CSP334 : Computer Networks
Lab Assignment No 4
Assignment on HTTP

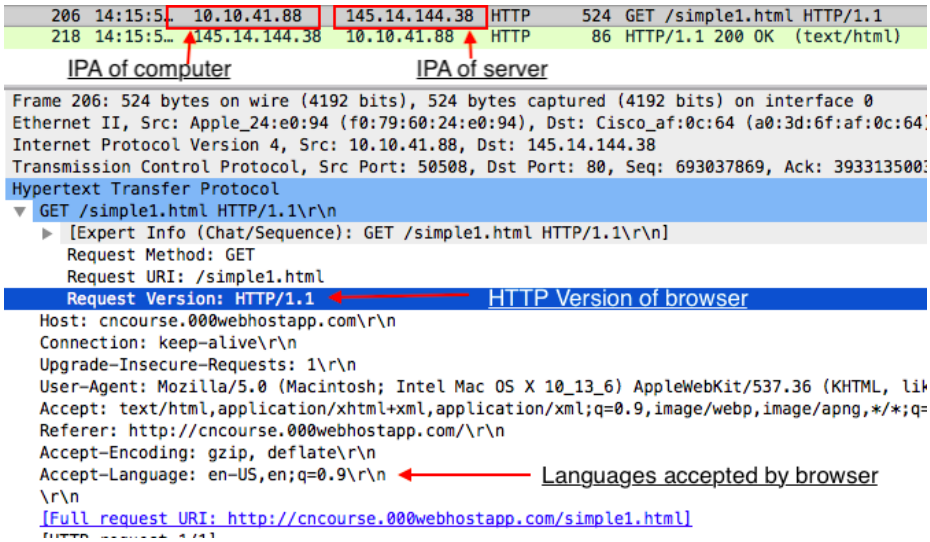
Sahil
2016UCS0008

September 26, 2018

1 The Basic HTTP GET/ Response Interaction:

File: *simple1.html*

1.1 HTTP Version of browser and server:



As highlighted, version **1.1** is used by both browser and server.

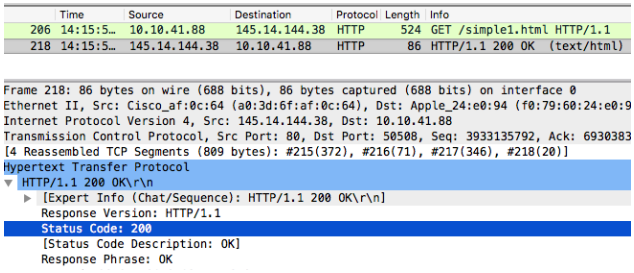
1.2 Languages accepted by the browser:

As shown above, **English** language is accepted.

1.3 IPA of computer and *cncourse* web server:

IPA of computer is 10.10.41.88 and that of server is 145.14.144.38.

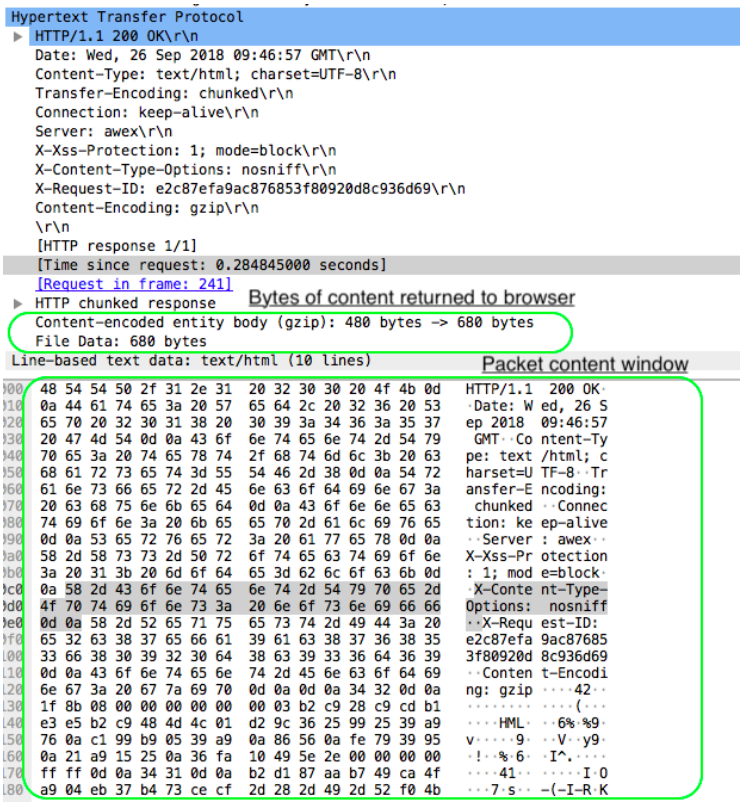
1.4 Status code returned from server to the browser: *200*



1.5 HTML file last modified at the server:

There is no *last-modified* header in the HTTP packet.

