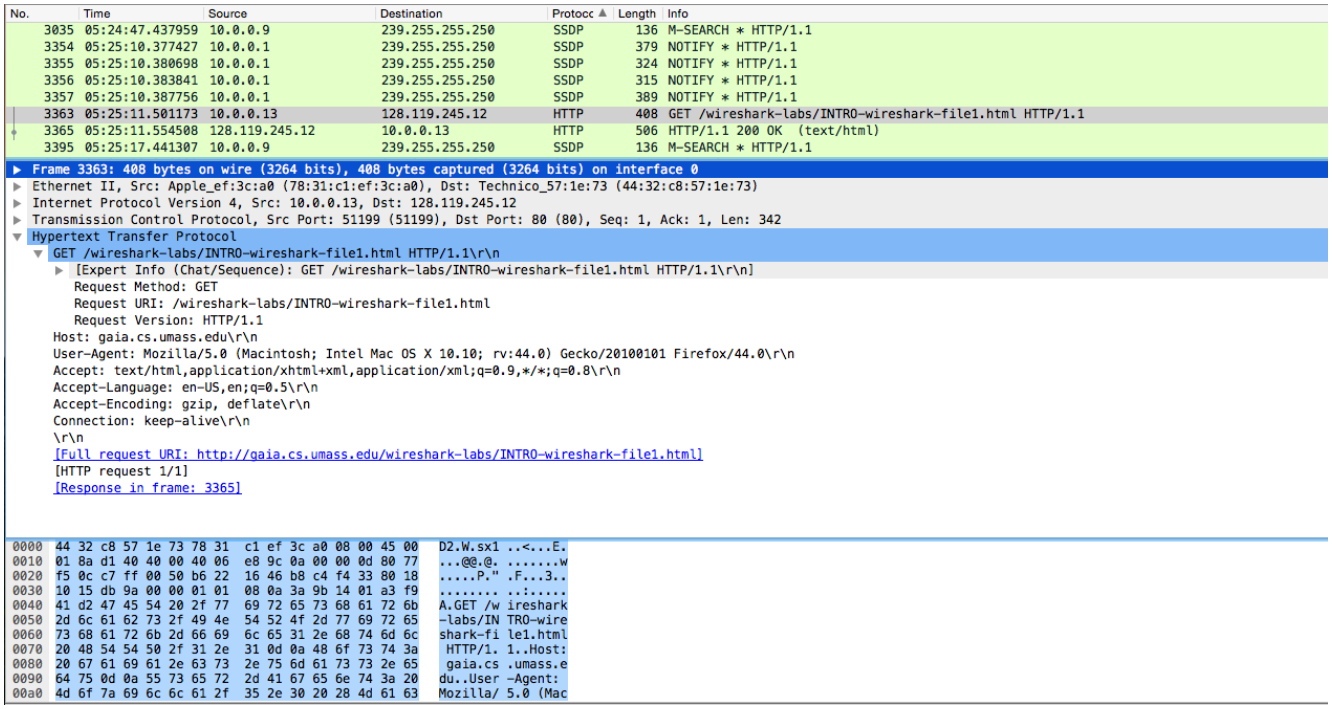
Wireshark Lab

lukas_borges

Part I

Capturing http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html packet



1. 3 protocols: Hyper Text Transfer Protocol (HTTP), Online Certificate Status Protocol (OCSP) and Simple Service Discovery Protocol (SSDP)
2. $.554508 - .501173 = .53335$ seconds
3. gaia.cs.umass.edu IP: `128.119.245.12`
   my IP: `10.0.0.13`

Part II

1. The basic HTTP GET/response interaction

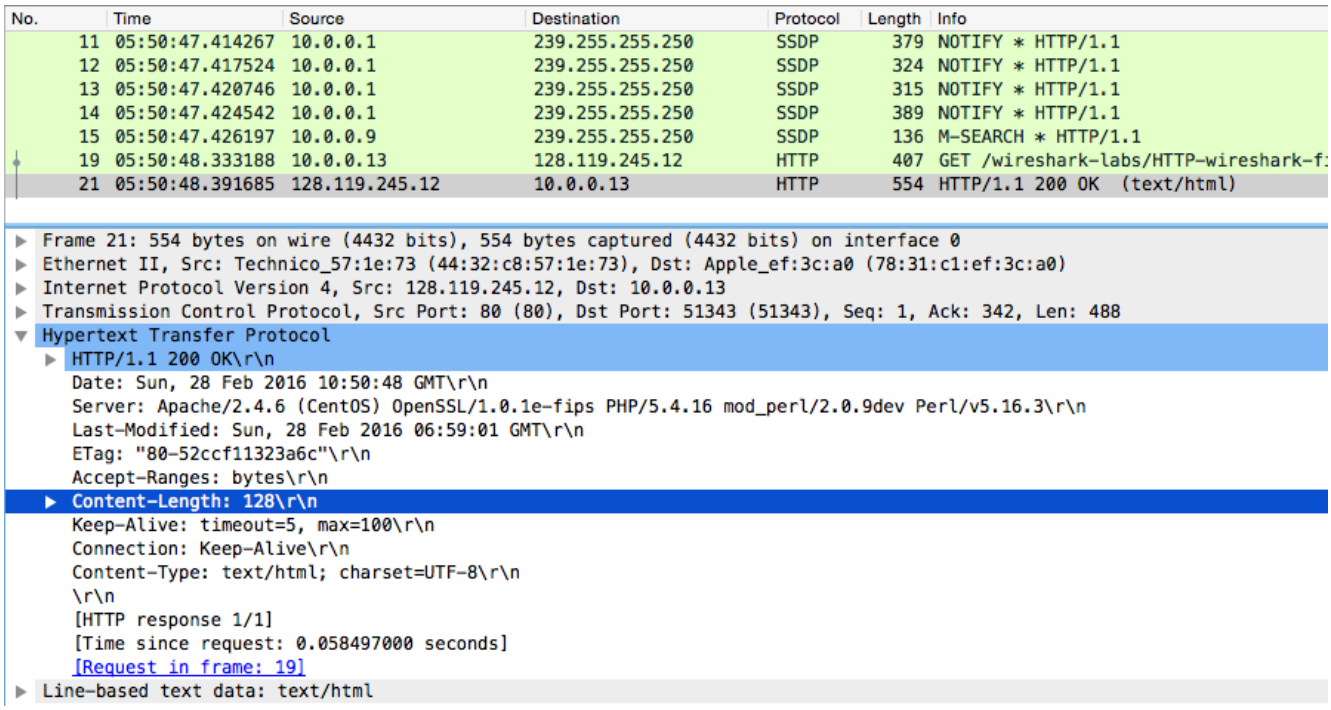
1. HTTP 1.1
2. No language specification (Accept-language or ‘en’) on header

