

- Assignment 3 -

HTTP+DNS (Wireshark)

Question 1 -

HTTP belongs to the layer application the seven layer of the OSI model.
It's in the layer application that we find most of the network program.
Usually, applications use udp or tcp, and are associated with a famous port.
For HTTP, it's the port TCP 80.

Question 2 -

GET /form.html HTTP/1.1

Accept: */*

Accept-Language: en-us

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1;
WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.
3; .NET4.0C; .NET4.0E)

Accept-Encoding: gzip, deflate

Host: 192.168.2.103

Connection: Keep-Alive

This text represent a GET message, differently said, a request-URI.

Explanation of the previous message:

GET is the method used to send the request. This method is cacheable and safe.

/form.html is the path.

Note that the path cannot be empty, if none is present in the original URI, it must be given as "/".

HTTP/1.1, is the latest version of Hypertext Transfer Protocol.

The accept request HTTP header advertises which content types (MIME types : Multipurpose Internet Mail Extension), the client is able to understand. Here “*/*” means any MIME type.

The Accept-Language request HTTP header advertises which languages the client is able to understand, and which locale variant is preferred. Here english and us (United states) are mentioned.

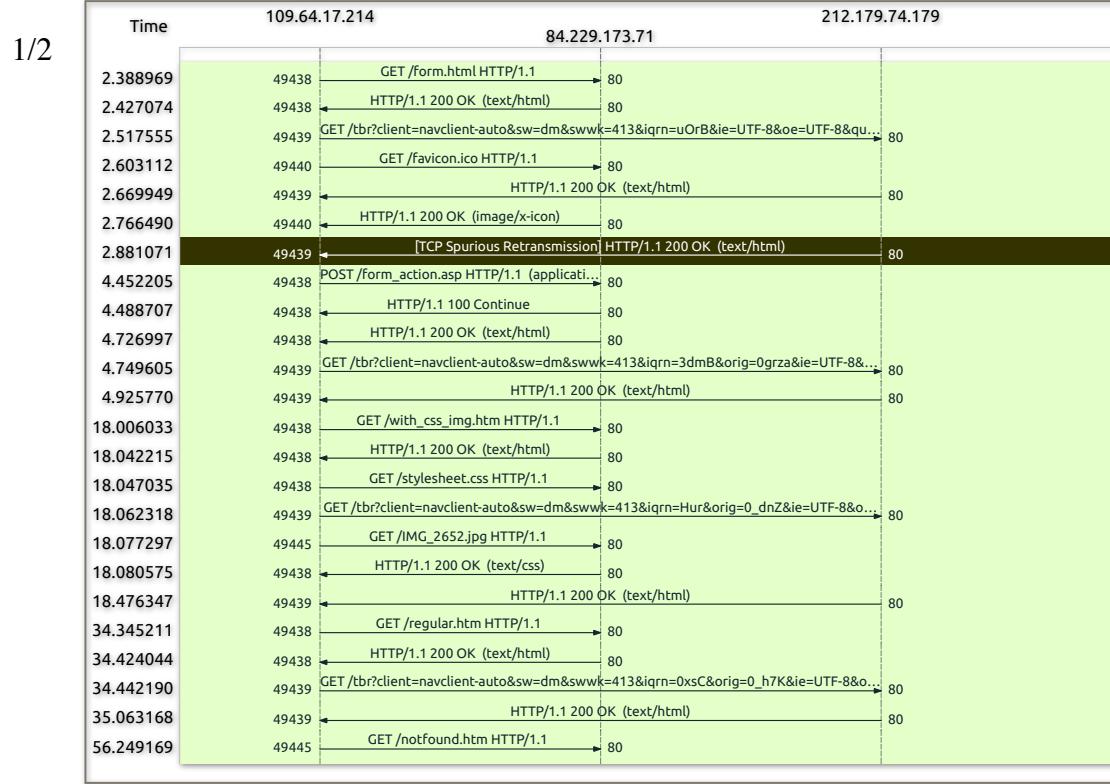
The User-Agent request header contains a characteristic string in our case : “Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)” that allows the network protocol peers to identify the application type, operating system, software vendor or software version of the requesting software user agent.

The Accept-Encoding request HTTP header advertises which content encoding, usually a compression algorithm, the client is able to understand. Here gzip and deflate are allowed.

The Host request header specifies the domain name of the serve. Here it's 192.168.2.103.

The Connection request header controls whether or not the network connection stays open after the current transaction finishes. Here the value sent is keep-alive, ie, the connection is persistent and not closed, allowing for subsequent requests to the same server to be done.

Question 3 -



Comment

```

HTTP: GET /form.html HTTP/1.1
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /tbr?client=navclient-auto&sw=dm...
HTTP: GET /favicon.ico HTTP/1.1
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: HTTP/1.1 200 OK (image/x-icon)
HTTP: [TCP Spurious Retransmission] HTTP/1....
HTTP: POST /form_action.asp HTTP/1.1 (appl...
HTTP: HTTP/1.1 100 Continue
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /tbr?client=navclient-auto&sw=dm...
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /with_css_img.htm HTTP/1.1
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /stylesheet.css HTTP/1.1
HTTP: GET /tbr?client=navclient-auto&sw=dm...
HTTP: GET /IMG_2652.jpg HTTP/1.1
HTTP: HTTP/1.1 200 OK (text/css)
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /regular.htm HTTP/1.1
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /tbr?client=navclient-auto&sw=dm...
HTTP: HTTP/1.1 200 OK (text/html)
HTTP: GET /notfound.htm HTTP/1.1

```

2/2

Screen-shot of sequence diagram from Wireshark

Question 4 -

From the trace, we can notice two types of requests : GET and POST.

The GET method requests a representation of the specified resource.

Requests using GET should only retrieve data.

The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server



Screen-shot from Wireshark who display the packet number from one of the gate message and the post message.

As we can see in the following pictures, we raise that the first get request get a packet with the number 6. The only post request of the trace get a packet with the number 66. This number provide from the legend, see like on the pictures :

Wireshark - Wireshark - Flow - HTTP-Full-Client

Time 109.64.17.214 84.229.173.71 212.179.74.179 Comment

2.388969 49438 GET /form.html HTTP/1.1 80 HTTP: GET /form.html HTTP/1.1
49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
2.427074 49438 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=uOrB&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=uOrB&ie=UTF-8&qu= 80
2.517555 49439 GET /favicon.ico HTTP/1.1 80 HTTP: GET /favicon.ico HTTP/1.1
2.603112 49440 GET / HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
2.669949 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
2.766490 49440 HTTP/1.1 200 OK (image/x-icon) 80 HTTP: HTTP/1.1 200 OK (image/x-icon)
2.881071 49439 [TCP Spurious Retransmission] HTTP/1.1 200 OK (text/html) 80 HTTP: [TCP Spurious Retransmission] HTTP/1.1 200 OK (text/html)
4.452205 49438 POST /form_action.asp HTTP/1.1 (application/x-www-form-urlencoded) 80 HTTP: POST /form_action.asp HTTP/1.1 (application/x-www-form-urlencoded)
4.488707 49438 HTTP/1.1 100 Continue 80 HTTP: HTTP/1.1 100 Continue
4.726997 49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
4.749605 49439 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=3dmB&orig=0grza&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=3dmB&orig=0grza&ie=UTF-8&qu= 80
4.925770 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
18.006033 49438 GET /with_css_img.htm HTTP/1.1 80 HTTP: GET /with_css_img.htm HTTP/1.1
18.042215 49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
18.047035 49438 GET /stylesheet.css HTTP/1.1 80 HTTP: GET /stylesheet.css HTTP/1.1
18.062318 49439 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=Hur&orig=0_dnZ&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=Hur&orig=0_dnZ&ie=UTF-8&qu= 80
18.077297 49445 GET /IMG_2652.jpg HTTP/1.1 80 HTTP: GET /IMG_2652.jpg HTTP/1.1
18.080575 49438 HTTP/1.1 200 OK (text/css) 80 HTTP: HTTP/1.1 200 OK (text/css)
18.476347 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
34.345211 49438 GET /regular.htm HTTP/1.1 80 HTTP: GET /regular.htm HTTP/1.1
34.424044 49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
34.442190 49439 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=0xsC&orig=0_h7K&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=0xsC&orig=0_h7K&ie=UTF-8&qu= 80
35.063168 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
56.249169 49445 GET /notfound.htm HTTP/1.1 80 HTTP: GET /notfound.htm HTTP/1.1

Packet 6: HTTP: GET /form.html HTTP/1.1

Show: Displayed packets ▾ Flow type: All Flows ▾ Addresses: An ▾ Help ▾ Close ▾

```
0130 76 65 2d 66 6c 61 73 68 2c 20 2a 2f 2a 0d 0a 41 ve-flash , /*..A
0140 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 29 ccept-La nguage:
0150 68 65 2d 49 4c 0d 0a 55 73 65 72 2d 41 67 65 6e he-IL..U ser-Agen
0160 74 3a 29 4d 6f 7a 69 6c 6c 61 2f 34 2e 30 20 28 t: Mozil la/4.0 (
0170 63 6f 6d 70 61 74 69 62 6c 65 3b 20 4d 53 49 45 compatib le; MSIE
0180 29 38 2e 30 3b 20 57 69 6e 64 6f 77 73 20 4e 54 8.0; Wi ndows NT
0190 29 36 2e 31 3b 20 54 72 69 64 65 6e 74 2f 34 2e 6.1; Tr ident/4.
01a0 30 3b 20 47 54 42 36 2e 36 3b 20 53 4c 43 43 32 0; GTB6. 6; SLC2
01b0 3b 20 2e 4e 45 54 20 43 4c 52 20 32 2e 30 2e 35 ; .NET C LR 2.0.5
```

Screen-shot from Wireshark. The rectangle focus on the packet number.