1.6 Bytes of content returned to the browser:



As highlighted, 480 bytes are returned to browser in *gzip* format which is then uncompressed to 680 bytes by the browser.

1.7 Headers not displayed in the packet-listing window:

As seen in the packet content window, all the headers shown in the packet-listing window are only displayed, so there is NO additional header present.

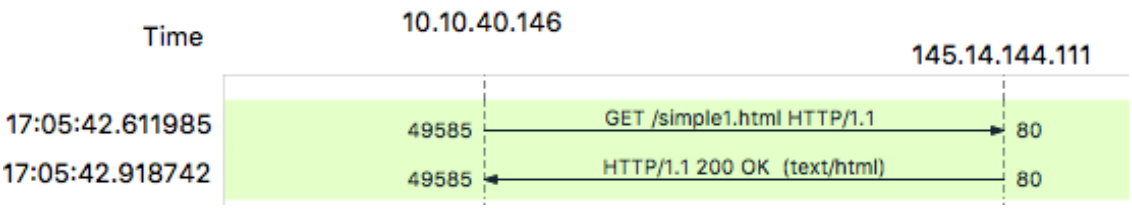
1.8 No. of request sent by browser and responded by server:

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
▼ HTTP Requests by HTTP Host	1	0.0035	100%	0.0100	5.839
▼ cncourse.000webhostapp.com	1	0.0035	100.00%	0.0100	5.839
/simple1.html	1	0.0035	100.00%	0.0100	5.839

There is one request sent by the browser to the *simple1.html* file on the server, and same is responded back by the server.

1.9 HTTP traffic flow graph showing the packet exchanges between the client and the server:

The *HTTP traffic flow* graph is obtained from the **Statistics - Flow Graph** option and then checking the option *Limit to display filter* as the filter chosen is HTTP.



1.10 File *simple2.html*:

Last modified information is not available for the *html* file but for the *GIF* image loaded along with this html, it is displayed below.

```
Hypertext Transfer Protocol
▶ HTTP/1.1 200 OK\r\n
  Server: nginx/1.12.2\r\n
  Date: Wed, 26 Sep 2018 09:17:53 GMT\r\n
  Content-Type: image/gif\r\n
▶ Content-Length: 9601\r\n
  Last-Modified: Fri, 03 Jun 2016 13:32:18 GMT\r\n
  Connection: keep-alive\r\n
  ETag: "575186e2-2581"\r\n
  Expires: Thu, 26 Sep 2019 09:17:53 GMT\r\n
  Cache-Control: max-age=31536000\r\n
  Accept-Ranges: bytes\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.411436000 seconds]
  [Request in frame: 323]
  File Data: 9601 bytes
Compuserve GIF, Version: GIF89a
```

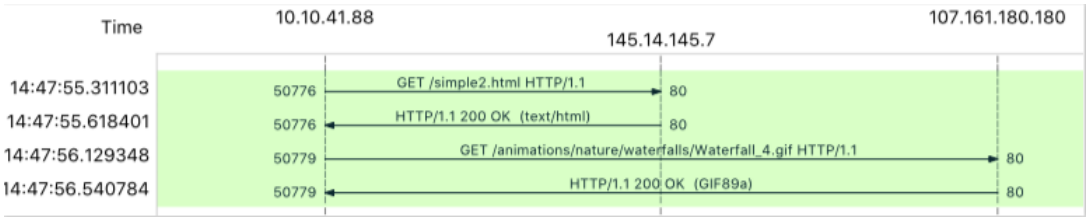
Bytes of content returned for the HTML file is 572 bytes. That for the GIF is 9601 bytes.

```
Hypertext Transfer Protocol
▶ HTTP/1.1 200 OK\r\n
  Date: Wed, 26 Sep 2018 09:17:52 GMT\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  Transfer-Encoding: chunked\r\n
  Connection: keep-alive\r\n
  Server: awex\r\n
  X-Xss-Protection: 1; mode=block\r\n
  X-Content-Type-Options: nosniff\r\n
  X-Request-ID: a370cb879592100b6ea0221a42134a4e\r\n
  Content-Encoding: gzip\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.307298000 seconds]
  [Request in frame: 295]
▶ HTTP chunked response
  Content-encoded entity body (gzip): 572 bytes -> 868 bytes
  File Data: 868 bytes
Line-based text data: text/html (13 lines)
```

No. of requests sent by the browser and responded by the server are both 2, as shown below.

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
▼ HTTP Requests by HTTP Host	2	0.0016	100%	0.0100	3.791
▼ cncourse.000webhostapp.com	1	0.0008	50.00%	0.0100	3.791
/simple2.html	1	0.0008	100.00%	0.0100	3.791
▼ gifgifs.com	1	0.0008	50.00%	0.0100	4.609
/animations/nature/waterfalls/Waterfall_4.gif	1	0.0008	100.00%	0.0100	4.609

HTTP Traffic Flow graph:



1.11 File *simple3.html*:

Last modified information is not available for the *simple3.html* file.

