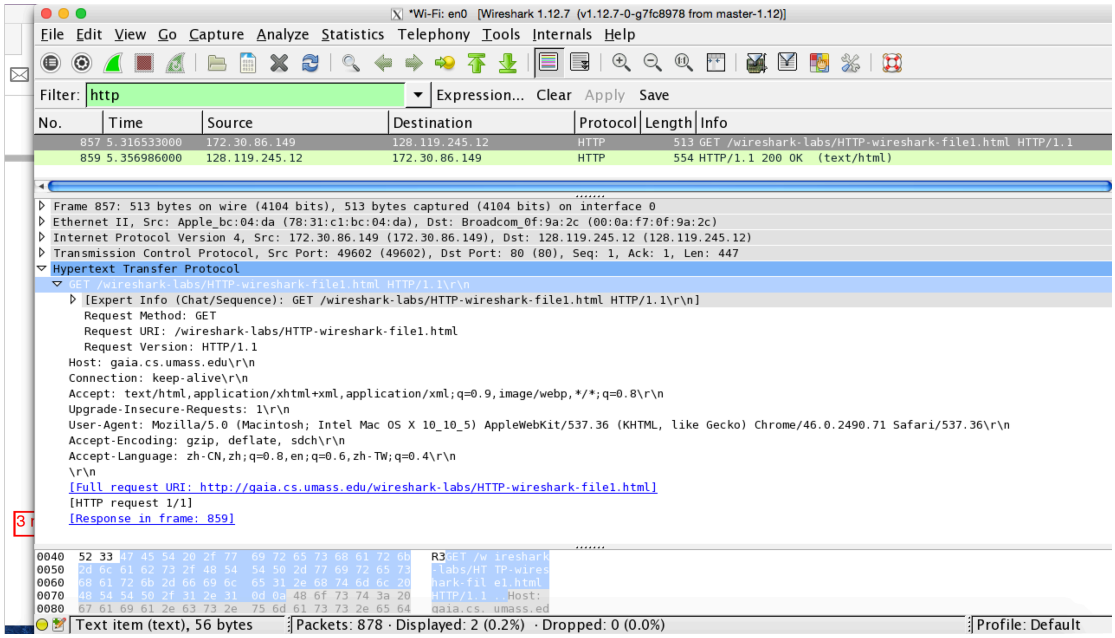
Lab2 Report – Xinyu Yun (xyun@uwo.ca)

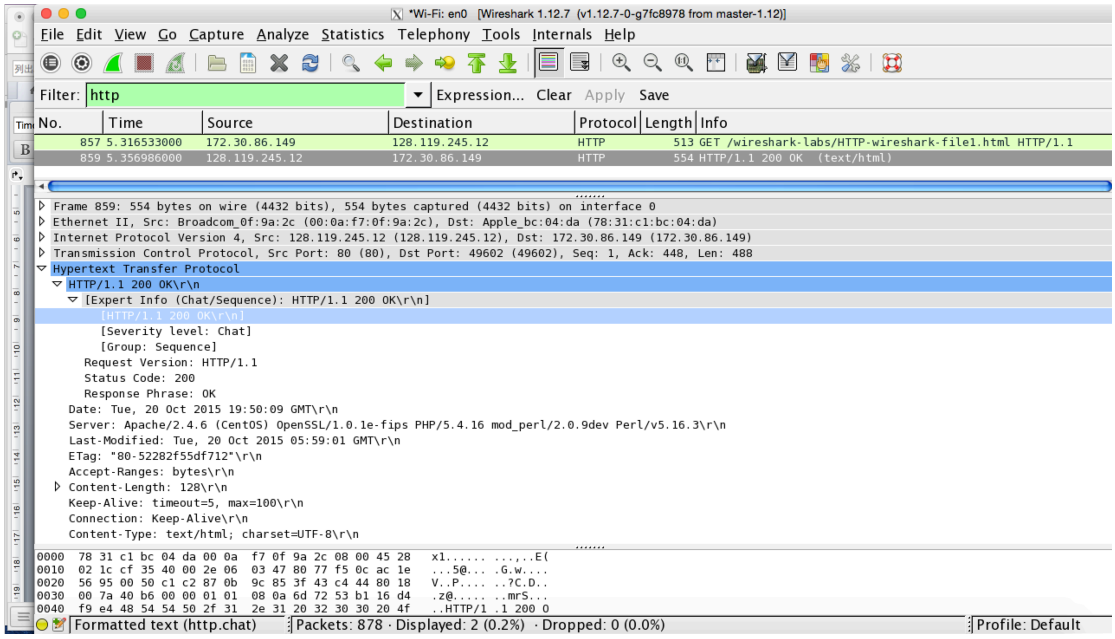
Wireshark Lab: HTTP v6.1

1. The Basic HTTP GET/response interaction

Request Details：



Response details：



1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

Answer: my browser is running HTTP 1.1; the response package also shows the HTTP version is 1.1.

2. What languages (if any) does your browser indicate that it can accept to the server?

   Answer: As showed in request header: "Accept-Language:zh-CN,zh;q=0.8,en;q=0.6,zh-TW;q=0.4", the browser will support both English and Chinese.

3. What is the IP address of your computer? Of the gaia.cs.umass.edu server?

   Answer: in the request package, my IP is the source "172.30.86.149"; the destination IP of the server is "128.119.245.12"

4. What is the status code returned from the server to your browser?

   Answer: As shown in response package: Status Code: 200

5. When was the HTML file that you are retrieving last modified at the server?

   Answer: Last-Modified: Tue, 20 Oct 2015 05:59:01 GMT

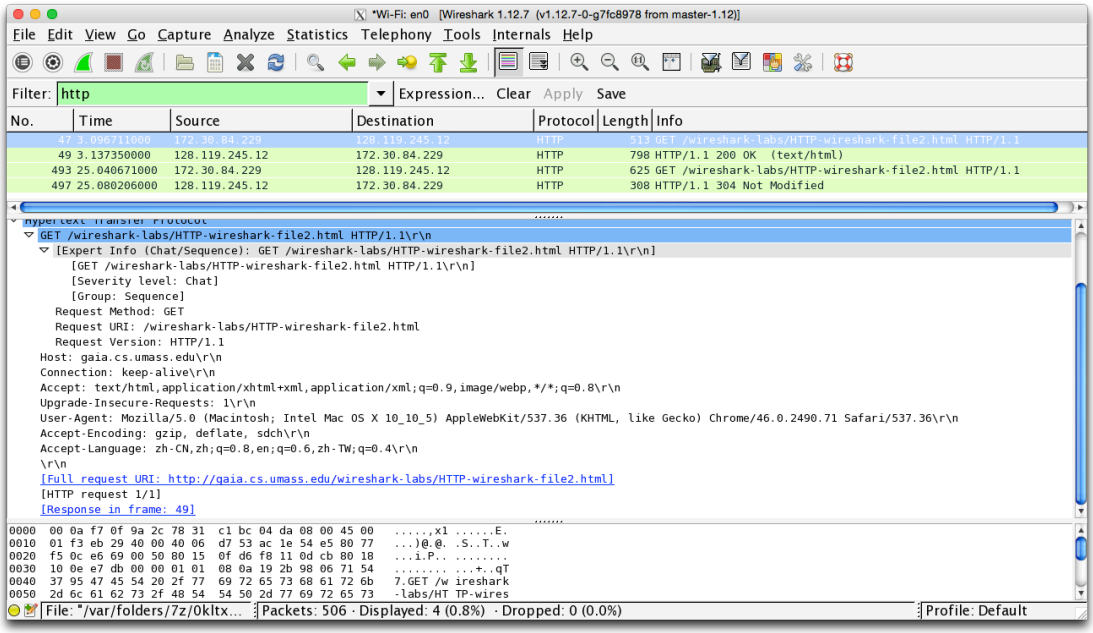
6. How many bytes of content are being returned to your browser?

   Answer：Content length: 128

7. By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

   Answer：As I check, all the message in [] will not be displayed in raw data.

2. The HTTP CONDITIONAL GET/response interaction

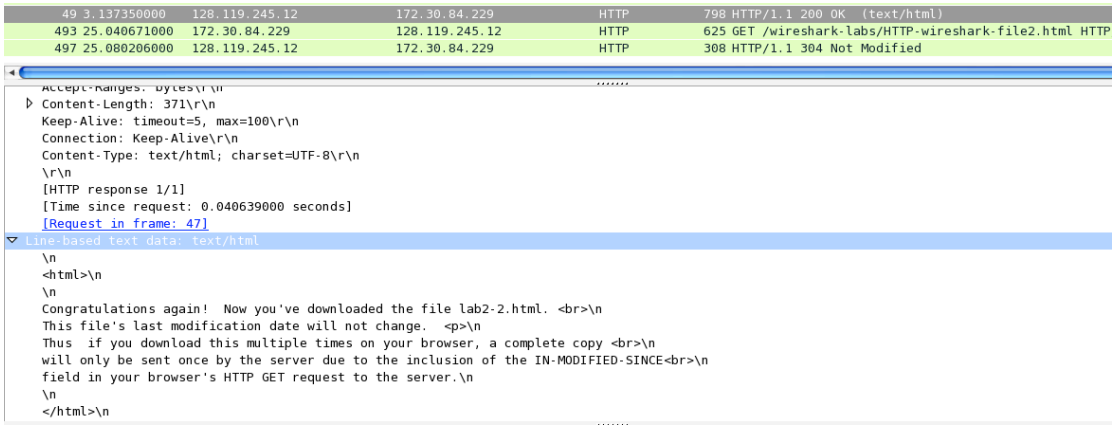


8. Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE” line in the HTTP GET?

Answer: No, I do not see the line in first HTTP GET message.

9. Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

Answer: yes, the server explicitly return the contents of HTML file. I can tell from the response context with <html> and 5 lines content.



10. Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

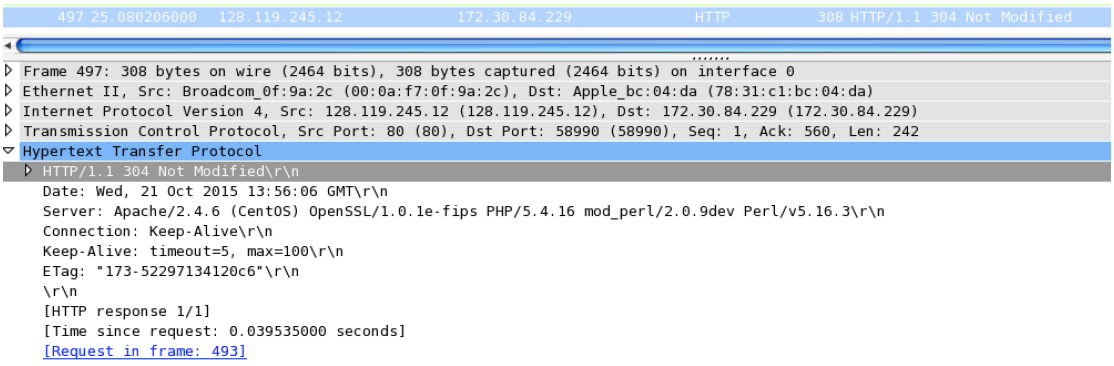
Answer: yes I see the line from second HTTP GET:

If-Modified-Since: Wed, 21 Oct 2015 05:59:01 GMT

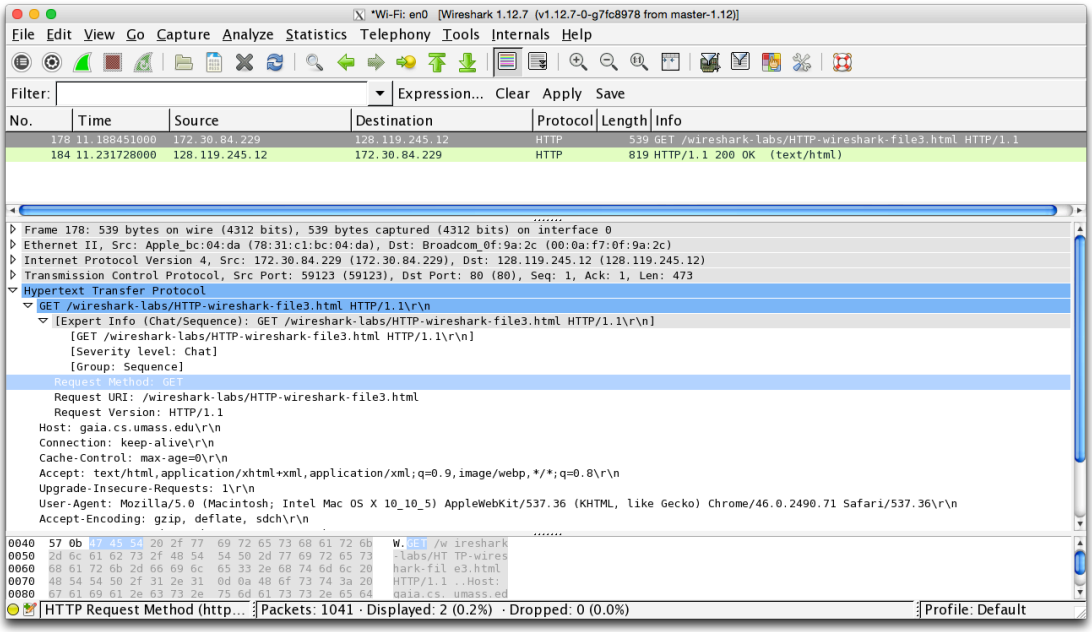
11. What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

Answer: in the second HTTP GET, the status is: HTTP/1.1 304 Not Modified

The server did not return any content for HTML file, judging from the length of response and details bellowed:



3. Retrieving Long Documents



12. How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill or Rights?

Answer: Only 1 HTTP GET message was sent, and the packet number is 178.

13. Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

Answer: NO 184 packet contains the status code to response the HTTP GET request.

14. What is the status code and phrase in the response?

Answer: Status Code: 200

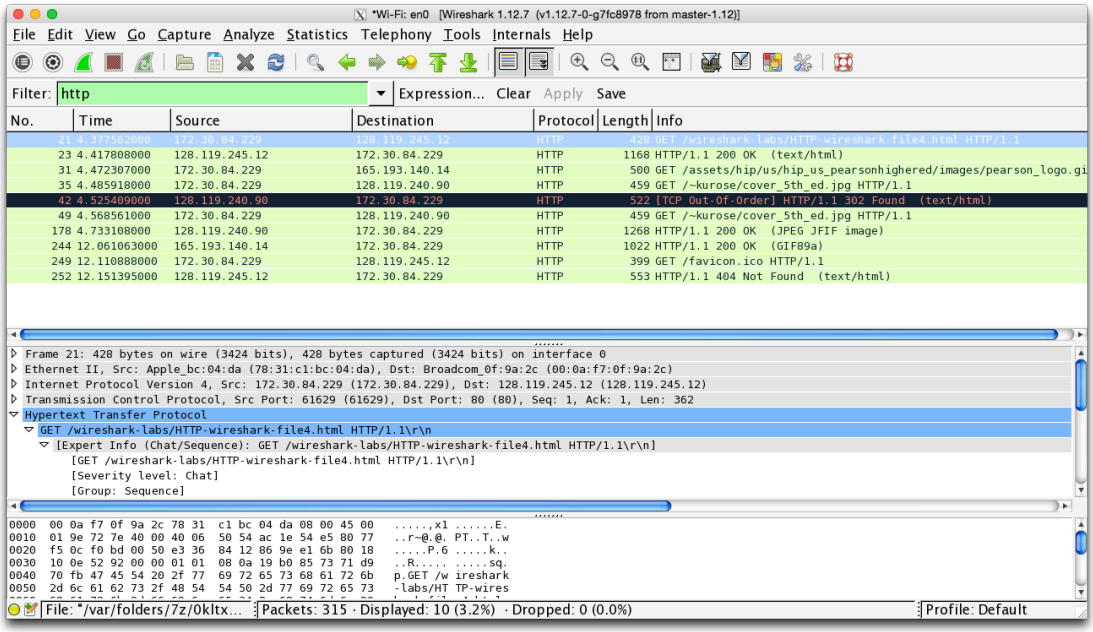
15. How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

Answer: as shown bellowed, 4 TCP segments carry all messages of the response and the text.

No.	Time	Source	Destination	Protocol
178	11.188451000	172.30.84.229	128.119.245.12	HTTP
184	11.231728000	128.119.245.12	172.30.84.229	HTTP

```
    Window size value: 122
    [Calculated window size: 15616]
    [Window size scaling factor: 128]
  ▷ Checksum: 0x446c [validation disabled]
    Urgent pointer: 0
  ▷ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▷ [SEQ/ACK analysis]
    TCP segment data (753 bytes)
▽ [4 Reassembled TCP Segments (4863 bytes): #180(1370), #181(1370), #183(1370), #184(75
    [Frame: 180, payload: 0-1369 (1370 bytes)]
    [Frame: 181, payload: 1370-2739 (1370 bytes)]
    [Frame: 183, payload: 2740-4109 (1370 bytes)]
    [Frame: 184, payload: 4110-4862 (753 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 4863]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a2057...]
▽ Hypertext Transfer Protocol
  ▽ HTTP/1.1 200 OK\r\n
    ▽ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
        [HTTP/1.1 200 OK\r\n]

0000  48 54 54 50 2f 31 2e 31  20 32 30 30 20 4f 4b 0d   HTTP/1.1  200 OK.
0010  0a 44 61 74 65 3a 20 57  65 64 2c 20 32 31 20 4f   .Date: W ed, 21 O
0020  63 74 20 32 30 31 35 20  31 34 3a 34 31 3a 34 35   ct 2015  14:41:45
```

4. HTML Documents with Embedded Objects



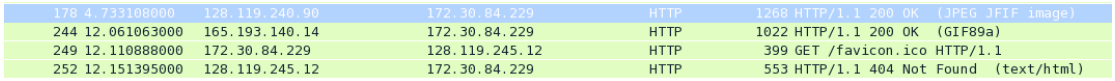
16. How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

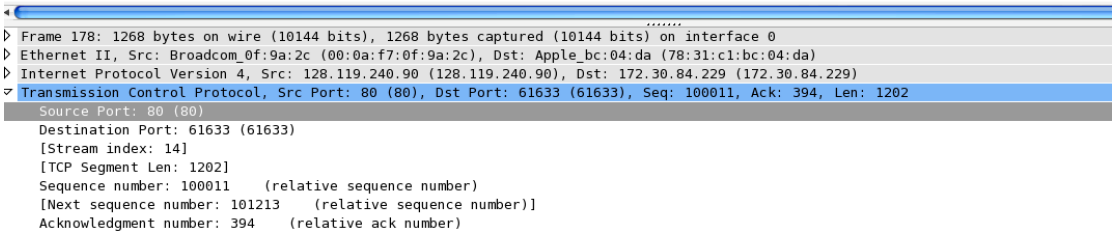
Answer: As shown from above screenshot, except the GET request to 'GET /favicon.ico' there are 4 HTTP GET requests have been sent to 3 destinations:

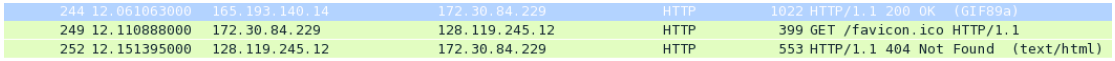
1. To 128.119.245.12 to get the html page
2. To 165.193.140.14 to retrieve the pearson_logo.gif
3. To 172.30.84.299 to retrieve cover_5th_ed.jpg, request has been sent twice because the first GET request got the '302 Found'

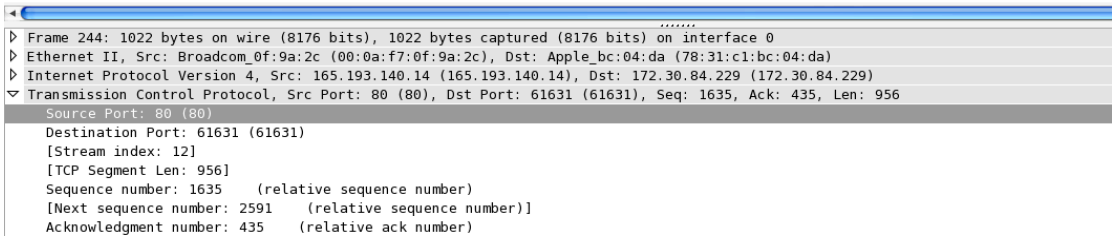
17. Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

Answer: As checked the image files are transferred in TCP layer through the same port 80, so the images downloading is serially.