Wireshark - Wireshark - Flow - HTTP-Full-Client

Time 109.64.17.214 84.229.173.71 212.179.74.179 Comment

2.388969 49438 GET /form.html HTTP/1.1 80 HTTP: GET /form.html HTTP/1.1
49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
2.427074 49438 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=uOrB&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=uOrB&ie=UTF-8&qu= 80
2.517555 49439 GET /favicon.ico HTTP/1.1 80 HTTP: GET /favicon.ico HTTP/1.1
2.603112 49440 GET / HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
2.669949 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
2.766490 49440 HTTP/1.1 200 OK (image/x-icon) 80 HTTP: HTTP/1.1 200 OK (image/x-icon)
2.881071 49439 [TCP Spurious Retransmission] HTTP/1.1 200 OK (text/html) 80 HTTP: [TCP Spurious Retransmission] HTTP/1.1 200 OK (text/html)
4.452205 49438 POST /form_action.asp HTTP/1.1 (application/x-www-form-urlencoded) 80 HTTP: POST /form_action.asp HTTP/1.1 (application/x-www-form-urlencoded)
4.488707 49438 HTTP/1.1 100 Continue 80 HTTP: HTTP/1.1 100 Continue
4.726997 49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
4.749605 49439 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=3dmB&orig=0grza&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=3dmB&orig=0grza&ie=UTF-8&qu= 80
4.925770 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
18.006033 49438 GET /with_css_img.htm HTTP/1.1 80 HTTP: GET /with_css_img.htm HTTP/1.1
18.042215 49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
18.047035 49438 GET /stylesheet.css HTTP/1.1 80 HTTP: GET /stylesheet.css HTTP/1.1
18.062318 49439 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=Hur&orig=0_dnZ&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=Hur&orig=0_dnZ&ie=UTF-8&qu= 80
18.077297 49445 GET /IMG_2652.jpg HTTP/1.1 80 HTTP: GET /IMG_2652.jpg HTTP/1.1
18.080575 49438 HTTP/1.1 200 OK (text/css) 80 HTTP: HTTP/1.1 200 OK (text/css)
18.476347 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
34.345211 49438 GET /regular.htm HTTP/1.1 80 HTTP: GET /regular.htm HTTP/1.1
34.424044 49438 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
34.442190 49439 GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=0xsC&orig=0_h7K&ie=UTF-8&qu= 80 HTTP: GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqn=0xsC&orig=0_h7K&ie=UTF-8&qu= 80
35.063168 49439 HTTP/1.1 200 OK (text/html) 80 HTTP: HTTP/1.1 200 OK (text/html)
56.249169 49445 GET /notfound.htm HTTP/1.1 80 HTTP: GET /notfound.htm HTTP/1.1

Packet 66: HTTP: POST /form_action.asp HTTP/1.1 (application/x-www-form-urlencoded)

Show: Displayed packets ▾ Flow type: All Flows ▾ Addresses: Any ▾ Help ▾ Close ▾ Save As... ▾

```
0130 76 65 2d 66 6c 61 73 68 2c 20 2a 2f 2a 0d 0a 41 ve-flash , /*..A
0140 63 63 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 29 ccept-La nguage:
0150 68 65 2d 49 4c 0d 0a 55 73 65 72 2d 41 67 65 6e he-IL..U ser-Agen
0160 74 3a 29 4d 6f 7a 69 6c 6c 61 2f 34 2e 30 20 28 t: Mozil la/4.0 (
0170 63 6f 6d 70 61 74 69 62 6c 65 3b 20 4d 53 49 45 compatib le; MSIE
0180 29 38 2e 30 3b 20 57 69 6e 64 6f 77 73 20 4e 54 8.0; Wi ndows NT
0190 29 36 2e 31 3b 20 54 72 69 64 65 6e 74 2f 34 2e 6.1; Tr ident/4.
01a0 30 3b 20 47 54 42 36 2e 36 3b 20 53 4c 43 43 32 0; GTB6. 6; SLC2
01b0 3b 20 2e 4e 45 54 20 43 4c 52 20 32 2e 30 2e 35 ; .NET C LR 2.0.5
```

Screen-shot from Wireshark. The rectangle focus on the packet number.

Here are all the HTTP request types except of GET and POST , HEAD, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH.

More explanations to follow :

The HEAD method asks for a response identical to that of a GET request, but without the response body.

The PUT method replaces all current representations of the target resource with the request payload.

The DELETE method deletes the specified resource.

The CONNECT method establishes a tunnel to the server identified by the target resource.

The OPTIONS method is used to describe the communication options for the target resource.

The TRACE method performs a message loop-back test along the path to the target resource.

The PATCH method is used to apply partial modifications to a resource.

Question 5 -

There are a lot of differences between the two method GET and POST.

The fundamental difference is that they correspond to different HTTP requests.

The submission process for both methods begins in the same way a form data set is constructed by the browser and then encoded in a manner specified by the enctype attribute. Here are some of the many differences :

The GET request can be cached whereas the POST is never cached.

The GET request remain in the browser history whereas the POST don't.

The GET request can be bookmarked whereas the POST can't.

The GET request have length restriction whereas the POST have no restriction on data length.

Question 6 -

GET is basically for just getting data whereas "POST" may involve anything, like storing or updating data, or ordering a product, or sending E-mail.

GET is recommended when submitting "idempotent" forms. In other words, forms that involve database queries only. Also recommended if the interaction is more like a question, such as query, read operation or lookup.

Another perspective is that several idempotent queries will have the same effect as a single query. If database updates or other actions such as triggering emails are involved, the usage of POST is recommended. In others words, if the interaction is more like an order, or the interaction changes the state of the ressource in a way that the user would perceive. Or if the user be held accountable for the result of the interaction.

Question 7 -

From the trace we notice three types of answers : 200 ok, 100 continue, 404 Object Not Found.

200 ok means the request has succeeded.

100 continue means the client should continue with its request.

404 Not found means that the server has not found anything matching the Request-URI.

4.452205	49438	POST /form_action.asp HTTP/1.1 (application/x-www-form-urlencoded)	80
4.488707	49438	HTTP/1.1 100 Continue	80
4.726997	49438	HTTP/1.1 200 OK (text/html)	80
4.749605	49439	GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqrn=3dmB&orig=0grza&ie=UTF-8&o... (text/html)	80
4.925770	49439	HTTP/1.1 200 OK (text/html)	80
18.006033	49438	GET /with_css_img.htm HTTP/1.1	80
18.042215	49438	HTTP/1.1 200 OK (text/html)	80
18.047035	49438	GET /stylesheet.css HTTP/1.1	80
18.062318	49439	GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqrn=Hur&orig=0_dnZ&ie=UTF-8&o... (text/html)	80
18.077297	49445	GET /IMG_2652.jpg HTTP/1.1	80
18.080575	49438	HTTP/1.1 200 OK (text/css)	80
18.476347	49439	HTTP/1.1 200 OK (text/html)	80
34.345211	49438	GET /regular.htm HTTP/1.1	80
34.424044	49438	HTTP/1.1 200 OK (text/html)	80
34.442190	49439	GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqrn=0xsC&orig=0_h7K&ie=UTF-8&o... (text/html)	80
35.063168	49439	HTTP/1.1 200 OK (text/html)	80
56.249169	49445	GET /notfound.htm HTTP/1.1	80
56.366756	49445	HTTP/1.1 404 Object Not Found (text/html)	80
56.403409	49439	GET /tbr?client=navclient-auto&sw=dm&swwk=413&iqrn=HqEC&orig=0PdMr&iqst=404&... (text/html)	80
56.535467	49439	HTTP/1.1 200 OK (text/html)	80

Packet 248: HTTP: HTTP/1.1 200 OK (text/html)

Screen-shot from wireshark. The numbers determines the packets numbers.

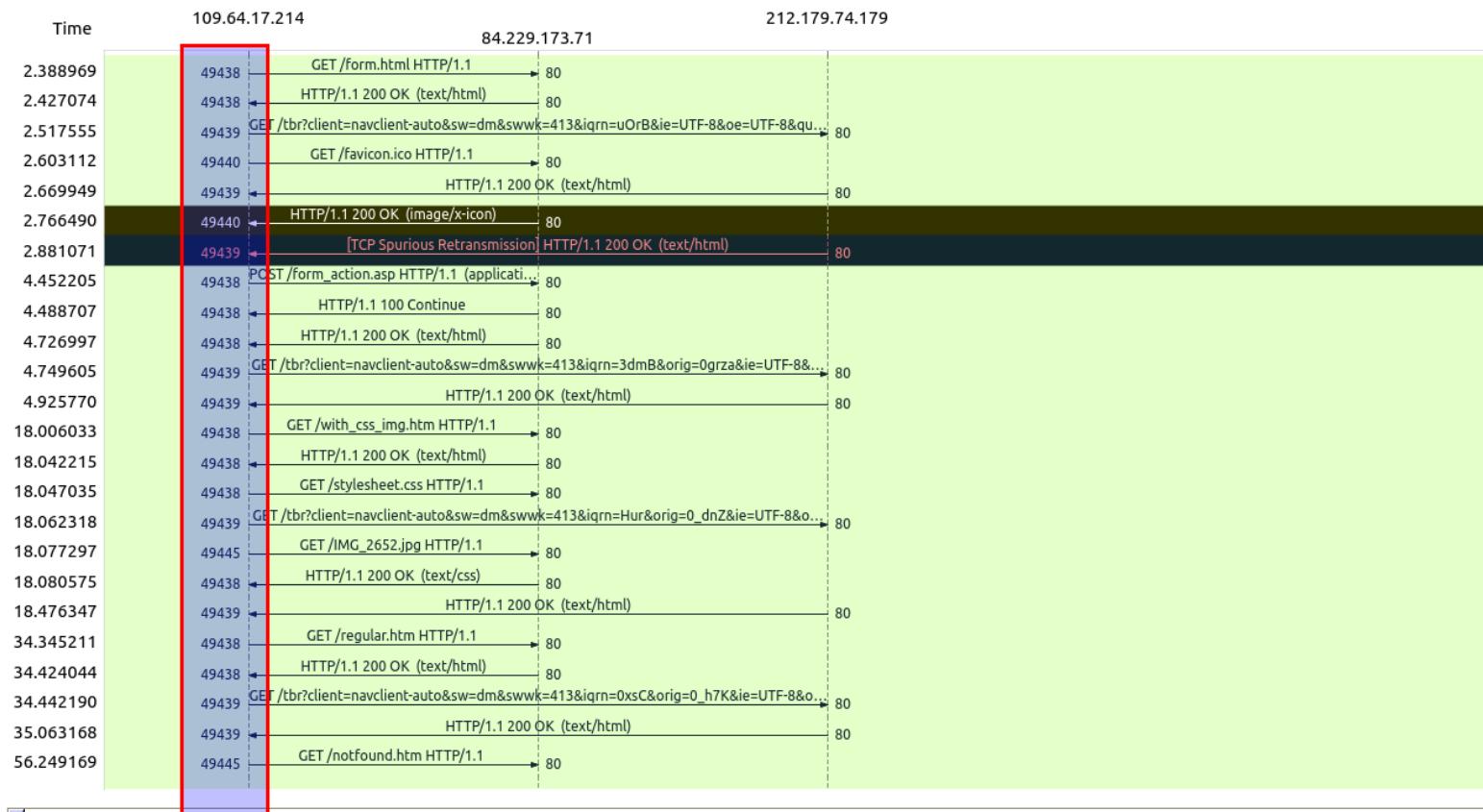
As we can see in the previous picture, the POST request has a packet number equal to 66, the response of this request is 100 continue, with a packet number 67.

