한 결과인데, 이를 보면 클라이언트의 IP 주소는 10.0.0.44, 서버의 IP 주소는 128.119.245.12라는 것을 알 수 있다.

6.



HTTP response message의 status line을 보면 200 OK이다.

7.



HTTP response message의 Last-Modified를 보면 2021년 1월 30일 토요일 6시 59분 02초이다. (Sat, 30 Jan 2021 06:59:02 GMT)

8.

```
Hypertext Transfer Protocol
> HTTP/1.1 200 OK\r\n
  Date: Sat, 30 Jan 2021 21:43:30 GMT\r\n
  Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3\r\n
  Last-Modified: Sat, 30 Jan 2021 06:59:02 GMT\r\n
  ETag: "80-5ba18a7e1c636"\r\n
  Accept-Ranges: bytes\r\n
> Content-Length: 128\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
  [HTTP response 1/1]
  [Time since request: 0.027278000 seconds]
  [Request in frame: 93]
  [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html]
  File Data: 128 bytes
```

HTTP response message의 Content-Length를 보면 128 bytes라는 것을 알 수 있다.

9.

```
http
No.   Time        Source          Destination     Protocol  Length  Info
    56 0.309961    10.0.0.44       128.119.245.12  HTTP      547 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
    58 0.334570    128.119.245.12  10.0.0.44       HTTP      796 HTTP/1.1 200 OK  (text/html)
   555 3.481656    10.0.0.44       128.119.245.12  HTTP      659 GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
   556 3.504286    128.119.245.12  10.0.0.44       HTTP      305 HTTP/1.1 304 Not Modified
```

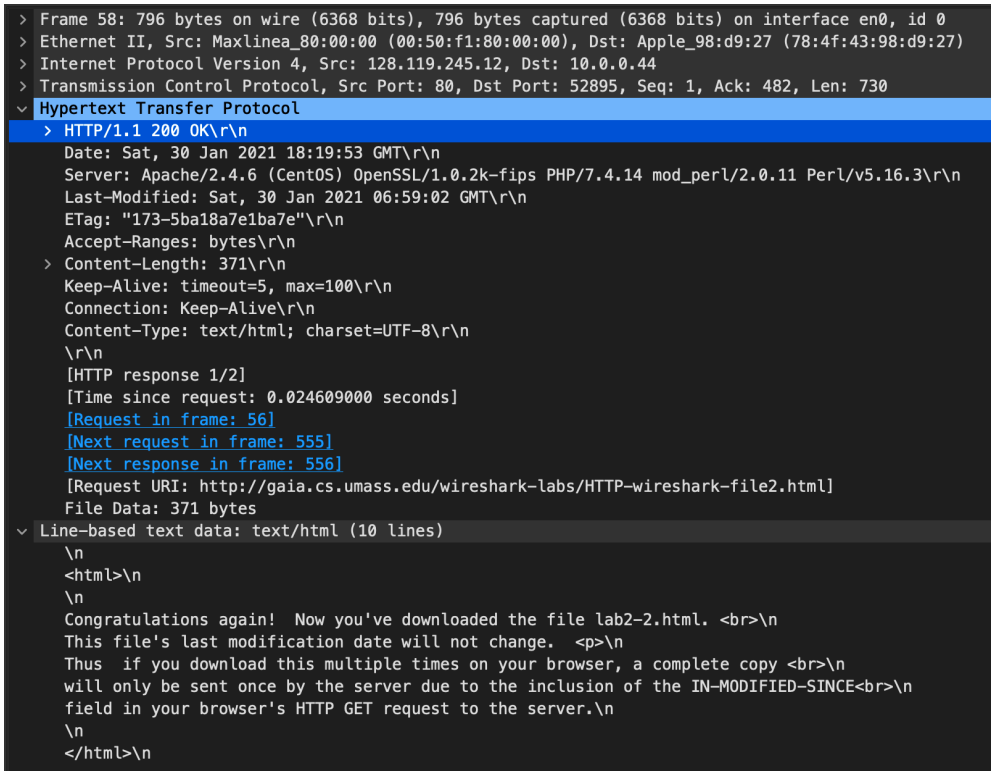
Display filtering 부분에 http를 적고 결과를 확인하면 위와 같다.

10.