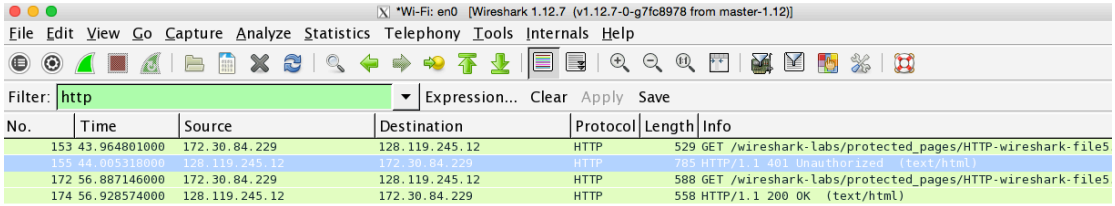


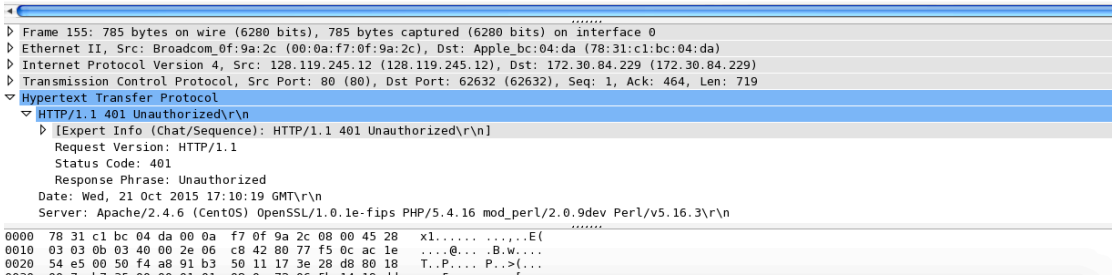


5 HTTP Authentications

18. What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

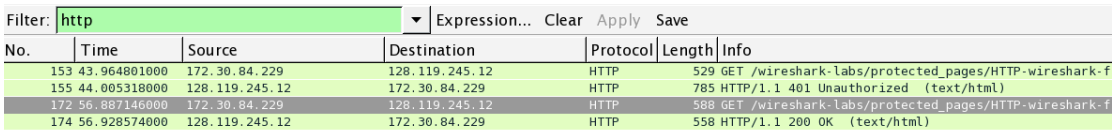
Answer: as shown bellowed, the initial response from destination server is: 401 Unauthorized





19. When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

Answer: compared to the first GET request, new field called Authorization is included with the basic token and credentials message:



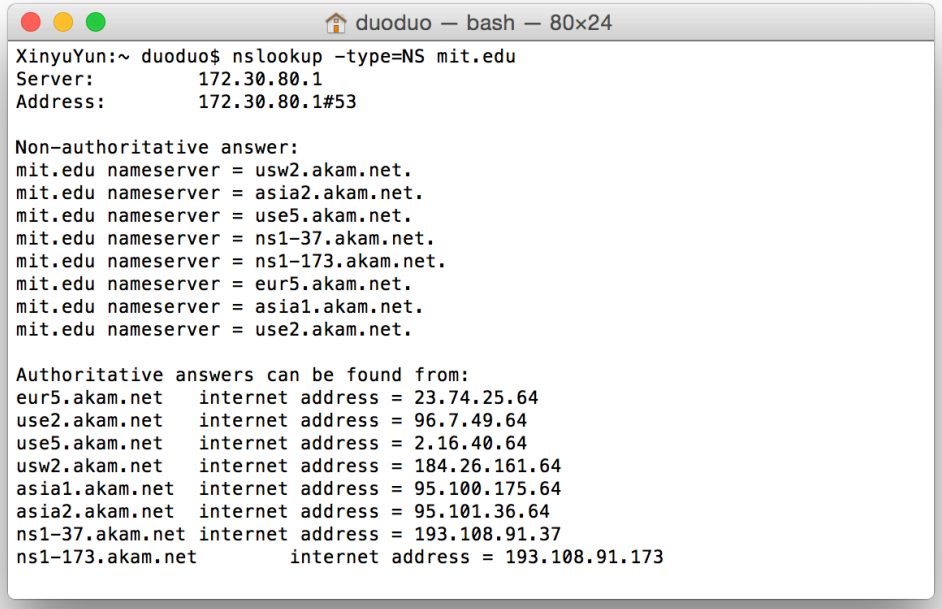
Wireshark Lab: DNS v6.01

1. Nslookup

- nslookup www.mit.edu

```
XinyuYun:~ duoduo$ nslookup
> www.mit.edu
Server:         172.30.80.1
Address:        172.30.80.1#53

Non-authoritative answer:
www.mit.edu     canonical name = www.mit.edu.edgekey.net.
www.mit.edu.edgekey.net canonical name = e9566.dscb.akamaiedge.net.
Name:   e9566.dscb.akamaiedge.net
Address: 184.86.32.128
```

- nslookup –type=NS mit.edu



- nslookup www.aiit.or.kr bitsy.mit.edu (**no servers could be reached!**)

1. Run nslookup to obtain the IP address of a Web server in Asia. What is the IP address of that server?

   Answer: I picked 'www.yhd.cn' to test, 2 mapping IP addresses are found as shown bellowed:

```
XinyuYun:~ duoduo$ nslookup www.yhd.cn
Server:         172.30.80.1
Address:        172.30.80.1#53

Non-authoritative answer:
Name:   www.yhd.cn
Address: 50.117.116.117
Name:   www.yhd.cn
Address: 50.117.120.253

XinyuYun:~ duoduo$
```

2. Run nslookup to determine the authoritative DNS servers for a university in Europe.

Answer：I select Humboldt-Universität in Berlin(hu-berlin.de) to test, the authoritative DNS servers listed as bellowed:

```
XinyuYun:~ duoduo$ nslookup -type=NS hu-berlin.de
Server:         192.168.199.1
Address:        192.168.199.1#53

Non-authoritative answer:
hu-berlin.de    nameserver = suncom.rz.hu-berlin.de.
hu-berlin.de    nameserver = ws-was.win-ip.dfn.de.
hu-berlin.de    nameserver = ns.tu-berlin.de.
hu-berlin.de    nameserver = hpcom.rz.hu-berlin.de.
hu-berlin.de    nameserver = buffy.cms.hu-berlin.de.

Authoritative answers can be found from:
ns.tu-berlin.de internet address = 130.149.7.7
buffy.cms.hu-berlin.de  internet address = 141.20.2.3
hpcom.rz.hu-berlin.de   internet address = 141.20.1.3
hpcom.rz.hu-berlin.de   has AAAA address 2001:638:813:1::3
suncom.rz.hu-berlin.de  internet address = 141.20.1.31
ws-was.win-ip.dfn.de    internet address = 193.174.75.110
ws-was.win-ip.dfn.de    internet address = 193.174.75.126
```

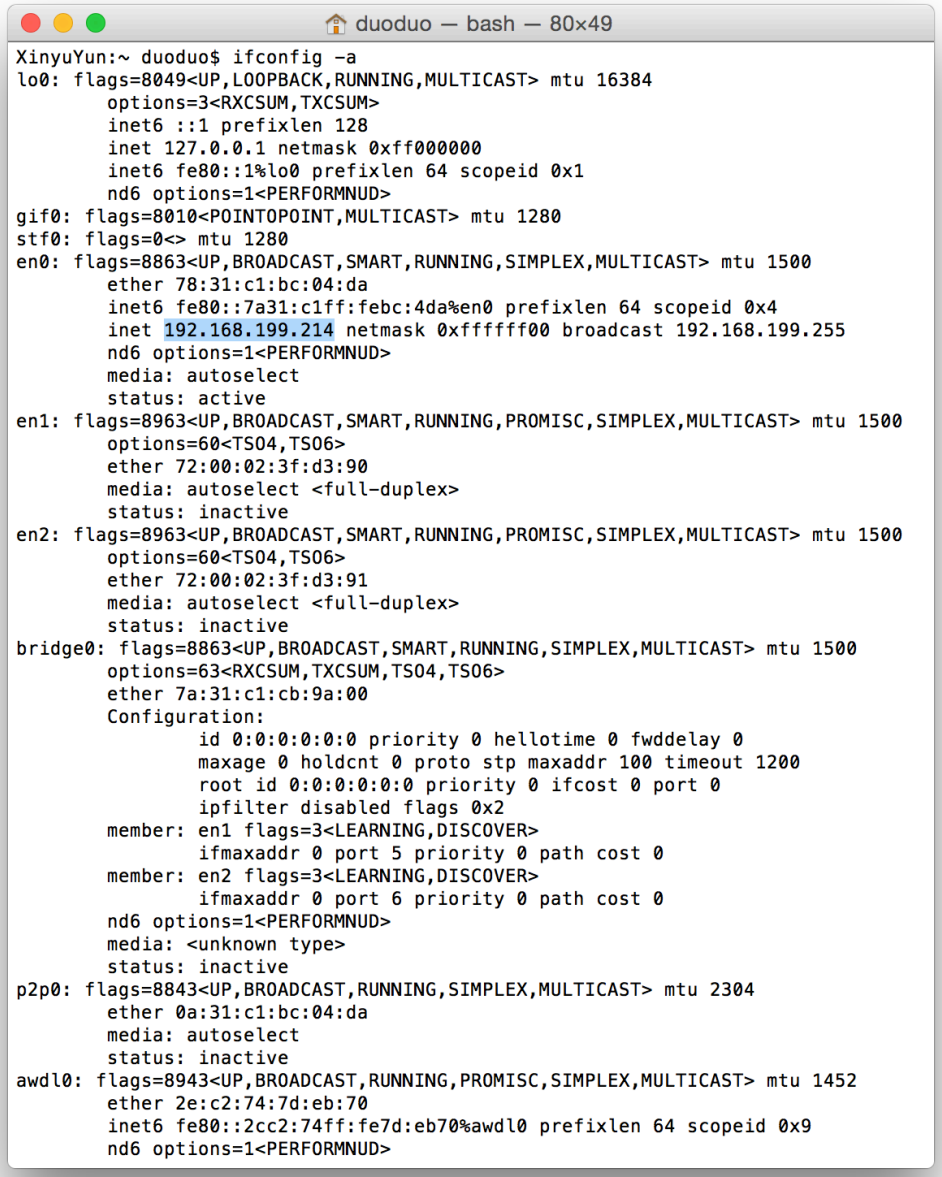
3. Run nslookup so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail. What is its IP address?