The GET/NotFound.html request has a packet number equal to 283, the response of this request is 404 Not found, with a packet number 67.

The GET request (the last of the trace) has a packet number equal to 291, the response of this request is 200 ok, with a packet number 295.

Note that all the packets numbers have been found like in the question 4.

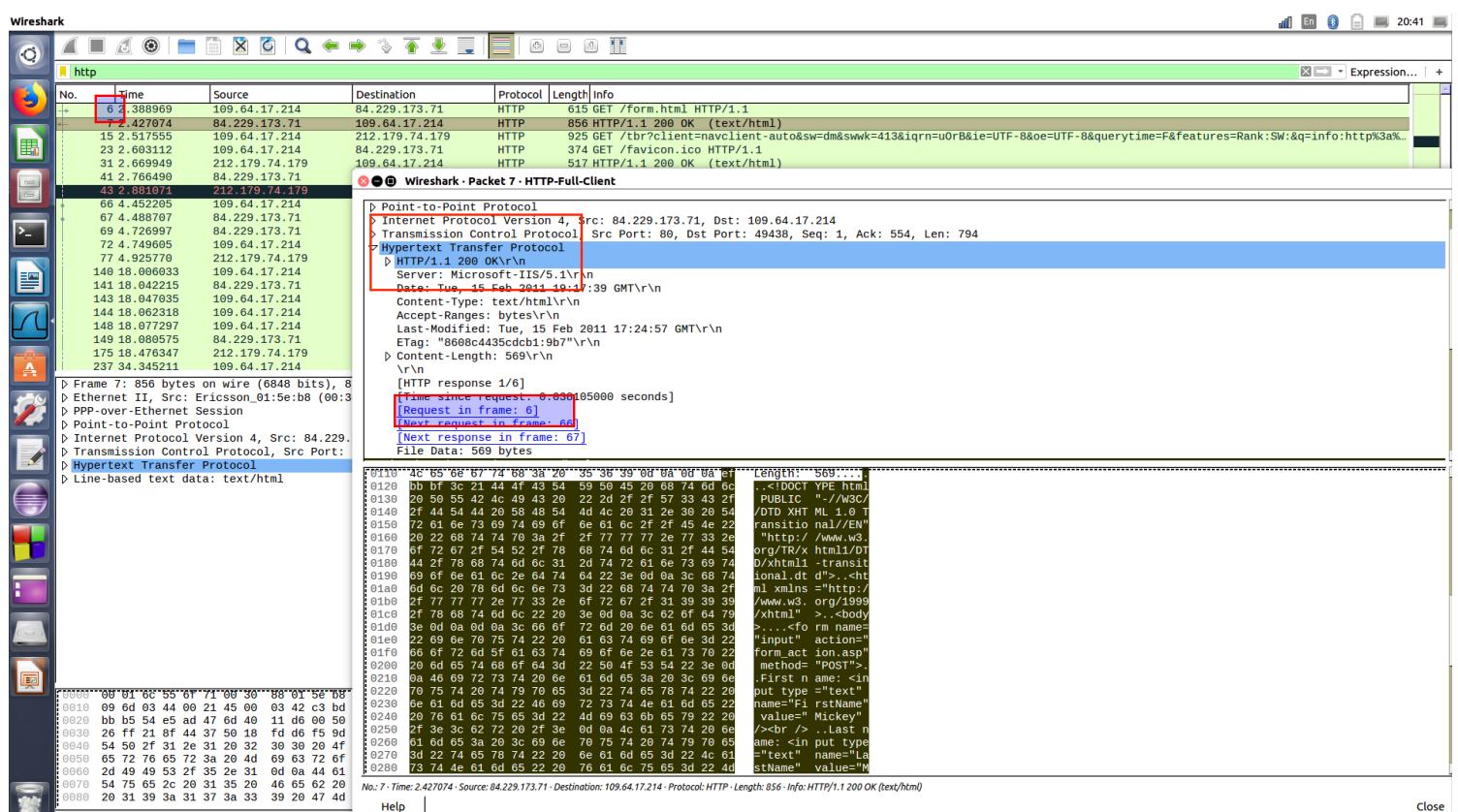
To know which request precede the answer, we look at the number of source port, which is the second number from the left side.



Screen-shot from wireshark. The rectangle focus on the DST port

If the source port of a request is the same that the source port of an answer, so they are connected. And of course, we're looking for the clothier time.

Another way to check this is by looking the information like in the picture :



Screen-shot from wireshark.

We denote 67 different answers (without care about the unofficial code). All the answers are determined by a number, (for example here : 200 ok.) Which belongs to 6 types of answer, the 100's, 200's, 300's, 400's, 500's, and the unofficial answer.

Question 8 -

There are six types of answer :

- 1xx specify a provisional response.
 - 2xx indicate that the server has received client's request, understands said request, accepts it and has processed it successfully.
 - 3xx indicate that the client is required to take additional steps to complete a request.
 - 4xx are meant for situations where the client has erred.
 - 5xx are responses to requests that servers fail to fulfill.
 - Unofficial codes are still utilized by third-party services to provide restful or semantic error responses.
-

Question 9 -

The HTTP belongs to the TCP/IP , differently said, the Internet protocol suite.

In the TCP/IP, we denote a lot of layers.

Every layers resolve some problems about the data transmission and give to the other layers some informations.

The highest layers are closer from the users and handle more abstract data by using the informations from the lowest layers. The lowest layers must configure the data in the goal that the data may be useful by the highest layers.

So we are now able to understand why does HTTP belong to another protocol. Like we explained, HTTP belongs to a layer : the application layer. And the HTTP protocol needs other layers to work fine, and the other protocol needs the HTTP protocol to work fine.

What the TCP/IP adds to the HTTP protocol ?

In the TCP/IP we denote four layers : application (which contains HTTP), transport, Internet, link.

The most important things for the application layer is the choice of the protocol into the transport layers. The HTTP protocol uses the TCP transport.

Question 10 -



Wireshark - Follow HTTP Stream (tcp.stream eq 0) - HTTP-Full-Client

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
Date: Tue, 15 Feb 2011 19:17:39 GMT
Content-Type: text/html
Accept-Ranges: bytes
Last-Modified: Tue, 15 Feb 2011 17:24:57 GMT
ETag: "8008c4435cdcb1:9b7"
Content-Length: 569

...<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<body>

<form name="input" action="form_action.asp" method="POST">
First name: <input type="text" name="FirstName" value="Mickey" />

Last name: <input type="text" name="LastName" value="Mouse" />

<input type="submit" value="Submit" />
</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called "html_form_action.asp".</p>

</body>
</html>

POST /form_action.asp HTTP/1.1
Accept: image/jpeg, application/x-ms-application, image/gif, application/xaml+xml, image/pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-shockwave-flash, */*
Referer: http://84.229.173.71/form.html
Accept-Language: he-IL
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; GTB6.6; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Accept-Encoding: gzip, deflate
Host: 84.229.173.71
Content-Length: 31
Connection: Keep-Alive
Cache-Control: no-cache

FirstName=Mickey&LastName=MouseHTTP/1.1 100 Continue
Server: Microsoft-IIS/5.1
Date: Tue, 15 Feb 2011 19:17:41 GMT

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
Date: Tue, 15 Feb 2011 19:17:41 GMT

Packet 239. 5 client pkts, 6 server pkts, 9 turns. Click to select.

Entire conversation (7423 bytes)

Show and save data as ASCII

Find Next

Help

Filter Out This Stream | Print | Save as... | Back | Close

Screen-shot from Wireshark of the beginning of the discussion.

The picture shows only the beginning of the discussion, because we thought that it wasn't appropriate to provide screen-shot of the entire conversation.

Here are the details :

- First, the client send a request message (GET message) to form.html
- Then, the server answers with sent it back. We can recognize the http header with the message “200 ok”, and then the doctype html.
- The client ask another time, with the help of a POST message, the server form_action.asp. There are a lot of information details into the post message, like the languages and the accepted format...
- Then, the answer of the server is “100 continue”, follow by all the http header, which means that the request has been received and that it will be consider.
- The client so send the rest, with a get request, to get the the css file and the picture.
- Then the server send it back with a 200 ok as header.
- To continue, the client send another request for the server regular.html.
- The server sent it back : the header with all the html doctype needed.
- Here the discussion over.