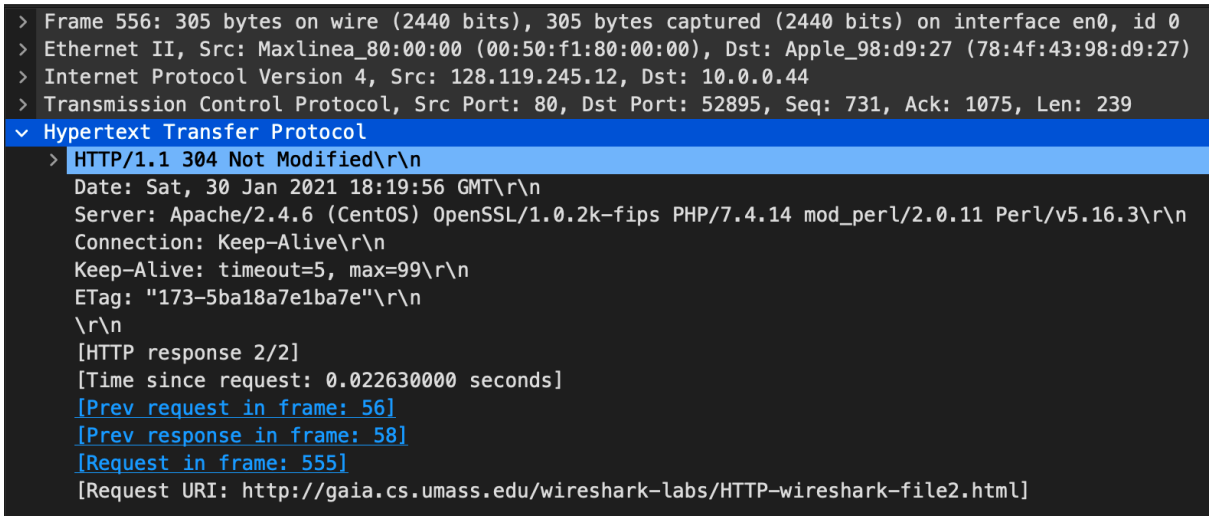
```
> Frame 56: 547 bytes on wire (4376 bits), 547 bytes captured (4376 bits) on interface en0, id 0
> Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Maxlinea_80:00:00 (00:50:f1:80:00:00)
> Internet Protocol Version 4, Src: 10.0.0.44, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 52895, Dst Port: 80, Seq: 1, Ack: 1, Len: 481
v Hypertext Transfer Protocol
  v GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /wireshark-labs/HTTP-wireshark-file2.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_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 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=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 1/2]
> Frame 555: 659 bytes on wire (5272 bits), 659 bytes captured (5272 bits) on interface en0, id 0
> Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: Maxlinea_80:00:00 (00:50:f1:80:00:00)
> Internet Protocol Version 4, Src: 10.0.0.44, Dst: 128.119.245.12
> Transmission Control Protocol, Src Port: 52895, Dst Port: 80, Seq: 482, Ack: 731, Len: 593
v Hypertext Transfer Protocol
  v GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /wireshark-labs/HTTP-wireshark-file2.html
      Request Version: HTTP/1.1
    Host: gaia.cs.umass.edu\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.96 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=0.9\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    If-None-Match: "173-5ba18a7e1ba7e"\r\n
    If-Modified-Since: Sat, 30 Jan 2021 06:59:02 GMT\r\n
```

56번 패킷에는 IF-MODIFIED-SINCE가 없지만, 555번 패킷에는 IF-MODIFIED-SINCE가 'Sat, 30 Jan 2021 06:59:02 GMT'로 있다.

11.