Answer: Here I selected the last DNS server ‘ws-was.win-ip.dfn.de’ from results in Question 2 as the specific DNS server to query the yahoo mail server ‘mail.yahoo.com’; the result bellowed shows the mail server’s IP is 188.125.80.138.

```
XinyuYun:~ duoduo$ nslookup mail.yahoo.com ws-was.win-ip.dfn.de
Server:         ws-was.win-ip.dfn.de
Address:        193.174.75.126#53

Non-authoritative answer:
mail.yahoo.com  canonical name = login.yahoo.com.
login.yahoo.com canonical name = fo-ds-ats.member.g02.yahoodns.net.
Name:   fo-ds-ats.member.g02.yahoodns.net
Address: 188.125.80.138
```

2. Ipconfig

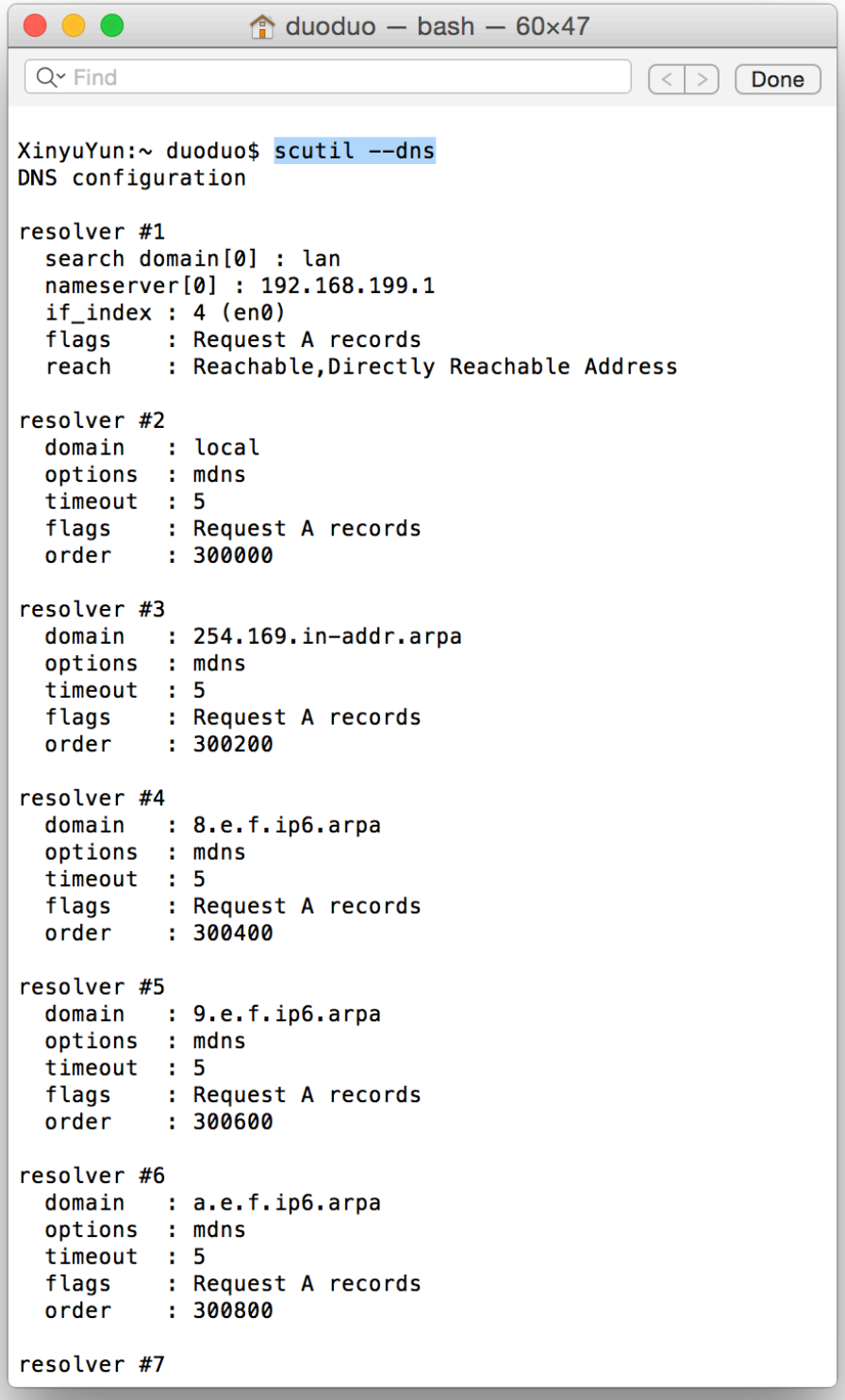
Answer: in Mac OS, I used ‘ifconfig’ to test:



In my laptop I checked the /etc/resolv.conf to find my DNS cache:

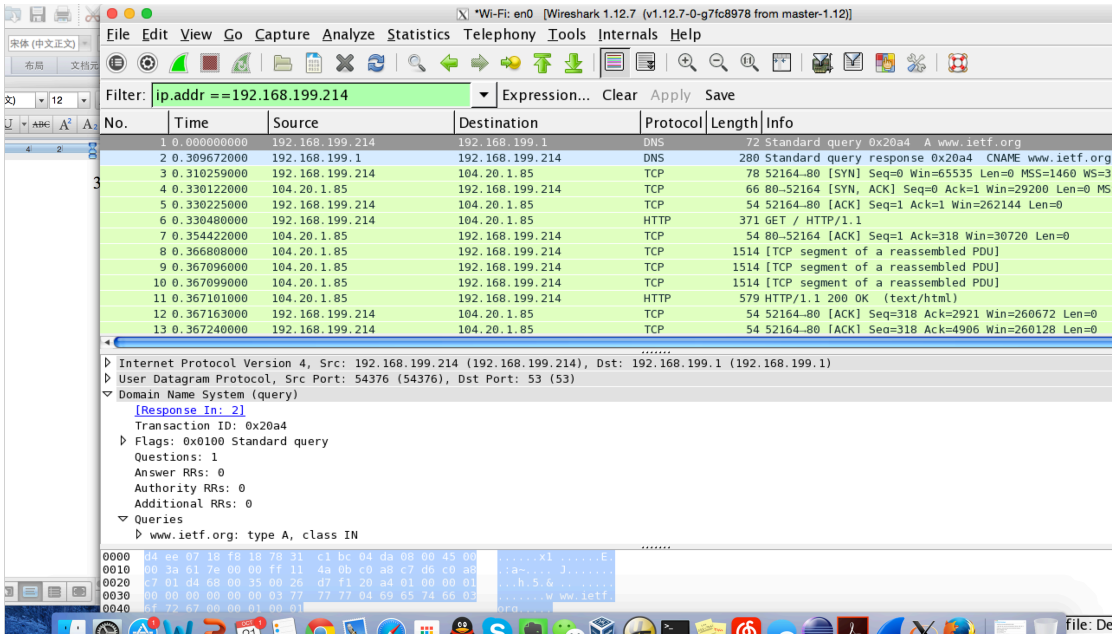
```
XinyuYun:~ duoduo$ cat /etc/resolv.conf
#
# Mac OS X Notice
#
# This file is not used by the host name and address resolution
# or the DNS query routing mechanisms used by most processes on
# this Mac OS X system.
#
# This file is automatically generated.
#
domain lan
nameserver 192.168.199.1
XinyuYun:~ duoduo$
```

and scutil –dns to display all DNS configurations:


duoduo — bash — 60×47
Find
Done
XinyuYun:~ duoduo$ scutil --dns
DNS configuration
resolver #1
search domain[0] : lan
nameserver[0] : 192.168.199.1
if_index : 4 (en0)
flags : Request A records
reach : Reachable,Directly Reachable Address
resolver #2
domain : local
options : mdns
timeout : 5
flags : Request A records
order : 300000
resolver #3
domain : 254.169.in-addr.arpa
options : mdns
timeout : 5
flags : Request A records
order : 300200
resolver #4
domain : 8.e.f.ip6.arpa
options : mdns
timeout : 5
flags : Request A records
order : 300400
resolver #5
domain : 9.e.f.ip6.arpa
options : mdns
timeout : 5
flags : Request A records
order : 300600
resolver #6
domain : a.e.f.ip6.arpa
options : mdns
timeout : 5
flags : Request A records
order : 300800
resolver #7