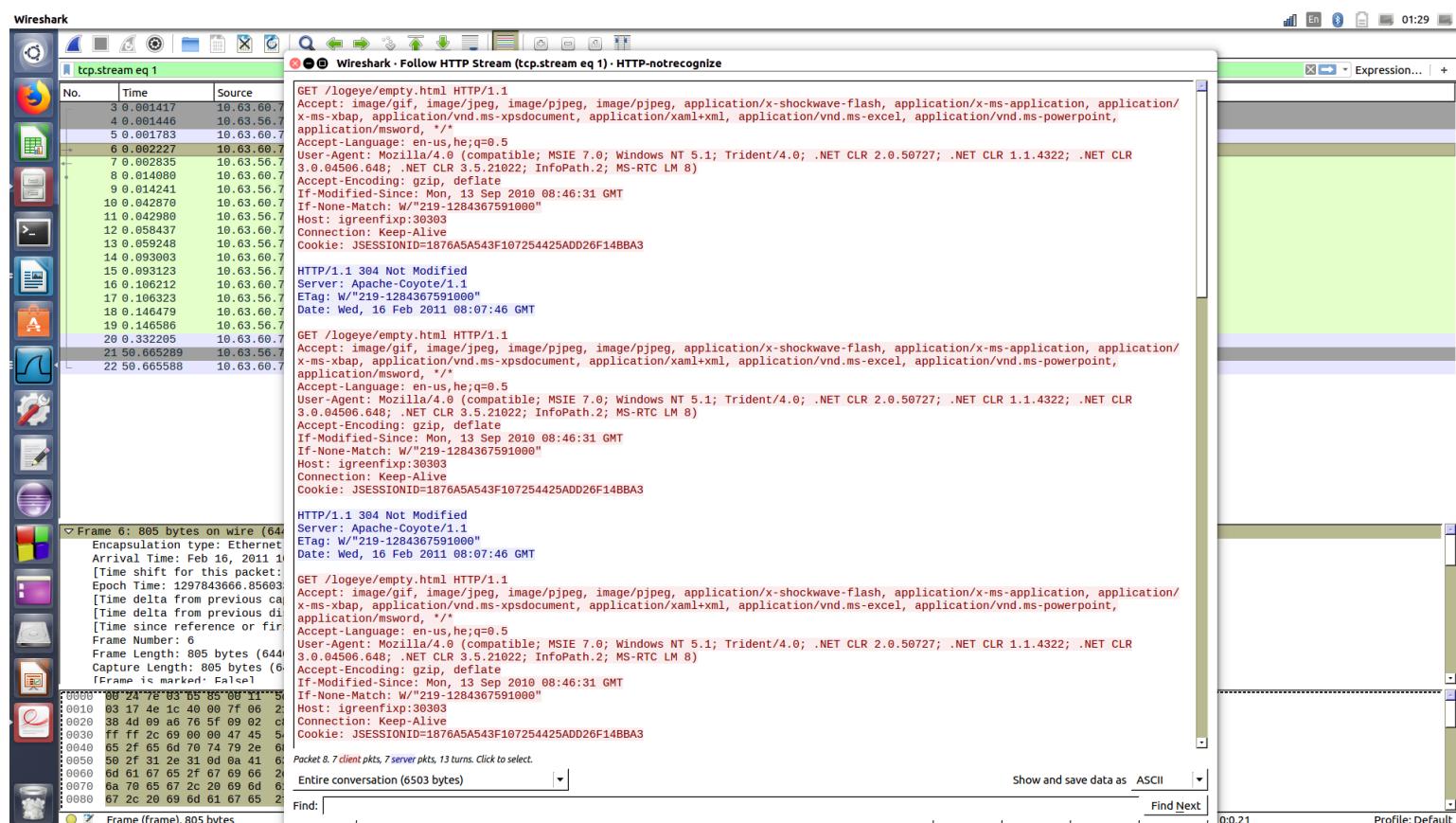
We can remark in the Wireshark packet with the filter http that, after a new request from the client, the server couldn't submit the answer. It's maybe why here is the end of the discussion.

Question 11 -

First of all, the request was a POST method, then, it's mean that it's susceptible to have side effect.

Then it does : The client has been asked to send another request because the server send the page back, and a http header with options for more data.

Question 12 -



Screen-shot from wireshark of the pcap file “notrecognizeit”, to display the discussion.

HTTP-notrecognize.pcap

http

No.	Time	Source	Destination	Protocol	Length	Info
6	0.002227	10.63.60.74	10.63.56.77	HTTP	805	GET /logeye/empty.html HTTP/1.1
7	0.002835	10.63.56.77	10.63.60.74	HTTP	176	HTTP/1.1 304 Not Modified
8	0.014680	10.63.60.74	10.63.56.77	HTTP	805	GET /logeye/empty.html HTTP/1.1
9	0.014241	10.63.56.77	10.63.60.74	HTTP	176	HTTP/1.1 304 Not Modified
10	0.042870	10.63.60.74	10.63.56.77	HTTP	805	GET /logeye/empty.html HTTP/1.1
11	0.042986	10.63.56.77	10.63.60.74	HTTP	176	HTTP/1.1 304 Not Modified
12	0.058437	10.63.60.74	10.63.56.77	HTTP	736	GET /logeye/security/common/keepAlive.jsp HTTP/1.1
13	0.059248	10.63.56.77	10.63.60.74	HTTP	441	HTTP/1.1 200 OK (text/html)
14	0.093003	10.63.60.74	10.63.56.77	HTTP	805	GET /logeye/empty.html HTTP/1.1
15	0.093123	10.63.56.77	10.63.60.74	HTTP	176	HTTP/1.1 304 Not Modified
16	0.106212	10.63.60.74	10.63.56.77	HTTP	736	GET /logeye/security/common/keepAlive.jsp HTTP/1.1
17	0.106323	10.63.56.77	10.63.60.74	HTTP	441	HTTP/1.1 200 OK (text/html)
18	0.146479	10.63.60.74	10.63.56.77	HTTP	805	GET /logeye/empty.html HTTP/1.1
19	0.146586	10.63.56.77	10.63.60.74	HTTP	176	HTTP/1.1 304 Not Modified

Frame 6: 805 bytes on wire (6440 bits), 805 bytes captured (6440 bits)

Encapsulation type: Ethernet (1)
Arrival Time: Feb 16, 2011 10:07:46.856033000 IST
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1297843666.856033000 seconds
[Time delta from previous captured frame: 0.000444000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.002227000 seconds]
Frame Number: 6
Frame Length: 805 bytes (6440 bits)
Capture Length: 805 bytes (6440 bits)
Frame is marked: False

```
00000000: 90 24 7e 03 b5 85 00 11 sd b9 d8 00 00 00 45 00 .S-.... ] ...E.
0001 03 17 4e 1c 40 00 7f 06 21 b0 0a 3f 3c 4a 0a 3f .N @... , .?<?
0002 38 4d 09 a6 76 5f 09 02 c8 31 73 d3 69 c7 50 18 8M .v... 1s.h.P.
0003 ff ff 2c 69 00 00 47 45 54 29 2f 6c 6f 67 65 79 ..i..GE T /logey
0004 05 2f 65 6d 70 74 79 2e 68 74 6d 6c 29 48 54 54 e/empty.html HTT
0005 50 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 29 69 P/1..A ccept: i
0006 0d 61 67 65 2f 67 69 66 2c 29 69 6d 61 67 65 2f mage/gif , image/
0007 0a 70 65 67 2c 29 69 6d 61 67 65 2f 70 6a 70 65 jpeg, im age/pjpeg
0008 07 2c 29 69 6d 61 67 65 2f 79 6a 70 65 67 2c 29 g, image /pjpeg,
```

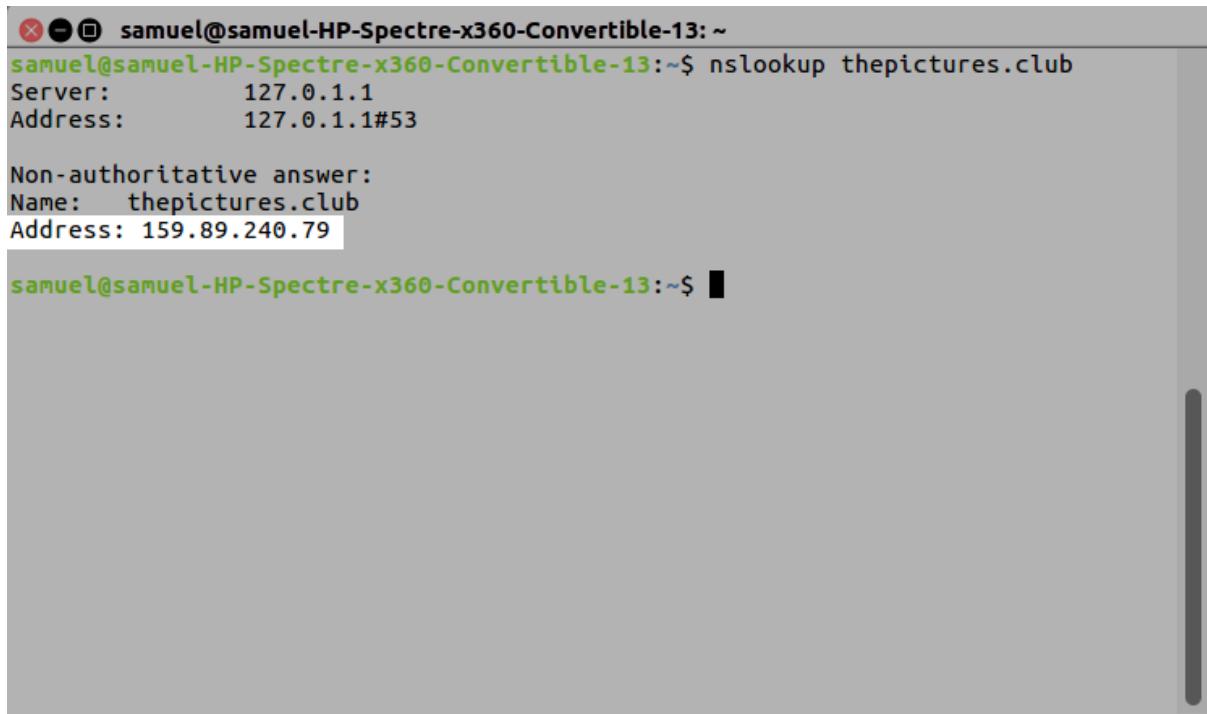
Packets: 22 - Displayed: 14 (63.6%) - Load time: 0:0.0 Profile: Default

Screen-shot from wireshark of the pcap file “notrecognizeit”

It's seem that Wireshark recognize the discussion.