3. My browser: `10.0.0.3`

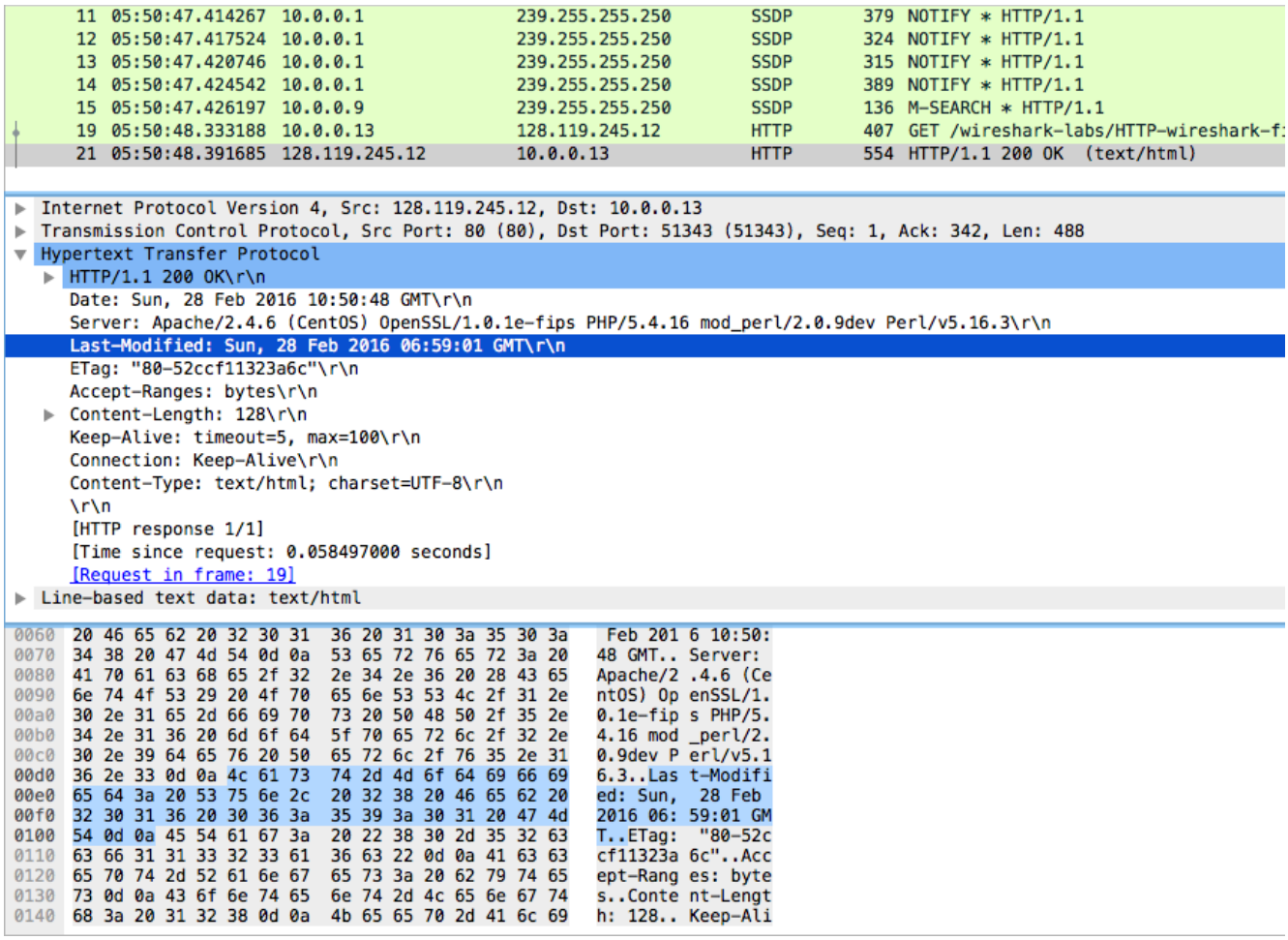
   Gaia: `128.119.245.12`

4. `200 OK`
5. Last-Modified: `Sun, 28 Feb 2016 06:59:01 GMT`
6. Content-length: `128 bytes`
7. No extra headers

Content-Length:

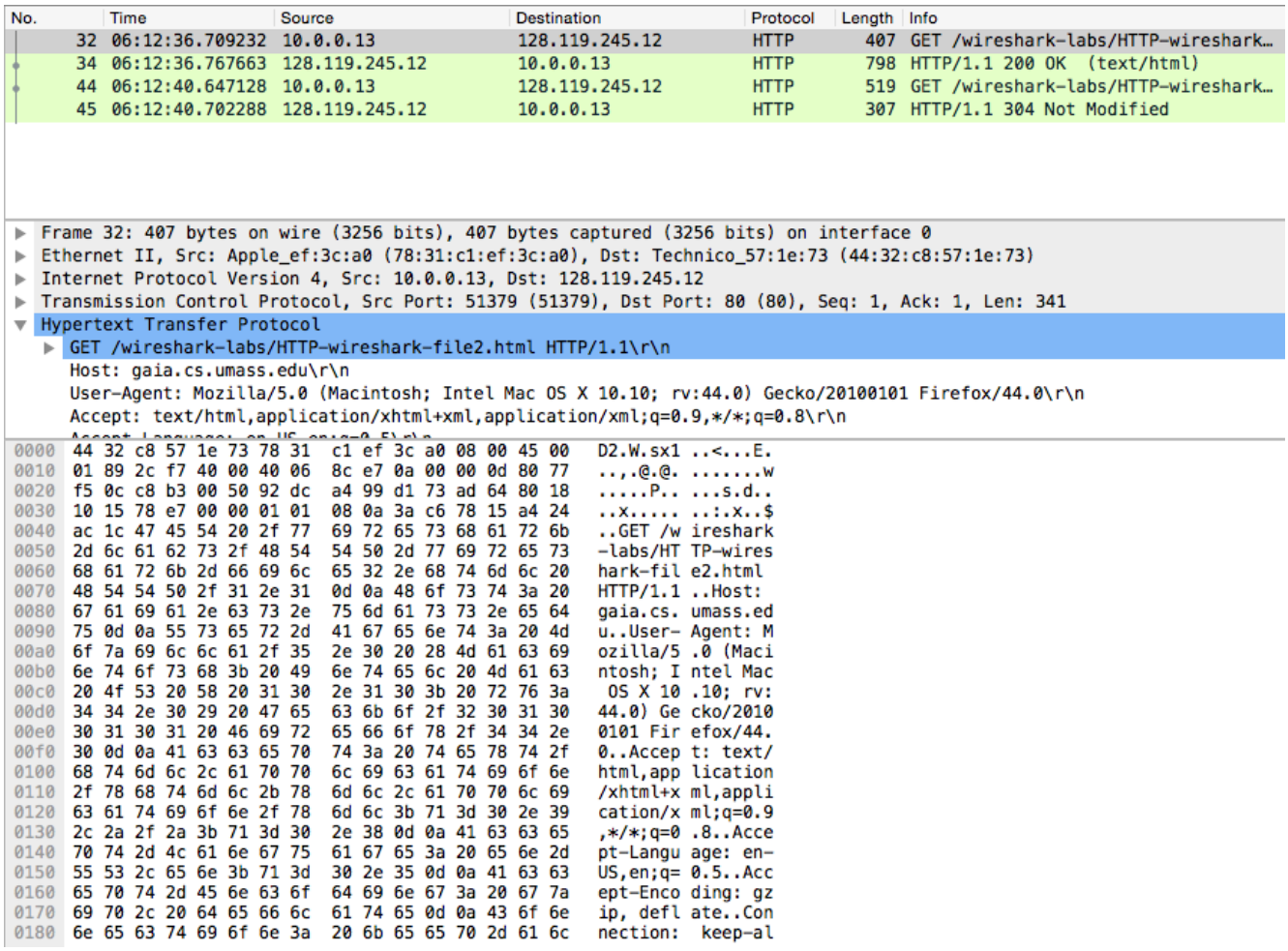


Last-modified:



2. HTTP Conditional GET/response interaction:

8. No `if-modified-since`:

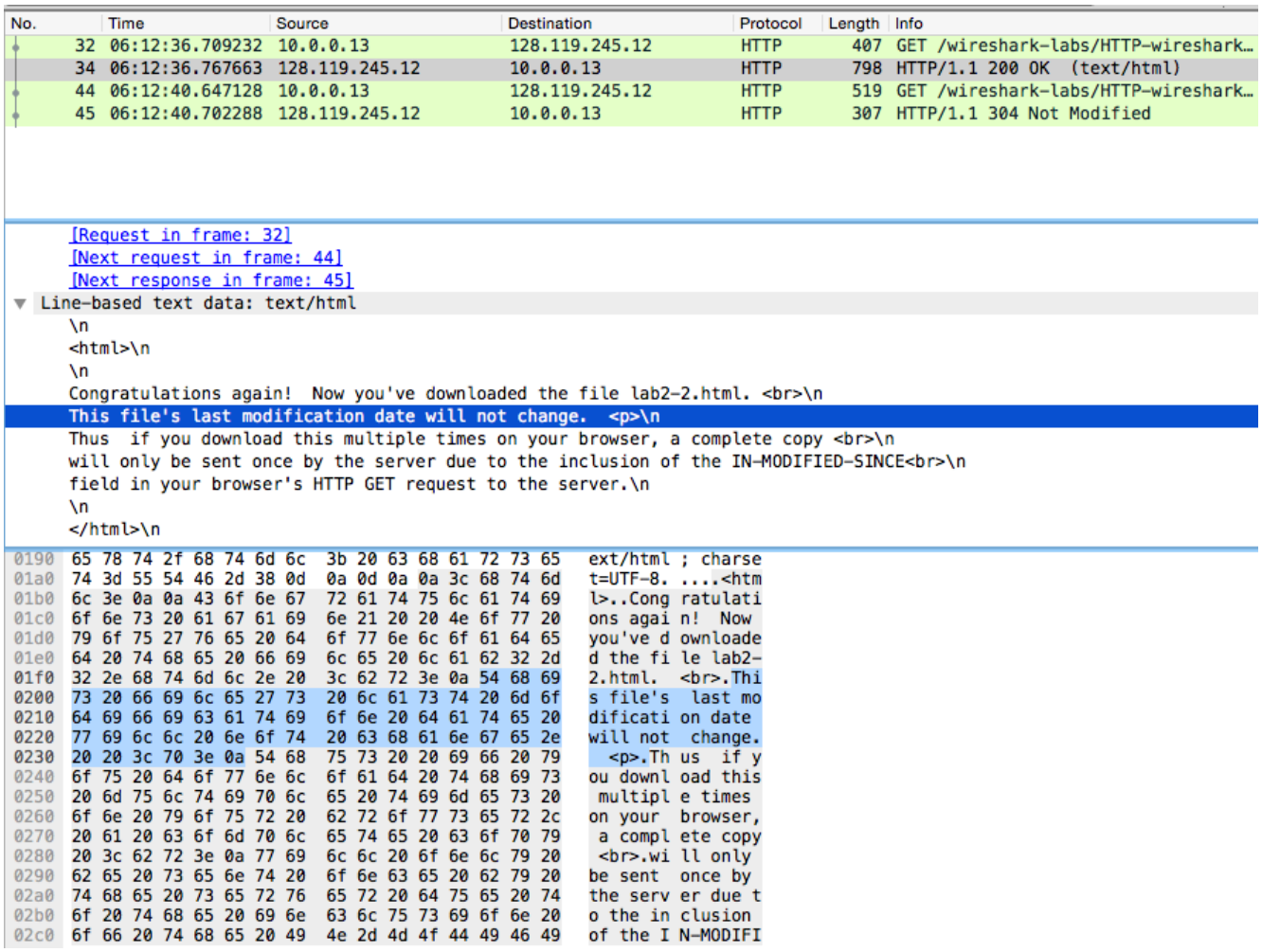


No.	Time	Source	Destination	Protocol	Length	Info
32	06:12:36.709232	10.0.0.13	128.119.245.12	HTTP	407	GET /wireshark-labs/HTTP-wireshark…
34	06:12:36.767663	128.119.245.12	10.0.0.13	HTTP	798	HTTP/1.1 200 OK (text/html)
44	06:12:40.647128	10.0.0.13	128.119.245.12	HTTP	519	GET /wireshark-labs/HTTP-wireshark…
45	06:12:40.702288	128.119.245.12	10.0.0.13	HTTP	307	HTTP/1.1 304 Not Modified

```
▶ Frame 32: 407 bytes on wire (3256 bits), 407 bytes captured (3256 bits) on interface 0
▶ Ethernet II, Src: Apple_ef:3c:a0 (78:31:c1:ef:3c:a0), Dst: Technico_57:1e:73 (44:32:c8:57:1e:73)
▶ Internet Protocol Version 4, Src: 10.0.0.13, Dst: 128.119.245.12
▶ Transmission Control Protocol, Src Port: 51379 (51379), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 341
▼ Hypertext Transfer Protocol
  ▶ GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:44.0) Gecko/20100101 Firefox/44.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
```

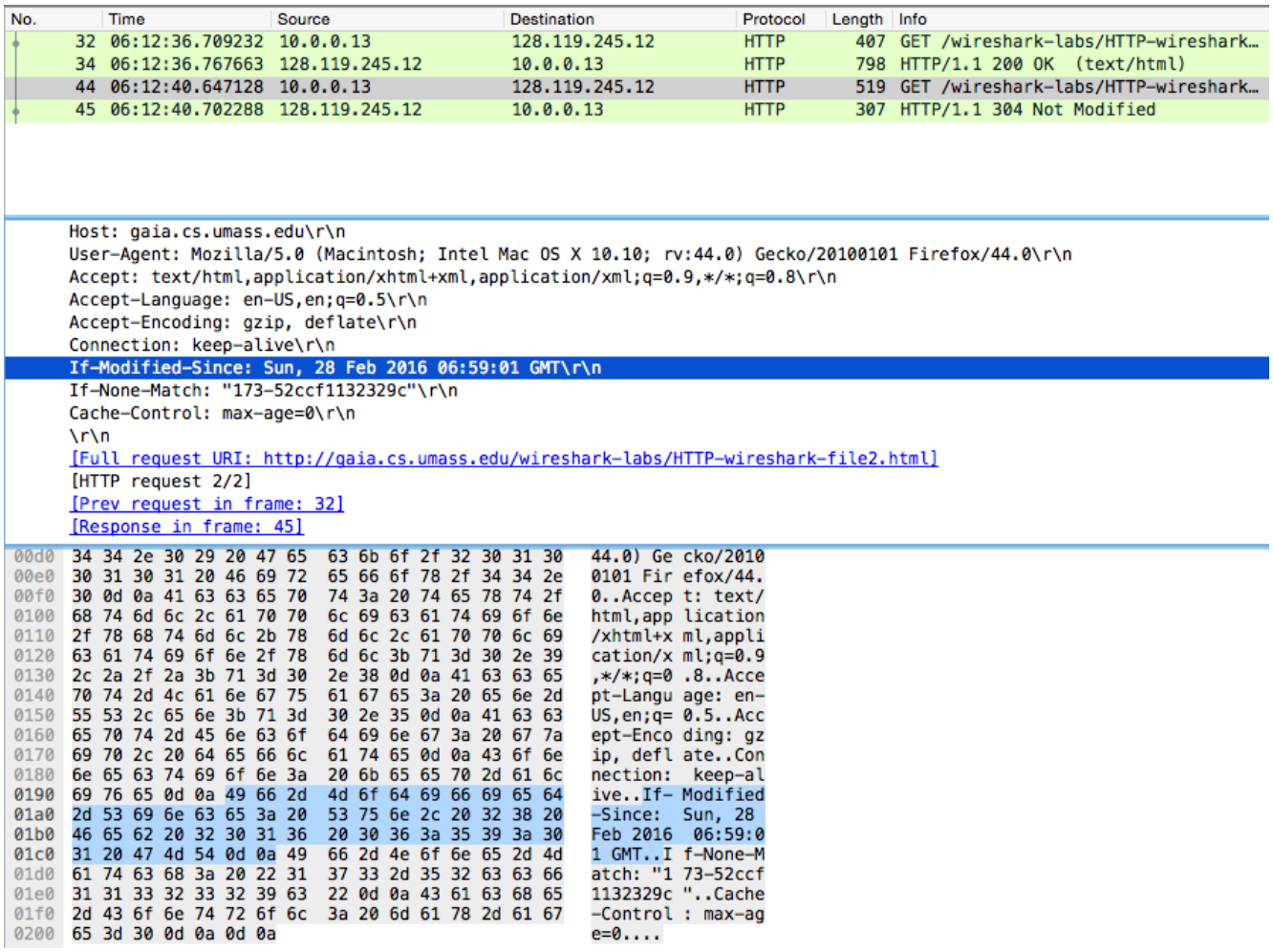
```
0000  44 32 c8 57 1e 73 78 31  c1 ef 3c a0 08 00 45 00   D2.W.sx1 ..<...E.
0010  01 89 2c f7 40 00 40 06  8c e7 0a 00 00 0d 80 77   ..,.@.@. .......w
0020  f5 0c c8 b3 00 50 92 dc  a4 99 d1 73 ad 64 80 18   .....P.. ...s.d..
0030  10 15 78 e7 00 00 01 01  08 0a 3a c6 78 15 a4 24   ..x..... ..:.x..$
0040  ac 1c 47 45 54 20 2f 77  69 72 65 73 68 61 72 6b   ..GET /w ireshark
0050  2d 6c 61 62 73 2f 48 54  54 50 2d 77 69 72 65 73   -labs/HT TP-wires
0060  68 61 72 6b 2d 66 69 6c  65 32 2e 68 74 6d 6c 20   hark-fil e2.html
0070  48 54 54 50 2f 31 2e 31  0d 0a 48 6f 73 74 3a 20   HTTP/1.1 ..Host:
0080  67 61 69 61 2e 63 73 2e  75 6d 61 73 73 2e 65 64   gaia.cs. umass.ed
0090  75 0d 0a 55 73 65 72 2d  41 67 65 6e 74 3a 20 4d   u..User- Agent: M
00a0  6f 7a 69 6c 6c 61 2f 35  2e 30 20 28 4d 61 63 69   ozilla/5 .0 (Maci
00b0  6e 74 6f 73 68 3b 20 49  6e 74 65 6c 20 4d 61 63   ntosh; I ntel Mac
00c0  20 4f 53 20 58 20 31 30  2e 31 30 3b 20 72 76 3a    OS X 10 .10; rv:
00d0  34 34 2e 30 29 20 47 65  63 6b 6f 2f 32 30 31 30   44.0) Ge cko/2010
00e0  30 31 30 31 20 46 69 72  65 66 6f 78 2f 34 34 2e   0101 Fir efox/44.
00f0  30 0d 0a 41 63 63 65 70  74 3a 20 74 65 78 74 2f   0..Accep t: text/
0100  68 74 6d 6c 2c 61 70 70  6c 69 63 61 74 69 6f 6e   html,app lication
0110  2f 78 68 74 6d 6c 2b 78  6d 6c 2c 61 70 70 6c 69   /xhtml+x ml,appli
0120  63 61 74 69 6f 6e 2f 78  6d 6c 3b 71 3d 30 2e 39   cation/x ml;q=0.9
0130  2c 2a 2f 2a 3b 71 3d 30  2e 38 0d 0a 41 63 63 65   ,*/*;q=0 .8..Acce
0140  70 74 2d 4c 61 6e 67 75  61 67 65 3a 20 65 6e 2d   pt-Langu age: en-
0150  55 53 2c 65 6e 3b 71 3d  30 2e 35 0d 0a 41 63 63   US,en;q= 0.5..Acc
0160  65 70 74 2d 45 6e 63 6f  64 69 6e 67 3a 20 67 7a   ept-Enco ding: gz
0170  69 70 2c 20 64 65 66 6c  61 74 65 0d 0a 43 6f 6e   ip, defl ate..Con
0180  6e 65 63 74 69 6f 6e 3a  20 6b 65 65 70 2d 61 6c   nection:  keep-al
```