In Mac OS 10.10, this command is used to flush DNS cache: sudo killall -HUP mDNSResponder

3. Tracing DNS with Wireshark

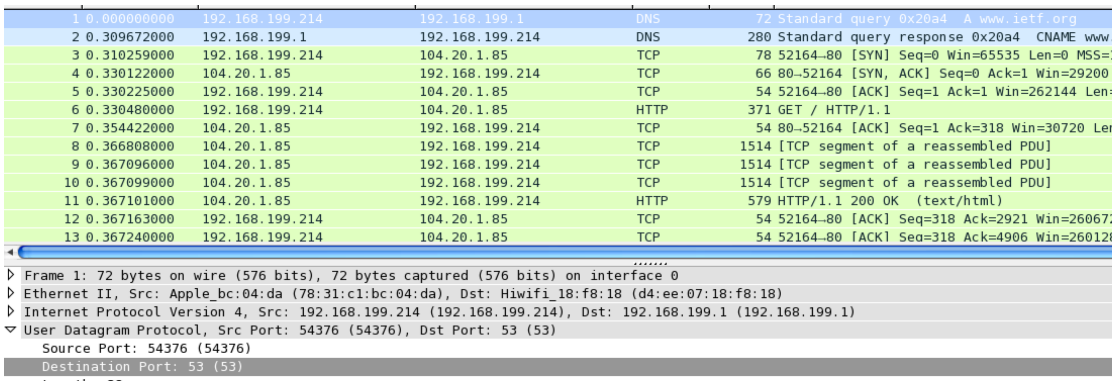


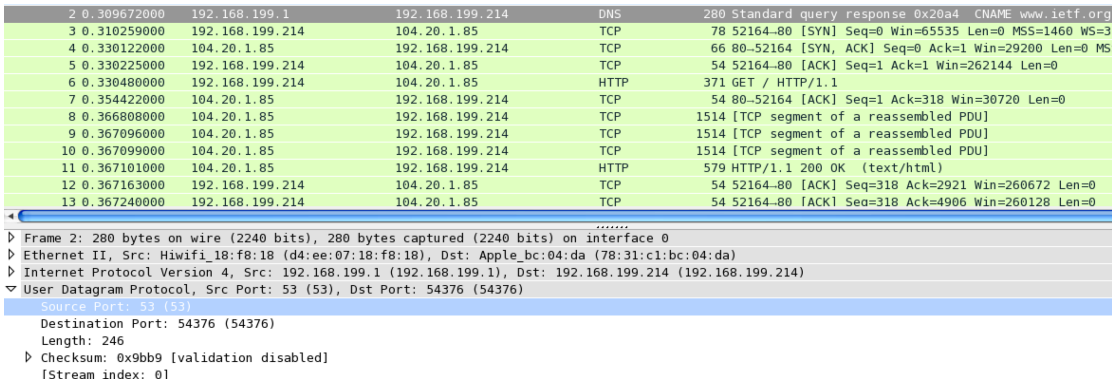
4. Locate the DNS query and response messages. Are then sent over UDP or TCP?

Answer： after the DNS query response, the messages then sent over UDP.

5. What is the destination port for the DNS query message? What is the source port of DNS response message?

Answer: the destination port for DNS query message is 53. So is the same port for DNS response message, 53.





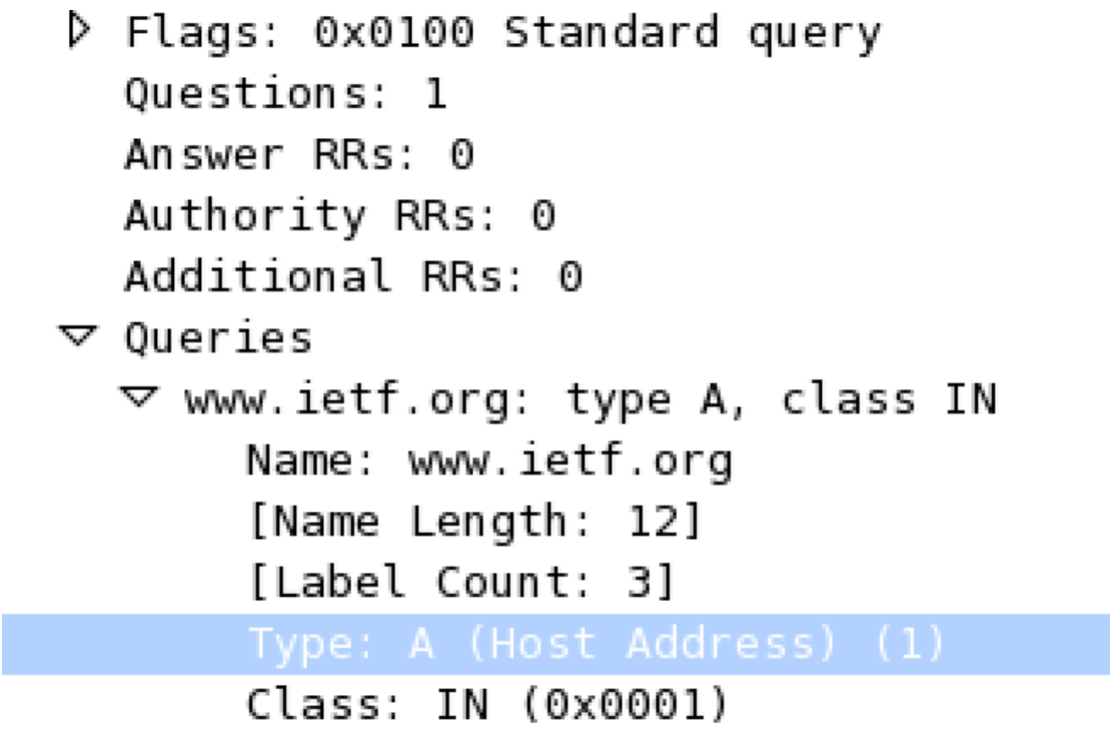
6. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?

Answer: the destination IP is 192.168.199.1 in DNS query, which is as same as the IP in DNS resolver file saved in my Mac OS:

```
XinyuYun:~ duoduo$ cat /etc/resolv.conf
#
# Mac OS X Notice
#
# This file is not used by the host name and address resolut
ion
# or the DNS query routing mechanisms used by most processes
 on
# this Mac OS X system.
#
# This file is automatically generated.
#
domain lan
nameserver 192.168.199.1
```

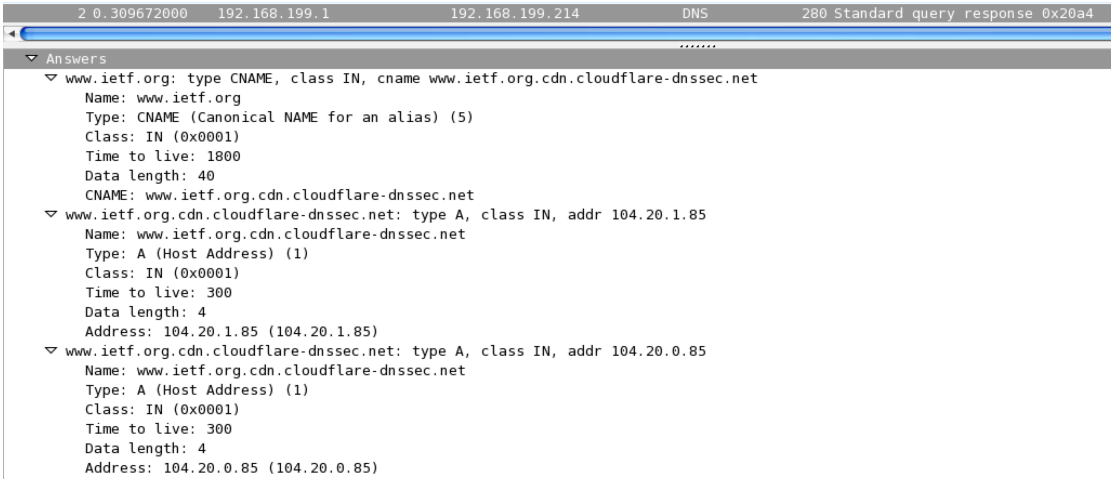
7. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

Answer: in query message, the type is A(host address)(1) meaning it will return a 32-bit IPv4 address, and the query does not contain answers.



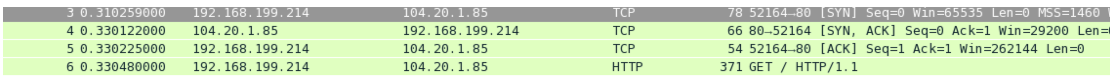
8. Examine the DNS response message. How many "answers" are provided? What does each of these answers contain?

Answer: In the response, 3 answers are provided, each of which contains Name, Type, Class, TTL, Data Length, and Address.



9. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?

Answer: the destination IP (104.20.0.85) in TCP [SYN] packet does correspond to the address in DNS query response answers.



10. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

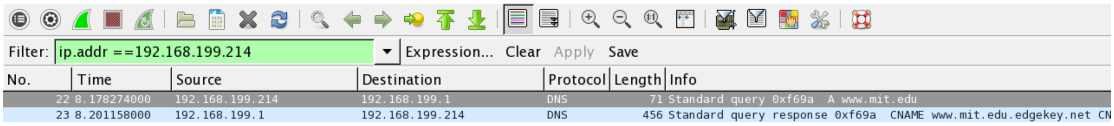
Answer: As checked the packets list, the image requests are all heading to the server 104.20.1.85, so there is no new DNS queries for GET these images.