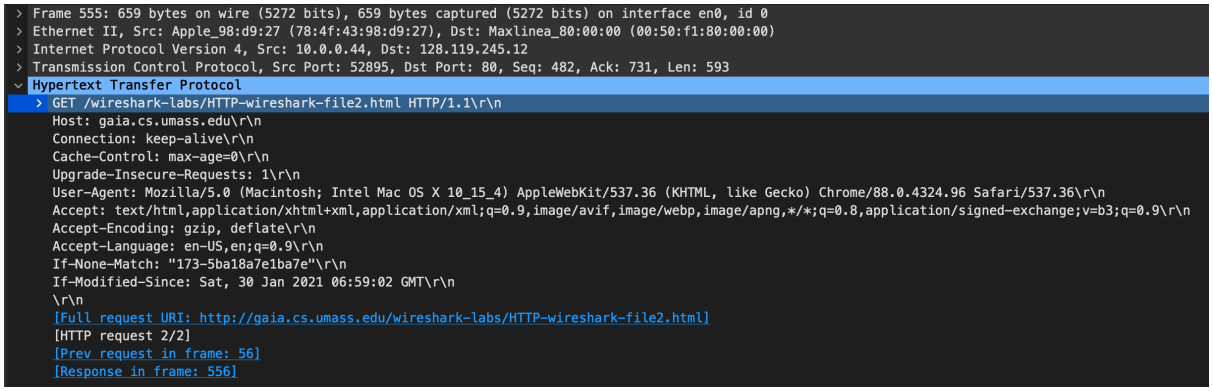


58번 패킷은 아래의 Line-based text data로 보내온 데이터를 확인할 수 있으므로 서버는 explicitly하게 파일의 컨텐츠를 return했다. Last-Modified 헤더에는 'Sat, 30 Jan 2021 06:59:02 GMT'이 있어 객체가 생성되거나 마지막으로 수정된 시간을 알려준다. ETag 헤더에는 '173-5ba18a7e1ba7e' 정보가 담겨 특정한 버전의 리소스를 식별하도록 한다.



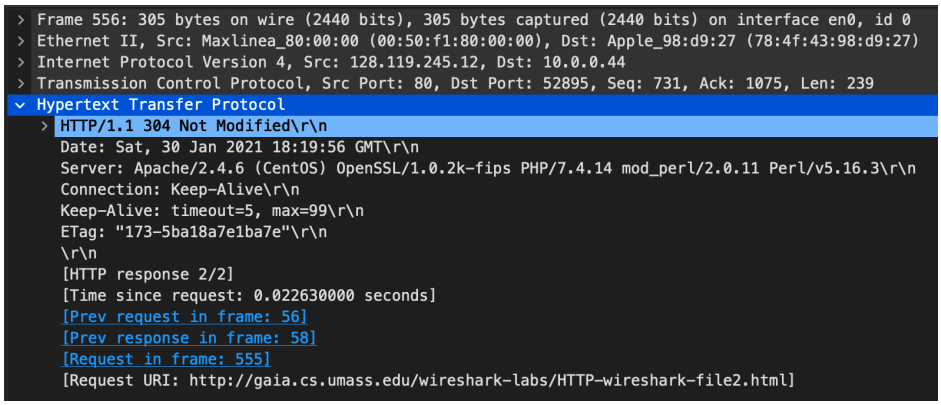
556번 패킷에는 explicitly returned contents가 없으며, Last-Modified 헤더도 없다. ETag에는 '173-5ba18a7e1ba7e'가 따라온다.

12.



IF-MODIFIED-SINCE에는 'Sat, 30 Jan 2021 06:59:02 GMT'가 담겨있고 If-None-Match에는 '173-5ba18a7e1ba7e'가 담겨있다. 이것은 ETag인데, 해당하는 ETag와 같지 않다면 요청을 처리하고, 같다면 304 Not Modified를 반환한다. 두 헤더 모두 캐시를 이용할 때 데이터가 바뀌었는지를 확인할 때 유용하다.

13.



556번 패킷은 304 Not Modified를 반환하였고, contents of file을 explicitly return하지 않았다. 그 이유는 이전의 58번 패킷에서 수정된 것이 없기 때문에 (ETag가 같음) 캐시된 것을 사용하면 되기 때문이다.

14.

HTTP/1.1에서 pipelining을 지원하게 되었지만 요청 순서대로 응답해야 하는 FCFS(First-Come-First-Served scheduling)에 의해 처음에 오래 걸리는 요청이 오게 되면 이후의 요청에 대한 응답이 길어지게 되는 HOL blocking이 발생하게 되었다. HTTP/2에서는 multiplexing으로 이를 해결하고자 하였는데, 하나의 TCP 연결에서 여러 메세지를 처리할 수 있게 되었고, 응답이 프레임으로 나누어지고 여러 프레임이 순서에 상관없이 stream 단위로 전송될 수 있게 되어 속도를 높였다.