Question 13 -

13.1 -

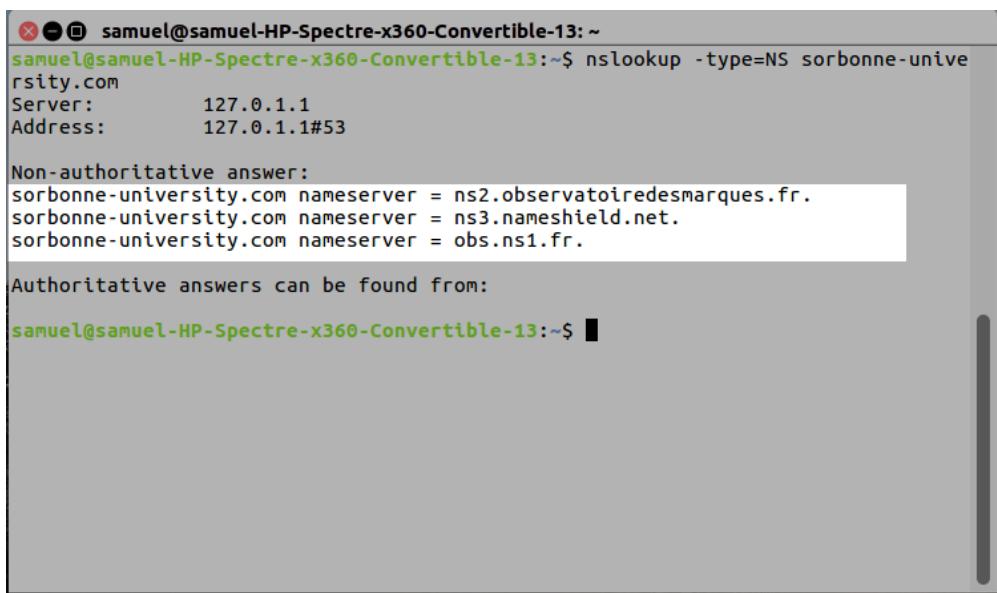


```
 samuel@samuel-HP-Spectre-x360-Convertible-13:~  
 samuel@samuel-HP-Spectre-x360-Convertible-13:~$ nslookup thepictures.club  
 Server:      127.0.1.1  
 Address:     127.0.1.1#53  
  
 Non-authoritative answer:  
 Name:   thepictures.club  
 Address: 159.89.240.79  
  
 samuel@samuel-HP-Spectre-x360-Convertible-13:~$
```

Screen-shot of the terminal who display the address of an Asiatic server using *nslookup*.

As we can read in the picture above, the ip address of the server is :
“159.89.240.79”.

13.2 -



```
 samuel@samuel-HP-Spectre-x360-Convertible-13:~  
 samuel@samuel-HP-Spectre-x360-Convertible-13:~$ nslookup -type=NS sorbonne-unive  
 rsity.com  
 Server:      127.0.1.1  
 Address:     127.0.1.1#53  
  
 Non-authoritative answer:  
 sorbonne-university.com nameserver = ns2.observatoiredesmarques.fr.  
 sorbonne-university.com nameserver = ns3.nameshield.net.  
 sorbonne-university.com nameserver = obs.ns1.fr.  
  
 Authoritative answers can be found from:  
  
 samuel@samuel-HP-Spectre-x360-Convertible-13:~$
```

Screen-shot who display the authoritative server of “la sorbonne” university using nslookup.

The name of the authoritative DNS server for La Sorbonne university are :

- ns2.observatoiredesmarques.fr
- ns3.nameshield.net
- obs.ns1.fr

as we can see in the picture above.

13.3 -

```
 samuel@samuel-HP-Spectre-x360-Convertible-13: ~
samuel@samuel-HP-Spectre-x360-Convertible-13:~$ nslookup -type=NS sorbonne-university.com
Server:      127.0.1.1
Address:     127.0.1.1#53

Non-authoritative answer:
sorbonne-university.com nameserver = ns2.observatoiredesmarques.fr.
sorbonne-university.com nameserver = ns3.nameshield.net.
sorbonne-university.com nameserver = obs.ns1.fr.

Authoritative answers can be found from:

samuel@samuel-HP-Spectre-x360-Convertible-13:~$ nslookup mail.yahoo.com obs.ns1.fr
Server:      obs.ns1.fr
Address:     81.92.84.146#53

** server can't find mail.yahoo.com: REFUSED

samuel@samuel-HP-Spectre-x360-Convertible-13:~$ █
```

Screen-shot who display the authoritative server of “la sorbonne” university using nslookup.

The ip address is : 81.92.84.146.

13.4 -

Wireshark capture showing DNS query message:

destination port of DNS query message: 53

Query port: 192.115.106.35

protocole UDP

```

0010 45 00 00 3b 9e 17 40 00 48 11 0a 06 01 b6 E...;..@. @.fH...
0020 c0 73 6a 23 8d e8 00 35 00 27 b7 c8 d1 01 00 .sJ#...5 .z...
0030 00 01 00 00 00 00 00 04 77 77 77 36 04 69 65 ..... www6.ie
0040 74 66 03 6f 72 67 00 00 01 00 00 01 00 00 01 tf.org...

```

Wireshark capture showing DNS response message:

source port of DNS response message: 53

protocole UDP

```

0010 45 00 00 59 70 a9 49 00 fb 11 d8 97 c0 73 6a 23 E...Yp@...Sj#
0020 0a 00 01 00 02 00 00 00 00 04 45 9c f1 7a d1 81 80 .....5 .E.z...
0030 00 01 00 02 00 00 00 00 04 77 77 77 36 04 69 65 ..... www6.ie
0040 74 66 03 6f 72 67 00 00 01 00 01 c0 00 05 00 00 tf.org...
0050 01 00 00 07 00 00 02 c0 11 c0 11 00 00 01 00 00 00 ..... ,
0060 00 07 00 00 04 1f c6 2c ..... ,

```

Screen-shots who from wireshark (up Query, down Response)

As we can see in the pictures above, the DNS query and response messages are sent over UDP (User Datagram Protocol).

13.5 -

By looking the same screen-shot from the question 4, we notice that :

source port of the DNS query message is 36328,

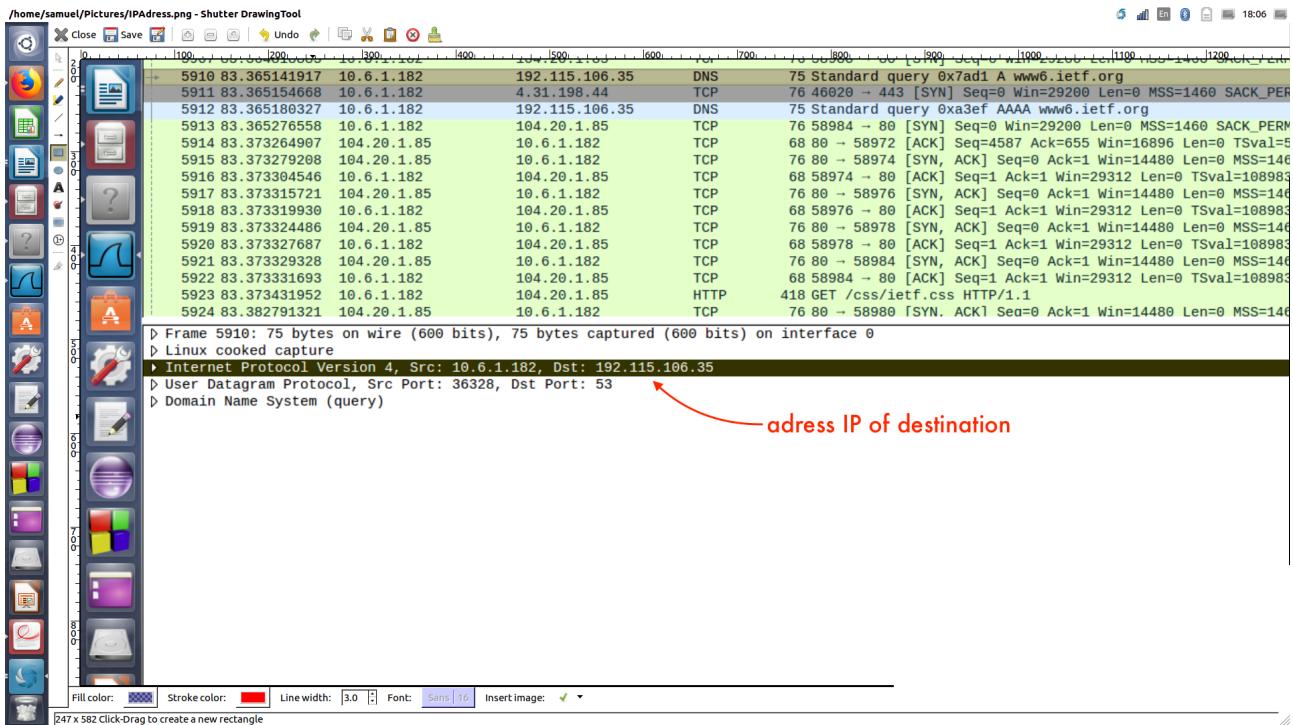
source port of the DNS query response is 53,

and reciprocally,

destination port of the DNS query message is 53,

destination port of the DNS query response is 36328,

13.6 -



Screen-shot who from wireshark.

The DNS query message send to the IP address 192.115.106.35, as we can see above.

```

samuel@samuel-HP-Spectre-x360-Convertible-13: ~
samuel@samuel-HP-Spectre-x360-Convertible-13:~$ nmcli dev show | grep DNS
IP4.DNS[1]:                               192.115.106.35
samuel@samuel-HP-Spectre-x360-Convertible-13:~$ █

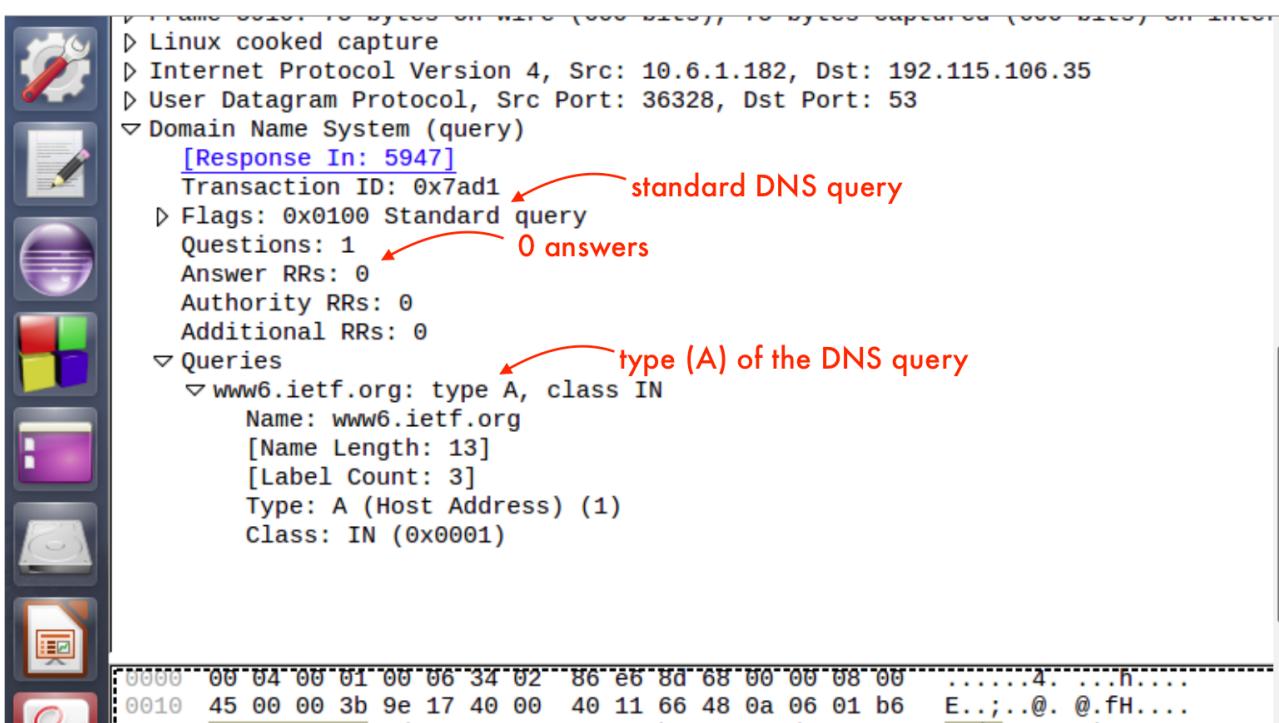
```

same IP address

Screen-shot from the terminal and the IP address of your local DNS server.

The two address are the same : 192.115.106.35.

13.7 -

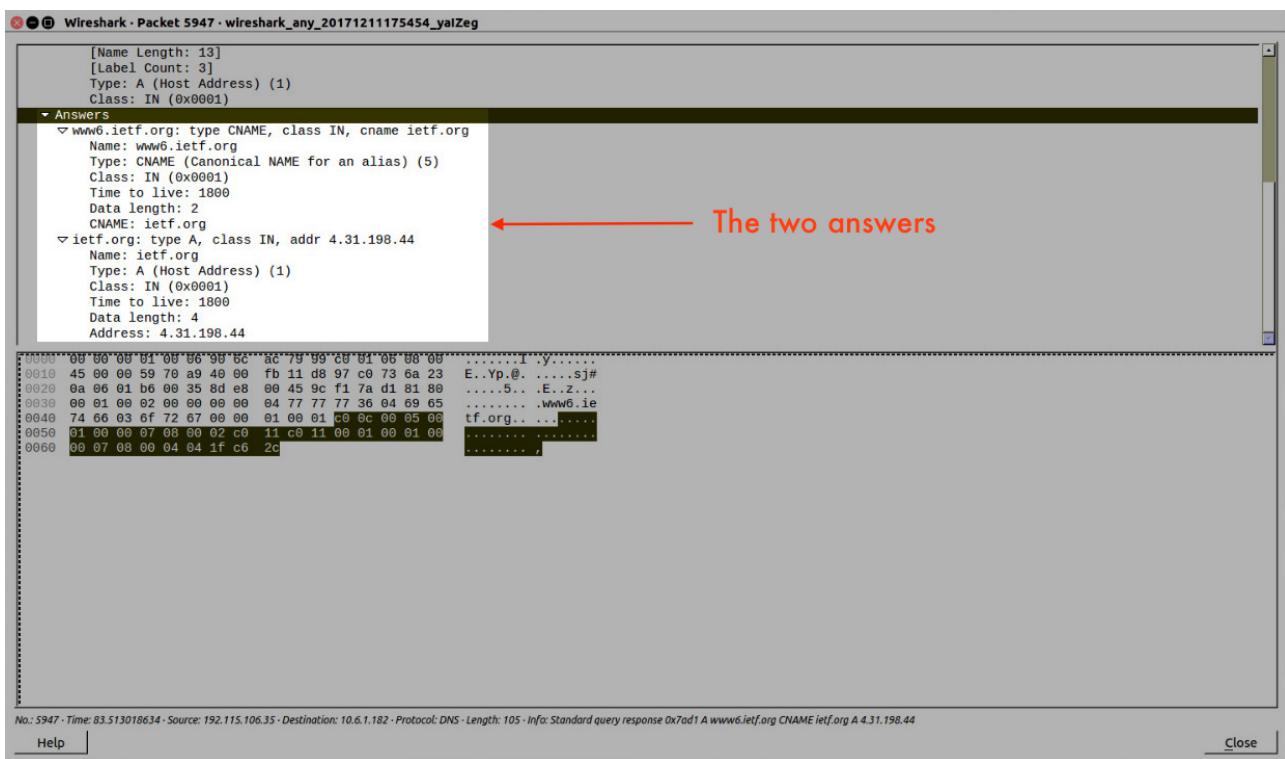


Screen-shot who from Wireshark who display the type of DNS query.

The type of DNS query is type A standard query.

The message query doesn't contains any answer.

13.8 -



Screen-shot who from wireshark who display the answers.

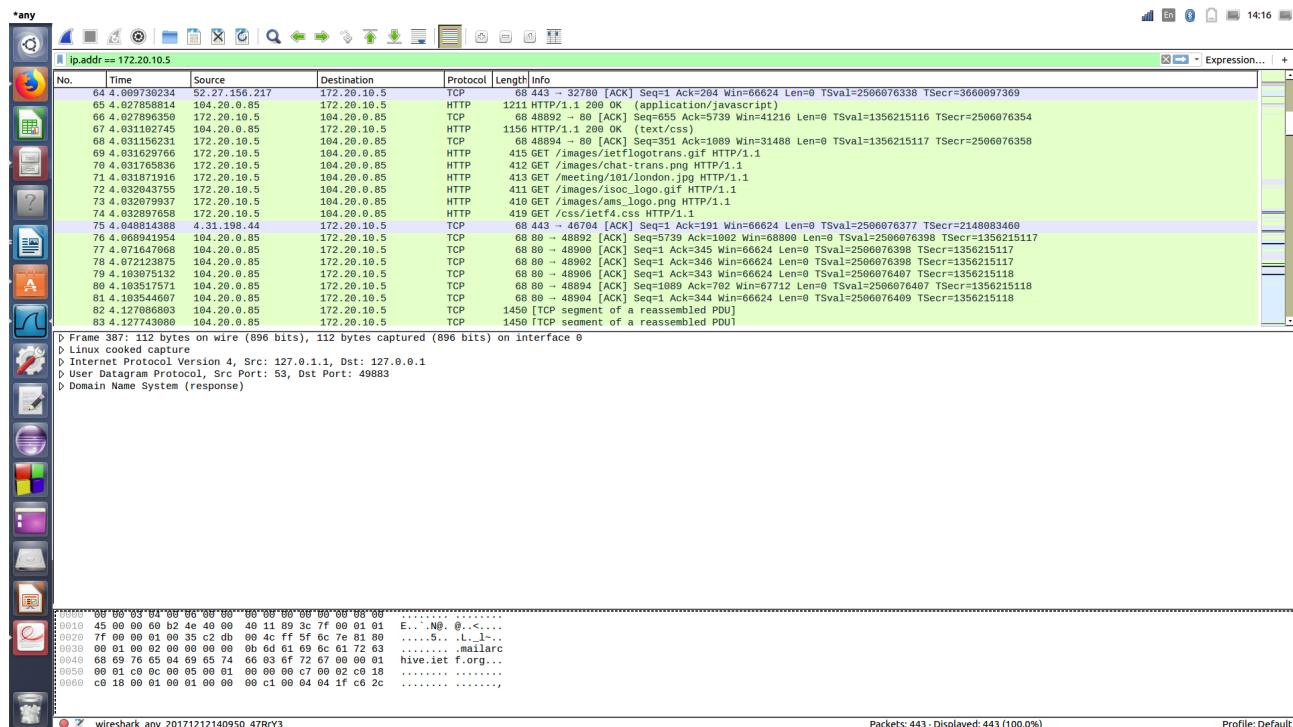
We notice two answers which contains information about the host, the type of the address, the class, the time to live, the data length, the name, and the IP address (4.31.198.44).

13.9 -

Yes it does.

The destination of the SYN packet is 4.31.198.44, is the same address that was provided in the DNS response message as the type “A” address of the webpage.

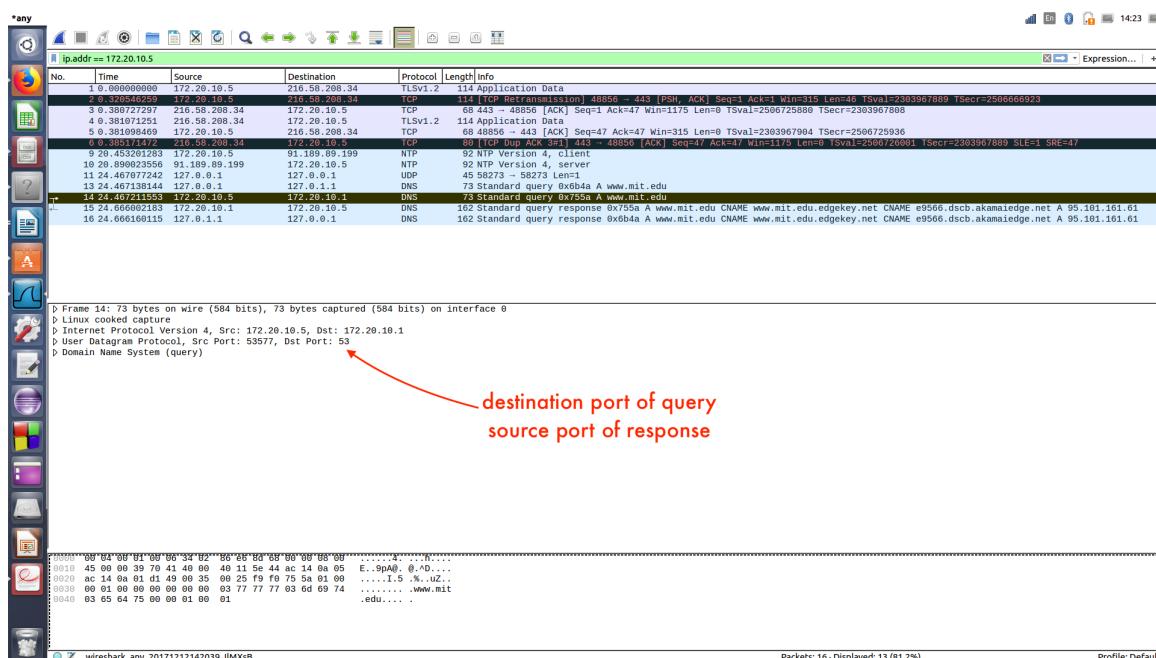
13.10 -



Screen-shot who from wireshark.