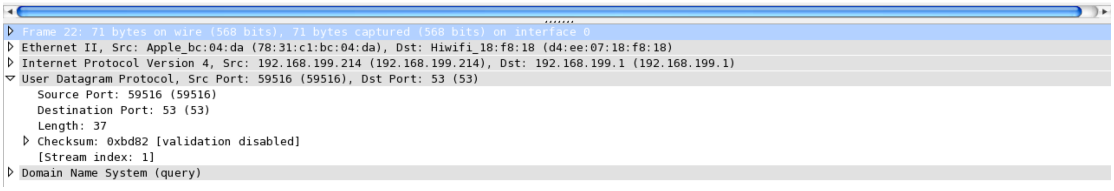
50 0.440772000	192.168.199.214	104.20.1.85	HTTP	458 GET /images/ietflogotrans.gif HTTP/1.1
51 0.440918000	192.168.199.214	104.20.1.85	HTTP	455 GET /images/chat-trans.png HTTP/1.1
52 0.441033000	192.168.199.214	104.20.1.85	HTTP	457 GET /images/openstand-md.png HTTP/1.1
53 0.441119000	192.168.199.214	104.20.1.85	HTTP	465 GET /meeting/94/images/yokohama2.jpg HTTP/1.1
54 0.441179000	192.168.199.214	104.20.1.85	HTTP	454 GET /images/isoc_logo.gif HTTP/1.1
55 0.441546000	192.168.199.214	104.20.1.85	HTTP	453 GET /images/ams_logo.png HTTP/1.1
80 0.468974000	104.20.1.85	192.168.199.214	HTTP	136 HTTP/1.1 200 OK (GIF89a) (GIF89a) (image/gif)
87 0.470112000	104.20.1.85	192.168.199.214	HTTP	914 HTTP/1.1 200 OK (PNG)
100 0.470768000	104.20.1.85	192.168.199.214	HTTP	1123 HTTP/1.1 200 OK (PNG)
130 0.490752000	104.20.1.85	192.168.199.214	HTTP	752 HTTP/1.1 200 OK (GIF89a) (GIF89a) (image/gif)
148 0.496141000	104.20.1.85	192.168.199.214	HTTP	1456 HTTP/1.1 200 OK (JPEG JFIF image)

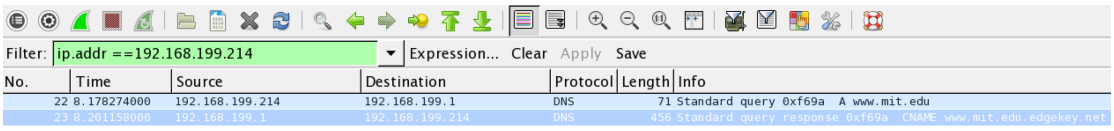
- **Start packet capture.**
- **Do an nslookup on www.mit.edu**
- **Stop packet capture.**

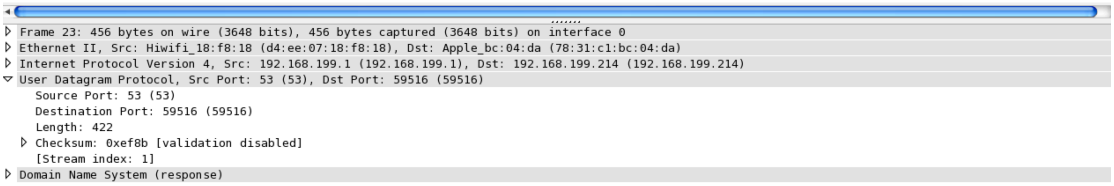
11. What is the destination port for the DNS query message? What is the source port of DNS response message?

Answer: destination port for DNS query message is 53 and so is the source ort of response message, being the port 53.







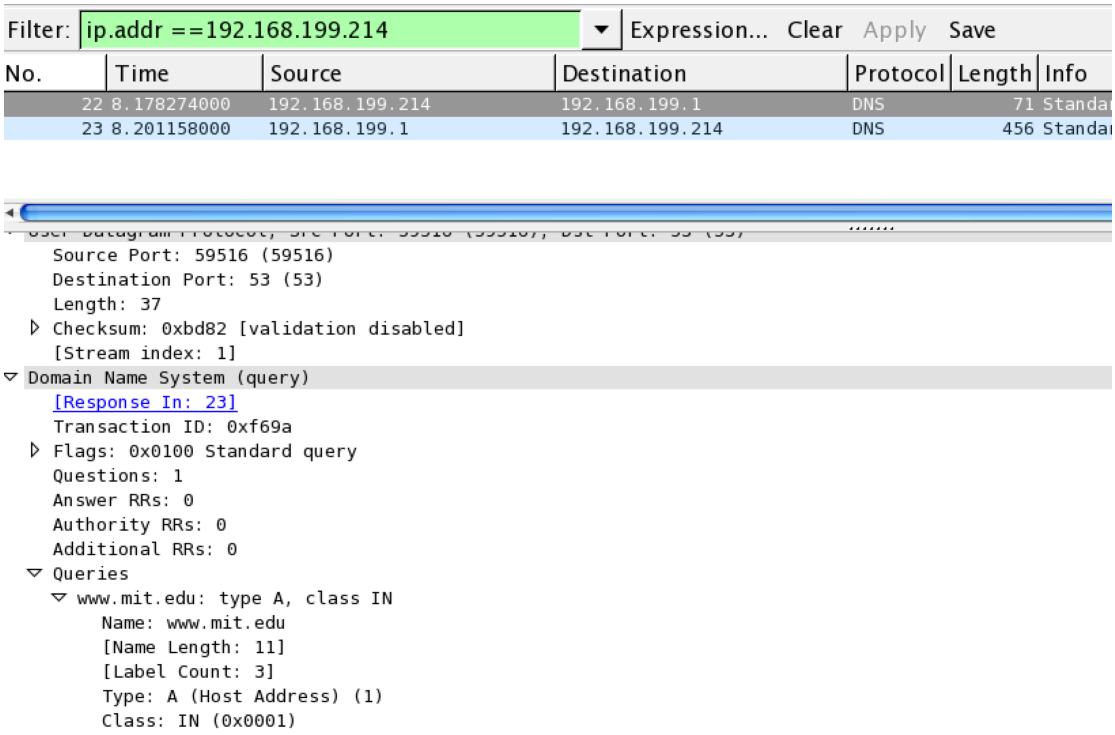


12. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

Answer: the destination IP is 192.168.199.1, which is the default DNS server.

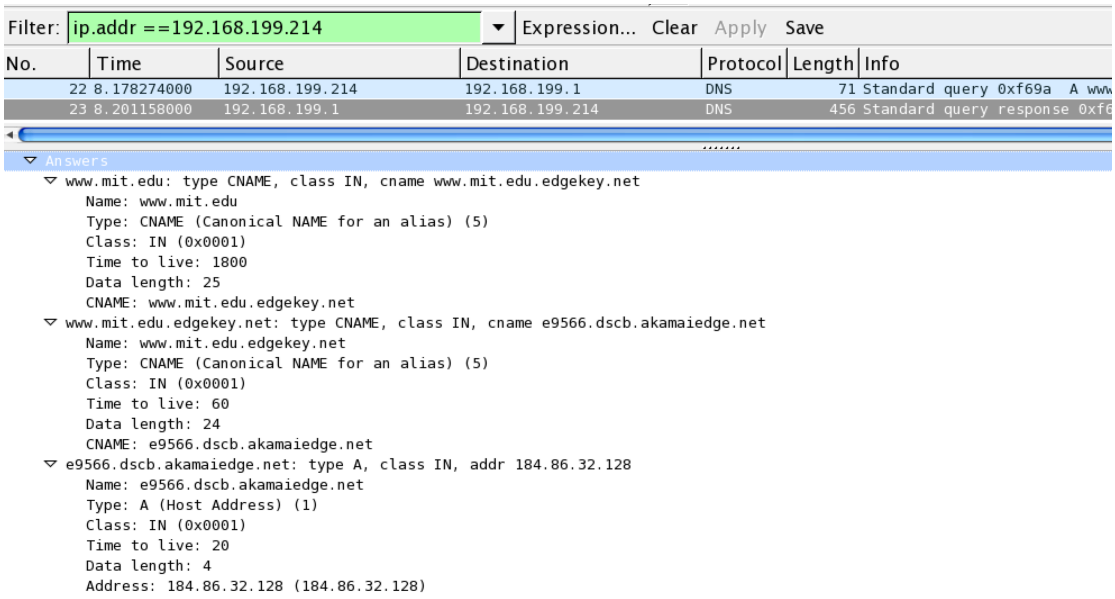
13. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

Answer: the type of query message is also ‘A’, and the query does not contain any answers.



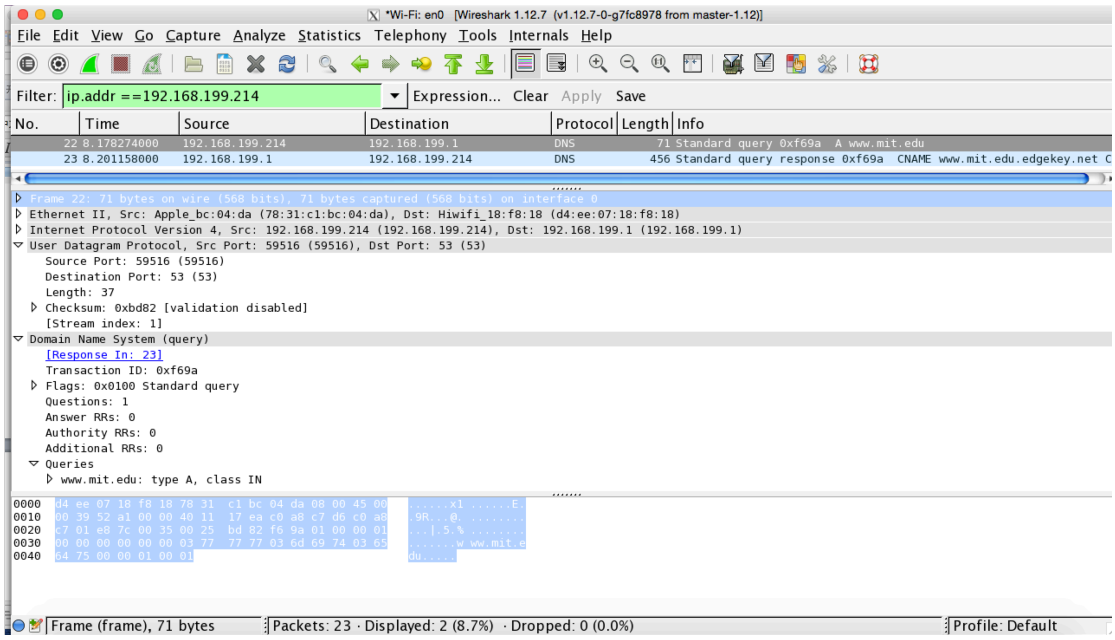
14. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?