9. Yes, the HTML code displayed on browser is present:



No.	Time	Source	Destination	Protocol	Length	Info
32	06:12:36.709232	10.0.0.13	128.119.245.12	HTTP	407	GET /wireshark-labs/HTTP-wireshark…
34	06:12:36.767663	128.119.245.12	10.0.0.13	HTTP	798	HTTP/1.1 200 OK (text/html)
44	06:12:40.647128	10.0.0.13	128.119.245.12	HTTP	519	GET /wireshark-labs/HTTP-wireshark…
45	06:12:40.702288	128.119.245.12	10.0.0.13	HTTP	307	HTTP/1.1 304 Not Modified

10. Yes: `Sun, 28 Feb 2016 06:59:01 GMT`

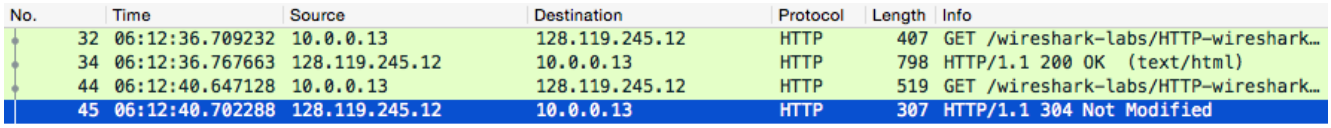


No.	Time	Source	Destination	Protocol	Length	Info
32	06:12:36.709232	10.0.0.13	128.119.245.12	HTTP	407	GET /wireshark-labs/HTTP-wireshark…
34	06:12:36.767663	128.119.245.12	10.0.0.13	HTTP	798	HTTP/1.1 200 OK (text/html)
44	06:12:40.647128	10.0.0.13	128.119.245.12	HTTP	519	GET /wireshark-labs/HTTP-wireshark…
45	06:12:40.702288	128.119.245.12	10.0.0.13	HTTP	307	HTTP/1.1 304 Not Modified

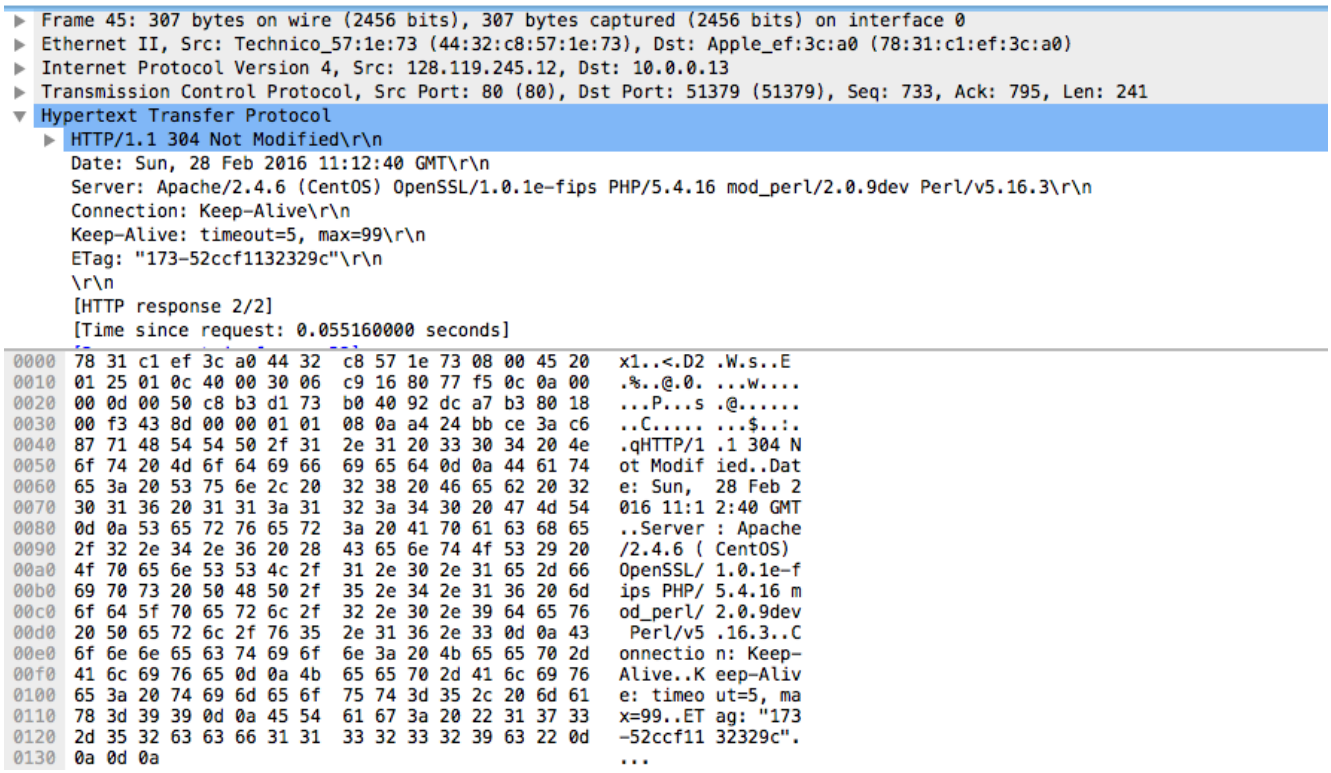
```
Host: gaia.cs.umass.edu\r\n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:44.0) Gecko/20100101 Firefox/44.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
If-Modified-Since: Sun, 28 Feb 2016 06:59:01 GMT\r\n
If-None-Match: "173-52ccf1132329c"\r\n
Cache-Control: max-age=0\r\n
\r\n
[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
[HTTP request 2/2]
[Prev request in frame: 32]
[Response in frame: 45]
```

```
00d0  34 34 2e 30 29 20 47 65  63 6b 6f 2f 32 30 31 30   44.0) Ge cko/2010
00e0  30 31 30 31 20 46 69 72  65 66 6f 78 2f 34 34 2e   0101 Fir efox/44.
00f0  30 0d 0a 41 63 63 65 70  74 3a 20 74 65 78 74 2f   0..Accep t: text/
0100  68 74 6d 6c 2c 61 70 70  6c 69 63 61 74 69 6f 6e   html,app lication
0110  2f 78 68 74 6d 6c 2b 78  6d 6c 2c 61 70 70 6c 69   /xhtml+x ml,appli
0120  63 61 74 69 6f 6e 2f 78  6d 6c 3b 71 3d 30 2e 39   cation/x ml;q=0.9
0130  2c 2a 2f 2a 3b 71 3d 30  2e 38 0d 0a 41 63 63 65   ,*/*;q=0 .8..Acce
0140  70 74 2d 4c 61 6e 67 75  61 67 65 3a 20 65 6e 2d   pt-Langu age: en-
0150  55 53 2c 65 6e 3b 71 3d  30 2e 35 0d 0a 41 63 63   US,en;q= 0.5..Acc
0160  65 70 74 2d 45 6e 63 6f  64 69 6e 67 3a 20 67 7a   ept-Enco ding: gz
0170  69 70 2c 20 64 65 66 6c  61 74 65 0d 0a 43 6f 6e   ip, defl ate..Con
0180  6e 65 63 74 69 6f 6e 3a  20 6b 65 65 70 2d 61 6c   nection:  keep-al
0190  69 76 65 0d 0a 49 66 2d  4d 6f 64 69 66 69 65 64   ive..If- Modified
01a0  2d 53 69 6e 63 65 3a 20  53 75 6e 2c 20 32 38 20   -Since:  Sun, 28
01b0  46 65 62 20 32 30 31 36  20 30 36 3a 35 39 3a 30   Feb 2016  06:59:0
01c0  31 20 47 4d 54 0d 0a 49  66 2d 4e 6f 6e 65 2d 4d   1 GMT..I f-None-M
01d0  61 74 63 68 3a 20 22 31  37 33 2d 35 32 63 63 66   atch: "1 73-52ccf
01e0  31 31 33 32 33 32 39 63  22 0d 0a 43 61 63 68 65   1132329c "..Cache
01f0  2d 43 6f 6e 74 72 6f 6c  3a 20 6d 61 78 2d 61 67   -Control : max-ag
0200  65 3d 30 0d 0a 0d 0a                              e=0....
```

11. The response was a `304 Not Modified`



No.	Time	Source	Destination	Protocol	Length	Info
32	06:12:36.709232	10.0.0.13	128.119.245.12	HTTP	407	GET /wireshark-labs/HTTP-wireshark…
34	06:12:36.767663	128.119.245.12	10.0.0.13	HTTP	798	HTTP/1.1 200 OK (text/html)
44	06:12:40.647128	10.0.0.13	128.119.245.12	HTTP	519	GET /wireshark-labs/HTTP-wireshark…
45	06:12:40.702288	128.119.245.12	10.0.0.13	HTTP	307	HTTP/1.1 304 Not Modified



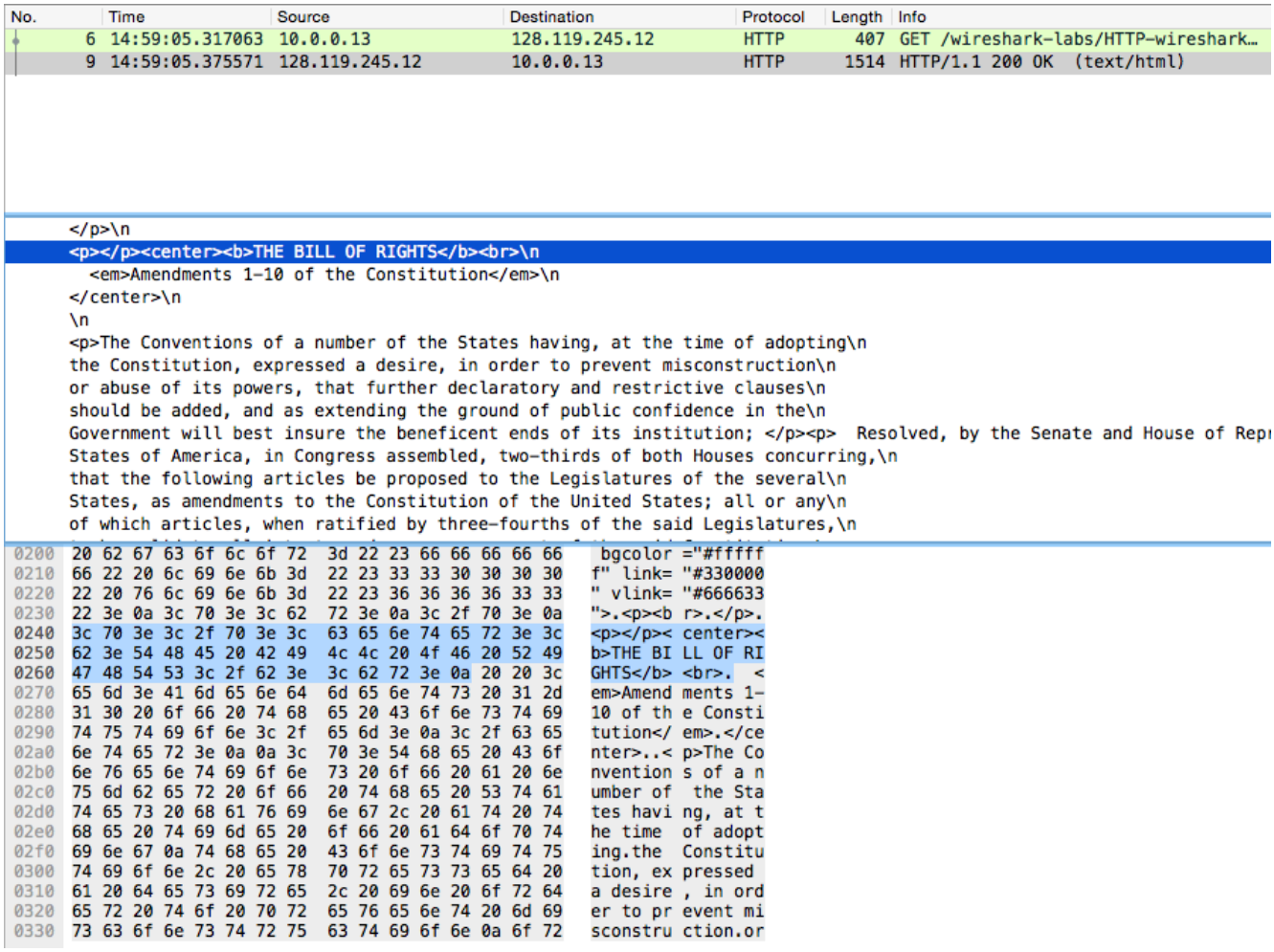
```
▶ Frame 45: 307 bytes on wire (2456 bits), 307 bytes captured (2456 bits) on interface 0
▶ Ethernet II, Src: Technico_57:1e:73 (44:32:c8:57:1e:73), Dst: Apple_ef:3c:a0 (78:31:c1:ef:3c:a0)
▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.13
▶ Transmission Control Protocol, Src Port: 80 (80), Dst Port: 51379 (51379), Seq: 733, Ack: 795, Len: 241
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 304 Not Modified\r\n
    Date: Sun, 28 Feb 2016 11:12:40 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips PHP/5.4.16 mod_perl/2.0.9dev Perl/v5.16.3\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=5, max=99\r\n
    ETag: "173-52ccf1132329c"\r\n
    \r\n
    [HTTP response 2/2]
    [Time since request: 0.055160000 seconds]
```

```
0000  78 31 c1 ef 3c a0 44 32  c8 57 1e 73 08 00 45 20   x1..<.D2 .W.s..E
0010  01 25 01 0c 40 00 30 06  c9 16 80 77 f5 0c 0a 00   .%..@.0. ...w....
0020  00 0d 00 50 c8 b3 d1 73  b0 40 92 dc a7 b3 80 18   ...P...s .@......
0030  00 f3 43 8d 00 00 01 01  08 0a a4 24 bb ce 3a c6   ..C..... ...$..:.
0040  87 71 48 54 54 50 2f 31  2e 31 20 33 30 34 20 4e   .qHTTP/1 .1 304 N
0050  6f 74 20 4d 6f 64 69 66  69 65 64 0d 0a 44 61 74   ot Modif ied..Dat
0060  65 3a 20 53 75 6e 2c 20  32 38 20 46 65 62 20 32   e: Sun,  28 Feb 2
0070  30 31 36 20 31 31 3a 31  32 3a 34 30 20 47 4d 54   016 11:1 2:40 GMT
0080  0d 0a 53 65 72 76 65 72  3a 20 41 70 61 63 68 65   ..Server : Apache
0090  2f 32 2e 34 2e 36 20 28  43 65 6e 74 4f 53 29 20   /2.4.6 ( CentOS)
00a0  4f 70 65 6e 53 53 4c 2f  31 2e 30 2e 31 65 2d 66   OpenSSL/ 1.0.1e-f
00b0  69 70 73 20 50 48 50 2f  35 2e 34 2e 31 36 20 6d   ips PHP/ 5.4.16 m
00c0  6f 64 5f 70 65 72 6c 2f  32 2e 30 2e 39 64 65 76   od_perl/ 2.0.9dev
00d0  20 50 65 72 6c 2f 76 35  2e 31 36 2e 33 0d 0a 43    Perl/v5 .16.3..C
00e0  6f 6e 6e 65 63 74 69 6f  6e 3a 20 4b 65 65 70 2d   onnectio n: Keep-
00f0  41 6c 69 76 65 0d 0a 4b  65 65 70 2d 41 6c 69 76   Alive..K eep-Aliv
0100  65 3a 20 74 69 6d 65 6f  75 74 3d 35 2c 20 6d 61   e: timeo ut=5, ma
0110  78 3d 39 39 0d 0a 45 54  61 67 3a 20 22 31 37 33   x=99..ET ag: "173
0120  2d 35 32 63 63 66 31 31  33 32 33 32 39 63 22 0d   -52ccf11 32329c".
0130  0a 0d 0a                                          ...
```