As we can see, there is no new query before retrieving image.

13.11 -

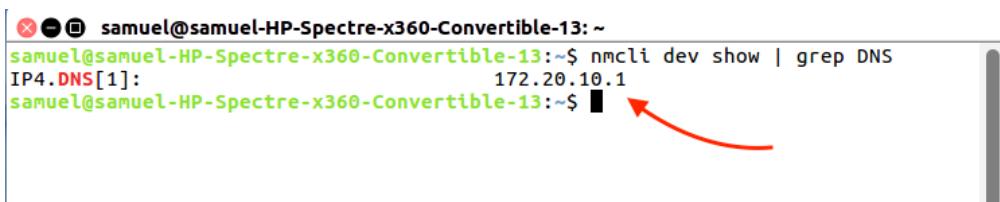


Screen-shot who from Wireshark.

The destination port for the DNS query message is 53,

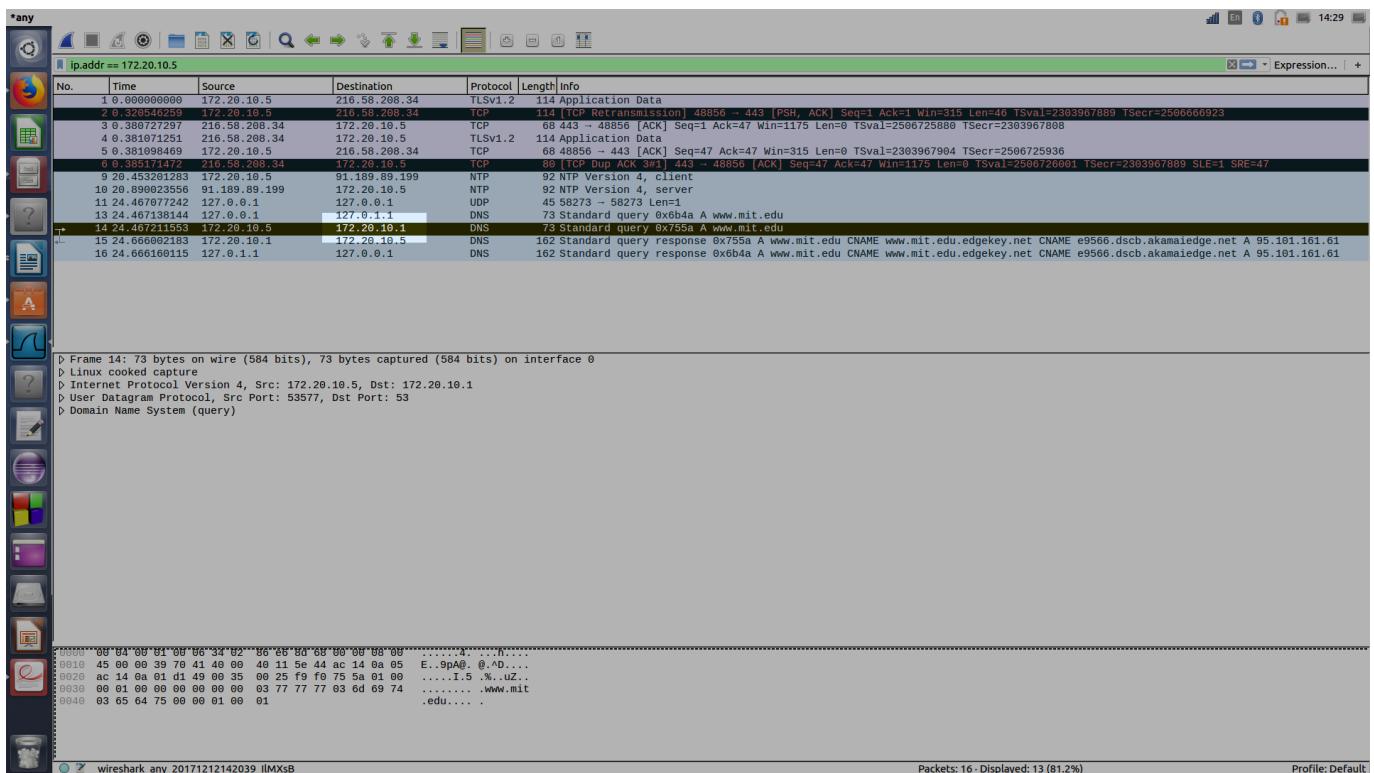
The source port for the response query is 53.

13.12 -



```
 samuel@samuel-HP-Spectre-x360-Convertible-13: ~
 samuel@samuel-HP-Spectre-x360-Convertible-13:~$ nmcli dev show | grep DNS
 IP4.DNS[1]:
          172.20.10.1
 samuel@samuel-HP-Spectre-x360-Convertible-13:~$
```

Screen-shot who from the terminal.



Screen-shot who from wireshark.

The IP of the DNS server is 172.20.10.1 same as my local DNS server.

13.13 -

The screenshot shows a Wireshark capture of a DNS query. The packet details pane indicates it's a standard query from 10.11.1.29 to 192.115.106.35. The DNS pane shows a query for 'www.mit.edu' of type 'A'. A red arrow points from the text 'www.mit.edu: type A, class IN' to the corresponding entry in the DNS pane. The bytes pane displays the raw hex and ASCII data of the DNS message.

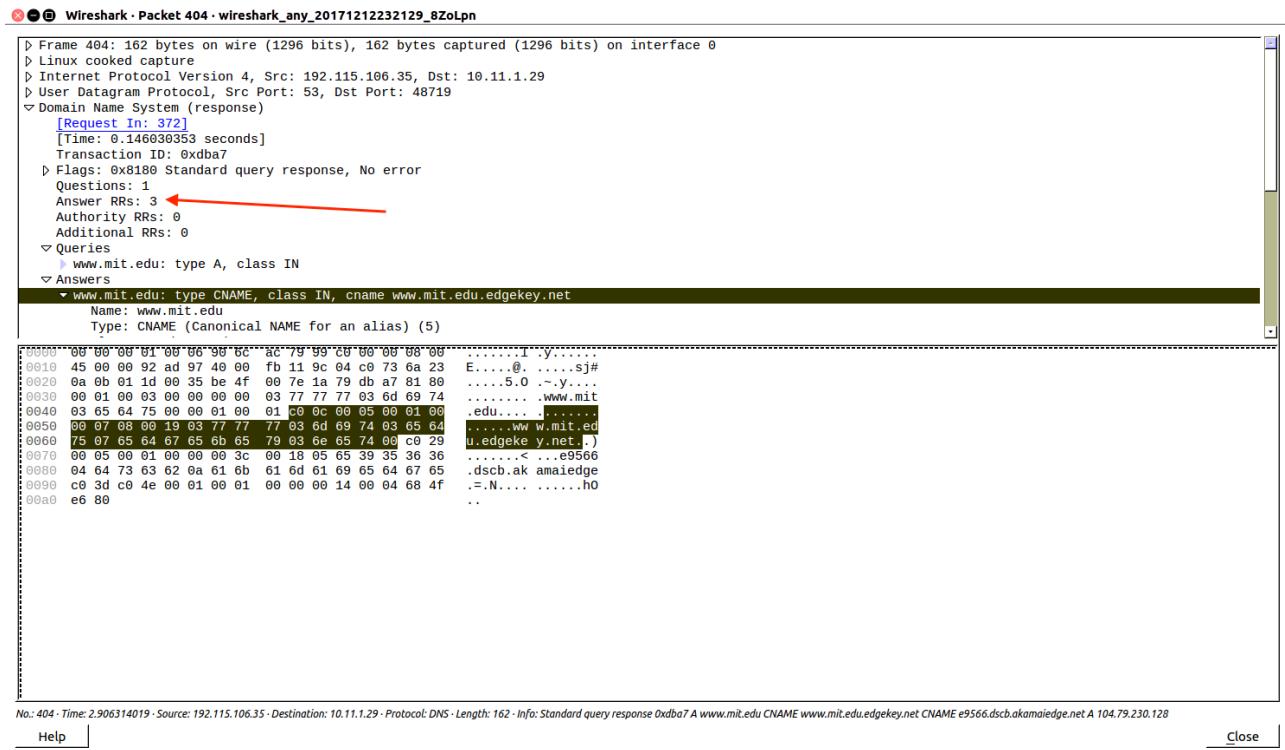
```
No.: 372 · Time: 2.760283666 · Source: 10.11.1.29 · Destination: 192.115.106.35 · Protocol: DNS · Length: 73 · Info: Standard query 0xd0ba7 A www.mit.edu
```

Hex	Dec	ASCII
0000	00 00 00 00 00 00 00 004.
0010	45 00 00 39 9c 46 40 00	E..9.F@. @.h....
0020	c0 73 6a 23 be 4f 00 35	.sj#.0.5 %.
0030	00 01 00 00 00 00 00 00 www.mit
0040	03 65 64 75 00 00 01 00	.edu....

Screen-shot who from wireshark, to display the type of the query message.

As we can see above in the picture, the message is type A, and there is no any answers.

13.14 -



Wireshark - Packet 404 · wireshark_any_20171212232129_8ZoLpn

Frame 404: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface 0
D Linux cooked capture
D Internet Protocol Version 4, Src: 192.115.106.35, Dst: 10.11.1.29
D User Datagram Protocol, Src Port: 53, Dst Port: 48719
Domain Name System (response)
[Request Id: 372]
[Time: 0.146030353 seconds]
Transaction ID: 0xdba7
Flags: 0x8100 Standard query response, No error
Questions: 1
Answer RRs: 3
Authority RRs: 0
Additional RRs: 0
Queries
www.mit.edu: type A, class IN
Answers
www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
Name: www.mit.edu
Type: CNAME (Canonical NAME for an alias) (5)

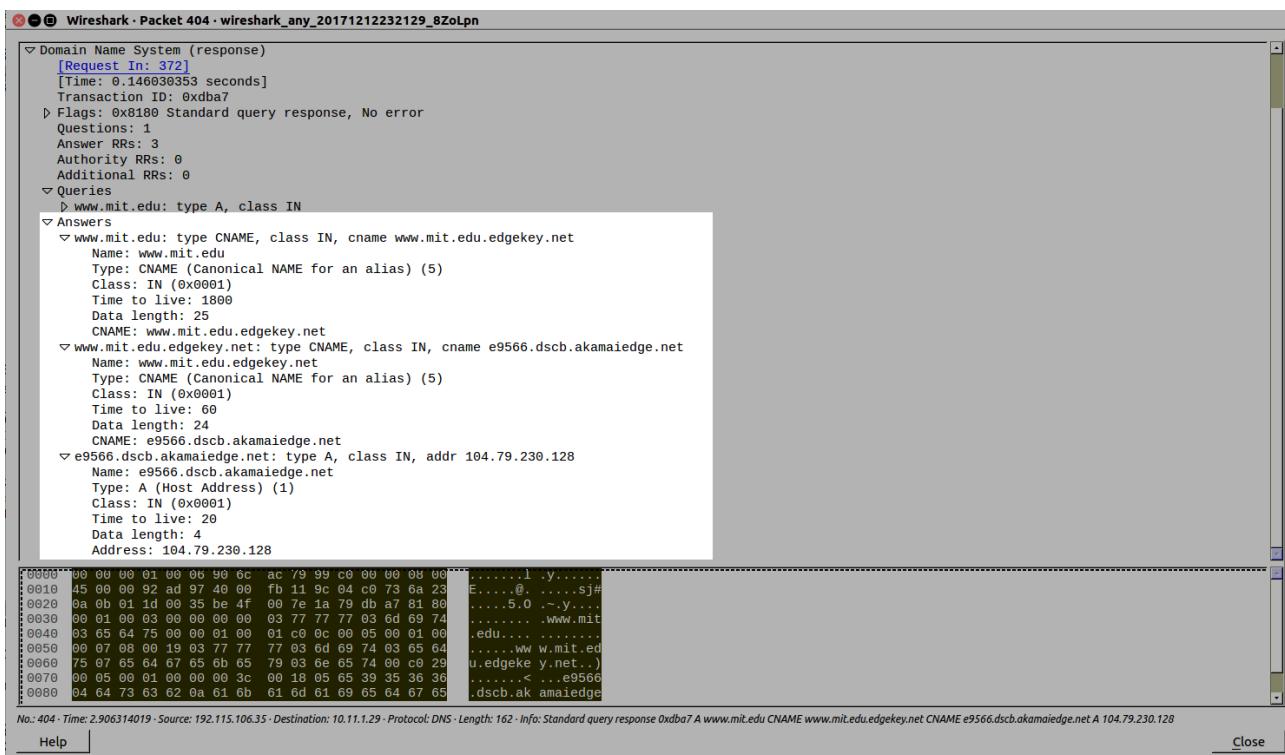
0000 00 00 00 01 00 00 90 6c ac 79 99 c0 00 00 08 001.y.....
0010 45 00 00 92 ad 97 40 00 fb 11 9c 04 c0 73 6a 23 E....@.sj#
0020 0a 0b 01 1d 00 35 be 4f 00 7e 1a 79 db a7 81 805.0 -.y....
0030 00 01 00 03 00 00 00 00 03 77 77 77 03 6d 69 74www.mit.
0040 03 65 64 75 00 00 01 00 01 c0 0c 00 05 00 01 00 .edu....
0050 00 07 08 00 19 03 77 77 77 03 6d 69 74 03 65 64 ..ww w.mit.ed
0060 75 07 65 64 67 65 6b 65 79 03 6e 65 74 00 c0 29 u.edgeke y.net..)
0070 00 05 00 01 00 00 00 3c 00 18 05 65 39 35 36 36< ...e9566
0080 04 64 73 63 62 0a 61 6b 61 6d 61 69 65 64 67 65 .dscb.ak amaiedge
0090 c0 3d c0 4e 00 01 00 01 00 00 00 14 00 04 68 4f .=N....ho
00a0 e6 80 ..

No.: 404 · Time: 2.906314019 · Source: 192.115.106.35 · Destination: 10.11.1.29 · Protocol: DNS · Length: 162 · Info: Standard query response 0xdba7 A www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e9566.dscb.akamaiedge.net A 104.79.230.128

Help Close

Screen-shot who from wireshark, to display the numbers of answers.

There are 3 answers as mentioned in the picture above.



Wireshark - Packet 404 · wireshark_any_20171212232129_8ZoLpn

Frame 404: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface 0
D Linux cooked capture
D Internet Protocol Version 4, Src: 192.115.106.35, Dst: 10.11.1.29
D User Datagram Protocol, Src Port: 53, Dst Port: 48719
Domain Name System (response)
[Request Id: 372]
[Time: 0.146030353 seconds]
Transaction ID: 0xdba7
Flags: 0x8100 Standard query response, No error
Questions: 1
Answer RRs: 3
Authority RRs: 0
Additional RRs: 0
Queries
www.mit.edu: type A, class IN
Answers
www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
Name: www.mit.edu
Type: CNAME (Canonical NAME for an alias) (5)
Class: IN (0x0001)
Time to live: 1800
Data length: 25
CNAME: www.mit.edu.edgekey.net
www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dscb.akamaiedge.net
Name: www.mit.edu.edgekey.net
Type: CNAME (Canonical NAME for an alias) (5)
Class: IN (0x0001)
Time to live: 60
Data length: 24
CNAME: e9566.dscb.akamaiedge.net
e9566.dscb.akamaiedge.net: type A, class IN, addr 104.79.230.128
Name: e9566.dscb.akamaiedge.net
Type: A (Host Address) (1)
Class: IN (0x0001)
Time to live: 20
Data length: 4
Address: 104.79.230.128

0000 00 00 00 01 00 00 90 6c ac 79 99 c0 00 00 08 001.y.....
0010 45 00 00 92 ad 97 40 00 fb 11 9c 04 c0 73 6a 23 E....@.sj#
0020 0a 0b 01 1d 00 35 be 4f 00 7e 1a 79 db a7 81 805.0 -.y....
0030 00 01 00 03 00 00 00 00 03 77 77 77 03 6d 69 74www.mit.
0040 03 65 64 75 00 00 01 00 01 c0 0c 00 05 00 01 00 .edu....
0050 00 07 08 00 19 03 77 77 77 03 6d 69 74 03 65 64 ..ww w.mit.ed
0060 75 07 65 64 67 65 6b 65 79 03 6e 65 74 00 c0 29 u.edgeke y.net..)
0070 00 05 00 01 00 00 00 3c 00 18 05 65 39 35 36 36< ...e9566
0080 04 64 73 63 62 0a 61 6b 61 6d 61 69 65 64 67 65 .dscb.ak amaiedge
00a0 e6 80 ..

No.: 404 · Time: 2.906314019 · Source: 192.115.106.35 · Destination: 10.11.1.29 · Protocol: DNS · Length: 162 · Info: Standard query response 0xdba7 A www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e9566.dscb.akamaiedge.net A 104.79.230.128

Help Close

Screen-shot who from wireshark, to display the answers.

1. The first response message is of type CNAME that point to a domain name : www.mit.edu.edgekey.net.
2. The second response message is another type CNAME that point to another domain name : e9566.dscb.akamaiedge.net.
3. The last response message is a type A that points to the ip address : 104.79.230.128.

Question – 15

All the interesting screen-shots have been provided.

Wireshark - Packet 404 · wireshark_any_20171212232129_8ZoLpn

Domain Name System (response)
 [Request ID: 372]
 [Time: 0.146030353 seconds]
 Transaction ID: 0xda7
 Flags: 0x8180 Standard query response, No error
 Questions: 1
 Answer RRs: 3
 Authority RRs: 0
 Additional RRs: 0
 Queries
 www.mit.edu: type A, class IN
 Answers
 www.mit.edu: type CNAME, class IN, cname www.mit.edu.edgekey.net
 Name: www.mit.edu
 Type: CNAME (Canonical NAME for an alias) (5)
 Class: IN (0x0001)
 Time to live: 1800
 Data length: 25
 CNAME: www.mit.edu.edgekey.net
 www.mit.edu.edgekey.net: type CNAME, class IN, cname e9566.dsdb.akamaiedge.net
 Name: www.mit.edu.edgekey.net
 Type: CNAME (Canonical NAME for an alias) (5)
 Class: IN (0x0001)
 Time to live: 60
 Data length: 24
 CNAME: e9566.dsdb.akamaiedge.net
 e9566.dsdb.akamaiedge.net: type A, class IN, addr 104.79.230.128
 Name: e9566.dsdb.akamaiedge.net
 Type: A (Host Address) (1)
 Class: IN (0x0001)
 Time to live: 20
 Data length: 4
 Address: 104.79.230.128

0000	00 00 00 01 00 06 90 6c ac 79 99 c0 00 00 00 00l.y.....
0010	45 00 00 92 ad 97 40 09 fb 11 9c 04 09 73 6a 23	E....@...Sj#
0020	0a 0b 01 1d 00 35 be 4f 00 7e 1a 79 db a7 81 805.0 ~.y...
0030	00 01 00 03 00 00 00 00 03 77 77 03 6d 69 74www.mit
0040	03 65 64 75 00 00 01 00 01 c0 0c 00 05 00 01 00	.edu.....
0050	00 07 08 00 19 03 77 77 03 60 69 74 03 65 64ww w.mit.ed
0060	75 07 65 64 67 65 6b 65 79 03 6e 65 74 00 c0 29	u.edgeke y.net..)
0070	00 05 00 01 00 00 00 3c 00 18 05 65 39 35 30 36< ...e9566
0080	04 64 73 63 62 0a 61 6b 61 6d 61 69 65 64 67 65	.dsdb.ak amailedge

No: 404 · Time: 2.906314019 · Source: 192.115.106.35 · Destination: 10.11.1.29 · Protocol: DNS · Length: 162 · Info: Standard query response Oxda7 A www.mit.edu CNAME www.mit.edu.edgekey.net CNAME e9566.dsdb.akamaiedge.net A 104.79.230.128

Help

Close