ADVANCED COMPUTER NETWORKS

ASSIGNMENT 1 : WIRESHARK

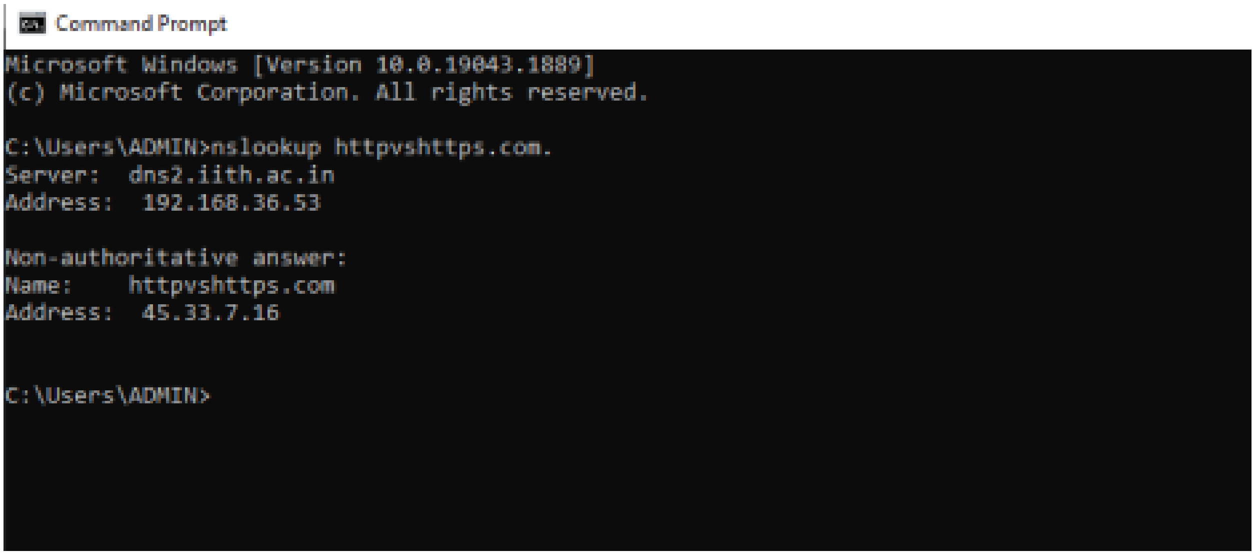
Name: **SHRUSTI**
Roll No : **CS22MTECH11017**

TASK 1:

Start the Wireshark packet sniffer and start capturing.
Enter the following URL into your browser http://httpvshttps.com. Click the HTTP tab on the top
right of the website if the browser has not opened it at the first load.

Stop the packet capture when you have all the information captured, which is required to answer all the questions below.

- IP address of the website is : **45.33.7.16**



```
Microsoft Windows [Version 10.0.19043.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ADMIN>nslookup httpvshttps.com.
Server:  dns2.iith.ac.in
Address:  192.168.36.53

Non-authoritative answer:
Name:    httpvshttps.com
Address:  45.33.7.16


C:\Users\ADMIN>
```

1. How many HTTP requests (Type and respective count of requests), responses (status code and phrase of each of the responses) did the browser send and receive ?

- Use the filter ip.addr == 45.33.7.16 to filter out packets only from/to http://httpvshttps.com.

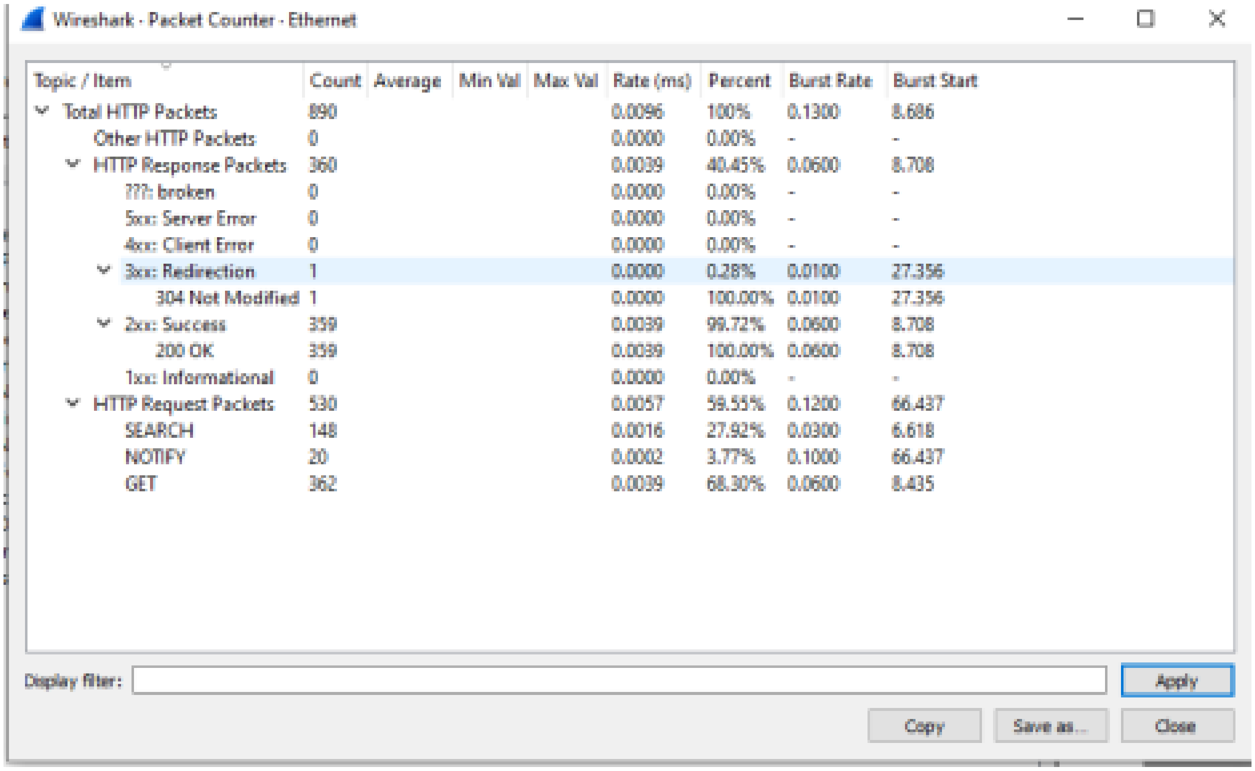
Using this filter and checking the Packet Counter of HTTP statistics, we see that a total of 890

packets are sent between the server and browser.
Out of these, there are 530 are request packets which are entirely composed of

- SEARCH requests | Count : 148
- NOTIFY requests | Count : 20
- GET requests | Count : 362

.
And, there are 360 HTTP Response packets, the distribution of these is as follows:

- Count: 1 | Status: 304 | Phrase: Not Modified
- Count: 359 | Status: 200 | Phrase: OK



Wireshark · Packet Counter · Ethernet

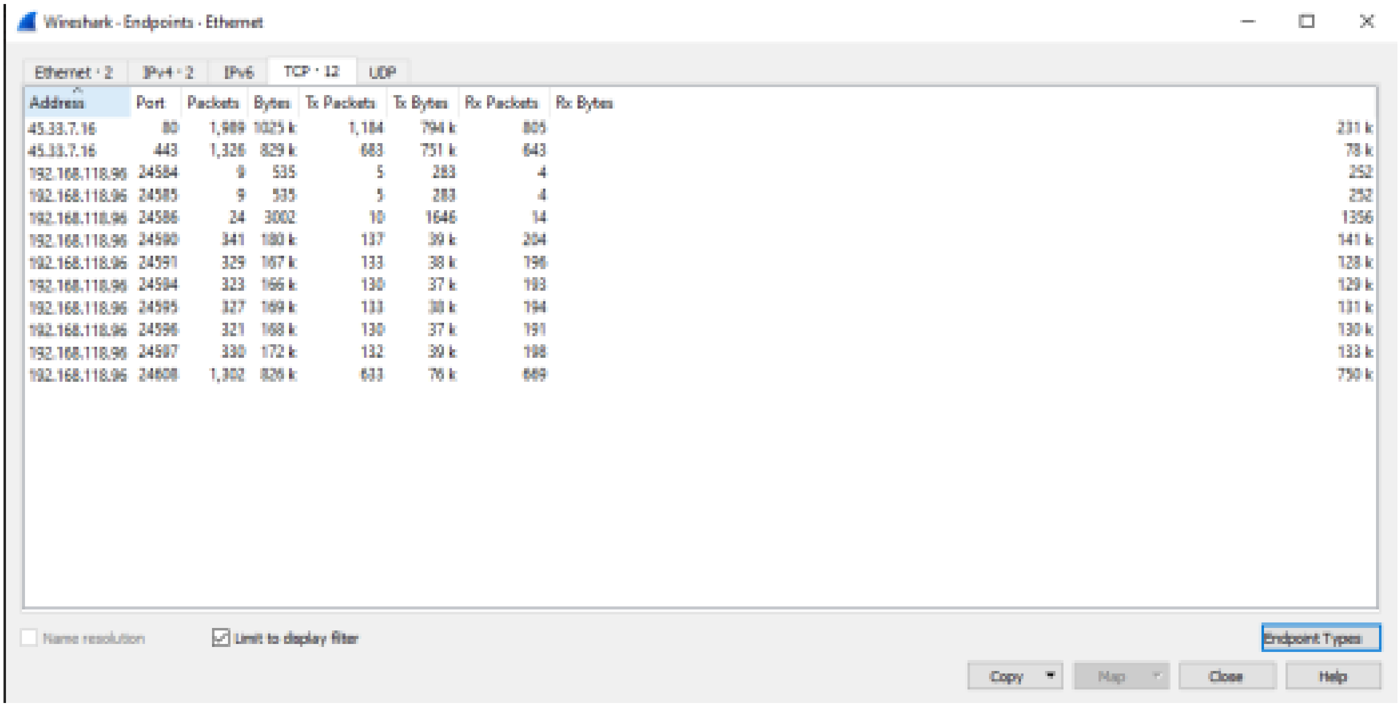
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total HTTP Packets	890				0.0096	100%	0.1300	8.686
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	360				0.0039	40.45%	0.0600	8.708
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	1				0.0000	0.28%	0.0100	27.356
304 Not Modified	1				0.0000	100.00%	0.0100	27.356
2xx: Success	359				0.0039	99.72%	0.0600	8.708
200 OK	359				0.0039	100.00%	0.0600	8.708
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	530				0.0057	59.55%	0.1200	66.437
SEARCH	148				0.0016	27.92%	0.0300	6.618
NOTIFY	20				0.0002	3.77%	0.1000	66.437
GET	362				0.0039	68.30%	0.0600	8.435

Display filter:

2. How many TCP Connections has the browser established overall ?

Use the filter ip.addr == 45.33.7.16 to filter out packets only from/to http://httpvshttps.com.

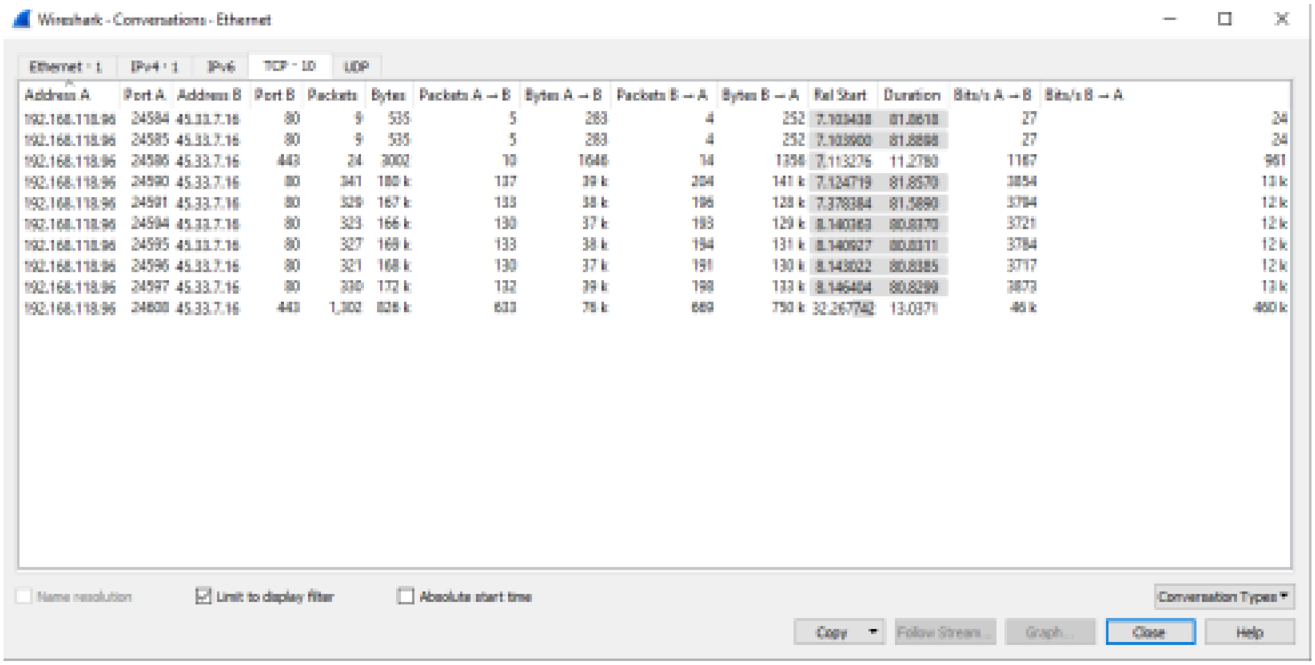
In the Endpoint statistics -> display to the filtered list, we find that the browser established **12** TCP connections between our system and the server.



Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
45.33.7.16	80	1,989	1025 k	1,184	794 k	805	231 k
45.33.7.16	443	1,326	829 k	683	751 k	643	78 k
192.168.118.96	24584	9	535	5	283	4	252
192.168.118.96	24585	9	535	5	283	4	252
192.168.118.96	24586	24	3002	10	1646	14	1356
192.168.118.96	24590	341	180 k	137	39 k	204	141 k
192.168.118.96	24591	329	167 k	133	38 k	196	128 k
192.168.118.96	24594	323	166 k	130	37 k	193	129 k
192.168.118.96	24595	327	169 k	133	38 k	194	131 k
192.168.118.96	24596	321	168 k	130	37 k	191	130 k
192.168.118.96	24597	330	172 k	132	39 k	198	133 k
192.168.118.96	24608	1,302	826 k	633	76 k	669	750 k

3. List the time taken to establish each TCP connection?

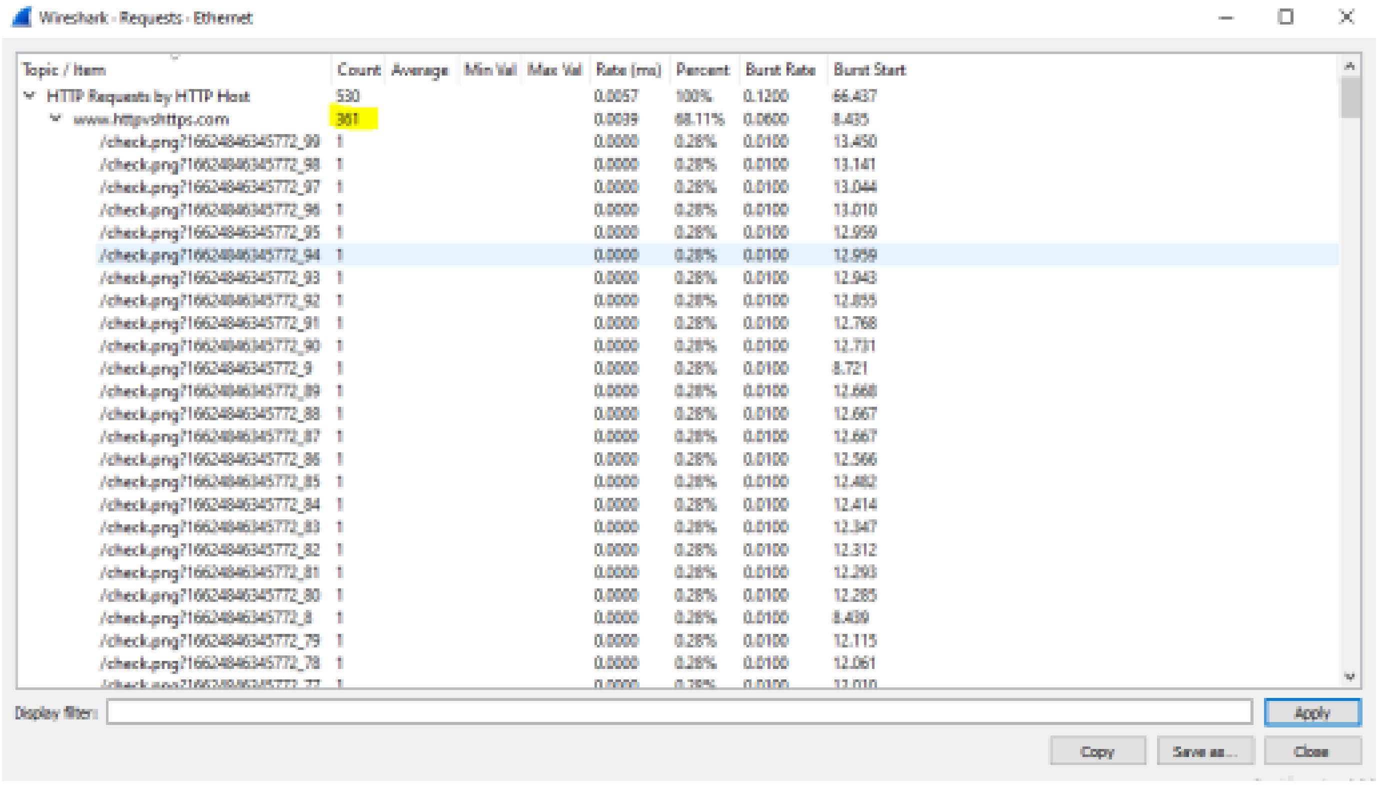
The time taken to establish each TCP connection can be seen in the ‘**Duration**’ column of Conversation statistics.



Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.118.96	24584	45.33.7.16	80	9	535	5	283	4	252	7.103438	81.8618	27	24
192.168.118.96	24585	45.33.7.16	80	9	535	5	283	4	252	7.103900	81.8888	27	24
192.168.118.96	24586	45.33.7.16	443	24	3002	10	1646	14	1356	7.113276	11.2780	1167	961
192.168.118.96	24590	45.33.7.16	80	341	180 k	137	39 k	204	141 k	7.124719	81.8570	3854	13 k
192.168.118.96	24591	45.33.7.16	80	329	167 k	133	38 k	196	128 k	7.378384	81.5890	3794	12 k
192.168.118.96	24594	45.33.7.16	80	323	166 k	130	37 k	193	129 k	8.140363	80.8370	3721	12 k
192.168.118.96	24595	45.33.7.16	80	327	169 k	133	38 k	194	131 k	8.140927	80.8311	3784	12 k
192.168.118.96	24596	45.33.7.16	80	321	168 k	130	37 k	191	130 k	8.143022	80.8385	3717	12 k
192.168.118.96	24597	45.33.7.16	80	330	172 k	132	39 k	198	133 k	8.146404	80.8299	3873	13 k
192.168.118.96	24608	45.33.7.16	443	1,302	826 k	633	76 k	669	750 k	32.267742	13.0371	46 k	460 k

4. Click the HTTP tab on the top right of the website if the browser has not opened it at the first load. How many objects/files are downloaded ?

Request Sequence under HTTP Statistics gives us the count of the number of files/objects that are downloaded.Here we can see that **361** objects/files are downloaded.



Wireshark · Requests · Ethernet

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
HTTP Requests by HTTP Host	530				0.0057	100%	0.1200	66.437
www.httpvshttps.com	361				0.0039	68.11%	0.0600	8.435
/check.png?16624846345772_99	1				0.0000	0.28%	0.0100	13.450
/check.png?16624846345772_98	1				0.0000	0.28%	0.0100	13.141
/check.png?16624846345772_97	1				0.0000	0.28%	0.0100	13.044
/check.png?16624846345772_96	1				0.0000	0.28%	0.0100	13.010
/check.png?16624846345772_95	1				0.0000	0.28%	0.0100	12.959
/check.png?16624846345772_94	1				0.0000	0.28%	0.0100	12.959
/check.png?16624846345772_93	1				0.0000	0.28%	0.0100	12.943
/check.png?16624846345772_92	1				0.0000	0.28%	0.0100	12.855
/check.png?16624846345772_91	1				0.0000	0.28%	0.0100	12.768
/check.png?16624846345772_90	1				0.0000	0.28%	0.0100	12.731
/check.png?16624846345772_9	1				0.0000	0.28%	0.0100	8.721
/check.png?16624846345772_89	1				0.0000	0.28%	0.0100	12.668
/check.png?16624846345772_88	1				0.0000	0.28%	0.0100	12.667
/check.png?16624846345772_87	1				0.0000	0.28%	0.0100	12.667
/check.png?16624846345772_86	1				0.0000	0.28%	0.0100	12.566
/check.png?16624846345772_85	1				0.0000	0.28%	0.0100	12.482
/check.png?16624846345772_84	1				0.0000	0.28%	0.0100	12.414
/check.png?16624846345772_83	1				0.0000	0.28%	0.0100	12.347
/check.png?16624846345772_82	1				0.0000	0.28%	0.0100	12.312
/check.png?16624846345772_81	1				0.0000	0.28%	0.0100	12.293
/check.png?16624846345772_80	1				0.0000	0.28%	0.0100	12.285
/check.png?16624846345772_8	1				0.0000	0.28%	0.0100	8.439
/check.png?16624846345772_79	1				0.0000	0.28%	0.0100	12.115
/check.png?16624846345772_78	1				0.0000	0.28%	0.0100	12.061
/check.png?16624846345772_77	1				0.0000	0.28%	0.0100	12.010

Display filter:

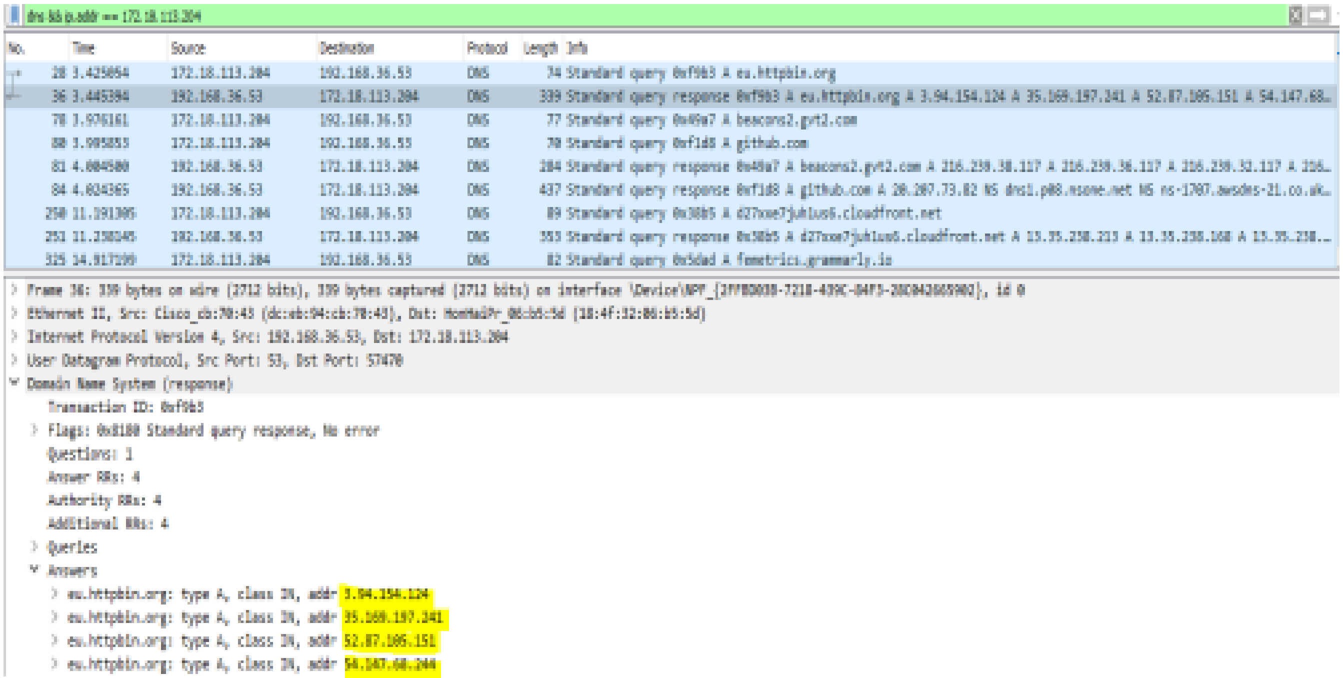
TASK 2:

Start the Wireshark packet sniffer and start capturing http://eu.httpbin.org shall be the website used. Note: If Google Chrome does not show an HTTP option in the schemes window at the top in the website, then use Firefox or other supportive browsers.

Enter the following URL into your browser http://eu.httpbin.org

Stop the packet capture when you have all the information captured, which is required to answer all the questions below.
1. What is/are the IP Addresses of this site ? How many DNS queries are sent from your browser (host machine) to DNS Server(s) ? How many DNS servers are involved ? Which DNS Server replies with actual IP Address(es). Do all DNS servers respond ? Clearly list the resource records involved in resolving the IP address of the site, mentioning, Name, value, type, TTL appropriately in the complete resolving process of this DNS conversation including query/queries and response/answer(s).

- IP address of the site are : **35.169.197.241 , 54.147.68.244, 3.94.154.14, 52.87.105.151** which can be found using the filter ‘dns’ and source IP address ‘172.18.113.204’ and observing the DNS packets.



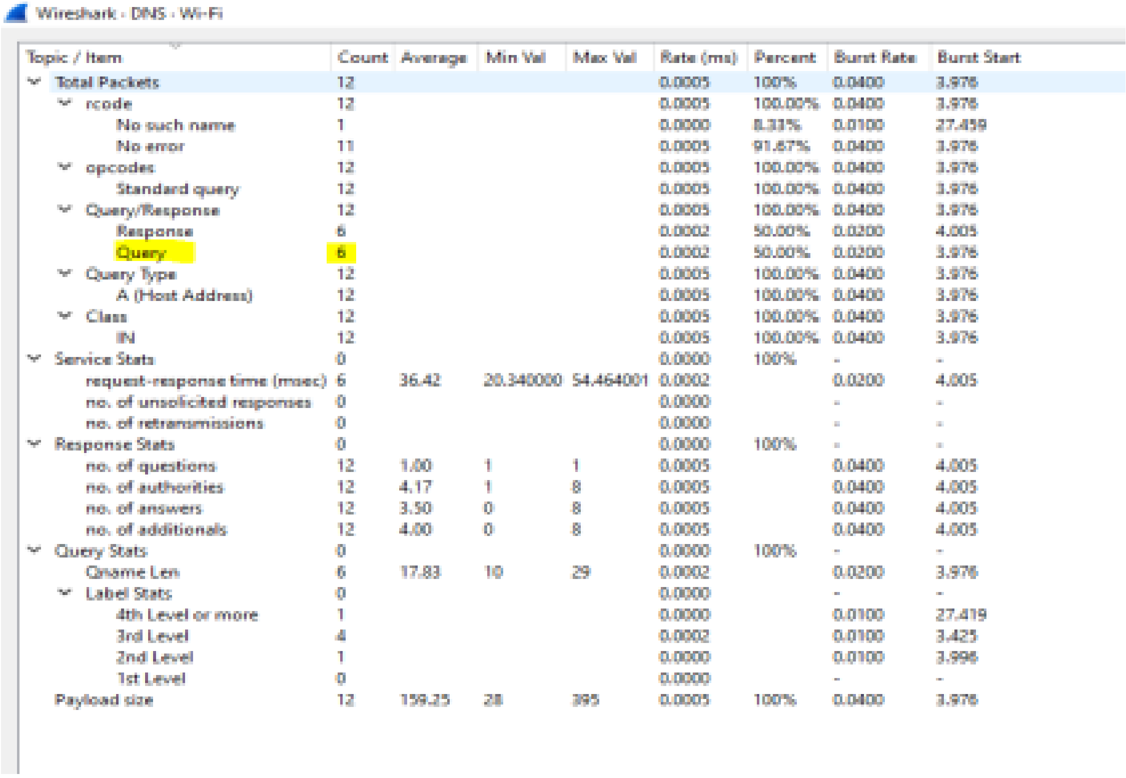
```
dns && ip.addr == 172.18.113.204

No.   Time        Source           Destination      Protocol Length Info
   28 3.425054    172.18.113.204   192.168.36.53    DNS      74 Standard query 0xf9b3 A eu.httpbin.org
   36 3.445394    192.168.36.53    172.18.113.204   DNS      339 Standard query response 0xf9b3 A eu.httpbin.org A 3.94.154.124 A 35.169.197.241 A 52.87.105.151 A 54.147.68…
   78 3.976161    172.18.113.204   192.168.36.53    DNS      77 Standard query 0x49a7 A beacons2.gvt2.com
   80 3.995853    172.18.113.204   192.168.36.53    DNS      70 Standard query 0xf1d8 A github.com
   81 4.004500    192.168.36.53    172.18.113.204   DNS      284 Standard query response 0x49a7 A beacons2.gvt2.com A 216.239.38.117 A 216.239.36.117 A 216.239.32.117 A 216…
   84 4.024365    192.168.36.53    172.18.113.204   DNS      437 Standard query response 0xf1d8 A github.com A 20.207.73.82 NS dns1.p08.nsone.net NS ns-1707.awsdns-21.co.uk…
  250 11.191305   172.18.113.204   192.168.36.53    DNS      89 Standard query 0x38b5 A d27xxe7juh1us6.cloudfront.net
  251 11.230145   192.168.36.53    172.18.113.204   DNS      353 Standard query response 0x38b5 A d27xxe7juh1us6.cloudfront.net A 13.35.238.213 A 13.35.238.168 A 13.35.238…
  325 14.917199   172.18.113.204   192.168.36.53    DNS      82 Standard query 0x5dad A femetrics.grammarly.io

> Frame 36: 339 bytes on wire (2712 bits), 339 bytes captured (2712 bits) on interface \Device\NPF_{2FFBD038-7218-439C-84F3-28D042665902}, id 0
> Ethernet II, Src: Cisco_cb:78:43 (dc:eb:94:cb:78:43), Dst: HonHaiPr_86:b5:5d (18:4f:32:86:b5:5d)
> Internet Protocol Version 4, Src: 192.168.36.53, Dst: 172.18.113.204
> User Datagram Protocol, Src Port: 53, Dst Port: 57470
v Domain Name System (response)
    Transaction ID: 0xf9b3
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 4
    Authority RRs: 4
    Additional RRs: 4
  > Queries
  v Answers
    > eu.httpbin.org: type A, class IN, addr 3.94.154.124
    > eu.httpbin.org: type A, class IN, addr 35.169.197.241
    > eu.httpbin.org: type A, class IN, addr 52.87.105.151
    > eu.httpbin.org: type A, class IN, addr 54.147.68.244
```

- Number of DNS queries sent from browser (host machine) to DNS Server i**s 6.**

This can be found in Statistics ->DNS by applying filter ‘**dns**’ and source IP address ‘**172.18.113.204**’ .

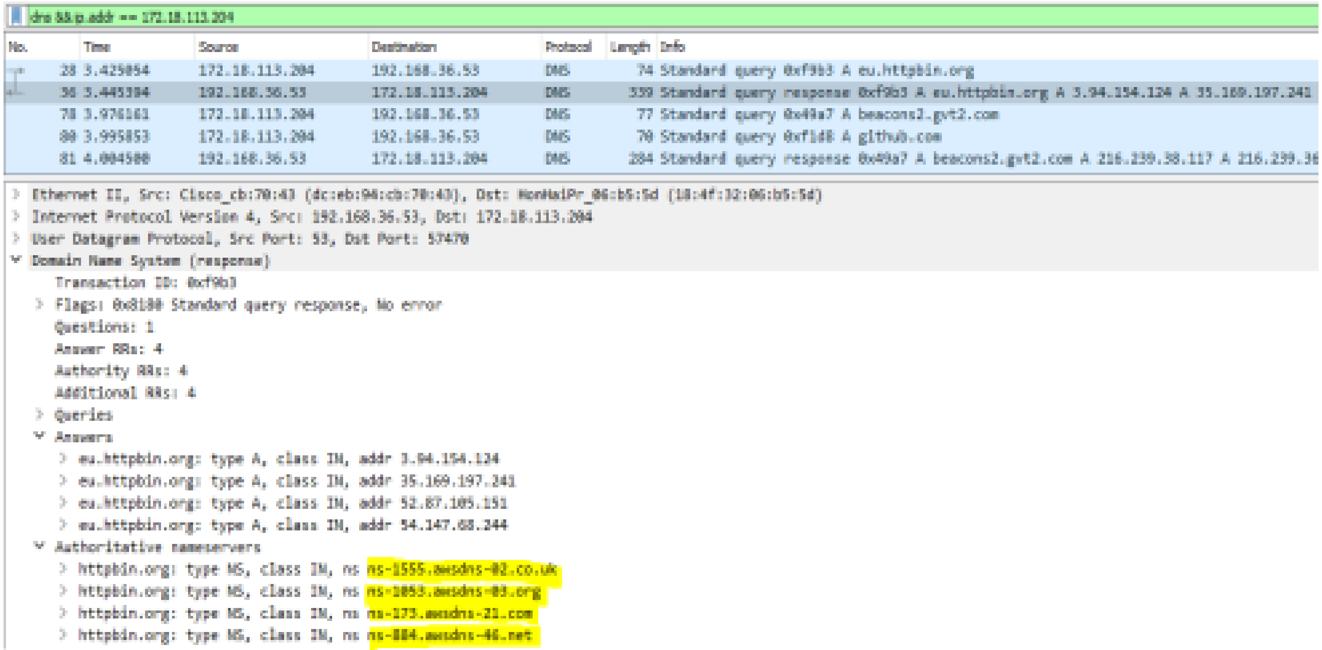


Wireshark · DNS · Wi-Fi

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total Packets	12				0.0005	100%	0.0400	3.976
rcode	12				0.0005	100.00%	0.0400	3.976
No such name	1				0.0000	8.33%	0.0100	27.459
No error	11				0.0005	91.67%	0.0400	3.976
opcodes	12				0.0005	100.00%	0.0400	3.976
Standard query	12				0.0005	100.00%	0.0400	3.976
Query/Response	12				0.0005	100.00%	0.0400	3.976
Response	6				0.0002	50.00%	0.0200	4.005
Query	6				0.0002	50.00%	0.0200	3.976
Query Type	12				0.0005	100.00%	0.0400	3.976
A (Host Address)	12				0.0005	100.00%	0.0400	3.976
Class	12				0.0005	100.00%	0.0400	3.976
IN	12				0.0005	100.00%	0.0400	3.976
Service Stats	0				0.0000	100%	-	-
request-response time (msec)	6	36.42	20.340000	54.464001	0.0002		0.0200	4.005
no. of unsolicited responses	0				0.0000		-	-
no. of retransmissions	0				0.0000		-	-
Response Stats	0				0.0000	100%	-	-
no. of questions	12	1.00	1	1	0.0005		0.0400	4.005
no. of authorities	12	4.17	1	8	0.0005		0.0400	4.005
no. of answers	12	3.50	0	8	0.0005		0.0400	4.005
no. of additionals	12	4.00	0	8	0.0005		0.0400	4.005
Query Stats	0				0.0000	100%	-	-
Qname Len	6	17.83	10	29	0.0002		0.0200	3.976
Label Stats	0				0.0000		-	-
4th Level or more	1				0.0000		0.0100	27.419
3rd Level	4				0.0002		0.0100	3.425
2nd Level	1				0.0000		0.0100	3.996
1st Level	0				0.0000		-	-
Payload size	12	159.25	28	395	0.0005	100%	0.0400	3.976

- Number of DNS Servers involved are **4**.

Apply the filter ‘dns’ and source IP address ‘172.18.113.204’ ,this can be found by observing ‘**Authoritative nameservers**’ under DNS .

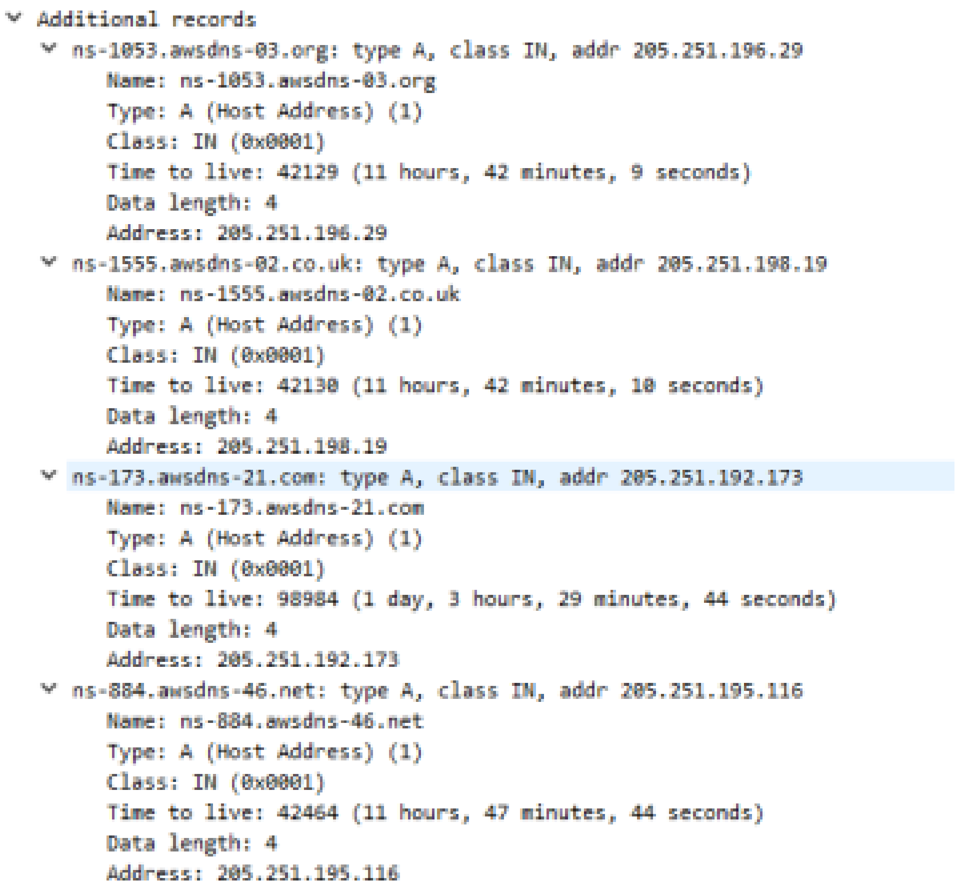


```
dns && ip.addr == 172.18.113.204

No.   Time        Source           Destination      Protocol  Length  Info
   28 3.425054    172.18.113.204   192.168.36.53    DNS          74   Standard query 0xf9b3 A eu.httpbin.org
   36 3.445394    192.168.36.53    172.18.113.204   DNS         339   Standard query response 0xf9b3 A eu.httpbin.org A 3.94.154.124 A 35.169.197.241
   78 3.976161    172.18.113.204   192.168.36.53    DNS          77   Standard query 0x49a7 A beacons2.gvt2.com
   80 3.995853    172.18.113.204   192.168.36.53    DNS          70   Standard query 0xf1d8 A github.com
   81 4.004500    192.168.36.53    172.18.113.204   DNS         284   Standard query response 0x49a7 A beacons2.gvt2.com A 216.239.38.117 A 216.239.36

> Ethernet II, Src: Cisco_cb:70:43 (dc:eb:94:cb:70:43), Dst: HonHaiPr_06:b5:5d (18:4f:32:06:b5:5d)
> Internet Protocol Version 4, Src: 192.168.36.53, Dst: 172.18.113.204
> User Datagram Protocol, Src Port: 53, Dst Port: 57470
v Domain Name System (response)
    Transaction ID: 0xf9b3
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 4
    Authority RRs: 4
    Additional RRs: 4
  > Queries
  v Answers
    > eu.httpbin.org: type A, class IN, addr 3.94.154.124
    > eu.httpbin.org: type A, class IN, addr 35.169.197.241
    > eu.httpbin.org: type A, class IN, addr 52.87.105.151
    > eu.httpbin.org: type A, class IN, addr 54.147.68.244
  v Authoritative nameservers
    > httpbin.org: type NS, class IN, ns ns-1555.awsdns-02.co.uk
    > httpbin.org: type NS, class IN, ns ns-1053.awsdns-03.org
    > httpbin.org: type NS, class IN, ns ns-173.awsdns-21.com
    > httpbin.org: type NS, class IN, ns ns-884.awsdns-46.net
```

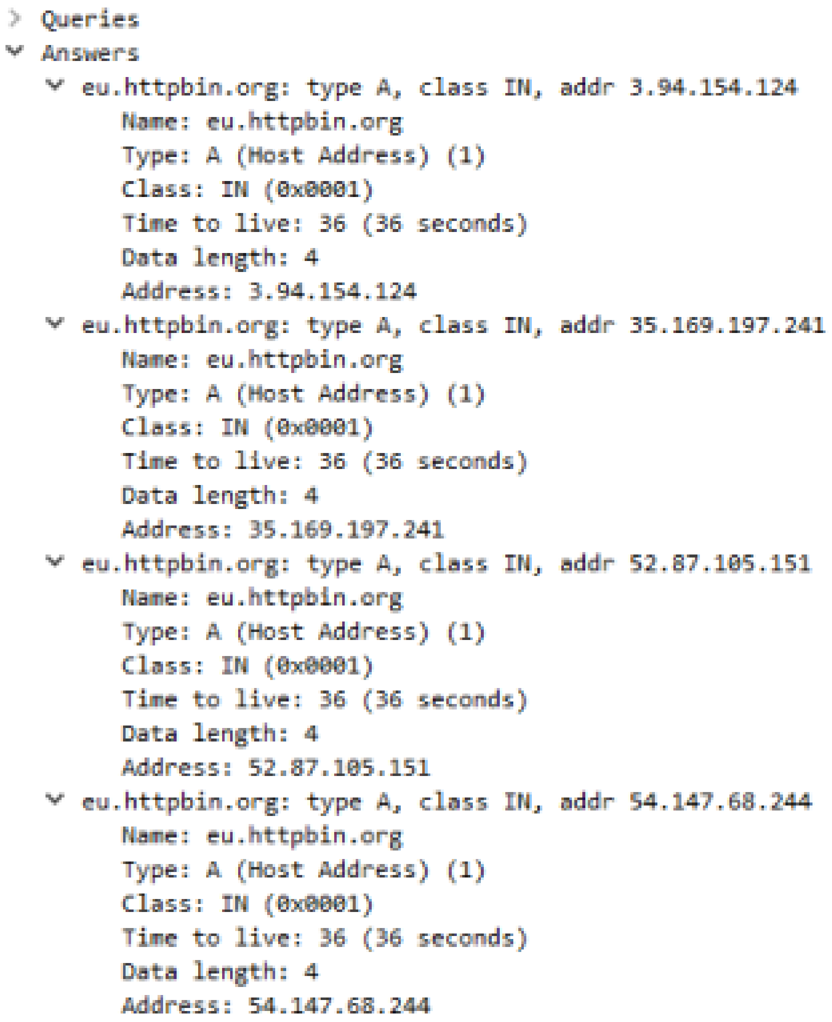
- The DNS servers which replies with actual IP Address are :



```
Additional records
  ns-1053.awsdns-03.org: type A, class IN, addr 205.251.196.29
    Name: ns-1053.awsdns-03.org
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 42129 (11 hours, 42 minutes, 9 seconds)
    Data length: 4
    Address: 205.251.196.29
  ns-1555.awsdns-02.co.uk: type A, class IN, addr 205.251.198.19
    Name: ns-1555.awsdns-02.co.uk
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 42130 (11 hours, 42 minutes, 10 seconds)
    Data length: 4
    Address: 205.251.198.19
  ns-173.awsdns-21.com: type A, class IN, addr 205.251.192.173
    Name: ns-173.awsdns-21.com
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 98984 (1 day, 3 hours, 29 minutes, 44 seconds)
    Data length: 4
    Address: 205.251.192.173
  ns-884.awsdns-46.net: type A, class IN, addr 205.251.195.116
    Name: ns-884.awsdns-46.net
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 42464 (11 hours, 47 minutes, 44 seconds)
    Data length: 4
    Address: 205.251.195.116
```

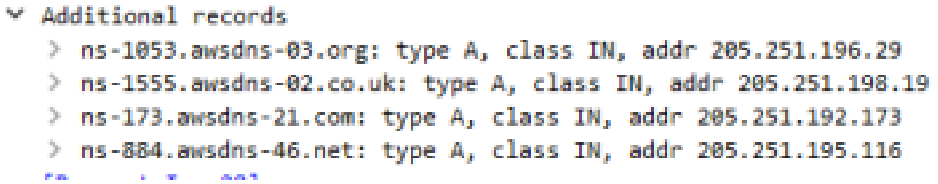
- Yes, all DNS servers respond.

- Name, value, type, TTL is given as follows:



```
> Queries
v Answers
  v eu.httpbin.org: type A, class IN, addr 3.94.154.124
      Name: eu.httpbin.org
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 36 (36 seconds)
      Data length: 4
      Address: 3.94.154.124
  v eu.httpbin.org: type A, class IN, addr 35.169.197.241
      Name: eu.httpbin.org
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 36 (36 seconds)
      Data length: 4
      Address: 35.169.197.241
  v eu.httpbin.org: type A, class IN, addr 52.87.105.151
      Name: eu.httpbin.org
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 36 (36 seconds)
      Data length: 4
      Address: 52.87.105.151
  v eu.httpbin.org: type A, class IN, addr 54.147.68.244
      Name: eu.httpbin.org
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 36 (36 seconds)
      Data length: 4
      Address: 54.147.68.244
```

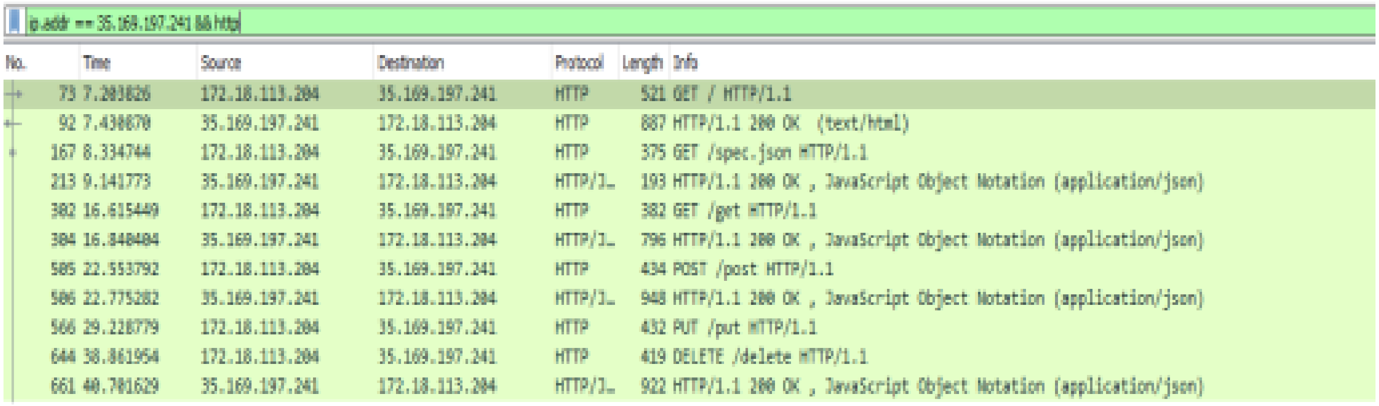
- Resource records involved in resolving the IP address are:



```
v Additional records
  > ns-1053.awsdns-03.org: type A, class IN, addr 205.251.196.29
  > ns-1555.awsdns-02.co.uk: type A, class IN, addr 205.251.198.19
  > ns-173.awsdns-21.com: type A, class IN, addr 205.251.192.173
  > ns-884.awsdns-46.net: type A, class IN, addr 205.251.195.116
```

2. Browse through various options in the site and perform GET/POST/PUT/DELETE operations on various request URIs provided by the site. How many HTTP requests (Type and respective count of requests), responses (status code and phrase of each of the responses) did the browser send and receive ? Clearly show the statistics of Wireshark.

- After performing GET/POST/PUT/DELETE operations, to know the packets apply the filter ‘http’ and ‘ip.addr == 35.169.197.241’



ip.addr == 35.169.197.241 && http

No.	Time	Source	Destination	Protocol	Length	Info
73	7.203826	172.18.113.204	35.169.197.241	HTTP	521	GET / HTTP/1.1
92	7.430870	35.169.197.241	172.18.113.204	HTTP	887	HTTP/1.1 200 OK (text/html)
167	8.334744	172.18.113.204	35.169.197.241	HTTP	375	GET /spec.json HTTP/1.1
213	9.141773	35.169.197.241	172.18.113.204	HTTP/J…	193	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
382	16.615449	172.18.113.204	35.169.197.241	HTTP	382	GET /get HTTP/1.1
384	16.840484	35.169.197.241	172.18.113.204	HTTP/J…	796	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
505	22.553792	172.18.113.204	35.169.197.241	HTTP	434	POST /post HTTP/1.1
506	22.775282	35.169.197.241	172.18.113.204	HTTP/J…	948	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)
566	29.228779	172.18.113.204	35.169.197.241	HTTP	432	PUT /put HTTP/1.1
644	38.861954	172.18.113.204	35.169.197.241	HTTP	419	DELETE /delete HTTP/1.1
661	40.701629	35.169.197.241	172.18.113.204	HTTP/J…	922	HTTP/1.1 200 OK , JavaScript Object Notation (application/json)

➢ Using this filter and checking the Packet Counter of HTTP statistics, we see that a total of **11** packets are sent between the server and browser.

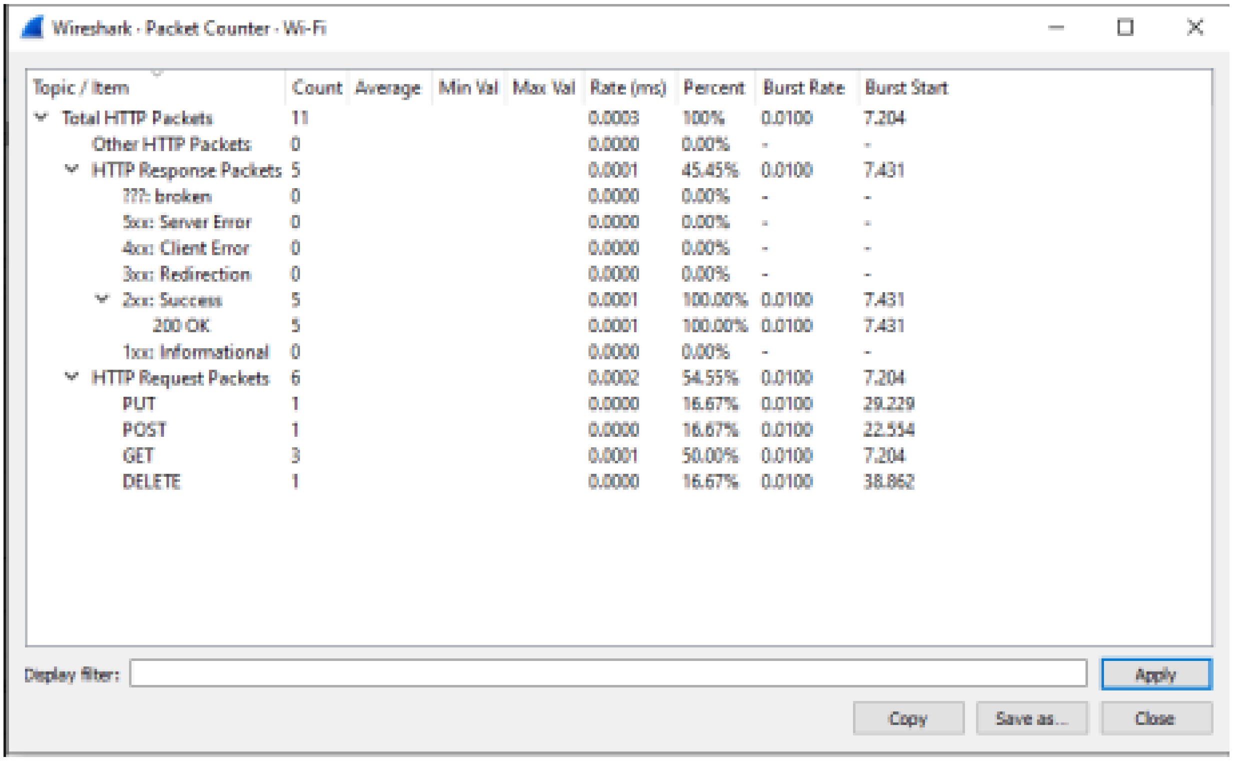
Out of these, there are 6 are request packets which are entirely composed of

- PUT requests | Count : 1
- POST requests | Count : 1
- GET requests | Count : 3
- DELETE requests | Count : 1

.

And, there are 5 HTTP Response packets, the distribution of these is as follows:

- Count: 5 | Status: 200 | Phrase: OK



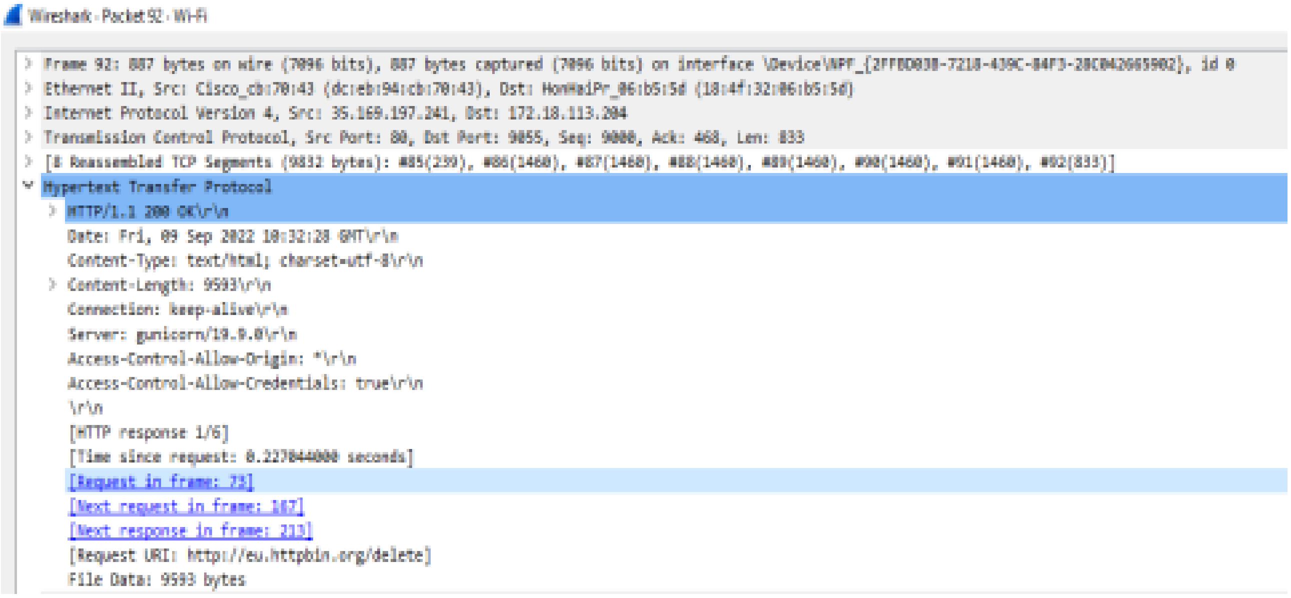
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total HTTP Packets	11				0.0003	100%	0.0100	7.204
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	5				0.0001	45.45%	0.0100	7.431
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	0				0.0000	0.00%	-	-
2xx: Success	5				0.0001	100.00%	0.0100	7.431
200 OK	5				0.0001	100.00%	0.0100	7.431
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	6				0.0002	54.55%	0.0100	7.204
PUT	1				0.0000	16.67%	0.0100	29.229
POST	1				0.0000	16.67%	0.0100	22.554
GET	3				0.0001	50.00%	0.0100	7.204
DELETE	1				0.0000	16.67%	0.0100	38.862

3. Make a detailed list including for each object/file downloaded what is the time taken for downloading the objects, the size of the object downloaded, object name, last modified

time at the server. At least 5 such objects' details shall be provided. Ensure to perform the enough number of operations in step#1 mentioned above to ensure that Wireshark has enough packets captured to answer this question.

To know the object/file downloaded, go to File-> Export Objects-> HTTP**.**

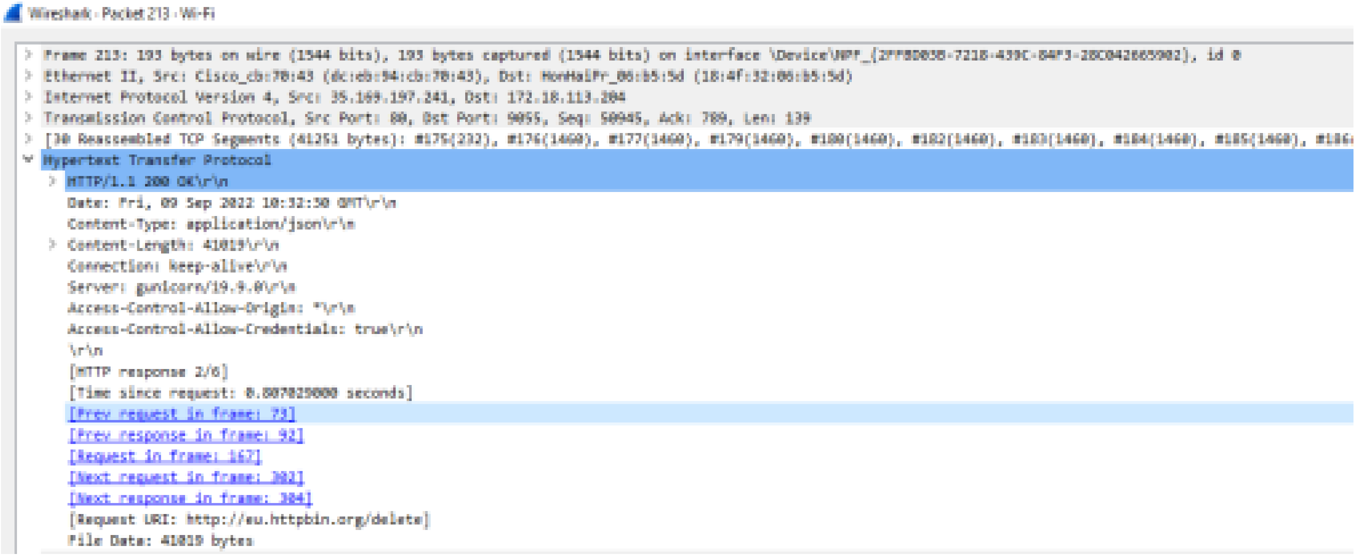
1. **Packet: 92 | Time taken : 0.227 sec | Size : 9593 bytes | Name: \**



Wireshark · Packet 92 · Wi-Fi

```
> Frame 92: 887 bytes on wire (7096 bits), 887 bytes captured (7096 bits) on interface \Device\NPF_{2FF8D038-7218-439C-84F3-28C042665902}, id 0
> Ethernet II, Src: Cisco_cb:70:43 (dc:eb:94:cb:70:43), Dst: HonHaiPr_06:b5:5d (18:4f:32:06:b5:5d)
> Internet Protocol Version 4, Src: 35.169.197.241, Dst: 172.18.113.204
> Transmission Control Protocol, Src Port: 80, Dst Port: 9055, Seq: 9000, Ack: 468, Len: 833
> [8 Reassembled TCP Segments (9832 bytes): #85(239), #86(1460), #87(1460), #88(1460), #89(1460), #90(1460), #91(1460), #92(833)]
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Fri, 09 Sep 2022 10:32:28 GMT\r\n
    Content-Type: text/html; charset=utf-8\r\n
  > Content-Length: 9593\r\n
    Connection: keep-alive\r\n
    Server: gunicorn/19.9.0\r\n
    Access-Control-Allow-Origin: *\r\n
    Access-Control-Allow-Credentials: true\r\n
    \r\n
    [HTTP response 1/6]
    [Time since request: 0.227844000 seconds]
    [Request in frame: 73]
    [Next request in frame: 167]
    [Next response in frame: 213]
    [Request URI: http://eu.httpbin.org/delete]
    File Data: 9593 bytes
```

2. **Packet: 213 | Time taken : 0.807 sec | Size : 41019 bytes | Name: spec.json**



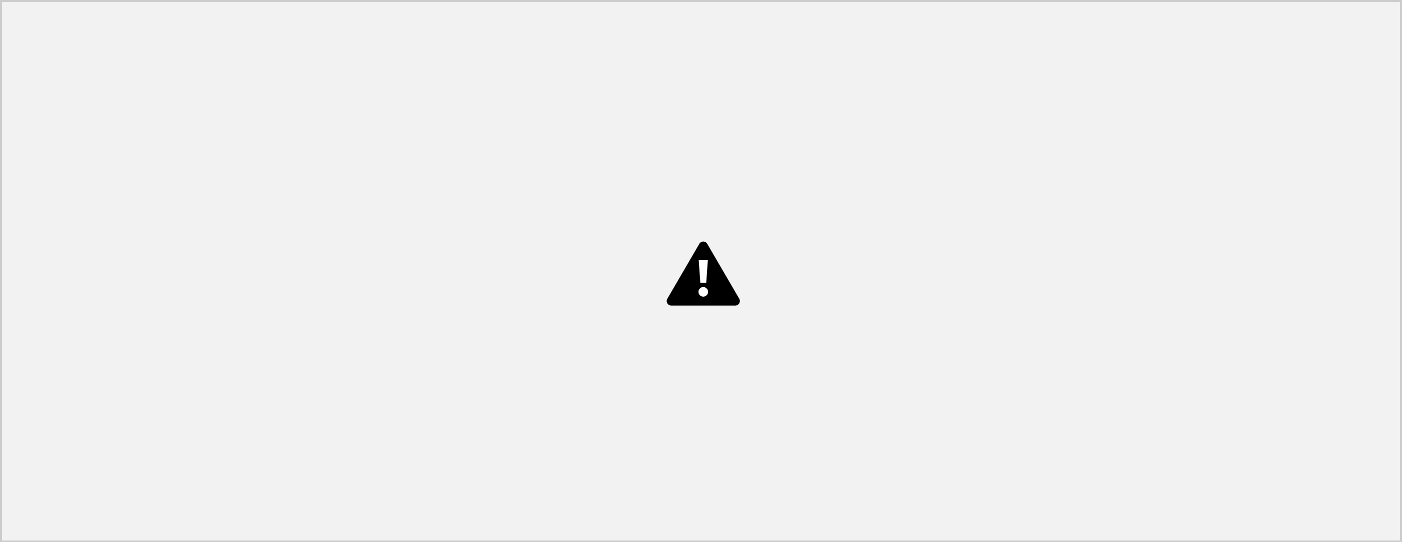
Wireshark · Packet 213 · Wi-Fi

```
> Frame 213: 193 bytes on wire (1544 bits), 193 bytes captured (1544 bits) on interface \Device\NPF_{2FF8D038-7218-439C-84F3-28C042665902}, id 0
> Ethernet II, Src: Cisco_cb:70:43 (dc:eb:94:cb:70:43), Dst: HonHaiPr_06:b5:5d (18:4f:32:06:b5:5d)
> Internet Protocol Version 4, Src: 35.169.197.241, Dst: 172.18.113.204
> Transmission Control Protocol, Src Port: 80, Dst Port: 9055, Seq: 50945, Ack: 789, Len: 139
> [30 Reassembled TCP Segments (41251 bytes): #175(232), #176(1460), #177(1460), #179(1460), #180(1460), #182(1460), #183(1460), #184(1460), #185(1460), #186
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Fri, 09 Sep 2022 10:32:30 GMT\r\n
    Content-Type: application/json\r\n
  > Content-Length: 41019\r\n
    Connection: keep-alive\r\n
    Server: gunicorn/19.9.0\r\n
    Access-Control-Allow-Origin: *\r\n
    Access-Control-Allow-Credentials: true\r\n
    \r\n
    [HTTP response 2/6]
    [Time since request: 0.807025000 seconds]
    [Prev request in frame: 73]
    [Prev response in frame: 92]
    [Request in frame: 167]
    [Next request in frame: 302]
    [Next response in frame: 304]
    [Request URI: http://eu.httpbin.org/delete]
    File Data: 41019 bytes
```

3. **Packet: 304 | Time taken : 0.224 sec | Size : 512 bytes | Name: get**

4. Packet: 506 | Time taken : 0.221 sec | Size : 644 bytes | Name: put

5. Packet: 661 | Time taken : 1.839 sec | Size : 618 bytes | Name: delete



4. How many times does the browser ask the site to keep the connection alive ?

Browser asked the site to keep the connection alive for **2 times.**

5. Which version of the HTTP is your browser running ?

The version of HTTP used is **HTTP 1.1**

TASK 3:

Before starting the steps below, make sure your browser’s cache is empty.

Steps:

Start the Wireshark packet sniffer and start capturing..

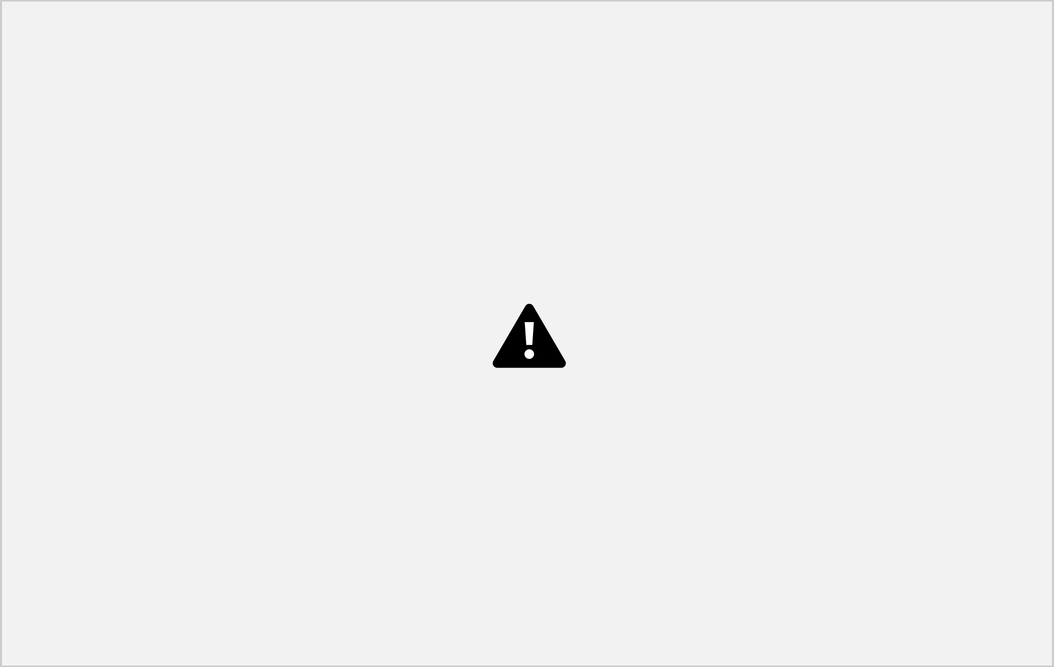
Enter the following URL into your browser http://eu.httpbin.org

Quickly enter the same URL into your browser again (or simply select the refresh button on your browser)
Stop Wireshark packet capture.

1. How many conditional GETs are sent by browser to the server ?

NOTE: As conditional GETs and its respective packets are not seen, I have used different website i.e; http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

According to this website, number of conditional GETs obtained is **1.**



2. Make a list for each of the file/object downloaded when the site was loaded, how many times the server sends the full contents of the respective file/object ?

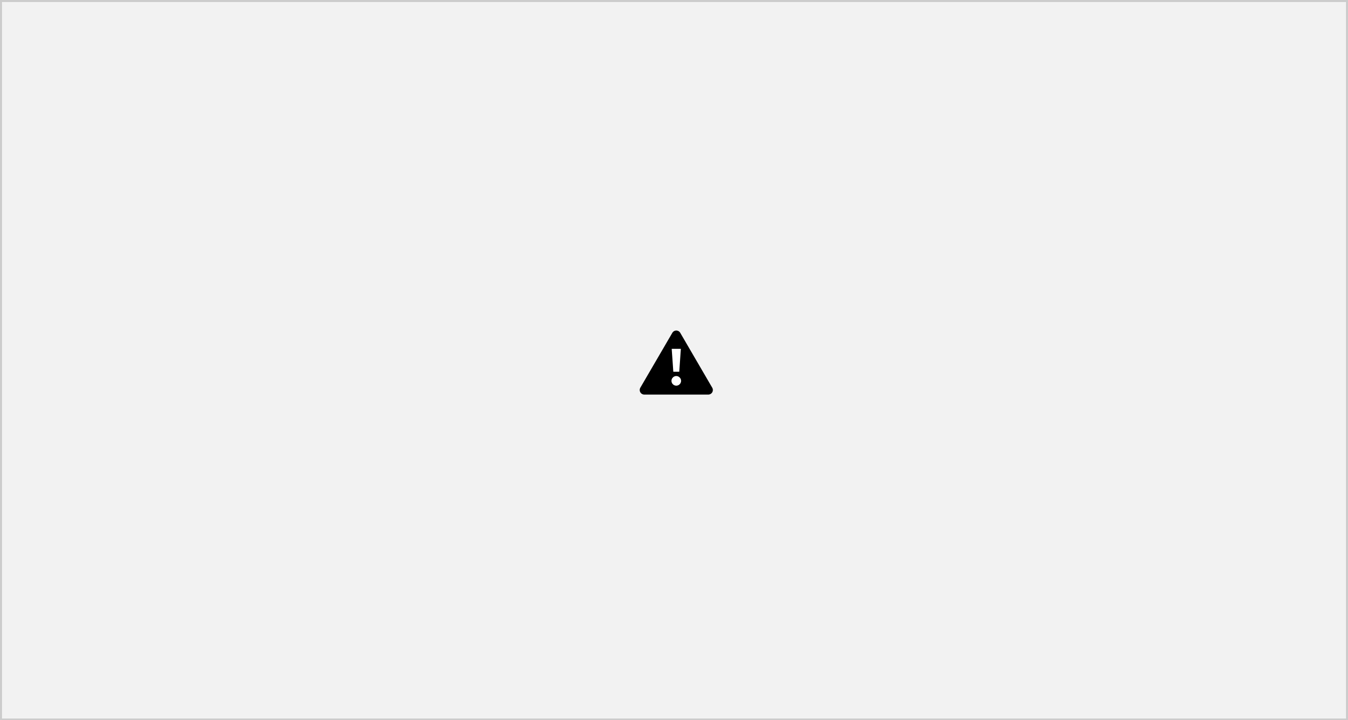
Number of file/object downloaded when the site was loaded is **1.**
Details is as follows:

Packet : 101 | Size: 371 bytes | Filename: HTTP-wireshark-file2.html

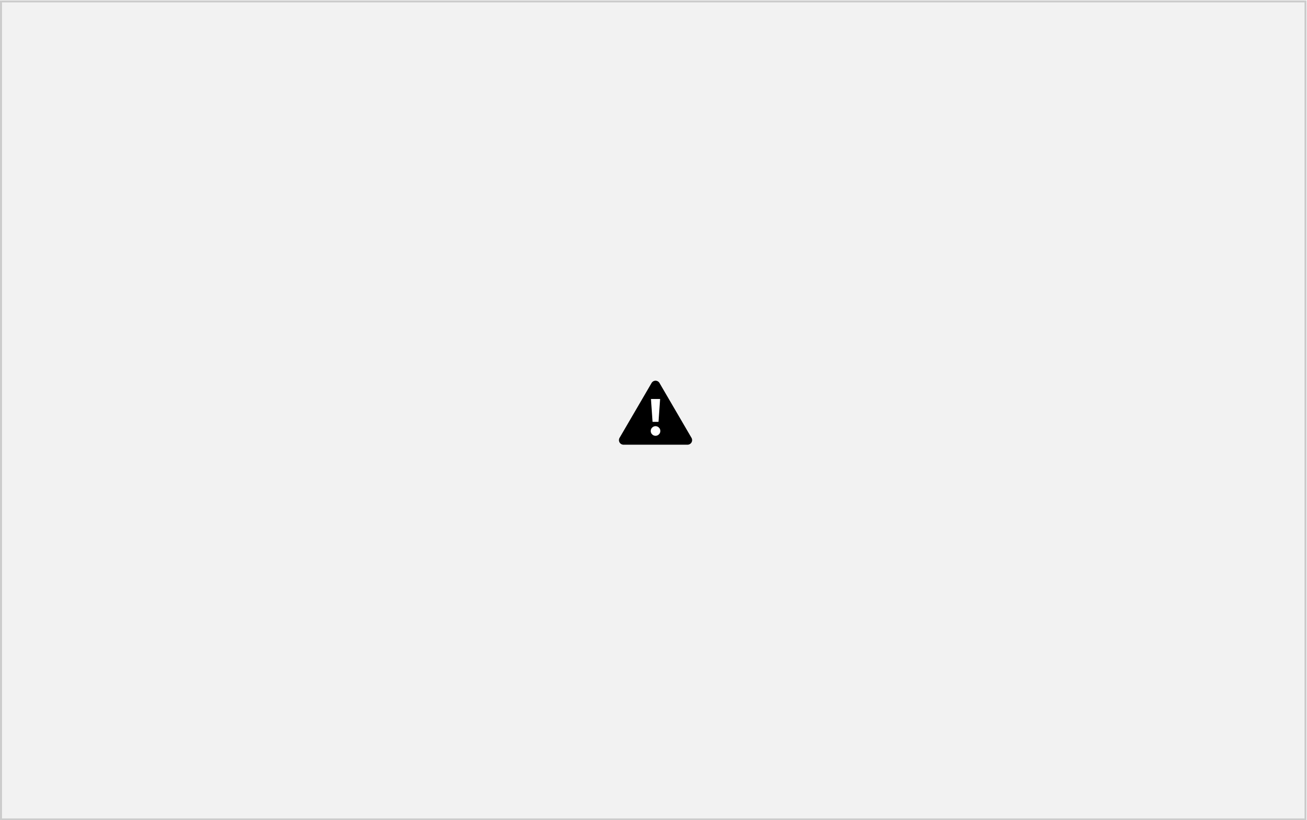
Only once the server sends the full contents of the respective file/object as it gets stored in cache and next time when we load the contents are obtained from the cache .

3. Explain in detail what is the difference in server's behavior between first and second request/browsing ?

When we send the request for the first time, text/html is downloaded with status 200 OK and this gets stored inside cache now.

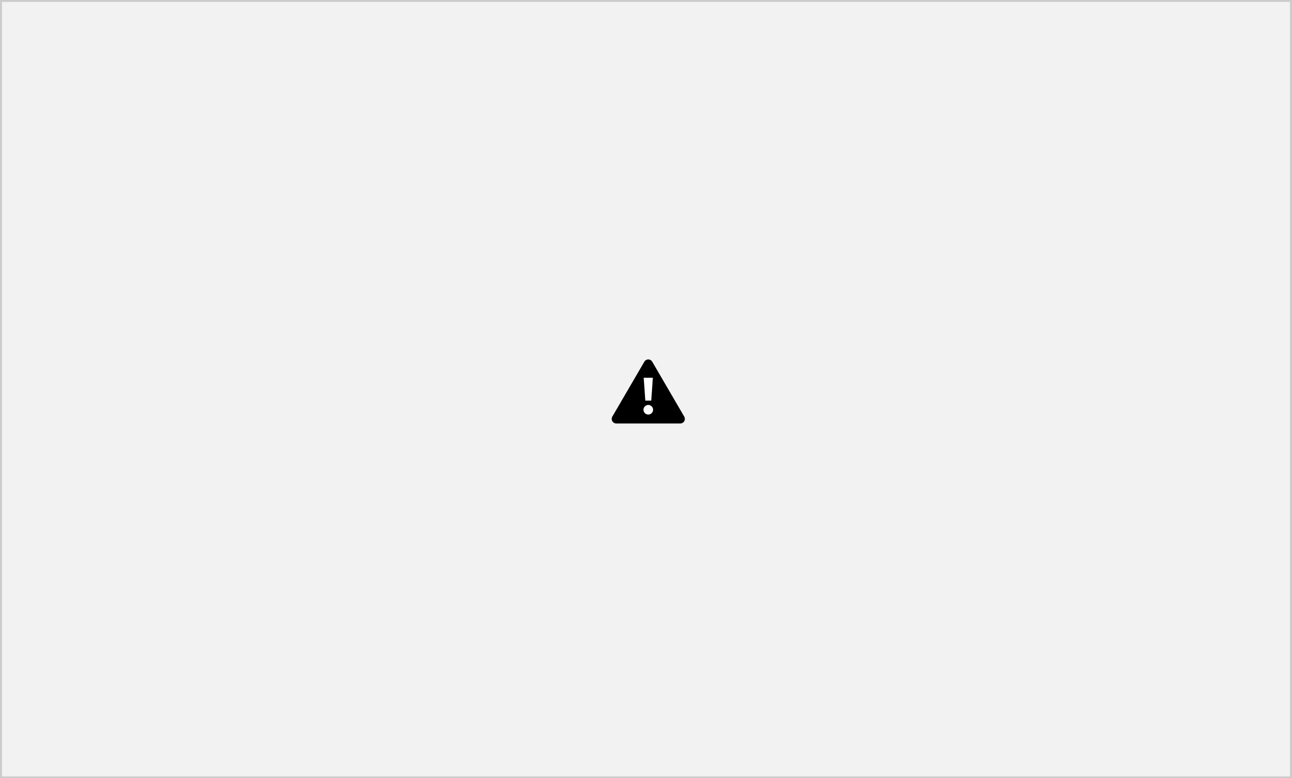


Whereas, when we run the website for second time, server captures the packets from the cache only and hence we can see the Response Phrase as '**Not Modified'**.



4. List the headers of HTTP which influence this functionality in question#3 above.

If HTTP request contains ‘**If-None-Match’** and ‘**If-Modified-Since**’ this means that the packet contains were taken from the cache.



ETag is shown if HTTP packet is not modified.

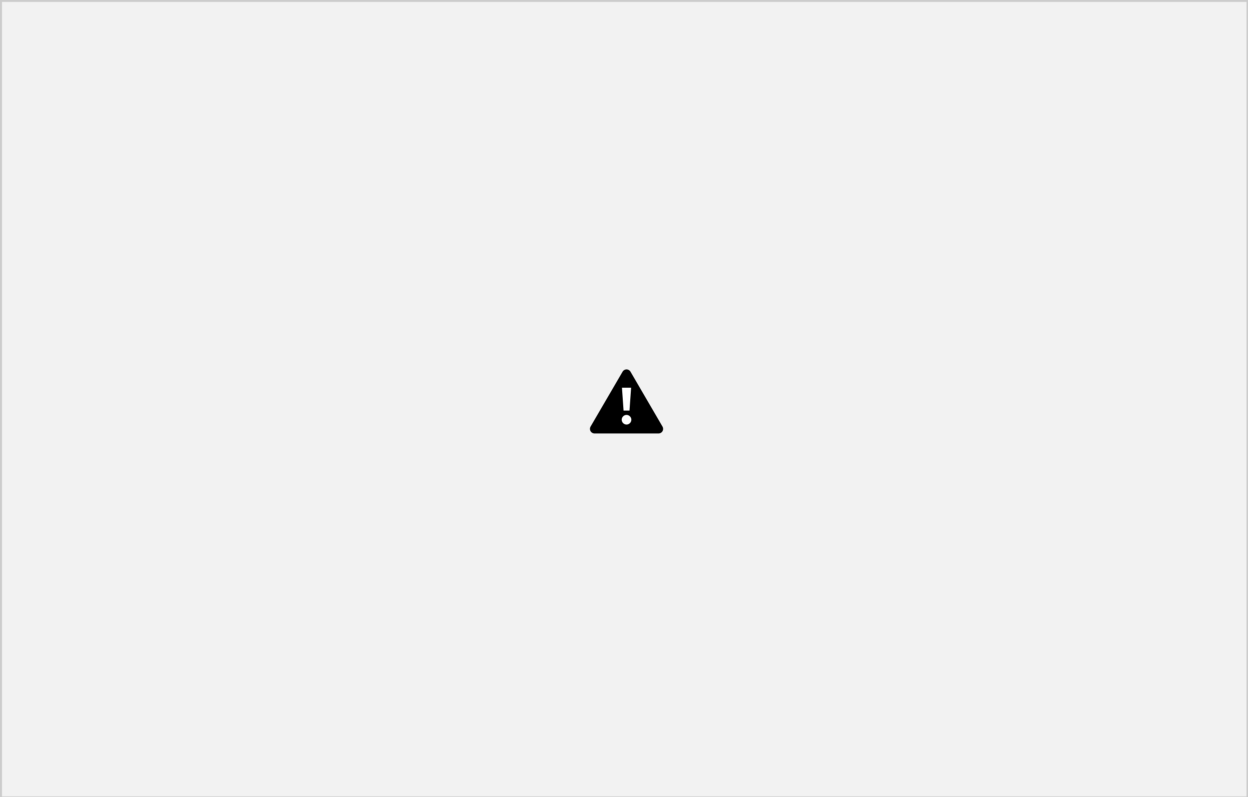
TASK 4:

Start the Wireshark packet sniffer and start capturing. You may find the IP address of the sites listed here using nslookup command and use it appropriately for wireshark filtering purposes.
Using the telnet command line interface, telnet to nghttp2.org, eu.httpbin.org, on port 80. Perform the following for each of these site names listed, for both HTTP 1.1 and 1.0

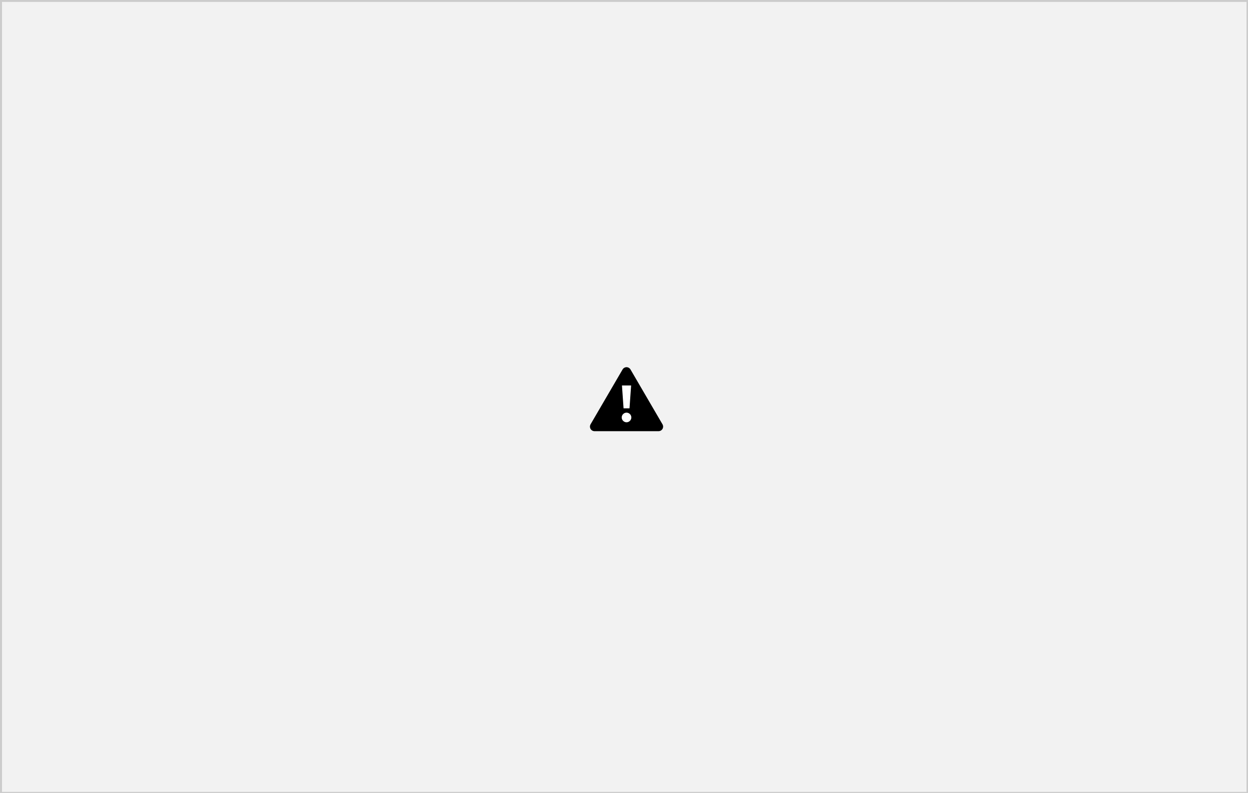
1. Inside the telnet interface shell, fetch the home page/index contents of the site using appropriate commands of telnet

The IP address of the site **nghttp2.org** is as follows:

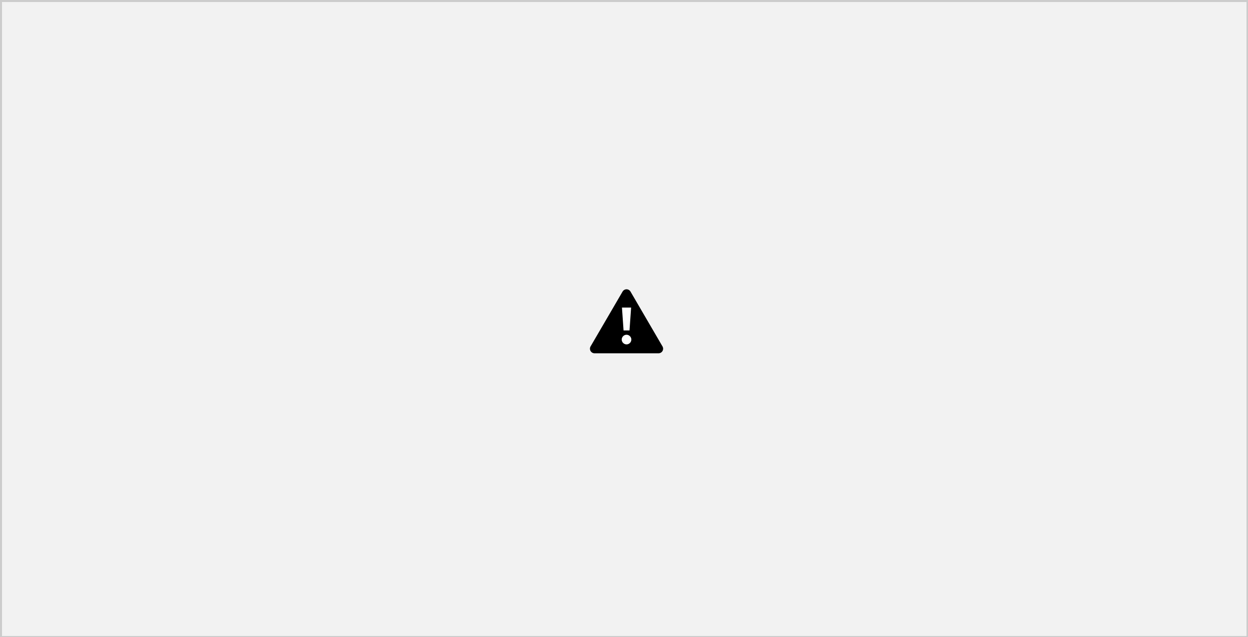
Using Telnet command to **nghttp2.org** on port 80 for HTTP 1.0



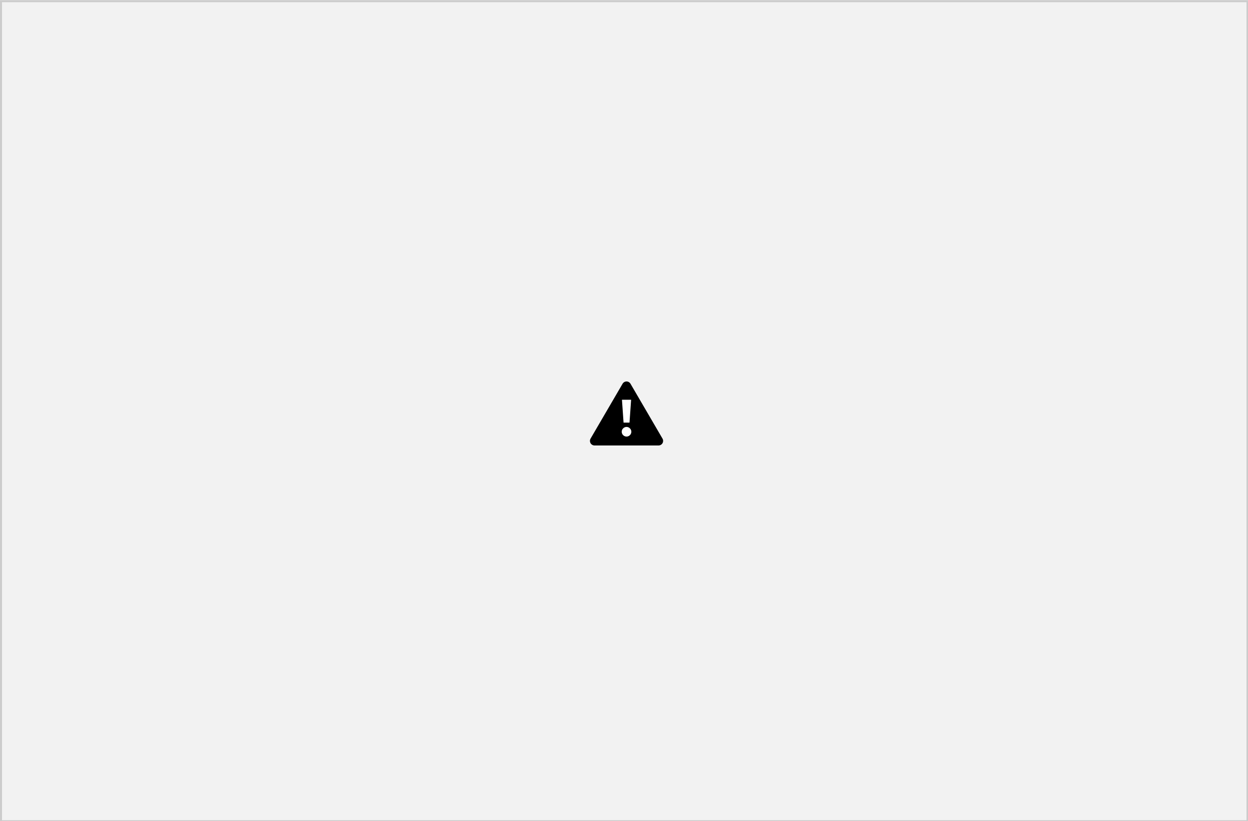
Using Telnet command to **nghttp2.org** on port 80 for HTTP 1.1



The IP address of the site **eu.httpbin.org** is as follows:



Using Telnet command to **eu.httpbin.org** on port 80 for HTTP 1.0



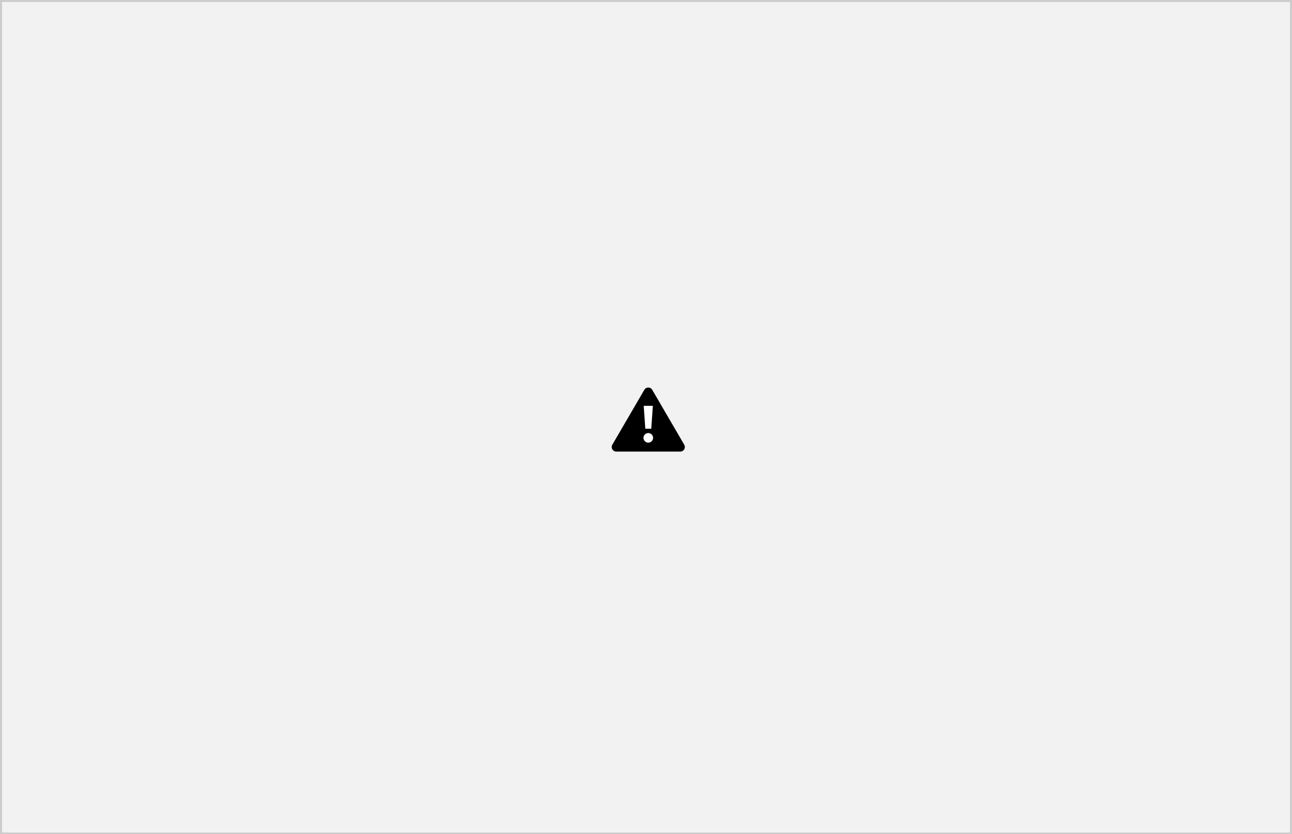
Using Telnet command to **eu.httpbin.org** on port 80 for HTTP 1.1

a. Is the site HTTP persistent ?

Both the sites are HTTP persistent as we know that HTTP 1.1 is a Persistent connection and it keeps the connection open and allows multiple requests using Keep Alive Timer

b. If the site is not persistent, what do you do to make it persistent from the telnet shell ?

We can use **HTTP keepalive or HTTP Connection** (as shown in the below screenshot) to make non-persistent to make persistent.



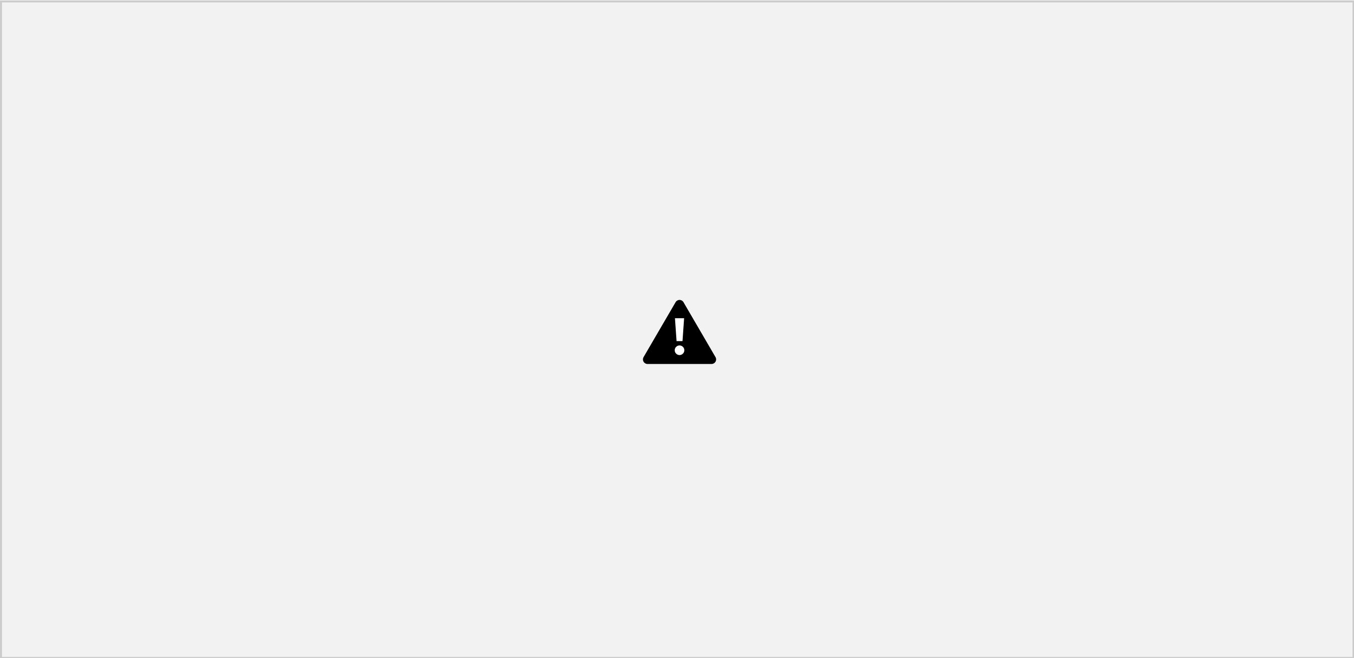
c. Once the required contents are fetched for analysis, is the connection to the site closed immediately. If so why/if not why ? Who is closing the connection ? Why ? And what is the time period before the connection termination is triggered

HTTP 1.0 is a Non Persistant connection .
Non persistent connection means connection means as soon as you receive the response the connection gets closed. The connection gets closed by using FIN command. To get new object you have to initiate the connection again.

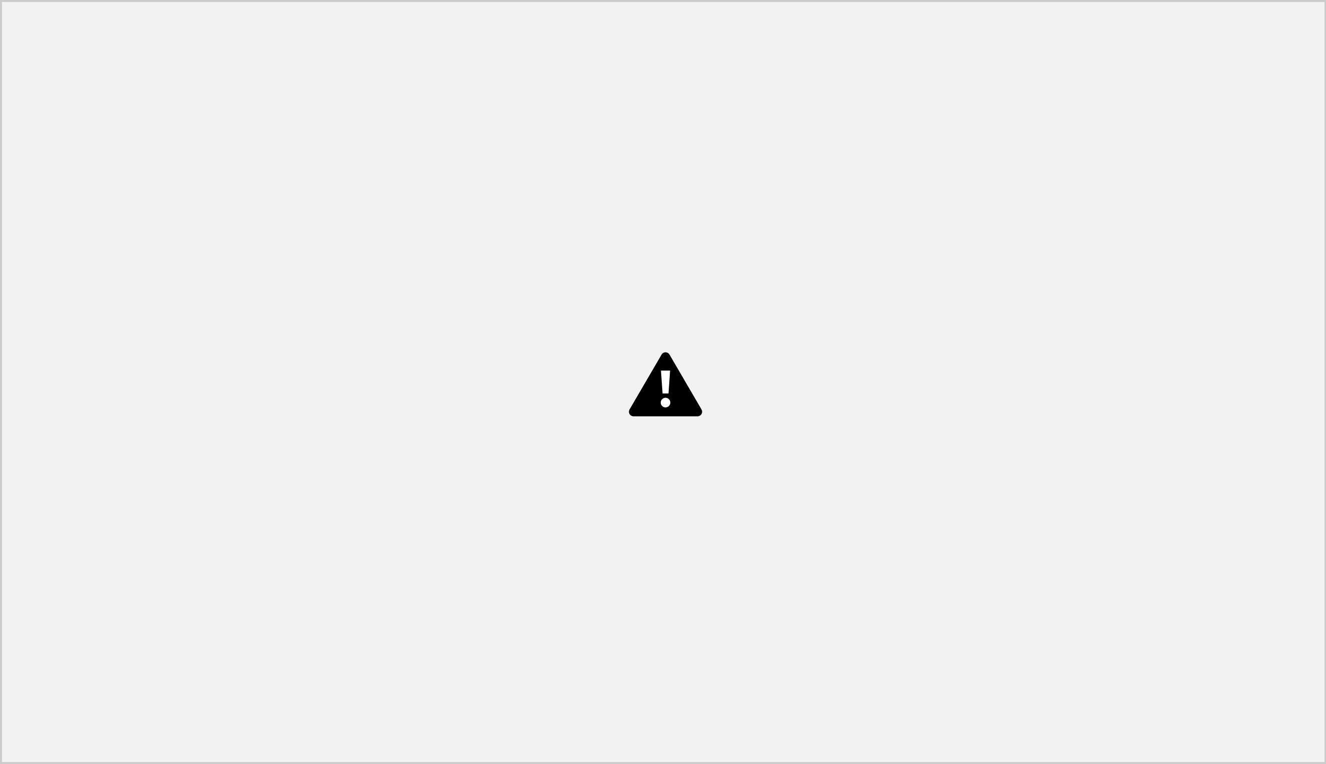
Whereas, HTTP 1.1 is persistent which means connections are kept open for some time called as Keep Alive Time which allow multiple requests to be sent in a single connection.

The time taken to close the connection the connection for **eu.httpbin.org** using HTTP 1.0 is **14.79 sec.**

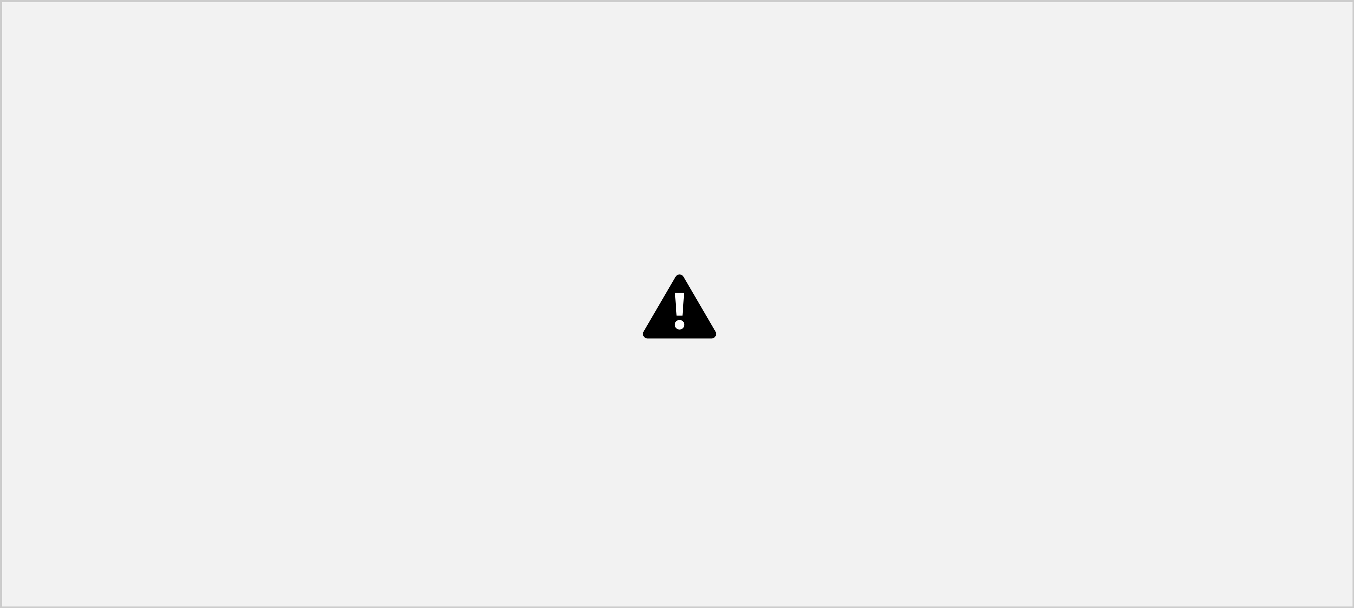
The time taken to close the connection the connection for **eu.httpbin.org** using HTTP 1.0 is **19.79 sec.**



The time taken to close the connection the connection for **nghttp2.org** using HTTP 1.1 is **86.914 sec**



Thetimetakentoclosetheconnectiontheconnectionfor**eu.httpbin.org**usingHTTP1.1 is **75.88sec**



PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not

previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honor violations by other students if I become aware of it.

Name of the student : SHRUSTI
Roll No: CS22MTECH11017