	Time	Source	Destination	Protocol	Length	Info
93	14:48:20…	10.10.41.88	145.14.145.7	HTTP	565	GET /simple3.html HTTP/1.1
103	14:48:20…	145.14.145.7	10.10.41.88	HTTP	86	HTTP/1.1 200 OK (text/html)
108	14:48:20…	10.10.41.88	145.14.145.7	HTTP	577	GET /simple1.html HTTP/1.1
109	14:48:20…	10.10.41.88	145.14.145.7	HTTP	577	GET /simple2.html HTTP/1.1
128	14:48:20…	145.14.145.7	10.10.41.88	HTTP	86	HTTP/1.1 200 OK (text/html)
137	14:48:20…	145.14.145.7	10.10.41.88	HTTP	86	HTTP/1.1 200 OK (text/html)

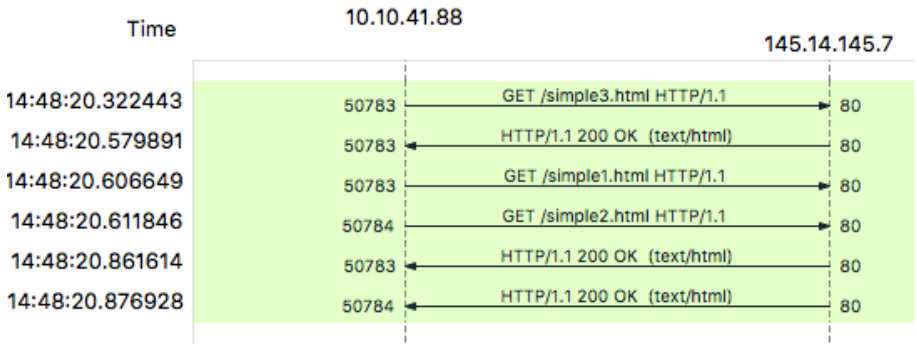
Bytes of content returned for the HTML file is 138 bytes for *simple3.html*. It also loads *simple1.html* and *simple2.html* whose sizes were mentioned earlier.

```
Hypertext Transfer Protocol
▶ HTTP/1.1 200 OK\r\n
  Date: Wed, 26 Sep 2018 09:18:17 GMT\r\n
  Content-Type: text/html; charset=UTF-8\r\n
  Transfer-Encoding: chunked\r\n
  Connection: keep-alive\r\n
  Server: awex\r\n
  X-Xss-Protection: 1; mode=block\r\n
  X-Content-Type-Options: nosniff\r\n
  X-Request-ID: 37e26d841c1af140b506268b5a254bee\r\n
  Content-Encoding: gzip\r\n
  \r\n
  [HTTP response 1/2]
  [Time since request: 0.257448000 seconds]
  [Request in frame: 93]
  [Next request in frame: 108]
  [Next response in frame: 128]
▶ HTTP chunked response
  Content-encoded entity body (gzip): 138 bytes -> 174 bytes
  File Data: 174 bytes
Line-based text data: text/html (10 lines)
```

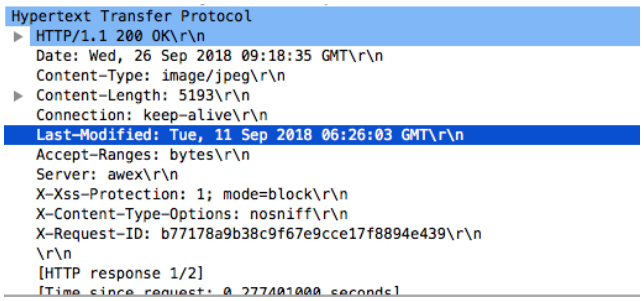
No. of requests sent by the browser and responded by the server are both 3, as shown below.

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
▼ HTTP Requests by HTTP Host	3	0.0054	100%	0.0200	4.409
▼ cncourse.000webhostapp.com	3	0.0054	100.00%	0.0200	4.409
/simple1.html	1	0.0018	33.33%	0.0100	4.409
/simple2.html	1	0.0018	33.33%	0.0100	4.415
/simple3.html	1	0.0018	33.33%	0.0100	4.125

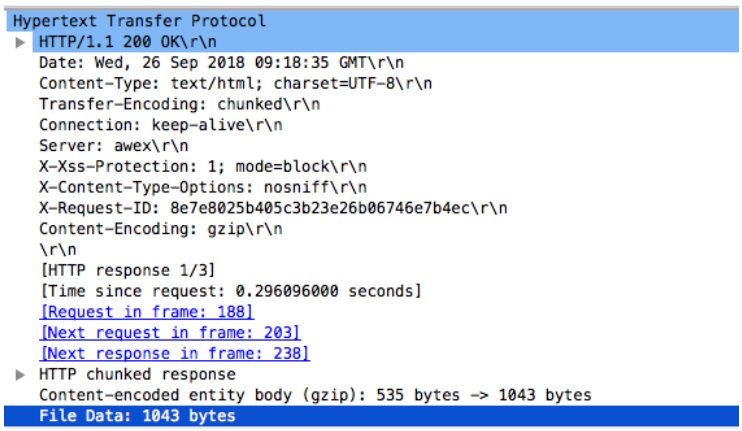
HTTP Traffic Flow graph:



1.12 File *simple4.html*:



Last modified information is not available for the *simple4.html* file but its present for the JPEG image files loaded from another server, as shown above for one such image.
Bytes of content returned for the HTML file is 535 bytes for *simple4.html*. It also loads 10 other image files.



No. of requests sent by the browser and responded by the server are both 11, as shown below.

Topic / Item	Count	Rate (ms)	Percent	Burst rate	Burst start
▼ HTTP Requests by HTTP Host	11	0.0089	100%	0.0600	8.509
▼ cncourse.000webhostapp.com	11	0.0089	100.00%	0.0600	8.509
/simple4.html	1	0.0008	9.09%	0.0100	7.930
/9.jpg	1	0.0008	9.09%	0.0100	8.509
/8.jpg	1	0.0008	9.09%	0.0100	8.259
/7.jpg	1	0.0008	9.09%	0.0100	8.256
/6.jpg	1	0.0008	9.09%	0.0100	8.517
/5.jpg	1	0.0008	9.09%	0.0100	8.516
/4.jpg	1	0.0008	9.09%	0.0100	8.509
/3.jpg	1	0.0008	9.09%	0.0100	8.528
/2.jpg	1	0.0008	9.09%	0.0100	8.856
/1.jpg	1	0.0008	9.09%	0.0100	8.541
/0.jpg	1	0.0008	9.09%	0.0100	8.860

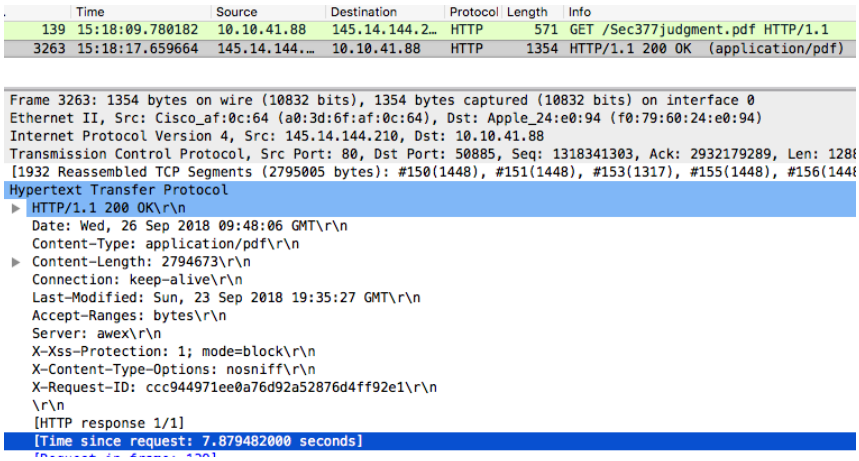
HTTP Traffic Flow graph:

Whether images downloaded serially or parallely:

The image downloads occurred *parallelly*. We can say so since in the above traffic flow graph, its clearly visible that the requests for the images 7.jpg, 8.jpg,

9.jpg, ..., all were made in a single go without waiting for response of either of the image to be available. The first response is available after 6 images were already requested.

1.13 Time required to access the pdf file:



The time required as highlighted = 7.88 seconds.

1.14 Table for time required to access each of the files:

S.No	File Name	Time Required to access the file (in seconds)
1.	simple1.html	0.284
2.	simple2.html	0.307
3.	simple3.html	0.257
4.	PDF file	7.88
5.	Large Text file	1.167

2 The HTTP CONDITIONAL GET/Response Interaction:

The 4 requests were captured as shown below:

Source	Destination	Protocol	Length	Info
10.10.41.88	13.75.89.224	HTTP	470	GET / HTTP/1.1
13.75.89.224	10.10.41.88	HTTP	839	HTTP/1.1 200 OK (text/html)
10.10.41.88	13.75.89.224	HTTP	668	GET / HTTP/1.1
13.75.89.224	10.10.41.88	HTTP	232	HTTP/1.1 304 Not Modified

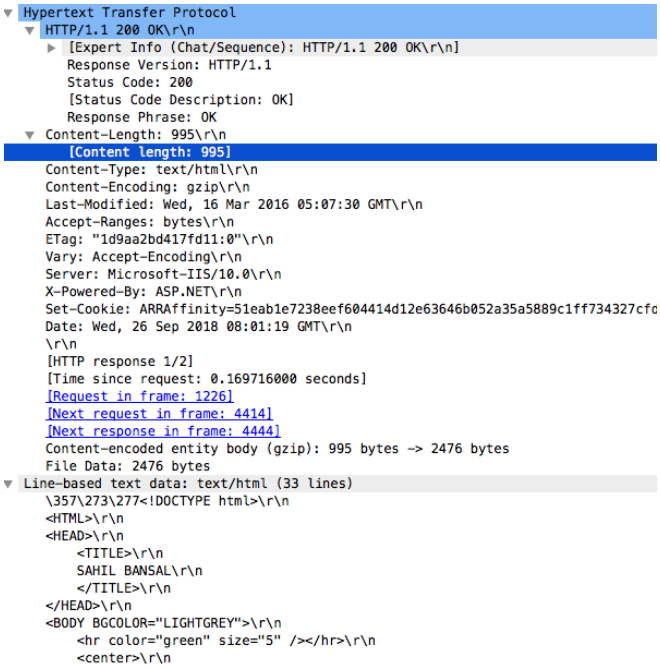
2.1 First HTTP GET request from the browser to the server:

	Time	Source	Destination	Protocol	Length	Info
1226	13:3…	10.10.41.88	13.75.89.224	HTTP	470	GET / HTTP/1.1
1345	13:3…	13.75.89.224	10.10.41.88	HTTP	839	HTTP/1.1 200 OK (text/html)
4414	13:3…	10.10.41.88	13.75.89.224	HTTP	668	GET / HTTP/1.1
4444	13:3…	13.75.89.224	10.10.41.88	HTTP	232	HTTP/1.1 304 Not Modified

```
Frame 1226: 470 bytes on wire (3760 bits), 470 bytes captured (3760 bits) on interface 0
Ethernet II, Src: Apple_24:e0:94 (f0:79:60:24:e0:94), Dst: Cisco_af:0c:64 (a0:3d:6f:af:0c:6
Internet Protocol Version 4, Src: 10.10.41.88, Dst: 13.75.89.224
Transmission Control Protocol, Src Port: 50297, Dst Port: 80, Seq: 873883957, Ack: 60751268
Hypertext Transfer Protocol
▼ GET / HTTP/1.1\r\n
  ▶ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /
    Request Version: HTTP/1.1
  Host: sahilbansal.azurewebsites.net\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, l
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: en-US,en;q=0.9\r\n
  \r\n
  [Full request URI: http://sahilbansal.azurewebsites.net/]
  [HTTP request 1/2]
  [Response in frame: 1345]
  [Next request in frame: 4414]
```

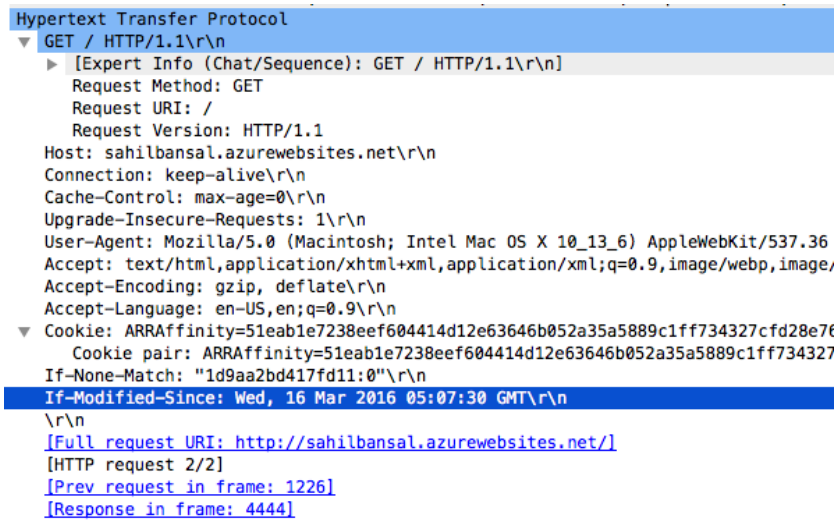
There is no **IF-MODIFIED-SINCE** header in this GET request.

2.2 Contents of the server response:



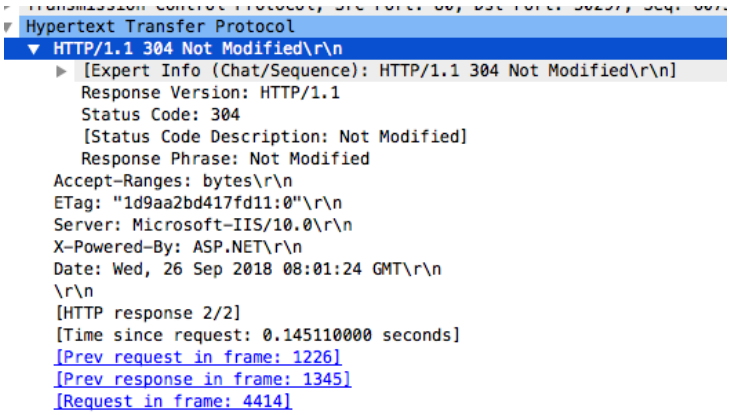
Yes, the server has explicitly responded with the content of the file since the *content length* has a positive value and also the HTML file is visible as shown.

2.3 Second HTTP GET request from the browser to the server:



Yes, now we can see an **IF-MODIFIED-SINCE** header in the HTTP GET request. The value of the header is the date when the file was last modified on the server side. It shows *Wed, Mar 16, 2016* as the last modified date.

2.4 HTTP status code and phrase returned from the server in response to 2nd HTTP GET:



The status code is **304** and the response phrase is **Not Modified** in response to the 2nd HTTP GET request. The server did not explicitly return the content of the file this time since there is no content length header visible, as well as no content visible after two cr-lf present after the Date Header.
This happens because the client sent an IF-MODIFIED GET request, in which the server responds with the content only when the file has been modified on the server after the date of previous entry of the file in the cache accessbile by client. Otherwise, it just sends a 304-Not Modified response.

3 Retrieving Long Documents:

3.1 HTTP GET request messages sent by the browser:

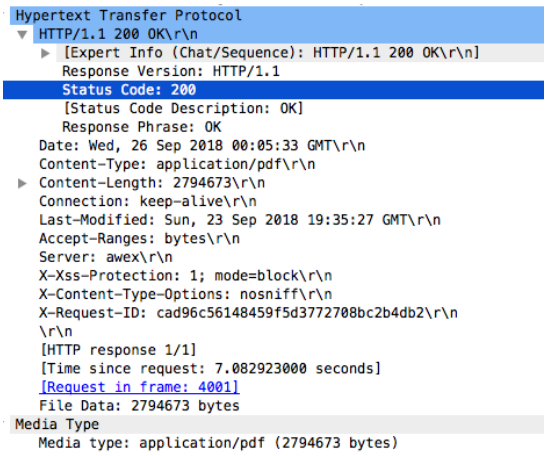
624	05:33:14.818792	10.10.41.88	145.14.144.188	HTTP	526	GET /largeText.html HTTP/1.1
637	05:33:15.082354	145.14.144.188	10.10.41.88	HTTP	77	[TCP Previous segment not captured] Continuation
641	05:33:15.082761	145.14.144.188	10.10.41.88	HTTP	1275	Continuation
642	05:33:15.082768	145.14.144.188	10.10.41.88	HTTP	960	Continuation
645	05:33:15.083051	145.14.144.188	10.10.41.88	HTTP	616	Continuation
646	05:33:15.083058	145.14.144.188	10.10.41.88	HTTP	555	Continuation
647	05:33:15.083145	145.14.144.188	10.10.41.88	HTTP	491	Continuation
648	05:33:15.083148	145.14.144.188	10.10.41.88	HTTP	456	Continuation
649	05:33:15.083149	145.14.144.188	10.10.41.88	HTTP	457	Continuation
672	05:33:15.329014	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
673	05:33:15.329019	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
677	05:33:15.339258	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
678	05:33:15.339444	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
680	05:33:15.339723	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
682	05:33:15.339887	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
683	05:33:15.340203	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
685	05:33:15.340403	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
687	05:33:15.340682	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
688	05:33:15.340882	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
692	05:33:15.341645	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
694	05:33:15.343412	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
695	05:33:15.343649	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
696	05:33:15.343653	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
699	05:33:15.343822	145.14.144.188	10.10.41.88	HTTP	1514	Continuation

1066	05:33:15.892655	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1070	05:33:15.892837	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1071	05:33:15.893104	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1072	05:33:15.893108	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1073	05:33:15.893110	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1074	05:33:15.893111	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1075	05:33:15.893113	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1079	05:33:15.893344	145.14.144.188	10.10.41.88	HTTP	1514	Continuation
1080	05:33:15.893347	145.14.144.188	10.10.41.88	HTTP	1032	Continuation
4001	05:35:35.987440	10.10.41.88	145.14.145.72	HTTP	530	GET /Sec377judgment.pdf HTTP/1.1
7342	05:35:43.070363	145.14.145.72	10.10.41.88	HTTP	1058	HTTP/1.1 200 OK (application/pdf)

The browser sent 2 HTTP GET requests, one for the pdf file and other for the large text file since both requests were made in the same wireshark session. Both the requests are highlighted above. The packet numbers 624 and 4001 in the trace contain the HTTP GET message.