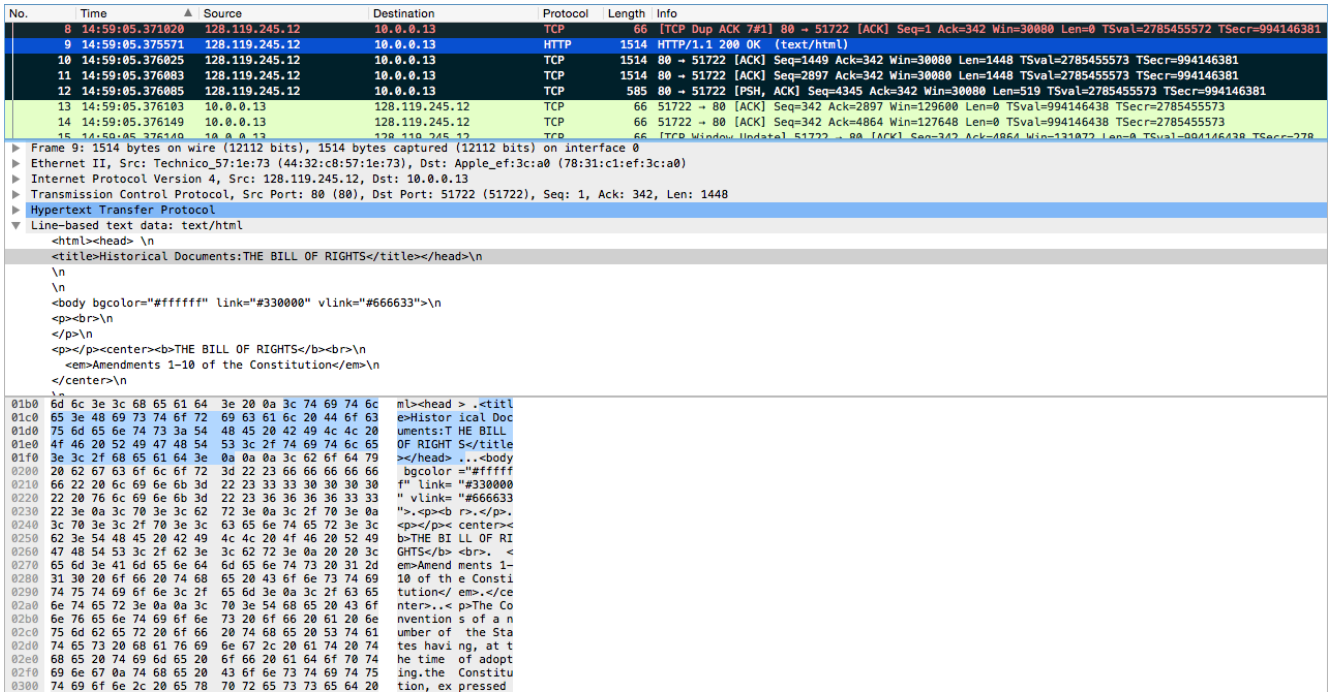
This happened because the file inside the server remains unchanged in comparison to the file cached. There is no need for of another costly round-trip-time to send me something I already have.

3. Retrieving Long Documents

12. One `HTTP GET`
13. Packet number 9



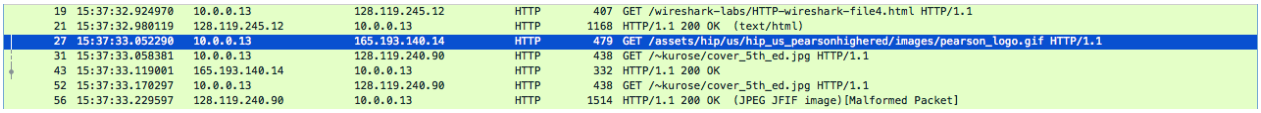
14. `HTTP/1.1 200 OK`
15. Data is reassembled between packets 9 through 12. 4 TCP segments total.



4. HTML Documents with Embedded Objects

16. 4 `HTTP GET` request messages to:

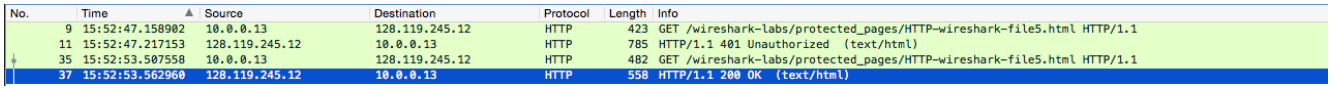
- `wireshark-labs/HTTP-wireshark-file.4html`
- `/assets/hip/us/hip_us_pearsonhighered/images/pearson_logo.gif`
- `/~kurose/cover_5th_ed.jpg`
- `/~kurose/cover_5th_ed.jpg`



19	15:37:32.924970	10.0.0.13	128.119.245.12	HTTP	407	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
21	15:37:32.980119	128.119.245.12	10.0.0.13	HTTP	1168	HTTP/1.1 200 OK (text/html)
27	15:37:33.052290	10.0.0.13	165.193.140.14	HTTP	479	GET /assets/hip/us/hip_us_pearsonhighered/images/pearson_logo.gif HTTP/1.1
31	15:37:33.058381	10.0.0.13	128.119.240.90	HTTP	438	GET /~kurose/cover_5th_ed.jpg HTTP/1.1
43	15:37:33.119001	165.193.140.14	10.0.0.13	HTTP	332	HTTP/1.1 200 OK
52	15:37:33.170297	10.0.0.13	128.119.240.90	HTTP	438	GET /~kurose/cover_5th_ed.jpg HTTP/1.1
56	15:37:33.229597	128.119.240.90	10.0.0.13	HTTP	1514	HTTP/1.1 200 OK (JPEG JFIF image)[Malformed Packet]

17. Browser behavior gives it away since the top image loaded before the bottom image. There is no GET message requesting two images at the same time. One GET request for each image and they are satisfied serially.

5. HTTP Authentication



No.	Time	Source	Destination	Protocol	Length	Info
9	15:52:47.158902	10.0.0.13	128.119.245.12	HTTP	423	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
11	15:52:47.217153	128.119.245.12	10.0.0.13	HTTP	785	HTTP/1.1 401 Unauthorized (text/html)
35	15:52:53.507558	10.0.0.13	128.119.245.12	HTTP	482	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
37	15:52:53.562960	128.119.245.12	10.0.0.13	HTTP	558	HTTP/1.1 200 OK (text/html)

18. `401 Unauthorized`
19. `200 OK`

Homework questions

lukas_borges

Chapter 2 Review questions: R1, R4, R11, R14, R25
Chapter 2 Problems: P1, P4, P5, P10, P11, P14, P15

Chapter 3 Review questions: R1, R2, R14, R15, R17
Chapter 3 Problems: P1, P3, P26, P27, P31, P40

Chapter 2:

Review questions:

R1.

#	application	protocol
1	email	SMTP (simple mail transfer protocol)
2	file transfer	FTP (file transfer protocol)
3	web	HTTP (hyper text transfer protocol)
4	remote login	telnet
5	streaming services	RTP (real-time transport protocol)

R4.
What happens in P2P is that every user is a client and a server at the same time. There is still a client-server relationship, but it is interchangeable.

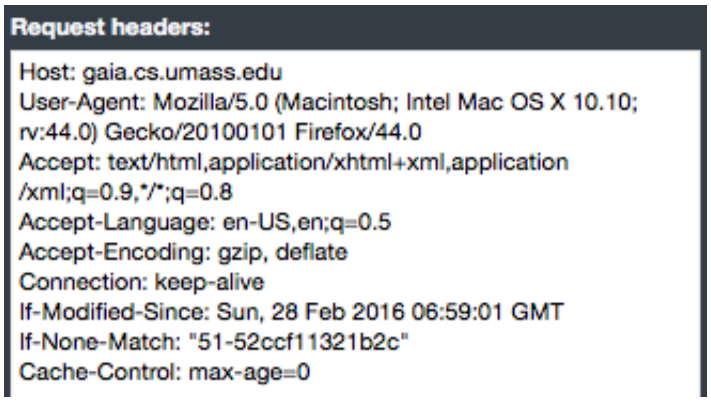
R11.
HTTP, FTP, SMTP and POP3 all require reliable data transfer. TCP provides schemes for data loss and corruption, while UDP works with a best-effort policy. Basic implementations of UDP do not guarantee information integrity, so important for the listed protocols.

R14.

telnet was not working as I expected, so I used Firefox developer tools instead:

first request

after selecting `Edit and Resend` , I manually edited the header:



this is the response:

Works as proposed.

R25.

1. Messaging
2. Real time communication (Skype)
3. File Sharing (torrent)
4. Distributed Computing

Problems

P1.

statement	answer
a	False
b	True
c	False
d	False
e	False

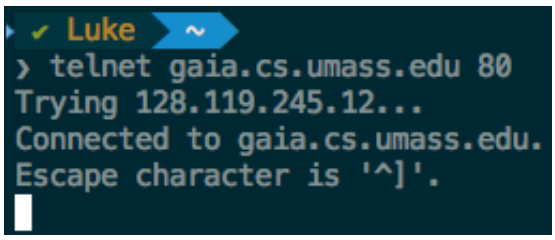
P4.

a. http://gaia.cs.umass.edu/cs453/index.html

b. HTTP/1.1

c. `Connection:keep-alive` , persistent

d.



`128.119.245.12`

e. Mozilla Firefox. Browser information is needed by the server in order to send different versions of the same object to different types of browsers.

P5.

a. Status code: `200 OK` .

`Date Tue 07 Mar 2008 12:39:45 GMT`

b. `Last-Modified: Sat, 10 Dec 2005 18:27:46 GMT`

c. `Content-Length: 3874` (bytes)

d. `<!doc` . Yes, according to `Connection: Keep-Alive`

P10.

persistent HTTP:

total time to receive all objects:

$$\left(\frac{200}{150}+T_p+\frac{200}{150}+T_p+\frac{200}{150}+T_p+\frac{100,000}{150}+T_p\right)+$$

$$\left(\frac{200}{150}+T_p+\frac{200}{150}+T_p+\frac{200}{150}+T_p+\frac{100,000}{150}+T_p\right)$$

$$=\left(\frac{200+200+200+100,000}{150}+4T_p\right)$$

$$+\left(\frac{200+200+200+100,000}{15}+4T_p\right)$$

$$=\left(\frac{100,600}{150}+4T_p\right)+\left(\frac{100,600}{15}+4T_p\right)$$

$$= (670 + 4T_p) + (6706 + 4T_p)$$

$$= 7377 + 8 * T_p \, seconds$$

time needed:

$$\left(\frac{200}{150} + T_p + \frac{200}{150} + T_p + \frac{200}{150} + T_p + \frac{100,000}{150} + T_p\right) +$$

$$10 * \left(\frac{200}{150} + T_p + \frac{100,000}{150} + T_p\right)$$

$$= \left(\frac{200 + 200 + 200 + 100,000}{150} + 4T_p\right) + 10 * \left(\frac{200 + 100,000}{150} + 2T_p\right)$$

$$= \left(\frac{100,600}{150} + 4T_p\right) + 10 * \left(\frac{100,200}{150} + 2T_p\right)$$

$$= (670 + 4T_p) + (6680 + 20T_p)$$

$$= 7350 + 24 * T_p$$

propagation speed of the medium: $300 * 10^6 m/sec$

$$T_p = \frac{10}{(300 * 10^6)} = 0.03 \, microseconds$$

For this case, persistent HTTP has no significant gain over non-persistent.

P11.
a. Yes. Bob's parallel connections will help him get pages quicker since he is using parallel instance of non-persistent HTTP while the others use non-persistent HTTP with parallel download.
b. Yes, even if all five users open five parallel instances of non-persistent HTTP. If Bob does not use parallel connections, he will get less bandwidth share. At this point, everyone (including Bob) should get equal bandwidth.

P14.

- SMTP uses period [.] to indicate end of message body.
- HTTP uses Content-Length header field to indicate end of message body.
  HTTP cannot use the same method because there is no format for message body

in HTTP.

P15.
MTA: Mail Transfer Agent.

Mail received from	Mail received by
barmail.cs.umass.edu [128.119.240.3]	cs.umass.edu [8.13.1/8.12.6]
asusus-4b96 [localhost (127.0.0.1)]	barmail.cs.umass.edu
asusus-4b96 [58.88.21.177]	barmail.cs.umass.edu
[58.88.21.177]	lnbnd55.exchangeddd.com

At `asusus-4b96[58.88.21.177]` the MTA does not report from where it receives the email (dishonest). If the mail is spam, only originator can be dishonest.

`asusus-4b96[58.88.21.177]` is the malicious host that generated spam.

Chapter 3:

Review questions:

R1.
a.
Transport protocol (TP):

- Accept chunk of data, Destination's host address and port number (given by the sender)
- Maximum size of data: 1196 bytes
- Add a 4-byte header to each chunk
- Number of destinations included in header
- Destination host address and resulting segment are passed to the network layer by our TP.
- Network layer delivers segment to TP at destination.
- At destination, port number from segment is analyzed by the TP, which catches

the data and sends it to the process with port number accordingly.

b.
Modified TP (mTP):

- Accept chunk of data, Destination's host address and port number (given by the sender)
- Maximum size of data: 1196 bytes
- Segment now has two header fields. One is used to specify source port and another to specify destination port.
- Add two 4-byte headers to each chunk. Include source port number and destination port number in two header fields.
- Destination host address, and resulting segment are passed to the network layer by mTCP
- Network layer delivers segment to mTCP at destination
- At destination, port number in the segment is examined by mTCP, which catches the data and sends it to the process with port number accordingly.
- It also gives the source port number to the application

c. No, transport layer only acts in the end systems.

R2.
a.

- Sender give delegate the letter along with the address of the destination's house and the name of the recipient.
- Delegate writes recipient's name is written on top of the letter.
- Letter is put into an envelope. Destination's house address is written by the delegate.
- Delegate passes letter to the planet's mail service.
- Receiver accepts letter from delegate. Envelope is opened and recipient's name written on top is stored.
- Delegate hand overs the letter according to the family name written on top.

b.
No, there is no need for the mail service to open the envelope and examine address. Envelope is opened by the delegate. Recipient's name written on top is noticed in order to deliver accordingly.

R14.

statement	answer
a	False
b	False
c	True
d	False
e	True
f	False
g	False

R15.

a.

first segment `SEQNUM = 90`

second segment `SEQNUM = 110`

data in first segment: $110 - 90 = 20\,bytes$

b.