Answer: There are 3 answers in response message; each of those contains name, type, class, TTL, data length, and address.

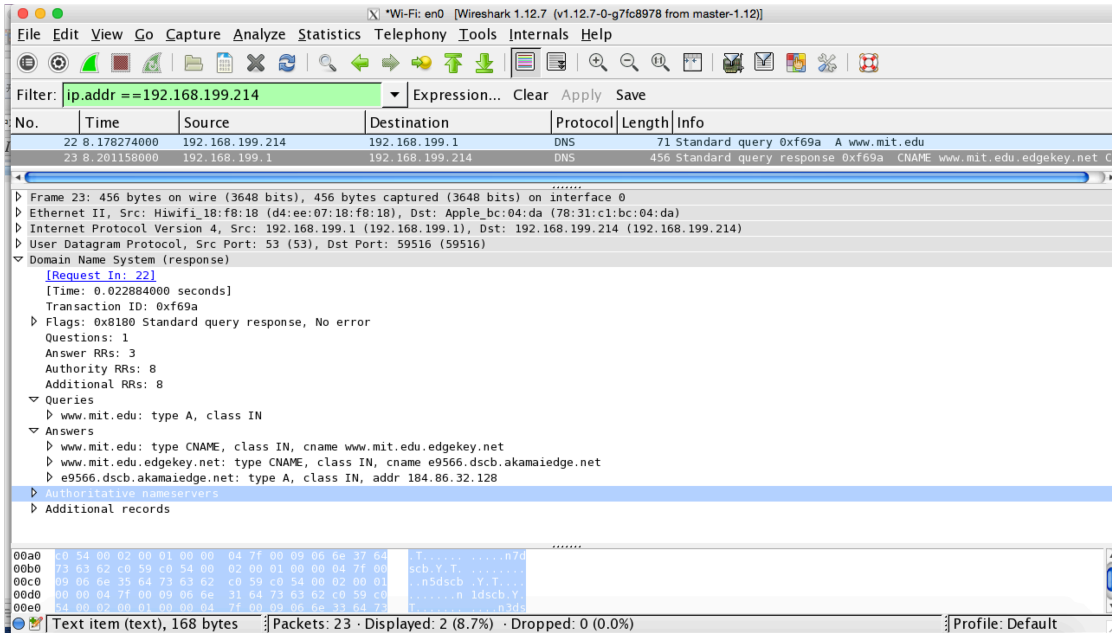


15. Provide a screenshot.

DNS query message:



DNS response message:



- **nslookup –type=NS mit.edu**

16. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?

Answer: the destination IP is also 192.168.199.1, which is my local default DNS server.

17. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

Answer: the type of query message is 'Type: NS (authoritative Name Server) (2)', and the query does not contain any answers either.

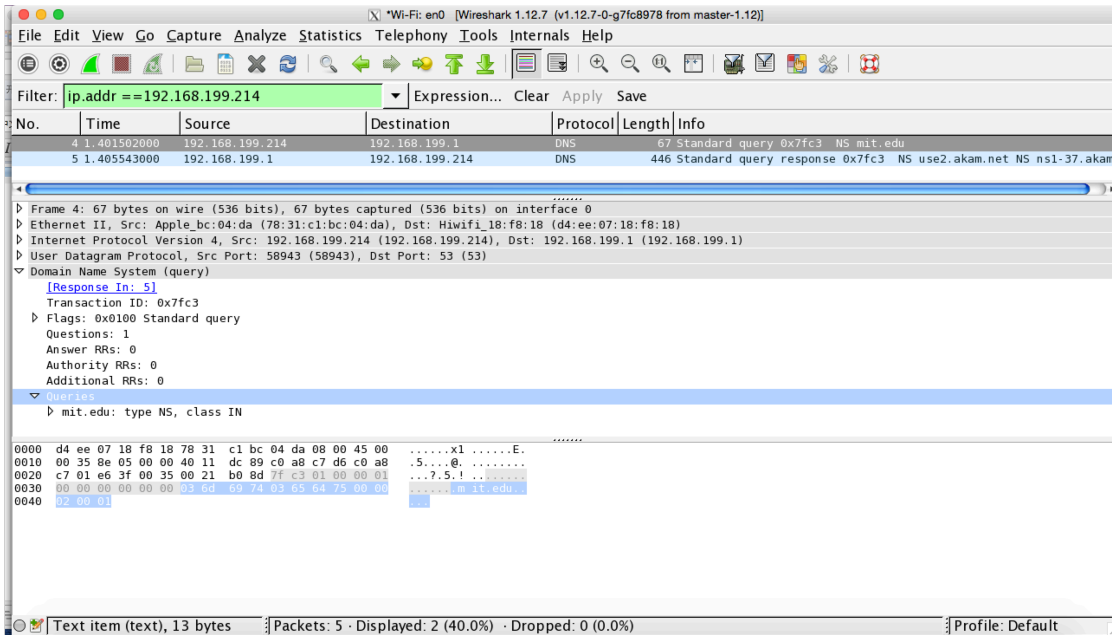
18. Examine the DNS response message. What MIT name servers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?

Answer: 8 name servers are provided as bellowed, and in the additional records section their IP addresses are provided.

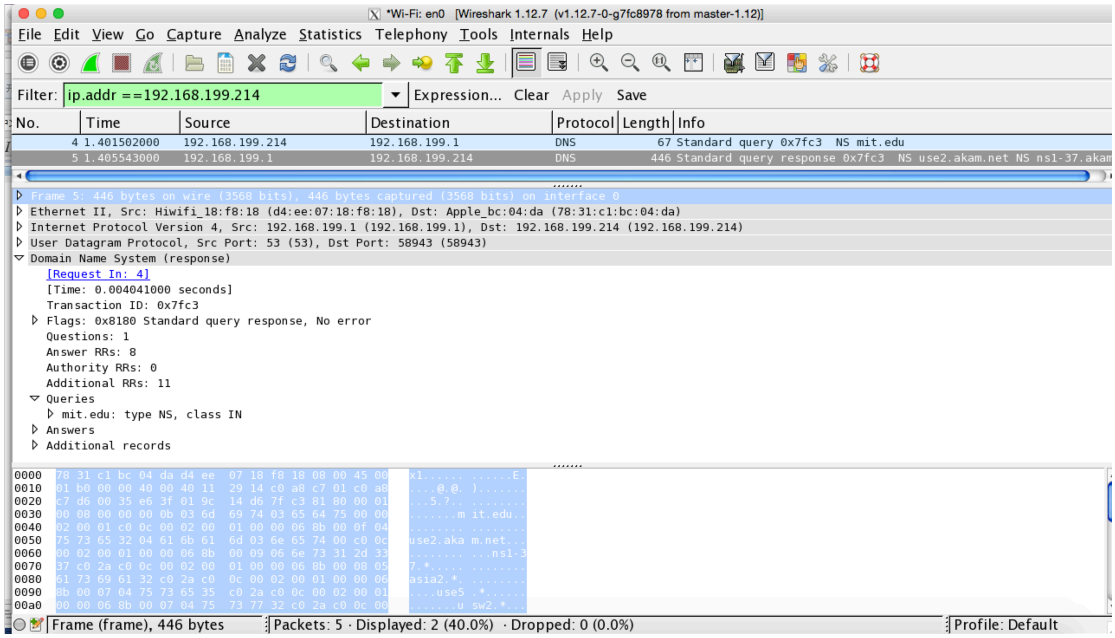
```
▽ Answers
  ▷ mit.edu: type NS, class IN, ns use2.akam.net
  ▷ mit.edu: type NS, class IN, ns ns1-37.akam.net
  ▷ mit.edu: type NS, class IN, ns asia2.akam.net
  ▷ mit.edu: type NS, class IN, ns use5.akam.net
  ▷ mit.edu: type NS, class IN, ns usw2.akam.net
  ▷ mit.edu: type NS, class IN, ns asia1.akam.net
  ▷ mit.edu: type NS, class IN, ns ns1-173.akam.net
  ▷ mit.edu: type NS, class IN, ns eur5.akam.net
▽ Additional records
  ▷ eur5.akam.net: type A, class IN, addr 23.74.25.64
  ▷ use2.akam.net: type A, class IN, addr 96.7.49.64
  ▷ use5.akam.net: type A, class IN, addr 2.16.40.64
  ▷ use5.akam.net: type AAAA, class IN, addr 2600:1401:1::40
  ▷ usw2.akam.net: type A, class IN, addr 184.26.161.64
  ▷ asia1.akam.net: type A, class IN, addr 95.100.175.64
  ▷ asia2.akam.net: type A, class IN, addr 95.101.36.64
  ▷ ns1-37.akam.net: type A, class IN, addr 193.108.91.37
  ▷ ns1-37.akam.net: type AAAA, class IN, addr 2600:1401:2::25
  ▷ ns1-173.akam.net: type A, class IN, addr 193.108.91.173
  ▷ ns1-173.akam.net: type AAAA, class IN, addr 2600:1401:2::ad
```