3.2 Packet no in trace containing the status code and phrase associated with the response to the HTTP GET request:

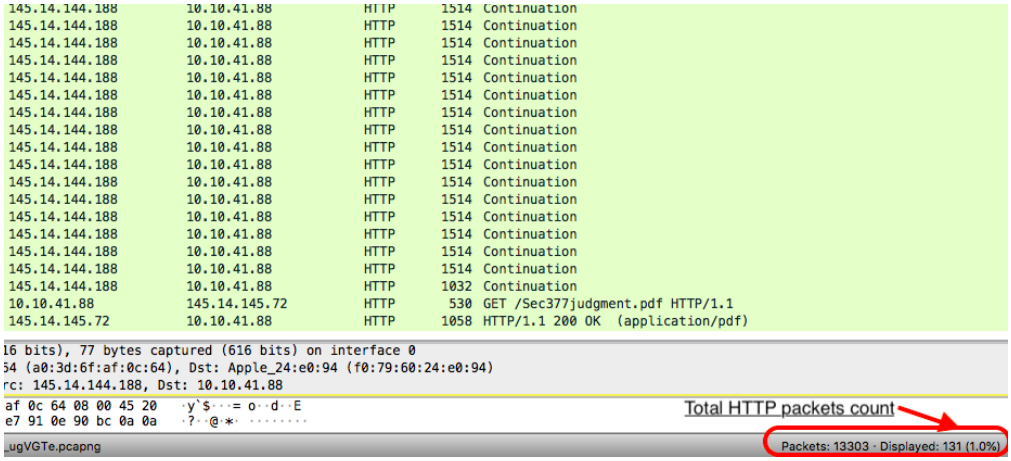
We receive *TCP Continuation* message for the large text file requested, as shown by packet no. 637 highlighted in black above. A **200 OK** response is received for the pdf file requested as shown below. The response is the packet no. 4001 in the above trace.



3.3 Status code and phrase in the response:

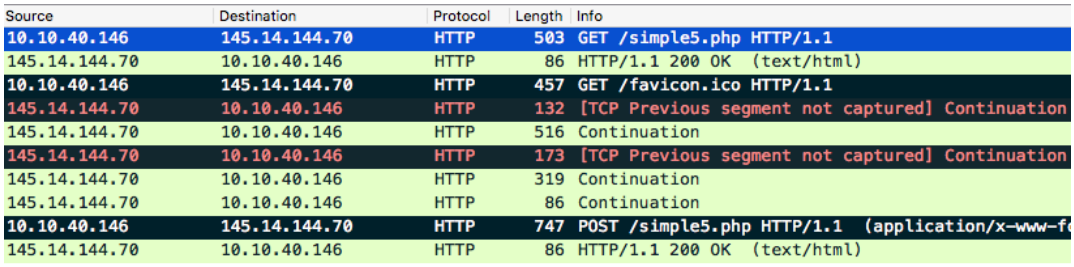
The status code is **200** and the response phrase is **OK**.

3.4 No. of data-containing TCP segments needed to carry the single HTTP response and the text of the file:



In total, there are 131 HTTP packets were displayed, as shown in bottom right corner here, out of which 2 packets are for the pdf file, whereas in the remaining 129 packets, 1 is for the GET request and 1 is for the HTTP response, so, in total 127 data containing TCP segments were needed to carry the text of the file.

4 HTML Documents with CGI Script:



4.1 Method used in the HTTP message:

GET method is used to get the simple5.php file whereas POST method is also used to send the input taken from user to the server.

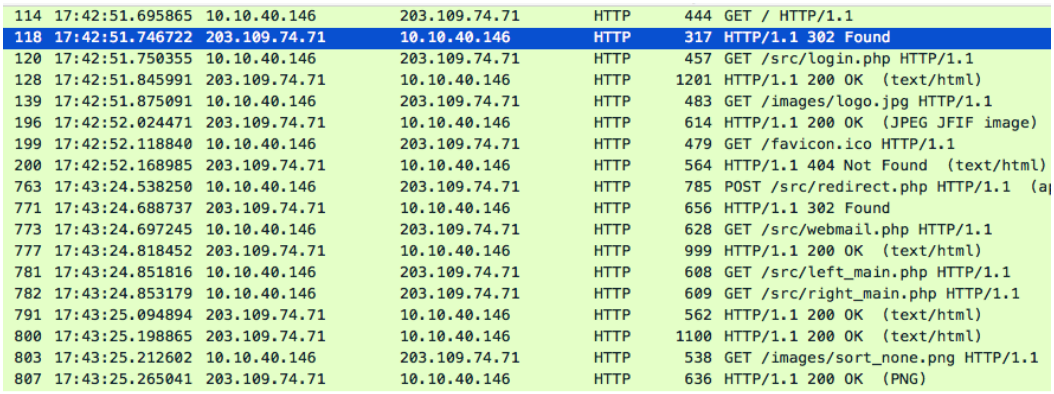
4.2 No. of HTTP request messages sent by the browser:

Total 3 HTTP request messages are sent by the browser.

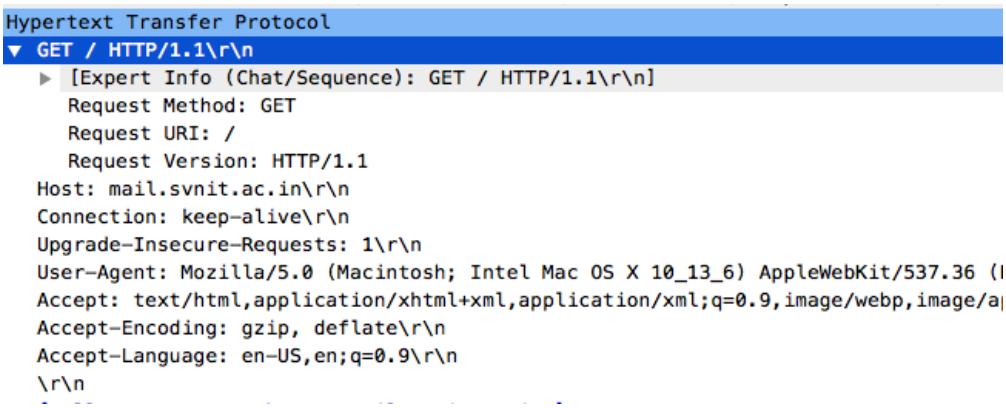
4.3 IPA to which the requests are sent:

The IPA to which the requests are sent is 145.14.144.70.

5 HTTP Authentication:

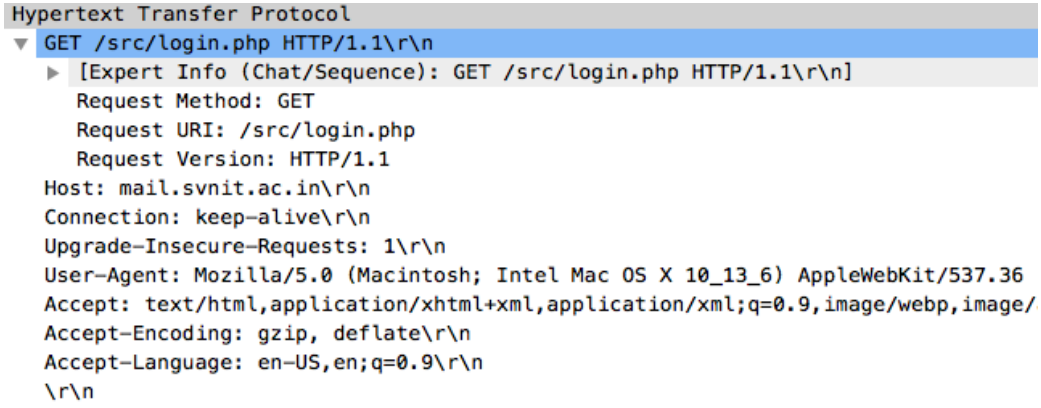


5.1 Server response to the initial HTTP GET message:



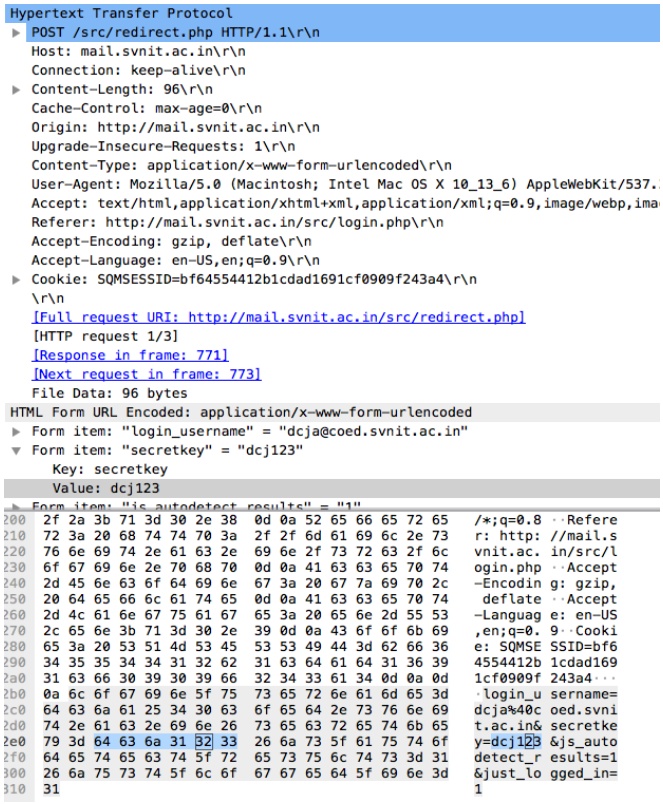
The status code is 302 and the response phrase is **Found**, in response to the initial GET message which is shown above.

5.2 New field included in HTTP GET message when browser sends request for 2nd time:



Considering the 2nd GET request sent by browser and comparing the HTTP message with previous one, we find that no new filed is included.

5.3 Content of packet in wireshark where password is displayed:



The password is highlighted in the packet content window with light blue color.