90 is the segment number of the first segment.

R17.

TCP shares Rate $\frac{R}{2}\,bps$ to each TCP connection

Problems:

P1.

prompt	connection	source port	destination port
a	A $\rightarrow$ S	467	23
b	B $\rightarrow$ S	513	23
c	S $\rightarrow$ S	23	467
d	S $\rightarrow$ B	23	513

e.

Yes, because IP addresses are also included.

f.
No, the server uses IP addresses to distinguish between hosts.

P3.
part 1:
first byte: `0101 0011`
second byte: `0110 0110`
third byte: `0111 0100`

first + second:
`0101 0011` +
`0110 0110`

_____
`1011 1001` (first sum)

first sum + third byte
`1011 1001` +
`0111 0100`

_____
wrap around: `(1) 0010 1101` (second sum)
`1`

_____
Total Sum: `0010 1110`
Checksum: `1101 0001`

1s complement of: `0010 1110 = 1101 0001`
Same as Checksum.

part 2: When the 1's complement of sum is performed, resultant will be the checksum. Receiving host uses the checksum so check for errors.

part 3: At receiving end, all bytes (including checksum) are added. Receiver looks at the checksum to for errors. If the sum contains all 1s, no errors. If there is at least one 0 bit, there is an error.

part 4: All one-bit errors will be detected. Two bit errors can be undetected since they could end up complementing each others' value.

P26.

a.

MSS $= 536$ bytes

TCP sequence number field $= 4$ bytes

$= 4 * 8 = 32\,bits = 2^{32}$

Maximum file size that can be sent is given by how many bytes can be represented by 2^{32}

$= 2^2 * 2^{30}$ bytes

$= 2^2$ Gbytes or 4 Gbytes

b.

Number of Segments $= \frac{2^{32}}{536} = 801299$

Number of bytes in each segment $= 66$

Total number of bytes sent over the 155Mbps link $= 8012999 * 66$ bytes

$= 528857934$ bytes $= 4.824 * 10^9$ bytes

time it takes:

$$\frac{4.824 * 10^9 * 8\,bits}{155 * 10^6\,bps}$$

$$\approx 249 seconds$$

P27.

a.

`SEQNUM = SEQNUM` of first segment + number of bytes of data in first segment

$127 + 80 = 207$

Source port: 302

Destination port: 80

b.

Acknowledgement number: 207

Source port: 80

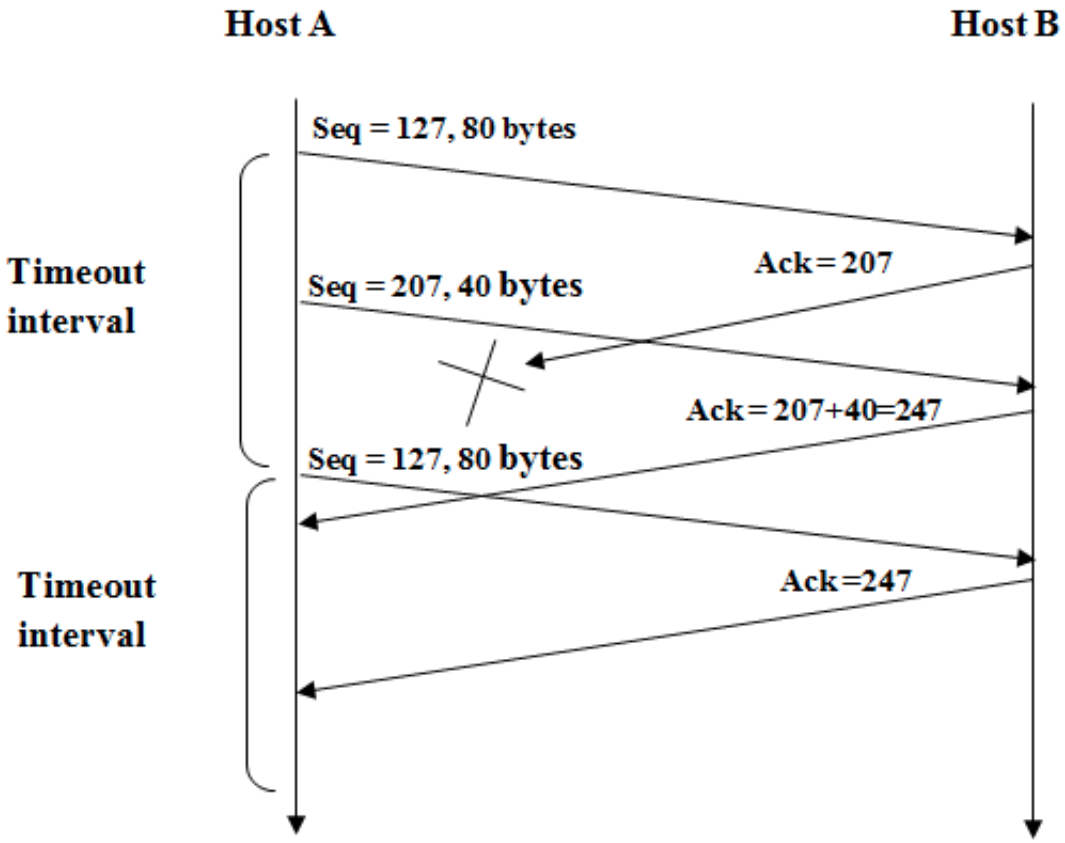
Destination port: 302

c.

Acknowledgement number: 127

Because the receive is waiting for 127 bytes of upcoming data.

d.



Even though ACK 207 is loss, when Host A receives ACK 247, it knows that packet 207 was received.

P31.

- EstimatedRTT $= (1 - \alpha) * EstimatedRTT + \alpha * SampleRTT$
- DevRTT = $(1 - \beta) * DevRTT + \beta * |SampleRTT - EstimatedRTT|$
- Timeout Interval $= EstimatedRTT + 4 * DevRTT$

First sample (106 ms):
EstimatedRTT: $0.875 * 100 + 0.125 * 106 = 100.75$
DevRTT: $0.75 * 5 + 0.25 * 5.25 = 5.06$
TimeOut Interval: $100.75 + 4 * 5.06 = 120.99$

Second sample (120 ms):
EstimatedRTT: $0.875 * 100 + 0.125 * 120 = 103.156$
DevRTT: $0.75 * 5.06 + 0.25 * 16.84 = 8$
TimeOut Interval: $103.156 + 4 * 8 = 135.156$

Third sample (140 ms):
EstimatedRTT: $0.875 * 103.156 + 0.125 * 140 = 107.761$
DevRTT: $0.75 * 8 + 0.25 * 32.239 = 14.05$
TimeOut Interval: $107.761 + 4 * 14.05 = 163.961$

Fourth sample (90 ms):
EstimatedRTT: $0.875 * 107.761 + 0.125 * 90 = 105.54$
DevRTT: $0.75 * 14.05 + 0.25 * (-15.54) = 6.73$
TimeOut Interval: $105.54 + 4 * 6.73 = 132.46$

$TimeoutInterval_{90} = 132.46$

SampleRTT (115 ms):
EstimatedRTT: $0.875 * 105.54 + 0.125 * 115 = 106.772$
DevRTT: $0.75 * 6.73 + 0.25 * |115 - 106.772| = 7.10$
TimeOut Interval: $106.772 + 4 * 7.10 = 135.122$

$TimeoutInterval_{115} = 135.122$

P40.

a.
[1,6] and [23, 26]

b.
[6,16] and [17,22]

c.
Yes, the segment loss after 16th transmission is indicated by triple-duplicate ACK.

d.
Yes, segment loss after 22nd transmission is indicated by a timeout.

e.
Initial threshold (segment size threshold) value at the first transmission is 32.

f.
Threshold value during 18th transmission is 21.

g.
Threshold value during 24th transmission round is 14 (approx).

h.

transmission	packet number
1	1
2	2 - 3
3	4 - 7
4	8 - 15
5	16 - 31
6	32 - 63
7	64 - 96

70th segment is transmitted in the 7th round.

i.
Threshold value after 26th transmission is 4.

j.
Threshold value at 19th transmission is 21 and congestion window size is 1.

k.
Total of 52 packets.

Transmission Round	Number of Packets
17	1
18	2
19	4
20	8
21	16
22	21
total	52