19. Provide a screenshot.

DNS query:



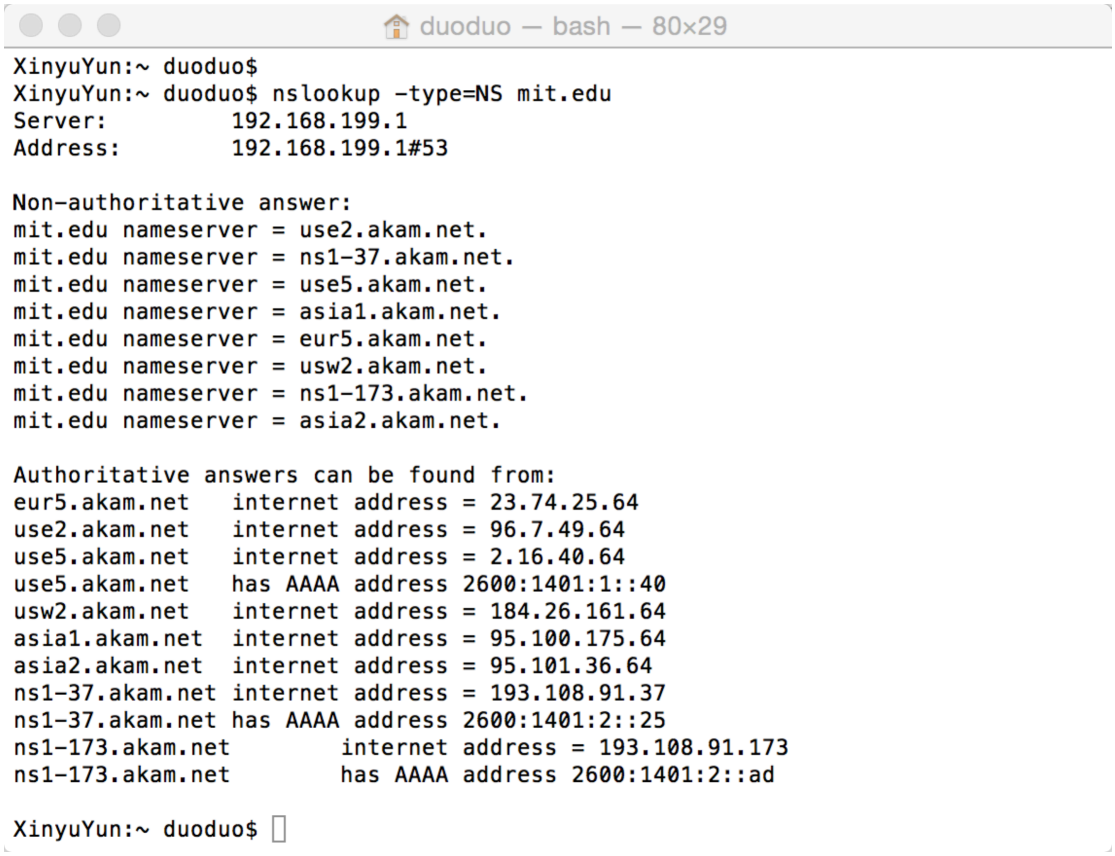
DNS response:



- **nslookup www.aiit.or.kr bitsy.mit.edu**

  Since neither the host ‘www.aiit.or.kr’ nor the name server ‘bitsy’ is available in my apartment Wifi with the Macbook.

  Firstly as shown the name servers I found in previous question, the bitsy server is not included:



Secondly, even I choose the found DNS servers to resolve 'www.ait.or.kr', the following message will returned to refuse the request.

```
XinyuYun:~ duoduo$ nslookup www.ait.or eur5.akam.net
Server:         eur5.akam.net
Address:        23.74.25.64#53

** server can't find www.ait.or.kr: REFUSED
```

As communicated with prof. and TA, I will use 'www.ait.or.kr' and another available server 'ws-was.win-ip.dfn.de' which I used in previous question to resolve the yahoo email server.

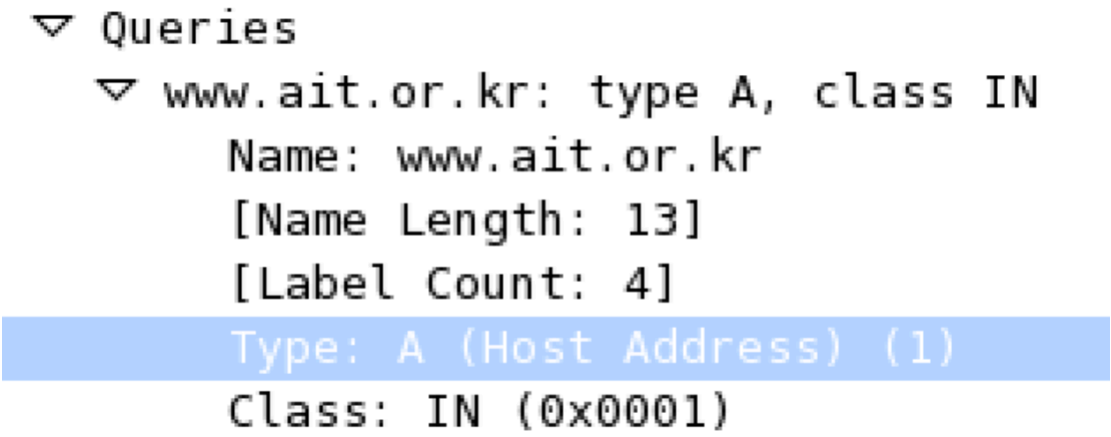
```
XinyuYun:~ duoduo$ nslookup www.ait.or.kr ws-was.win-ip.dfn.de
Server:         ws-was.win-ip.dfn.de
Address:        193.174.75.110#53

Non-authoritative answer:
Name:   www.ait.or.kr
Address: 116.127.123.32

XinyuYun:~ duoduo$
```

20. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?

Answer: the IP where the query sent is 193.174.75.110, which is not my default local DNS server (192.168.199.1). The new IP corresponds to the DNS server I use in the nslookup command.

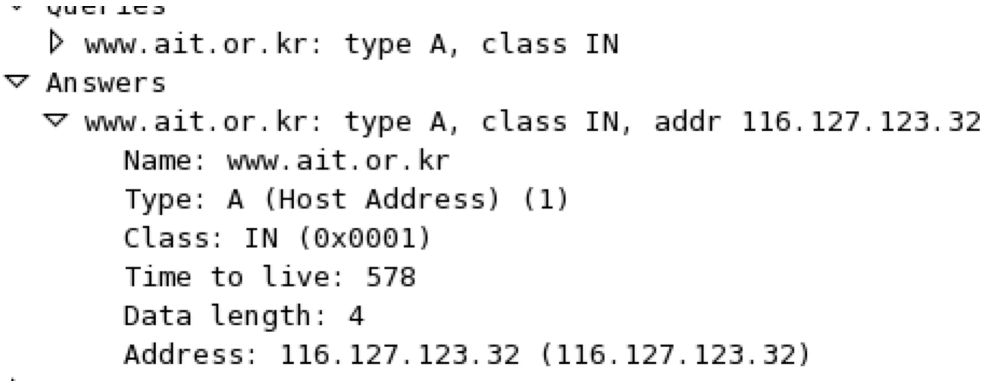
21. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?

Answer: the type is A and the query does not contain any answers.



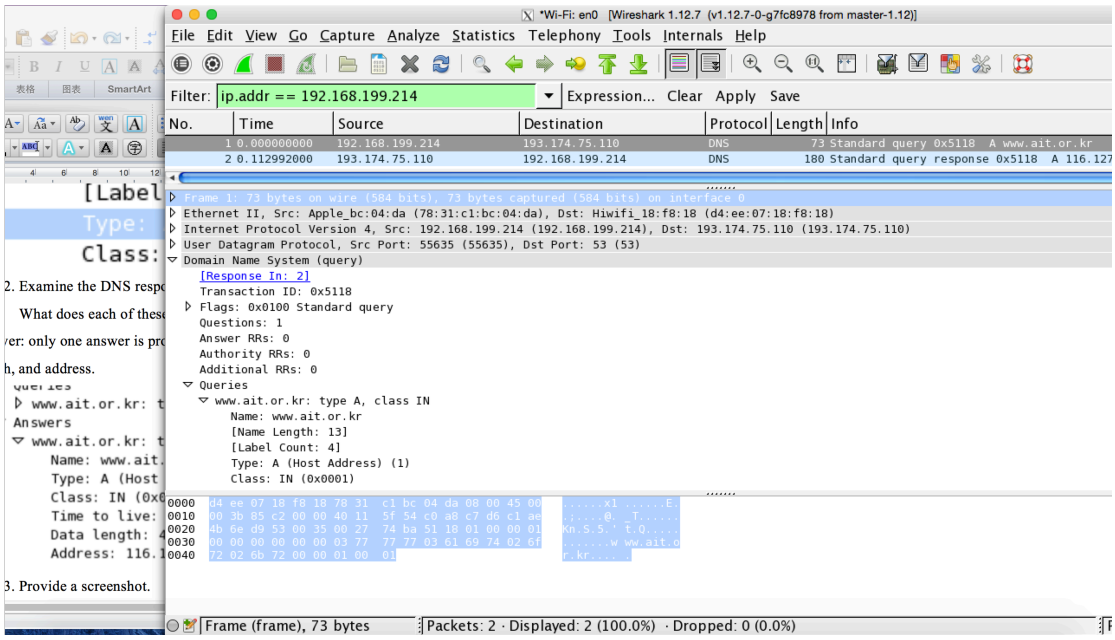
22. Examine the DNS response message. How many "answers" are provided? What does each of these answers contain?

Answer: only one answer is provided in which there are name, type, class, TTL, data length, and address.



23. Provide a screenshot.

DNS query:



DNS response: