



University of Tehran
College of Engineering
School of Electrical and Computer Engineering



Computer Networks

Dr.Shah-Mansouri

Wireshark Lab 2

Soroush Mesforush Mashhad

SN:810198472

Tir 01

Contents

1	DNS	4
1.1	Q1	4
1.2	Q2	6
1.3	Q3	8
1.4	Q4	9
1.5	Q5	10
1.6	Q6	11
1.7	Q7	12
2	HTTP	14
2.1	Q1	14
2.2	Q2	15
2.3	Q3	15
2.3.1	First GET response	15
2.3.2	Second GET response	16
2.3.3	Third GET response	17

Abstract

In this assignment, our goal is to get familiar with Ethernet and ARP(address resolution protocol).

In the first part, we shall attempt to capture and analyze the IP headers as instructed in the assignment description using Wireshark.

In the next section we go on to observe the ARP protocol in action, we pay attention that the ARP protocol normally maintains a cache of IP-to-Ethernet address translation pairs in our computer. We then go on to satisfy the assignment's requirements.

In the final part we shall observe the DHCP protocol, we need to carry out this part in a place where we have a dynamically assigned IP address.

1 DNS

First of all, I performed a quick search on the internet to find a website still using the HTTP protocol, surprisingly the university of Washington with the following address uses this protocol. <http://washington.edu/>

Now we go on to perform the desired tasks accordingly, our command line after cleaning the DNS history is as follows.

```
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Soroush>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\Soroush>
```

Figure 1: DNS flushing

Now we go on to answer the questions.

1.1 Q1

A snip of the captured DNS sequences are as follows.

No.	Time	Source	Destination	Protocol	Length	Info
...	1.945117	172.18.170.2	192.168.20.15	DNS	86	Standard query 0xf6d9 A encrypted-tbn0.gstatic.com
...	1.946888	192.168.20.15	172.18.170.2	DNS	357	Standard query response 0xf6d9 A encrypted-tbn0.gstatic.com A 142.250.187.142 NS ns4.google.com NS ns1.google.com NS ns3.google.com
...	4.440766	172.18.170.2	192.168.20.15	DNS	95	Standard query 0xd18f A optimizationguide-pa.googleapis.com
...	4.442821	192.168.20.15	172.18.170.2	DNS	366	Standard query response 0xd18f A optimizationguide-pa.googleapis.com A 216.58.206.202 NS ns2.google.com NS ns1.google.com NS ns4.google.com
...	5.765105	172.18.170.2	192.168.20.15	DNS	83	Standard query 0x8bc2 A www.msftconnecttest.com
...	5.767627	192.168.20.15	172.18.170.2	DNS	287	Standard query response 0x8bc2 A www.msftconnecttest.com CNAME ncsi-geo.trafficmanager.net CNAME v4ncsi.msedge.net CNAME ncsi.4-c-00
...	9.173203	172.18.170.2	192.168.20.15	DNS	74	Standard query 0x4a67 A washington.edu
...	9.426000	192.168.20.15	172.18.170.2	DNS	355	Standard query response 0x4a67 A washington.edu A 128.95.155.135 A 128.95.155.134 A 128.95.155.198 A 128.95.155.197 A 34.127.31.83 N
...	9.930385	172.18.170.2	192.168.20.15	DNS	78	Standard query 0xf536 A www.washington.edu
...	10.179218	192.168.20.15	172.18.170.2	DNS	251	Standard query response 0xf536 A www.washington.edu A 128.95.155.197 A 128.95.155.134 A 128.95.155.198 A 128.95.155.135 NS dnsload11.s.uw.edu NS dns1
...	10.702306	172.18.170.2	192.168.20.15	DNS	87	Standard query 0xdafa A safebrowsing.googleapis.com

Figure 2: DNS sequences

In the following snip I have highlighted the standard query and its response accordingly.

[udp.port=53 udp.cport=53]						
No.	Time	Source	Destination	Protocol	Length	Info
-	1.945117	172.18.170.2	192.168.20.15	DNS	86	Standard query 0xf6d9 A encrypted-tbn0.gstatic.com
-	1.946888	192.168.20.15	172.18.170.2	DNS	357	Standard query response 0xf6d9 A encrypted-tbn0.gstatic.com A 142.250.187.142 NS ns4.google.com NS ns1.google.com NS ns3.google.com
-	4.440766	172.18.170.2	192.168.20.15	DNS	95	Standard query 0xd18f A optimizationguide-pa.googleapis.com
-	4.442821	192.168.20.15	172.18.170.2	DNS	366	Standard query response 0xd18f A optimizationguide-pa.googleapis.com A 216.58.206.202 NS ns2.google.com NS ns1.google.com NS ns4.google.com
-	5.765105	172.18.170.2	192.168.20.15	DNS	83	Standard query 0x8bc2 A www.msftconnecttest.com
-	5.767627	192.168.20.15	172.18.170.2	DNS	287	Standard query response 0x8bc2 A www.msftconnecttest.com CNAME ncsi-geo.trafficmanager.net CNAME v4ncsi.wedge.net CNAME ncsi.4-c-00
-	9.173203	172.18.170.2	192.168.20.15	DNS	174	Standard query 0x4a67 A washington.edu
-	9.426000	192.168.20.15	172.18.170.2	DNS	355	Standard query response 0x4a67 A washington.edu A 128.95.155.135 A 128.95.155.134 A 128.95.155.198 A 128.95.155.197 A 34.127.31.83 N
-	9.930385	172.18.170.2	192.168.20.15	DNS	78	Standard query 0xf536 A www.washington.edu
-	10.179218	192.168.20.15	172.18.170.2	DNS	251	Standard query response 0xf536 A www.washington.edu A 128.95.155.197 A 128.95.155.134 A 128.95.155.135 NS dnsload11.s.uw.edu NS dns1
-	10.702306	172.18.170.2	192.168.20.15	DNS	87	Standard query 0xda4a A safebrowsing.googleapis.com

Figure 3: DNS sequences highlighted

First we begin with some basic explanations about DNS. We know that DNS is a global system utilized for translating IP addresses to domain names which are human-readable.

So when we try to enter a website such as the one demonstrated, our application which is typically a browser performs a DNS query, in other words the browser asks the DNS server the numeric IP address by providing it with the hostname, the DNS server finds the needed IP and gives it to the browser so we can connect to the site.

Now I draw your attention to some facts about the included snips.

As we can see the letter **A** is seen before the address of the given website, something like this:

A washington.edu

This mysterious **A** stands behind all queries and is an indicator of a translation record.

When we check we query packet, we see something called the transaction ID as follows.

```
> Frame 403: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{DFD65439-D4DC-43A1-AA37-DE0A363737F6}, id 0
> Ethernet II, Src: LiteonTe_ae:ea:73 (f8:28:19:ae:ea:73), Dst: Cisco_90:f2:d4 (f0:b2:e5:90:f2:d4)
> Internet Protocol Version 4, Src: 172.18.170.2, Dst: 192.168.20.15
> User Datagram Protocol, Src Port: 59416, Dst Port: 53
v Domain Name System (query)
  Transaction ID: 0x4a67
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
```

Figure 4: Transaction ID for query

This is a random number generated by the nameserver which initiates the query, after we get the response, the same transaction ID shall be set.

We shall have more explanation about the query and response packets in the following questions.

1.2 Q2

First of all we shall take a look at the query packet.

```
> Frame 403: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{DFD65439-D4DC-43A1-AA37-DE0A363737F6}, id 0
> Ethernet II, Src: LiteonTe_ae:ea:73 (f8:28:19:ae:ea:73), Dst: Cisco_90:f2:d4 (f0:b2:e5:90:f2:d4)
> Internet Protocol Version 4, Src: 172.18.170.2, Dst: 192.168.20.15
> User Datagram Protocol, Src Port: 59416, Dst Port: 53
v Domain Name System (query)
  Transaction ID: 0x4a67
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
```

Figure 5: Query packet overall

Now we take a look at it with more detail.

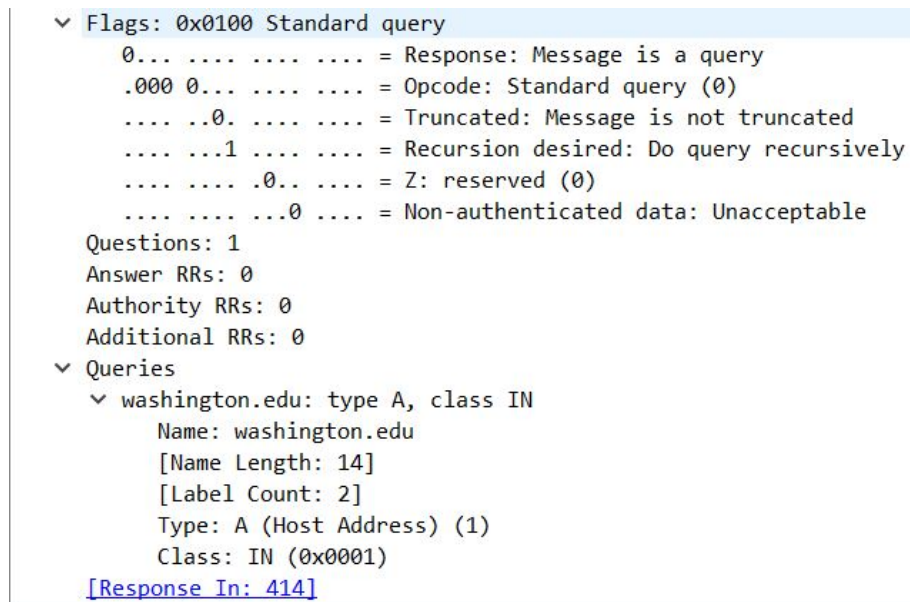


Figure 6: Query packet flags and queries

We take a closer look at the queries part:

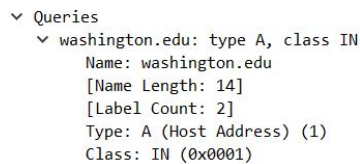


Figure 7: Query packet queries section

Type A: It is mostly used for a 32-bit IP address, commonly to map hostnames to an IP address of said host, but it also has other usages like storing subnet masks and so on.

Class IN: Expresses that is in the Internet class.

1.3 Q3

The response of the DNS packet is as follows.

```

> Frame 414: 355 bytes on wire (2840 bits), 355 bytes captured (2840 bits) on interface \Device\NPF_{DFD65439-D4DC-43A1-AA37-DE0A363737F6}, id 0
> Ethernet II, Src: Cisco 90:f2:d4 (f8:b2:e5:90:f2:d4), Dst: LiteonTe_ae:ea:73 (f8:28:19:ae:ea:73)
> Internet Protocol Version 4, Src: 192.168.20.15, Dst: 172.18.170.2
> User Datagram Protocol, Src Port: 53, Dst Port: 59416
▼ Domain Name System (response)
  Transaction ID: 0x4a67
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 5
    Authority RRs: 3
    Additional RRs: 6
  > Queries
  > Answers
  > Authoritative nameservers
  > Additional records

```

Figure 8: Response packet

We shall first take a closer look at the flags section.

```

▼ Flags: 0x8180 Standard query response, No error
  1... .. = Response: Message is a response
  .000 0... .. = Opcode: Standard query (0)
  ....0... .. = Authoritative: Server is not an authority for domain
  ....0... .. = Truncated: Message is not truncated
  ....1... .. = Recursion desired: Do query recursively
  ....1... .. = Recursion available: Server can do recursive queries
  ....0... .. = Z: reserved (0)
  ....0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
  ....0... .. = Non-authenticated data: Unacceptable
  ....0000 = Reply code: No error (0)

```

Figure 9: Flag section of response packet

Now we give an explanation of the meaning of each part.

The **first bit** indicates if we have a query or a response.

The 4-bit **Opcode** identifies the request type. The different opcode types are as follows.

- **QUERY**
- **STATUS**
- **UPDATE**
- **IQUERY**
- **NOTIFY**

In our case we have a standard request.

Authoritative refers to DNS servers that have a complete copy of the domain's information. This information can be passed to the DNS server by an administrator or transferred from a primary server.

Truncation occurs when the message length is longer than the maximum permitted amount for the type of transport we plan to use. It is good to note that TCP doesn't have a length limit for messages, whilst UDP messages are limited to 512 bytes.

Recursion: When a client asks the local DNS server to perform its needed requests we have a recursive DNS query. The client performs a recursive request by flagging a particular bit in the flag section of the DNS query. The server will then in turn confirm or not confirm whether it can support the recursive DNS query or not.

Z means reserved for future usage.

Answer authentication Implies whether the answer/authority is authenticated by the DNS server or not.

Reply code: In order to troubleshoot DNS-related problems we must get an overview of all response codes, if we get a 0 it means we have no errors.

The answers are as follows.

```
▼ Answers
> washington.edu: type A, class IN, addr 128.95.155.135
> washington.edu: type A, class IN, addr 128.95.155.134
> washington.edu: type A, class IN, addr 128.95.155.198
> washington.edu: type A, class IN, addr 128.95.155.197
> washington.edu: type A, class IN, addr 34.127.31.83
```

Figure 10: Answers in response

1.4 Q4

As indicated in the lectures and the previous assignment, **TTL** is short for **Time-to-live**. This field indicates the time which the record in the cache

should be considered valid, After its expiration, the record must be updated and renewed or discarded.

This field is set by the administrator of the DNS server(An authoritative DNS server), the amount of time which TTL indicates differs between just a few seconds to days long at times.

In Wireshark we can find the TTL field i=under in answer part as follows.

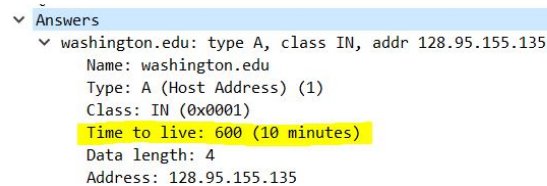


Figure 11: Time to live

1.5 Q5

As indicated in the question, some DNS packets that are the query for other websites other the one we opened are seen.

We know that there are three different queries used in DNS messages, they are:

- **Recursive**
- **Iterative**
- **Non-Recursive**

We must also know the meaning and usage of a **DNS resolver**, a DNS resolver is responsible for checking if the hostname is available in the local cache,if not, it contacts other DNS Servers, until it eventually receives the IP of the webpage or whatever we were trying to reach.

In our case we perform a recursive query, this query is initiated by a DNS resolver, it checks the local DNS cache to find the needed IP address to our

hostname, if it doesn't work it initiates a wider search consisting of the other DNS servers, TLD(Top level domain) and lastly Root DNS servers until it finds the needed IP7 so the other websites we see are actually DNS servers placed in the path of our queries.

1.6 Q6

In our case 5 answers containing information are available.

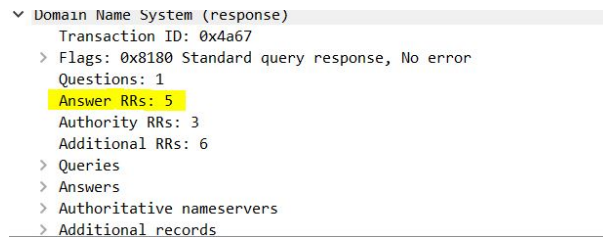


Figure 12: The number of answers

All of the answers have their, own specific Name, class, TTL, and Data length.

```

  ▾ Answers
    ▾ washington.edu: type A, class IN, addr 128.95.155.135
      Name: washington.edu
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 600 (10 minutes)
      Data length: 4
      Address: 128.95.155.135
    ▾ washington.edu: type A, class IN, addr 128.95.155.134
      Name: washington.edu
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 600 (10 minutes)
      Data length: 4
      Address: 128.95.155.134
    ▾ washington.edu: type A, class IN, addr 128.95.155.198
      Name: washington.edu
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 600 (10 minutes)
      Data length: 4
      Address: 128.95.155.198
    ▾ washington.edu: type A, class IN, addr 128.95.155.197
      Name: washington.edu
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 600 (10 minutes)
      Data length: 4
      Address: 128.95.155.197
    ▾ washington.edu: type A, class IN, addr 34.127.31.83
      Name: washington.edu
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 600 (10 minutes)
      Data length: 4
      Address: 34.127.31.83

```

Figure 13: The number of answers

If the answer type is CNAME, the answer shall have a CNAME field.

1.7 Q7

We shall open the command window as instructed and enter the following statement.

nslookup -type=NS washington.edu

But before doing so, we shall explain what the output shall give us.

NSLOOKUP is used to query name servers so that we can get information about nodes, examine the contents of a name-server database and establish accessibility of name servers.

NSLOOKUP is also used for troubleshooting.

Server name & internet address: Indicates the destination DNS, defined as a server name or IP address.

Domain name & address: Indicates the destination domain, defined as a domain name or IP address.

Non-Authoritative: It means that the answer has come from the cache of another server instead of an authoritative one.

```
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Soroush>nslookup -type=NS washington.edu
Server:      UnKnown
Address:     192.168.20.15

Non-authoritative answer:
washington.edu nameserver = hanna.cac.washington.edu
washington.edu nameserver = holly.s.uw.edu
washington.edu nameserver = marge.cac.washington.edu

holly.s.uw.edu internet address = 173.250.227.69
holly.s.uw.edu AAAA IPv6 address = 2607:4000:301:1::69
marge.cac.washington.edu internet address = 140.142.5.13
marge.cac.washington.edu AAAA IPv6 address = 2607:4000:200:43::13
hanna.cac.washington.edu internet address = 140.142.5.5
hanna.cac.washington.edu AAAA IPv6 address = 2607:4000:200:42::5

C:\Users\Soroush>
```

Figure 14: NSLOOKUP output

2 HTTP

We do all the steps as instructed, the webpage is as follows:

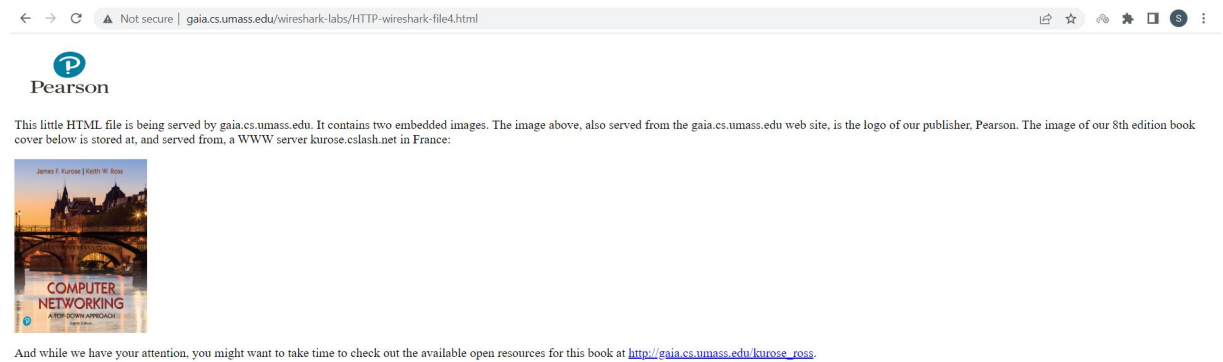


Figure 15: The webpage

2.1 Q1

The HTTP requests are seen as follows (The GET requests have been highlighted)

No.	Time	Source	Destination	Protocol	Length	Info
1	598416	172.18.170.2	128.119.245.12	HTTP	535	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
2	777989	128.119.245.12	172.18.170.2	HTTP	1355	HTTP/1.1 200 OK (text/html)
3	131729	172.18.170.2	128.119.245.12	HTTP	481	GET /pearson.png HTTP/1.1
4	274139	172.18.170.2	178.79.137.164	HTTP	448	GET /8E_cover_small.jpg HTTP/1.1
5	412468	178.79.137.164	172.18.170.2	HTTP	225	HTTP/1.1 301 Moved Permanently
6	835072	172.18.170.2	128.119.245.12	HTTP	481	GET /favicon.ico HTTP/1.1
7	809569	128.119.245.12	172.18.170.2	HTTP	538	HTTP/1.1 404 Not Found (text/html)

Figure 16: GET requests highlighted

$$GET\# = 4$$

So we have sent 4 HTTP GET request messages.

2.2 Q2

Now we go on to explain the purposes of the GET messages.

We know that GET is used to retrieve and request data from a specific resource. We have the following:

- **First GET:** To get the base file.
- **Second GET:** To get the Pearson logo.
- **Third Get:** To get the textbook cover(COMPUTER NETWORK-ING, A TOP DOWN APPROACH).
- **Fourth Get:** To get favicon.ico which is the small photo that appears behind the http(s) in some websites¹.

2.3 Q3

Now we go on to explain the GET responses.

2.3.1 First GET response

The first response is for the base file and PNG photo which is as follows.

```
> Frame 91: 1355 bytes on wire (10840 bits), 1355 bytes captured (10840 bits) on interface \Device\NPF_{DFD65439-D4DC-43A1-AA37-DE0A363737F6}, id 0
> Ethernet II, Src: Cisco_90:f2:d4 (f0:b2:e5:90:f2:d4), Dst: LiteonTe_ae:ea:73 (f8:28:19:ae:ea:73)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.18.170.2
> Transmission Control Protocol, Src Port: 80, Dst Port: 61080, Seq: 1, Ack: 482, Len: 1301
< Hypertext Transfer Protocol
  < HTTP/1.1 200 OK\r\n
    Date: Sat, 02 Jul 2022 14:28:04 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.28 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Last-Modified: Sat, 02 Jul 2022 05:59:01 GMT\r\n
    ETag: "3ae-5e2cc3314b051"\r\n
    Accept-Ranges: bytes\r\n
  > Content-Length: 942\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 1/3]
  [Time since request: 0.187493000 seconds]
  [Request in frame: 60]
```

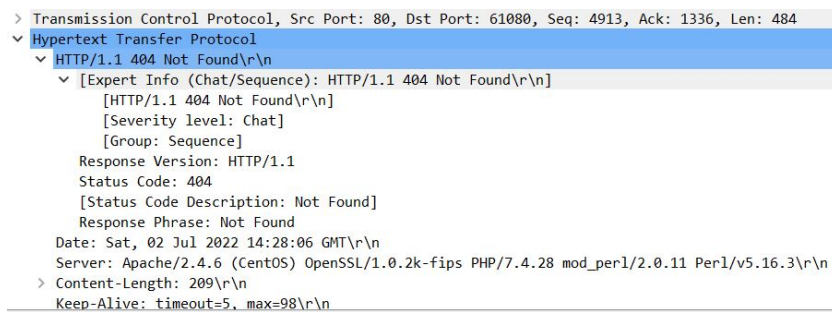
Figure 17: First GET response

¹For more information about favicon.ico check the following link:[Info about favicon](#)

As we see we have gotten moved permanently which indicates that the requested resource has been definitively moved to the URL given by the location headers.

2.3.3 Third GET response

This is in response to the favicon.ico, WE have:



```
> Transmission Control Protocol, Src Port: 80, Dst Port: 61080, Seq: 4913, Ack: 1336, Len: 484
  Hypertext Transfer Protocol
    HTTP/1.1 404 Not Found\r\n
      [Expert Info (Chat/Sequence): HTTP/1.1 404 Not Found\r\n]
        [HTTP/1.1 404 Not Found\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Response Version: HTTP/1.1
      Status Code: 404
      [Status Code Description: Not Found]
      Response Phrase: Not Found
      Date: Sat, 02 Jul 2022 14:28:06 GMT\r\n
      Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.28 mod_perl/2.0.11 Perl/v5.16.3\r\n
    > Content-Length: 209\r\n
    Keep-Alive: timeout=5, max=98\r\n
```

Figure 20: Third GET response

As we can see we have gotten 404 Not found, this means that the server cannot find the resources being requested by the client.