Data Communications

Assignment Element 1 & 2

Student ID no. 21359035

Introduction

Basically we have a network which is devices such as computers connected to each other. One of these networks is a wide area network called the internet. Devices such as computers communicate with each other when connected. Since it is interesting how networks work Iwill go into further detail using terms or words to describe how they work.

There are devices in a network called servers which connect computers to communicate with each other for example sending emails. A mail server is used for this common method of communication.

Devices ae connected to each other to send data to each other which is how the internet and email works. There are terms such as protocols and packets which describe how data is sent between devices in a network. A protocol is the way a message is sent between devices on a network such as email messages. It is the type of message being sent between computers (such as the format of the message), the way it is sent and received and the response to the message when it is received. In a network such as the internet the protocols are the machines which are the hardware and software, communicating with each other. Protocols are the way devices communicate with each other when connected.

Examples of protocols being used between connected devices are TCP (transmission control protocol), HTTP (hypertext transfer protocol) and SMTP (simple mail transfer protocol). The transmission control protocol and hyper text transfer protocol are used by a computer connected to the internet. A user connected to the internet goes to a website by typing in the web address in the browser software and the HTTP protocol is the way the website is sent to the user by the server. The protocol is the request of the website sent by the user to server and the server responding to the request by sending a live copy of the website to the user.

The simple mail transfer protocol is a simple protocol used for sending emails between computers on a network. The protocol is the email being sent from a computer to the computer it is connected to via a mail server. Once the connected computer receives the email action is taken such as a reply to the email message and then the reply is sent via the mail server to the sender. The SMTP can also be a response to an email message sent from a computer as a receipt telling the sender if the message was received or not. The Transmission Control protocol is the transmission of messages sent between devices on a network. It is the way emails are transmitted with the SMTP protocol while being sent and responded to. When a user on the internet requests a webpage with images the TCP is the transmission of the webpage with the images sent as a reply from the server to the user.

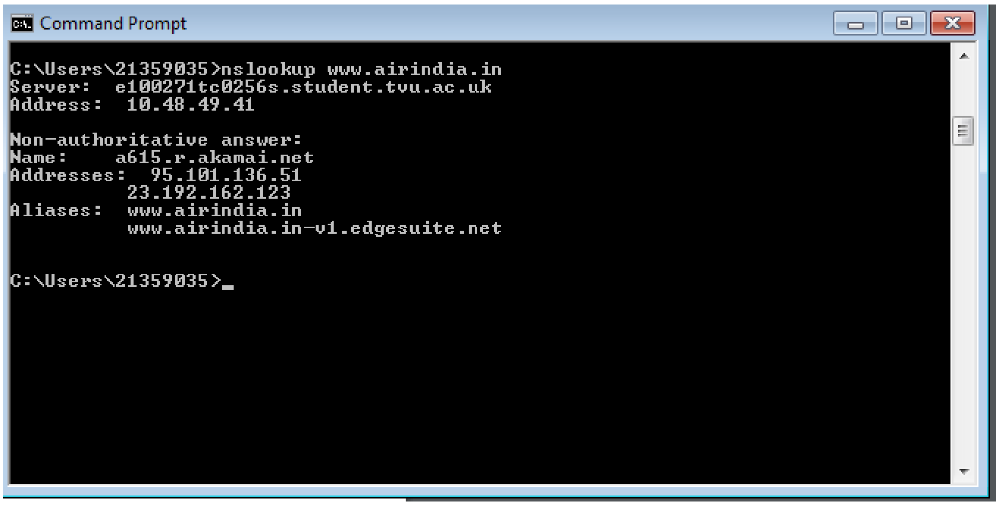
Another type of protocol is DNS which stands for Domain Name System. This is about the address of a webpage requestsed by a user connected to the internet. When the user requests a webpage a database known as a DNS database is connected to to get the IP address of the web address stored in the database while the rquested webpage is being sent to the user. Suppose the user requests the website www.afternerd.com by typing it in their browser. The address www.afternerd.com is the domain name and afternedr.com is the subdomain. The web address has an IP address say like 123.234.34.55 stored in the DNS database which is stored in a server known as a DNS server. Infact every device connected to a network is identified by an IP address so that they can be accessed.

The other term packets or data packets are the pieces of data being sent between devices. Data being sent between computers on a network are broken down into chunks known as packets and the purpose of the first part of this assignment is to show data is being sent and received by devices connected to each other. To do this we first need a software on the computer called Wireshark which detects data packets while the computer is connected to a network such as the internet. The software captures data packets being sent to and received by the computer it is installed on. It is basically a packet sniffing tool or “packet sniffer” that ‘sniffs’ chunks of data being sent and received by the computer. I will be using this software in this assignment to show the packets captured by it and mention what the details of the captured packet show.

You can check what network or devices your computer is connected to and details of these connections such as servers by typing commands on the Command Prompt. A user can type commands as queries to get details of servers including their name and IP addresses from DNS servers. There are tools you can use in the prompt by typing the command in all one word and in all one word with options.

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

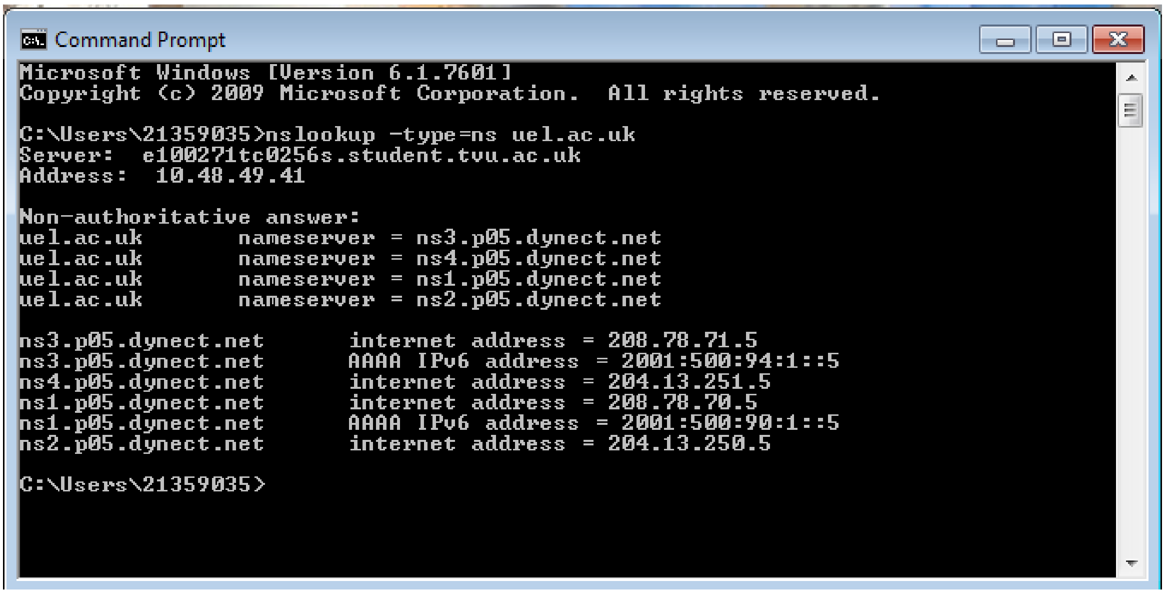
So here we start using the nslookup tool in Windows in the Command Prompt to get the IP address of a server holding a web page in another location. This is by typing a command known as nslookup followed by the web page address on the command line shown in the screenshot below.



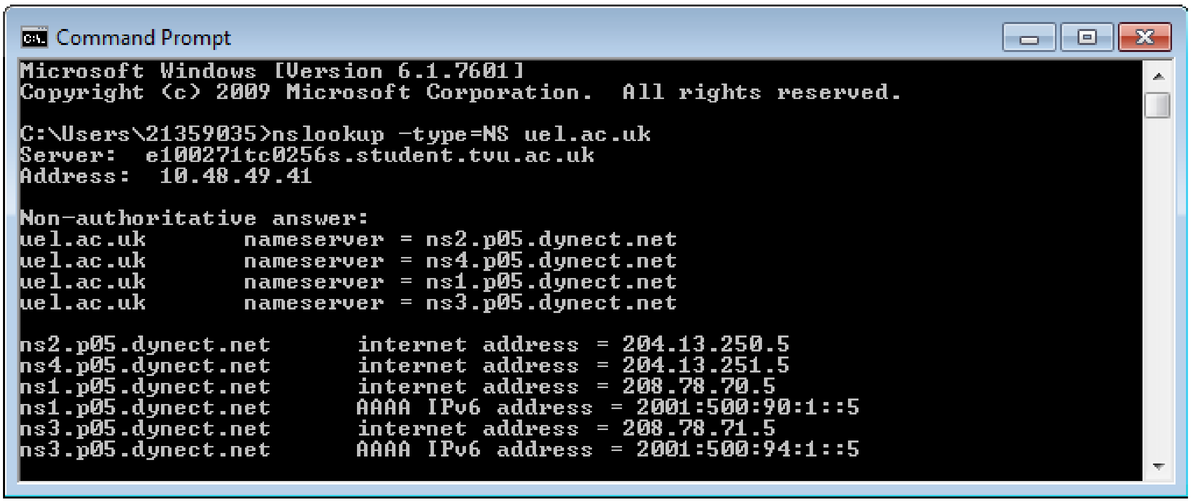
```
C:\Users\21359035>nslookup www.airindia.in
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
Name:    a615.r.akamai.net
Addresses:  95.101.136.51
          23.192.162.123
Aliases:  www.airindia.in
          www.airindia.in-v1.edgesuite.net


C:\Users\21359035>_
```

The screenshot above shows the IP address of a web server obtained in Asia is 95.101.136.51(which is usually the first one even though I have got the addresses of two web servers).

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



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\21359035>nslookup -type=ns uel.ac.uk
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
uel.ac.uk       nameserver = ns3.p05.dynect.net
uel.ac.uk       nameserver = ns4.p05.dynect.net
uel.ac.uk       nameserver = ns1.p05.dynect.net
uel.ac.uk       nameserver = ns2.p05.dynect.net

ns3.p05.dynect.net      internet address = 208.78.71.5
ns3.p05.dynect.net      AAAA IPv6 address = 2001:500:94:1::5
ns4.p05.dynect.net      internet address = 204.13.251.5
ns1.p05.dynect.net      internet address = 208.78.70.5
ns1.p05.dynect.net      AAAA IPv6 address = 2001:500:90:1::5
ns2.p05.dynect.net      internet address = 204.13.250.5

C:\Users\21359035>
```

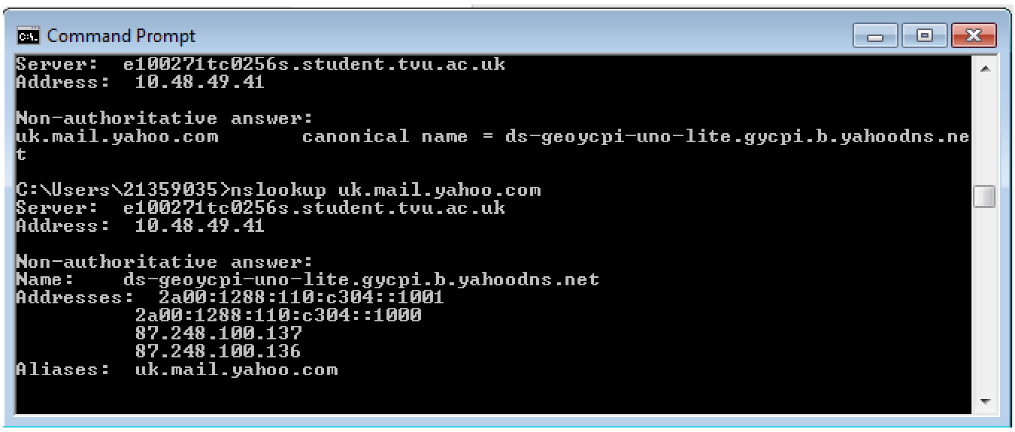


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\21359035>nslookup -type=NS uel.ac.uk
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
uel.ac.uk       nameserver = ns2.p05.dynect.net
uel.ac.uk       nameserver = ns4.p05.dynect.net
uel.ac.uk       nameserver = ns1.p05.dynect.net
uel.ac.uk       nameserver = ns3.p05.dynect.net

ns2.p05.dynect.net      internet address = 204.13.250.5
ns4.p05.dynect.net      internet address = 204.13.251.5
ns1.p05.dynect.net      internet address = 208.78.70.5
ns1.p05.dynect.net      AAAA IPv6 address = 2001:500:90:1::5
ns3.p05.dynect.net      internet address = 208.78.71.5
ns3.p05.dynect.net      AAAA IPv6 address = 2001:500:94:1::5
```

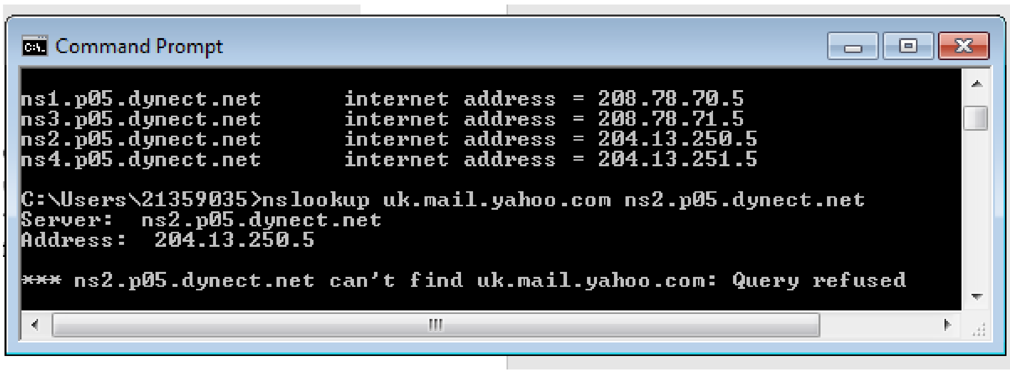
The command used here was nslookup with the option "–type=NS" and the domain uel.ac.uk(the address without using www.).

The nslookup command sends a query to the local DNS server for a record of authoritative DNS servers for the hosts at the uel site. The list of host names are provided here of the authoritative DNS servers with their IP addresses. The record is indicated as Non-authoritative by nslookup since the record has come from the cache of a server which is not an authoritative uel DNS server. The cache is a table of DNS records, held by servers and clients, of servers with their names and IP address which have recently been received.

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?

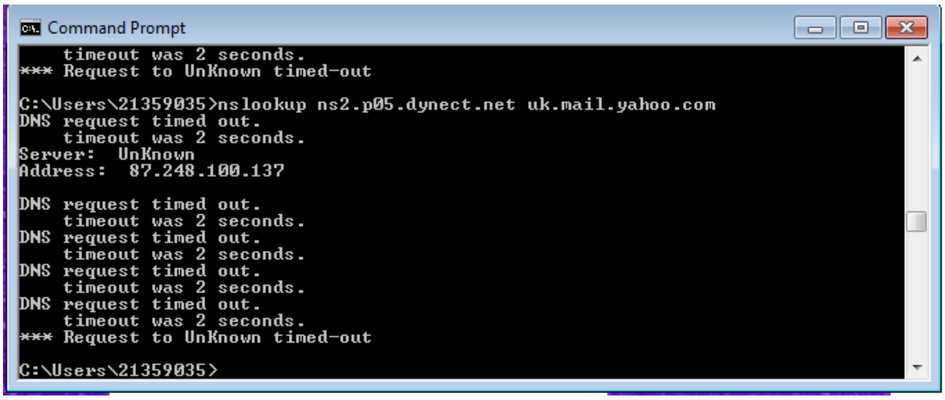


```
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
uk.mail.yahoo.com       canonical name = ds-geoycpi-uno-lite.gycpi.b.yahoodns.ne
t

C:\Users\21359035>nslookup uk.mail.yahoo.com
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
Name:    ds-geoycpi-uno-lite.gycpi.b.yahoodns.net
Addresses:  2a00:1288:110:c304::1001
          2a00:1288:110:c304::1000
          87.248.100.137
          87.248.100.136
Aliases:  uk.mail.yahoo.com
```



```
ns1.p05.dynect.net      internet address = 208.78.70.5
ns3.p05.dynect.net      internet address = 208.78.71.5
ns2.p05.dynect.net      internet address = 204.13.250.5
ns4.p05.dynect.net      internet address = 204.13.251.5

C:\Users\21359035>nslookup uk.mail.yahoo.com ns2.p05.dynect.net
Server:  ns2.p05.dynect.net
Address:  204.13.250.5

*** ns2.p05.dynect.net can't find uk.mail.yahoo.com: Query refused
```

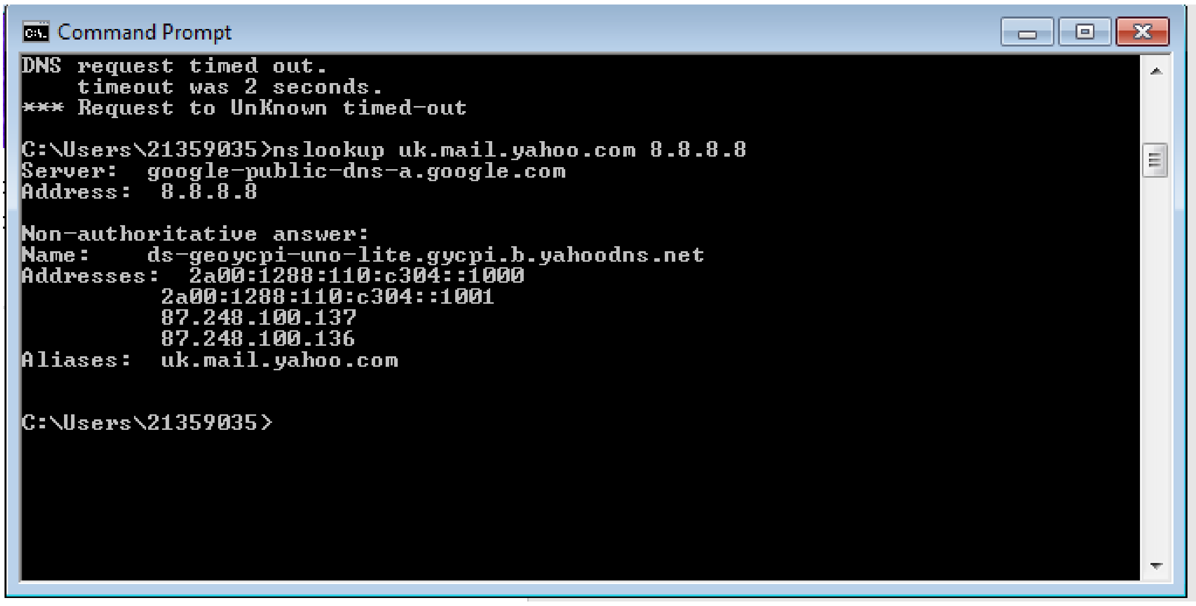
Here the command nslookup uk.mail.yahoo.com ns2.p05.dynect.net was used to query one of the DNS servers from question 2 for the mail servers for Yahoo! Mail but I got the result saying ns2.p05.dynect.net could not find the mail server for yahoo mail and the query was refused. This must mean the query would not return a result when sent to the uel DNS server ns2.p05.dynect.net if the server cannot find the Yahpo mail server. The server ns2.p05.dynect.net would not return the IP address of the host uk.mail.yahoo.com if it could not find this host.



```
    timeout was 2 seconds.
*** Request to UnKnown timed-out

C:\Users\21359035>nslookup ns2.p05.dynect.net uk.mail.yahoo.com
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  87.248.100.137

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out

C:\Users\21359035>
```

I then did it the other way round with ns2.p05.dynect.net as the host sending the query to uk.mail.yahoo.com but the result was the DNS request being timed out and the address of an unknown server rather than the Yahoo mail server.

To fix this problem I used the google public server 8.8.8.8 with the nslookup command and uk.mail.yahoo.com as nslookup uk.mail.yahoo.com 8.8.8.8 shown below.

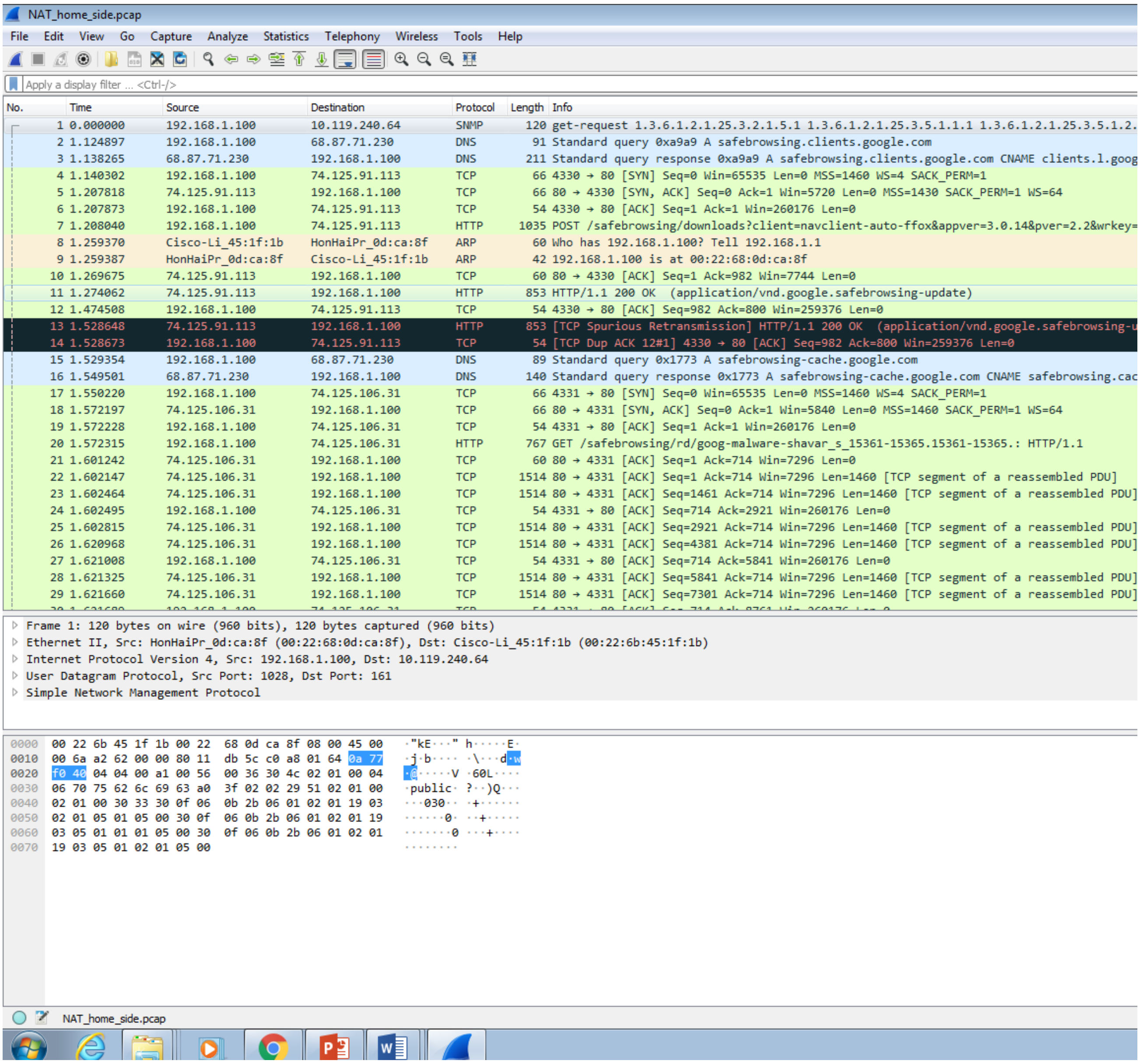


```
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out

C:\Users\21359035>nslookup uk.mail.yahoo.com 8.8.8.8
Server:  google-public-dns-a.google.com
Address:  8.8.8.8

Non-authoritative answer:
Name:    ds-geoycpi-uno-lite.gycpi.b.yahoodns.net
Addresses:  2a00:1288:110:c304::1000
          2a00:1288:110:c304::1001
          87.248.100.137
          87.248.100.136
Aliases:  uk.mail.yahoo.com


C:\Users\21359035>
```

The IP address is 87.248.100.137(which is usually the first one).

Back to the software on the computer Wireshark, that is used to capture or ‘sniff’data packets. We start here looking at lists of packets captured by the software while the computer was connected to the internet and a website on the internet was accessed. Data packets of the website accessed were captured as messages sent and received by the computer.

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



NAT_home_side.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

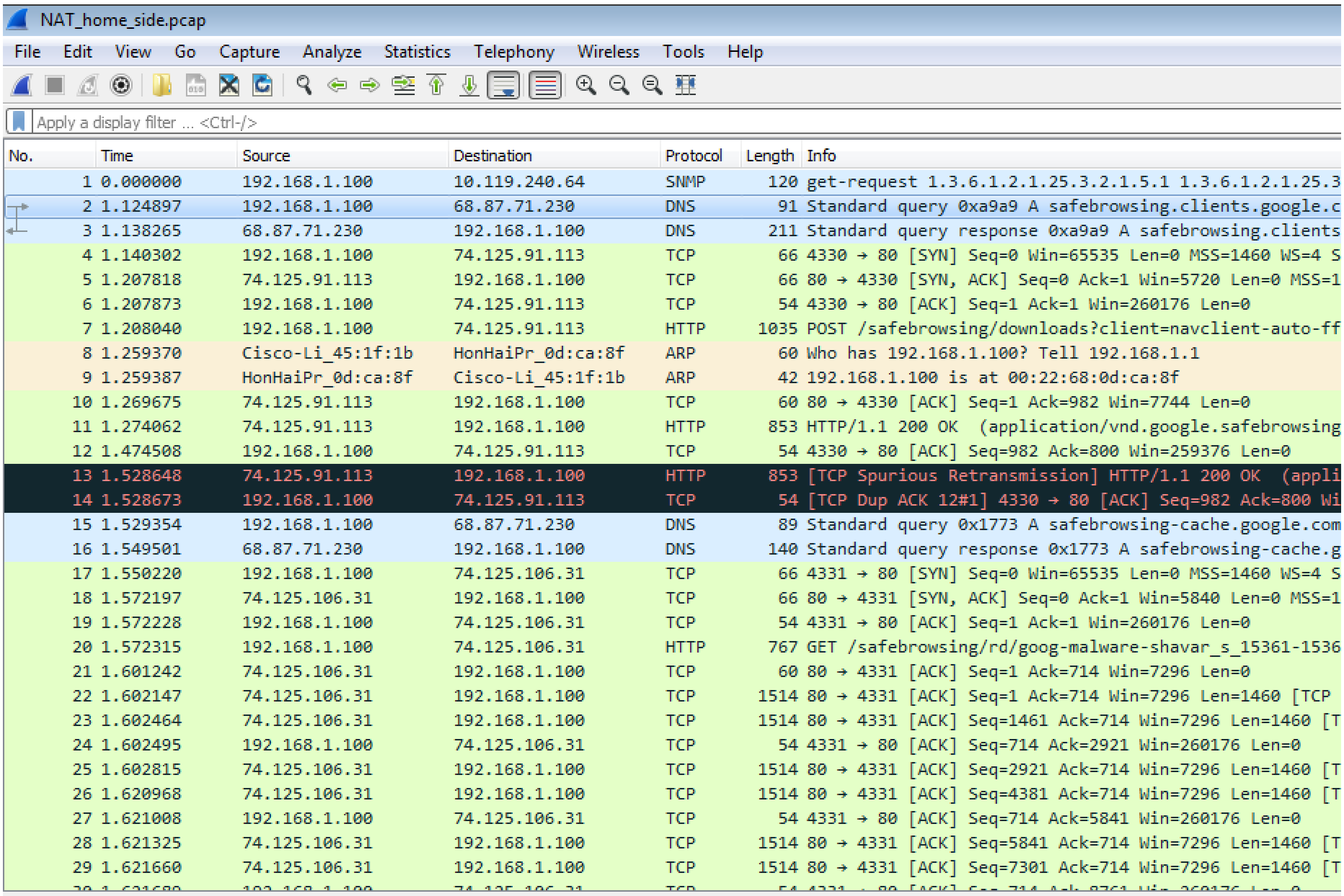
```
No.   Time      Source             Destination        Protocol  Length Info
    1 0.000000  192.168.1.100      10.119.240.64      SNMP       120 get-request 1.3.6.1.2.1.25.3.2.1.5.1 1.3.6.1.2.1.25.3.5.1.1.1 1.3.6.1.2.1.25.3.5.1.2.
    2 1.124897  192.168.1.100      68.87.71.230       DNS         91 Standard query 0xa9a9 A safebrowsing.clients.google.com
    3 1.138265  68.87.71.230       192.168.1.100      DNS        211 Standard query response 0xa9a9 A safebrowsing.clients.google.com CNAME clients.l.goog
    4 1.140302  192.168.1.100      74.125.91.113      TCP         66 4330 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM=1
    5 1.207818  74.125.91.113      192.168.1.100      TCP         66 80 → 4330 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=1 WS=64
    6 1.207873  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
    7 1.208040  192.168.1.100      74.125.91.113      HTTP      1035 POST /safebrowsing/downloads?client=navclient-auto-ffox&appver=3.0.14&pver=2.2&wrkey=
    8 1.259370  Cisco-Li_45:1f:1b  HonHaiPr_0d:ca:8f  ARP         60 Who has 192.168.1.100? Tell 192.168.1.1
    9 1.259387  HonHaiPr_0d:ca:8f  Cisco-Li_45:1f:1b  ARP         42 192.168.1.100 is at 00:22:68:0d:ca:8f
   10 1.269675  74.125.91.113      192.168.1.100      TCP         60 80 → 4330 [ACK] Seq=1 Ack=982 Win=7744 Len=0
   11 1.274062  74.125.91.113      192.168.1.100      HTTP       853 HTTP/1.1 200 OK  (application/vnd.google.safebrowsing-update)
   12 1.474508  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
   13 1.528648  74.125.91.113      192.168.1.100      HTTP       853 [TCP Spurious Retransmission] HTTP/1.1 200 OK  (application/vnd.google.safebrowsing-u
   14 1.528673  192.168.1.100      74.125.91.113      TCP         54 [TCP Dup ACK 12#1] 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
   15 1.529354  192.168.1.100      68.87.71.230       DNS         89 Standard query 0x1773 A safebrowsing-cache.google.com
   16 1.549501  68.87.71.230       192.168.1.100      DNS        140 Standard query response 0x1773 A safebrowsing-cache.google.com CNAME safebrowsing.cac
   17 1.550220  192.168.1.100      74.125.106.31      TCP         66 4331 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM=1
   18 1.572197  74.125.106.31      192.168.1.100      TCP         66 80 → 4331 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=64
   19 1.572228  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
   20 1.572315  192.168.1.100      74.125.106.31      HTTP       767 GET /safebrowsing/rd/goog-malware-shavar_s_15361-15365.15361-15365.: HTTP/1.1
   21 1.601242  74.125.106.31      192.168.1.100      TCP         60 80 → 4331 [ACK] Seq=1 Ack=714 Win=7296 Len=0
   22 1.602147  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=1 Ack=714 Win=7296 Len=1460 [TCP segment of a reassembled PDU]
   23 1.602464  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=1461 Ack=714 Win=7296 Len=1460 [TCP segment of a reassembled PDU]
   24 1.602495  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=714 Ack=2921 Win=260176 Len=0
   25 1.602815  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=2921 Ack=714 Win=7296 Len=1460 [TCP segment of a reassembled PDU]
   26 1.620968  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=4381 Ack=714 Win=7296 Len=1460 [TCP segment of a reassembled PDU]
   27 1.621008  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=714 Ack=5841 Win=260176 Len=0
   28 1.621325  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=5841 Ack=714 Win=7296 Len=1460 [TCP segment of a reassembled PDU]
   29 1.621660  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=7301 Ack=714 Win=7296 Len=1460 [TCP segment of a reassembled PDU]
```

```
▷ Frame 1: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
▷ Ethernet II, Src: HonHaiPr_0d:ca:8f (00:22:68:0d:ca:8f), Dst: Cisco-Li_45:1f:1b (00:22:6b:45:1f:1b)
▷ Internet Protocol Version 4, Src: 192.168.1.100, Dst: 10.119.240.64
▷ User Datagram Protocol, Src Port: 1028, Dst Port: 161
▷ Simple Network Management Protocol
```

```
0000  00 22 6b 45 1f 1b 00 22  68 0d ca 8f 08 00 45 00   ·"kE···" h·····E·
0010  00 6a a2 62 00 00 80 11  db 5c c0 a8 01 64 0a 77   ·j·b···· ·\···d·w
0020  f0 40 04 04 00 a1 00 56  00 36 30 4c 02 01 00 04   ·@·····V ·60L····
0030  06 70 75 62 6c 69 63 a0  3f 02 02 29 51 02 01 00   ·public· ?··)Q···
0040  02 01 00 30 33 30 0f 06  0b 2b 06 01 02 01 19 03   ···030·· ·+······
0050  02 01 05 01 05 00 30 0f  06 0b 2b 06 01 02 01 19   ······0· ··+·····
0060  03 05 01 01 01 05 00 30  0f 06 0b 2b 06 01 02 01   ·······0 ···+····
0070  19 03 05 01 02 01 05 00                            ········
```

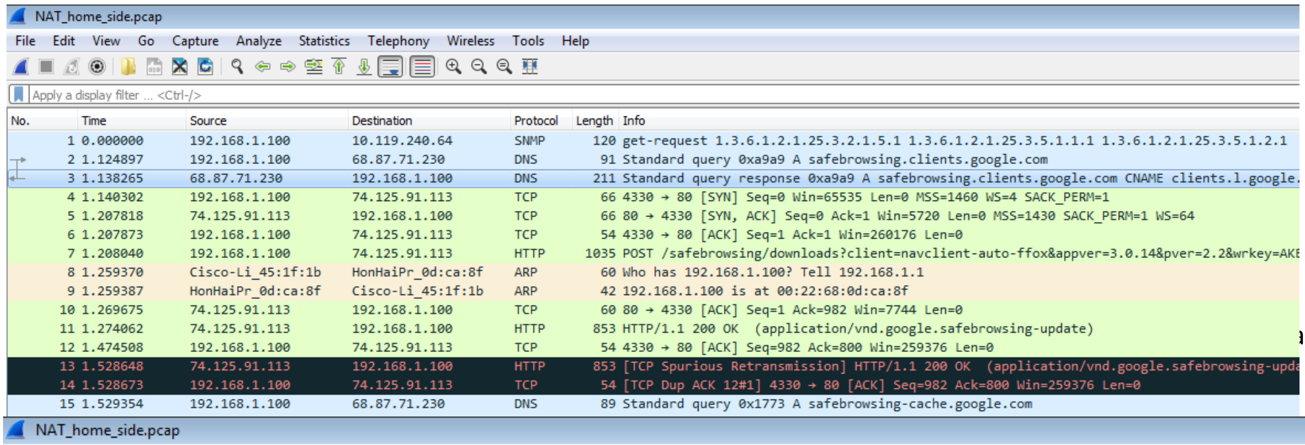
The screenshot above shows a packet trace file of the packet that was captured using the packet sniffing software called Wireshark.

Details of each packet is listed including the number time taken to capture the packet, the source i.e. where it’s coming from and the destination i.e. where it’s going, the protocol used to exchange the message and the information(Info) such as whether they are queries or response messages or use the GET or POST methods to send information to the client from the server.



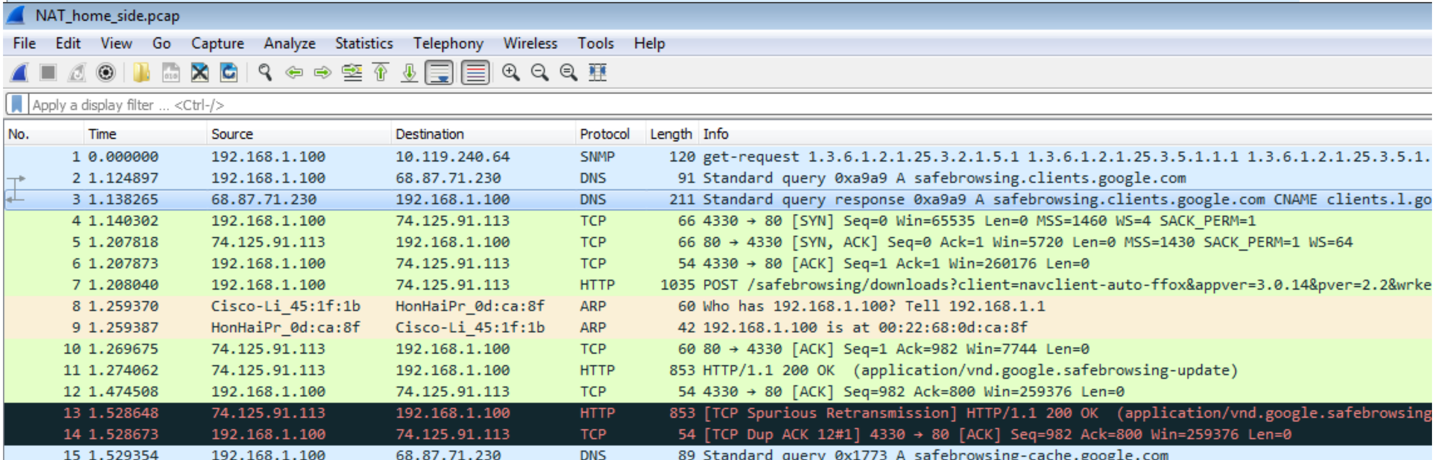
```
No.  Time      Source             Destination        Protocol  Length Info
   1 0.000000  192.168.1.100      10.119.240.64      SNMP       120 get-request 1.3.6.1.2.1.25.3.2.1.5.1 1.3.6.1.2.1.25.3
   2 1.124897  192.168.1.100      68.87.71.230       DNS         91 Standard query 0xa9a9 A safebrowsing.clients.google.c
   3 1.138265  68.87.71.230       192.168.1.100      DNS        211 Standard query response 0xa9a9 A safebrowsing.clients
   4 1.140302  192.168.1.100      74.125.91.113      TCP         66 4330 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 S
   5 1.207818  74.125.91.113      192.168.1.100      TCP         66 80 → 4330 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1
   6 1.207873  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
   7 1.208040  192.168.1.100      74.125.91.113      HTTP      1035 POST /safebrowsing/downloads?client=navclient-auto-ff
   8 1.259370  Cisco-Li_45:1f:1b  HonHaiPr_0d:ca:8f  ARP         60 Who has 192.168.1.100? Tell 192.168.1.1
   9 1.259387  HonHaiPr_0d:ca:8f  Cisco-Li_45:1f:1b  ARP         42 192.168.1.100 is at 00:22:68:0d:ca:8f
  10 1.269675  74.125.91.113      192.168.1.100      TCP         60 80 → 4330 [ACK] Seq=1 Ack=982 Win=7744 Len=0
  11 1.274062  74.125.91.113      192.168.1.100      HTTP       853 HTTP/1.1 200 OK  (application/vnd.google.safebrowsing
  12 1.474508  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
  13 1.528648  74.125.91.113      192.168.1.100      HTTP       853 [TCP Spurious Retransmission] HTTP/1.1 200 OK  (appli
  14 1.528673  192.168.1.100      74.125.91.113      TCP         54 [TCP Dup ACK 12#1] 4330 → 80 [ACK] Seq=982 Ack=800 Wi
  15 1.529354  192.168.1.100      68.87.71.230       DNS         89 Standard query 0x1773 A safebrowsing-cache.google.com
  16 1.549501  68.87.71.230       192.168.1.100      DNS        140 Standard query response 0x1773 A safebrowsing-cache.g
  17 1.550220  192.168.1.100      74.125.106.31      TCP         66 4331 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 S
  18 1.572197  74.125.106.31      192.168.1.100      TCP         66 80 → 4331 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1
  19 1.572228  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
  20 1.572315  192.168.1.100      74.125.106.31      HTTP       767 GET /safebrowsing/rd/goog-malware-shavar_s_15361-1536
  21 1.601242  74.125.106.31      192.168.1.100      TCP         60 80 → 4331 [ACK] Seq=1 Ack=714 Win=7296 Len=0
  22 1.602147  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=1 Ack=714 Win=7296 Len=1460 [TCP
  23 1.602464  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=1461 Ack=714 Win=7296 Len=1460 [T
  24 1.602495  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=714 Ack=2921 Win=260176 Len=0
  25 1.602815  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=2921 Ack=714 Win=7296 Len=1460 [T
  26 1.620968  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=4381 Ack=714 Win=7296 Len=1460 [T
  27 1.621008  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=714 Ack=5841 Win=260176 Len=0
  28 1.621325  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=5841 Ack=714 Win=7296 Len=1460 [T
  29 1.621660  74.125.106.31      192.168.1.100      TCP       1514 80 → 4331 [ACK] Seq=7301 Ack=714 Win=7296 Len=1460 [T
  30 1.621680  192.168.1.100      74.125.106.31      TCP         54 4331 → 80 [ACK] Seq=714 Ack=8761 Win=260176 Len=0
```

A close up of the packets captured on the trace file. The protocols used here are DNS, TCP and HTTP. DNS stands for Domain Name System which is used for the address of the website. The transfer control protocol os the TCP which is the protocol used to transfer the packet to and from the IP addresses.



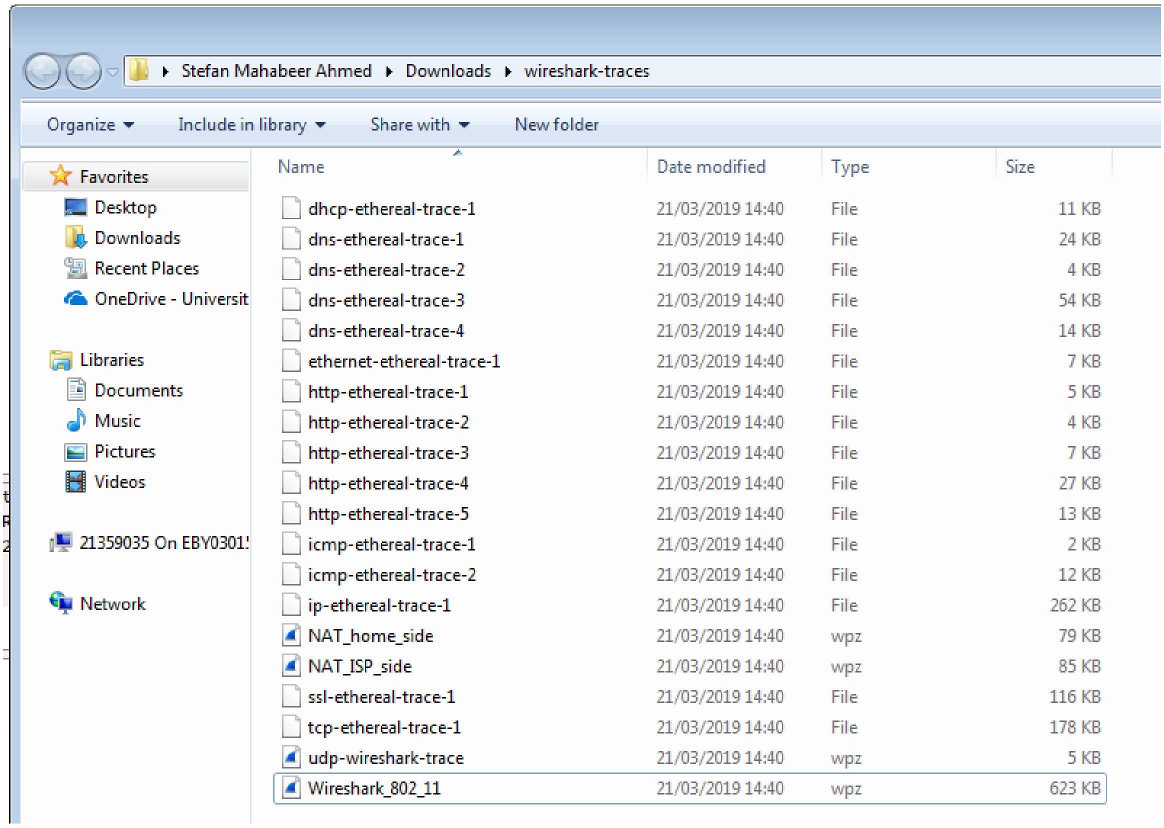
```
No.  Time      Source             Destination        Protocol  Length Info
   1 0.000000  192.168.1.100      10.119.240.64      SNMP       120 get-request 1.3.6.1.2.1.25.3.2.1.5.1 1.3.6.1.2.1.25.3.5.1.1.1 1.3.6.1.2.1.25.3.5.1.2.1
   2 1.124897  192.168.1.100      68.87.71.230       DNS         91 Standard query 0xa9a9 A safebrowsing.clients.google.com
   3 1.138265  68.87.71.230       192.168.1.100      DNS        211 Standard query response 0xa9a9 A safebrowsing.clients.google.com CNAME clients.l.google.
   4 1.140302  192.168.1.100      74.125.91.113      TCP         66 4330 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM=1
   5 1.207818  74.125.91.113      192.168.1.100      TCP         66 80 → 4330 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=1 WS=64
   6 1.207873  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
   7 1.208040  192.168.1.100      74.125.91.113      HTTP      1035 POST /safebrowsing/downloads?client=navclient-auto-ffox&appver=3.0.14&pver=2.2&wrkey=AKE
   8 1.259370  Cisco-Li_45:1f:1b  HonHaiPr_0d:ca:8f  ARP         60 Who has 192.168.1.100? Tell 192.168.1.1
   9 1.259387  HonHaiPr_0d:ca:8f  Cisco-Li_45:1f:1b  ARP         42 192.168.1.100 is at 00:22:68:0d:ca:8f
  10 1.269675  74.125.91.113      192.168.1.100      TCP         60 80 → 4330 [ACK] Seq=1 Ack=982 Win=7744 Len=0
  11 1.274062  74.125.91.113      192.168.1.100      HTTP       853 HTTP/1.1 200 OK  (application/vnd.google.safebrowsing-update)
  12 1.474508  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
  13 1.528648  74.125.91.113      192.168.1.100      HTTP       853 [TCP Spurious Retransmission] HTTP/1.1 200 OK  (application/vnd.google.safebrowsing-upda
  14 1.528673  192.168.1.100      74.125.91.113      TCP         54 [TCP Dup ACK 12#1] 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
  15 1.529354  192.168.1.100      68.87.71.230       DNS         89 Standard query 0x1773 A safebrowsing-cache.google.com
```

ıt



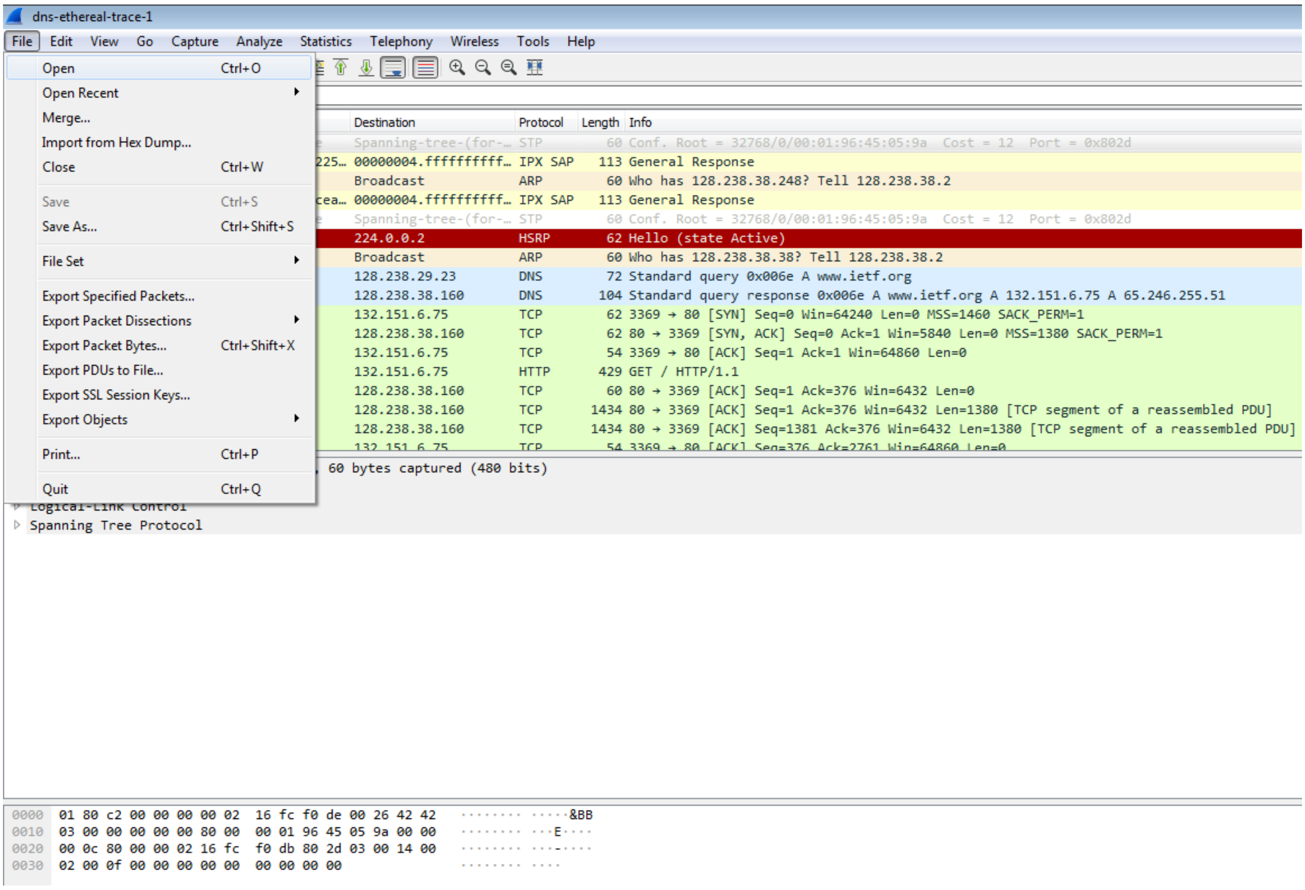
```
No.  Time      Source             Destination        Protocol  Length Info
   1 0.000000  192.168.1.100      10.119.240.64      SNMP       120 get-request 1.3.6.1.2.1.25.3.2.1.5.1 1.3.6.1.2.1.25.3.5.1.1.1 1.3.6.1.2.1.25.3.5.1.
   2 1.124897  192.168.1.100      68.87.71.230       DNS         91 Standard query 0xa9a9 A safebrowsing.clients.google.com
   3 1.138265  68.87.71.230       192.168.1.100      DNS        211 Standard query response 0xa9a9 A safebrowsing.clients.google.com CNAME clients.l.go
   4 1.140302  192.168.1.100      74.125.91.113      TCP         66 4330 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM=1
   5 1.207818  74.125.91.113      192.168.1.100      TCP         66 80 → 4330 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM=1 WS=64
   6 1.207873  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
   7 1.208040  192.168.1.100      74.125.91.113      HTTP      1035 POST /safebrowsing/downloads?client=navclient-auto-ffox&appver=3.0.14&pver=2.2&wrke
   8 1.259370  Cisco-Li_45:1f:1b  HonHaiPr_0d:ca:8f  ARP         60 Who has 192.168.1.100? Tell 192.168.1.1
   9 1.259387  HonHaiPr_0d:ca:8f  Cisco-Li_45:1f:1b  ARP         42 192.168.1.100 is at 00:22:68:0d:ca:8f
  10 1.269675  74.125.91.113      192.168.1.100      TCP         60 80 → 4330 [ACK] Seq=1 Ack=982 Win=7744 Len=0
  11 1.274062  74.125.91.113      192.168.1.100      HTTP       853 HTTP/1.1 200 OK  (application/vnd.google.safebrowsing-update)
  12 1.474508  192.168.1.100      74.125.91.113      TCP         54 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
  13 1.528648  74.125.91.113      192.168.1.100      HTTP       853 [TCP Spurious Retransmission] HTTP/1.1 200 OK  (application/vnd.google.safebrowsing
  14 1.528673  192.168.1.100      74.125.91.113      TCP         54 [TCP Dup ACK 12#1] 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
  15 1.529354  192.168.1.100      68.87.71.230       DNS         89 Standard query 0x1773 A safebrowsing-cache.google.com
```

The DNS Query and response messages are shown here. It says if the DNS message is a query or a response under the information(Info) column.



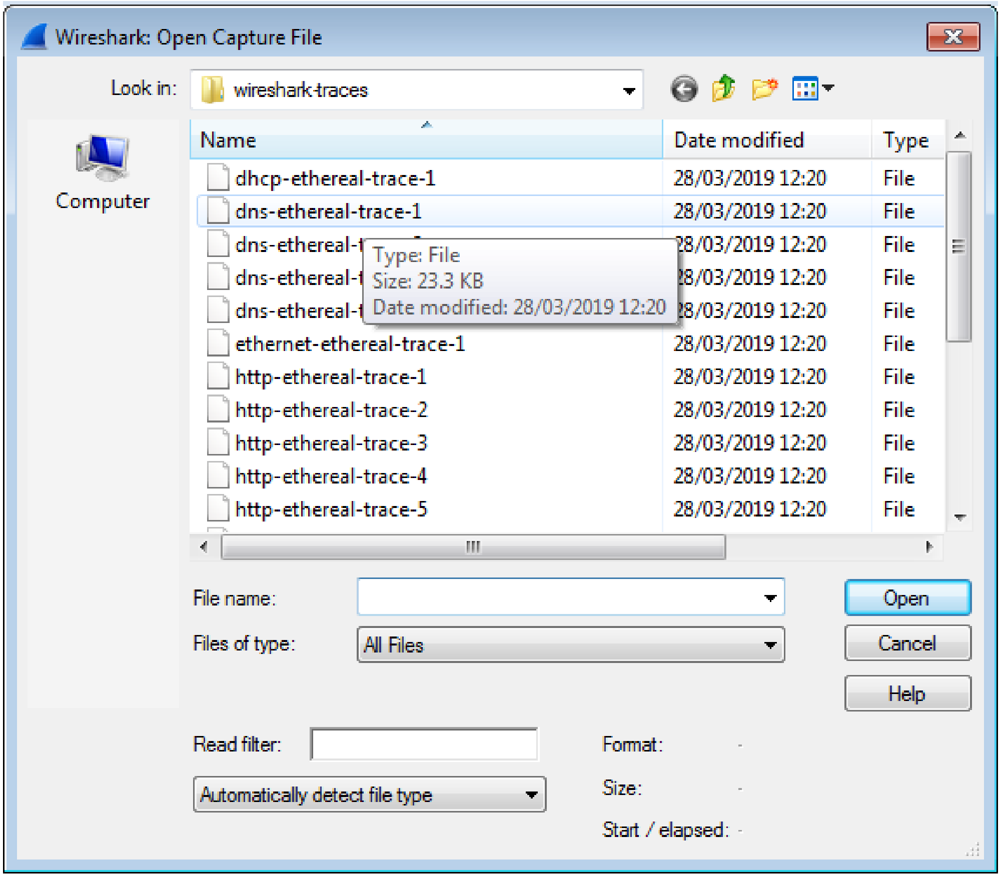
Name	Date modified	Type	Size
dhcp-ethereal-trace-1	21/03/2019 14:40	File	11 KB
dns-ethereal-trace-1	21/03/2019 14:40	File	24 KB
dns-ethereal-trace-2	21/03/2019 14:40	File	4 KB
dns-ethereal-trace-3	21/03/2019 14:40	File	54 KB
dns-ethereal-trace-4	21/03/2019 14:40	File	14 KB
ethernet-ethereal-trace-1	21/03/2019 14:40	File	7 KB
http-ethereal-trace-1	21/03/2019 14:40	File	5 KB
http-ethereal-trace-2	21/03/2019 14:40	File	4 KB
http-ethereal-trace-3	21/03/2019 14:40	File	7 KB
http-ethereal-trace-4	21/03/2019 14:40	File	27 KB
http-ethereal-trace-5	21/03/2019 14:40	File	13 KB
icmp-ethereal-trace-1	21/03/2019 14:40	File	2 KB
icmp-ethereal-trace-2	21/03/2019 14:40	File	12 KB
ip-ethereal-trace-1	21/03/2019 14:40	File	262 KB
NAT_home_side	21/03/2019 14:40	wpz	79 KB
NAT_ISP_side	21/03/2019 14:40	wpz	85 KB
ssl-ethereal-trace-1	21/03/2019 14:40	File	116 KB
tcp-ethereal-trace-1	21/03/2019 14:40	File	178 KB
udp-wireshark-trace	21/03/2019 14:40	wpz	5 KB
Wireshark_802_11	21/03/2019 14:40	wpz	623 KB

To answer the question here is the folder of trace files collected by Wireshark while following lab tasks on an author’s computer. The dns-ethereal-trace-1 file located at the top is the file to use which has all the details of the DNS packet that was captured.



```
     Destination               Protocol Length Info
     Spanning-tree-(for-… STP        60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Port = 0x802d
225… 00000004.ffffffffff… IPX SAP   113 General Response
     Broadcast            ARP        60 Who has 128.238.38.248? Tell 128.238.38.2
cea… 00000004.ffffffffff… IPX SAP   113 General Response
     Spanning-tree-(for-… STP        60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Port = 0x802d
     224.0.0.2            HSRP       62 Hello (state Active)
     Broadcast            ARP        60 Who has 128.238.38.38? Tell 128.238.38.2
     128.238.29.23        DNS        72 Standard query 0x006e A www.ietf.org
     128.238.38.160       DNS       104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
     132.151.6.75         TCP        62 3369 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
     128.238.38.160       TCP        62 80 → 3369 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM=1
     132.151.6.75         TCP        54 3369 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
     132.151.6.75         HTTP      429 GET / HTTP/1.1
     128.238.38.160       TCP        60 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=0
     128.238.38.160       TCP      1434 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
     128.238.38.160       TCP      1434 80 → 3369 [ACK] Seq=1381 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
     132.151.6.75         TCP        54 3369 → 80 [ACK] Seq=376 Ack=2761 Win=64860 Len=0

, 60 bytes captured (480 bits)
Logical-Link Control
Spanning Tree Protocol

0000  01 80 c2 00 00 00 00 02  16 fc f0 de 00 26 42 42   ········ ·····&BB
0010  03 00 00 00 00 00 80 00  00 01 96 45 05 9a 00 00   ········ ···E····
0020  00 0c 80 00 00 02 16 fc  f0 db 80 2d 03 00 14 00   ········ ···-····
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ········ ····
```

Back to Wireshark I go to File, Open to open a trace file called dns-ethereal-trace-1.



Look in: wireshark-traces

Name	Date modified	Type
dhcp-ethereal-trace-1	28/03/2019 12:20	File
dns-ethereal-trace-1	28/03/2019 12:20	File
dns-ethereal-t…	28/03/2019 12:20	File
dns-ethereal-t…	28/03/2019 12:20	File
dns-ethereal-t…	28/03/2019 12:20	File
ethernet-ethereal-trace-1	28/03/2019 12:20	File
http-ethereal-trace-1	28/03/2019 12:20	File
http-ethereal-trace-2	28/03/2019 12:20	File
http-ethereal-trace-3	28/03/2019 12:20	File
http-ethereal-trace-4	28/03/2019 12:20	File
http-ethereal-trace-5	28/03/2019 12:20	File

Type: File
Size: 23.3 KB
Date modified: 28/03/2019 12:20

File name:

Files of type: All Files

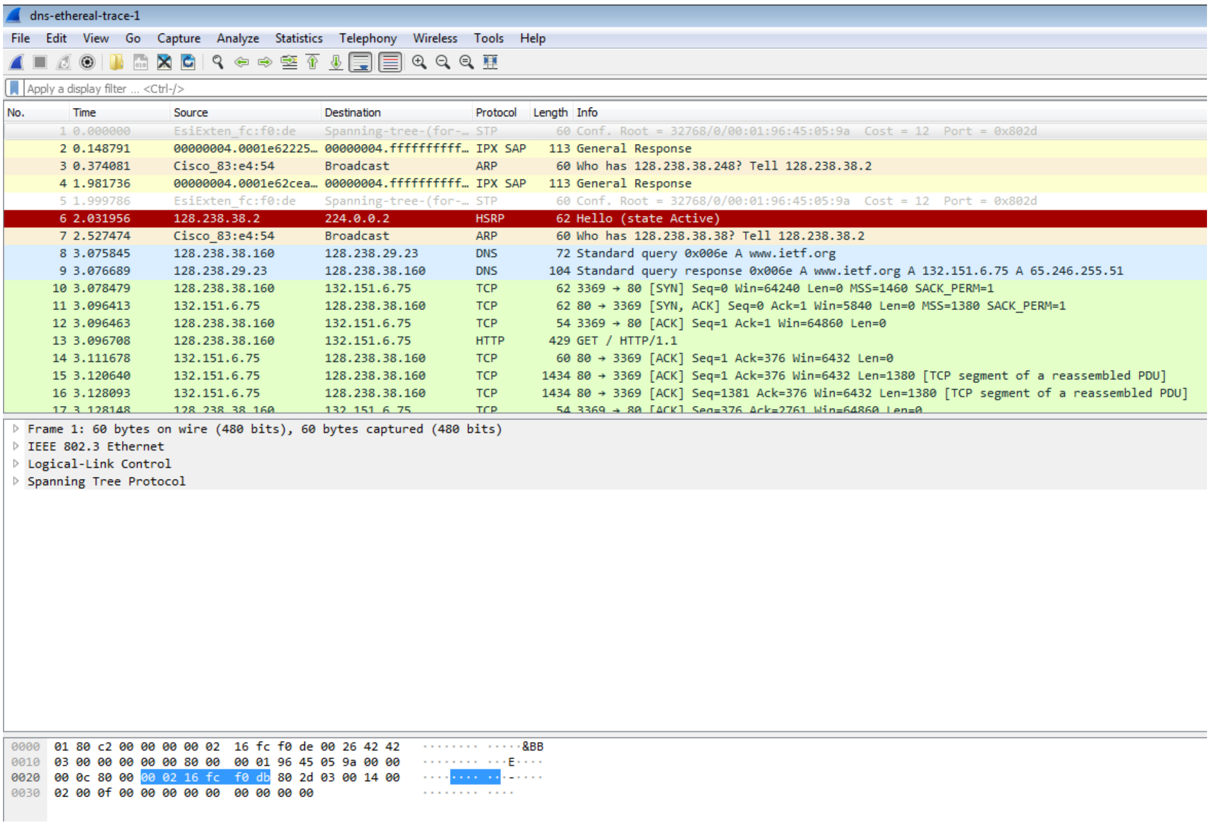
Read filter:

Automatically detect file type

Format: -

Size: -

Start / elapsed: -

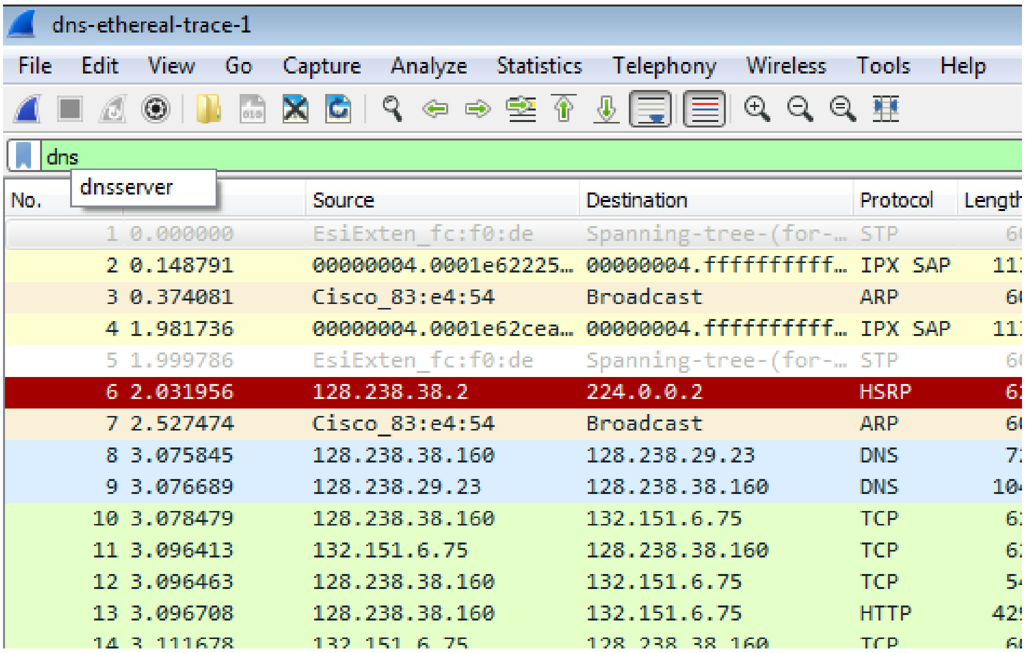

dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time      Source                 Destination            Protocol Length Info
   1 0.000000  EsiExten_fc:f0:de      Spanning-tree-(for-…   STP        60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Port = 0x802d
   2 0.148791  00000004.0001e62225…   00000004.ffffffffff…   IPX SAP   113 General Response
   3 0.374081  Cisco_83:e4:54         Broadcast              ARP        60 Who has 128.238.38.248? Tell 128.238.38.2
   4 1.981736  00000004.0001e62cea…   00000004.ffffffffff…   IPX SAP   113 General Response
   5 1.999786  EsiExten_fc:f0:de      Spanning-tree-(for-…   STP        60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Port = 0x802d
   6 2.031956  128.238.38.2           224.0.0.2              HSRP       62 Hello (state Active)
   7 2.527474  Cisco_83:e4:54         Broadcast              ARP        60 Who has 128.238.38.38? Tell 128.238.38.2
   8 3.075845  128.238.38.160         128.238.29.23          DNS        72 Standard query 0x006e A www.ietf.org
   9 3.076689  128.238.29.23          128.238.38.160         DNS       104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
  10 3.078479  128.238.38.160         132.151.6.75           TCP        62 3369 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
  11 3.096413  132.151.6.75           128.238.38.160         TCP        62 80 → 3369 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM=1
  12 3.096463  128.238.38.160         132.151.6.75           TCP        54 3369 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
  13 3.096708  128.238.38.160         132.151.6.75           HTTP      429 GET / HTTP/1.1
  14 3.111678  132.151.6.75           128.238.38.160         TCP        60 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=0
  15 3.120640  132.151.6.75           128.238.38.160         TCP      1434 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
  16 3.128093  132.151.6.75           128.238.38.160         TCP      1434 80 → 3369 [ACK] Seq=1381 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
  17 3.128148  128.238.38.160         132.151.6.75           TCP        54 3369 → 80 [ACK] Seq=376 Ack=2761 Win=64860 Len=0
```

```
▷ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▷ IEEE 802.3 Ethernet
▷ Logical-Link Control
▷ Spanning Tree Protocol
```

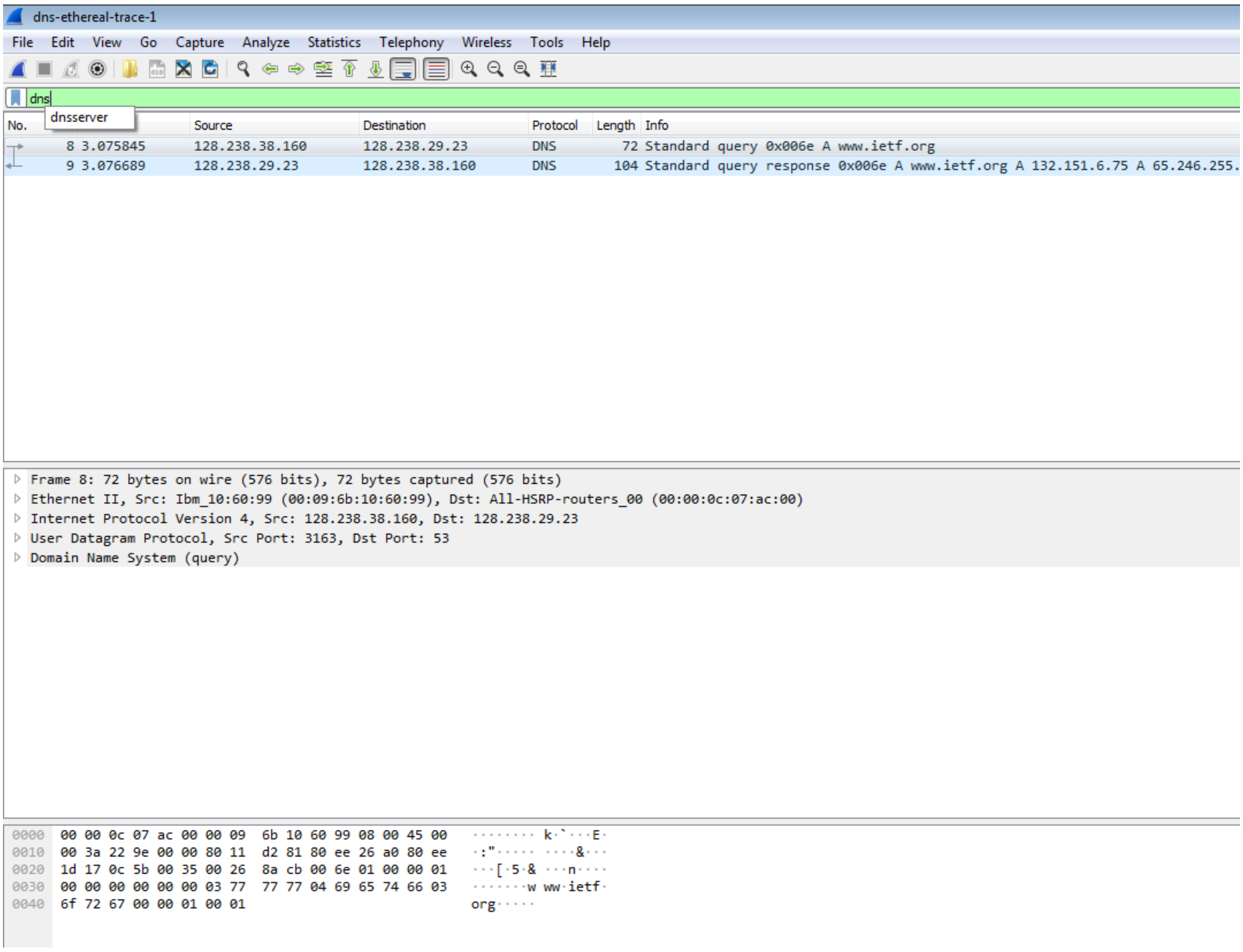
```
0000  01 80 c2 00 00 00 00 02  16 fc f0 de 00 26 42 42   ........ .....&BB
0010  03 00 00 00 00 00 80 00  00 01 96 45 05 9a 00 00   ........ ...E....
0020  00 0c 80 00 00 02 16 fc  f0 db 80 2d 03 00 14 00   ........ ..-.....
0030  02 00 0f 00 00 00 00 00  00 00 00 00               ........ ....
```

Details of the trace file are shown including a frame mentioning the size of the message captured in bits and bytes (in a shorter form of bits) and the type of interface it was sent over which in this case is Ethernet.


dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
dnsserver

```
No.  Time      Source                 Destination            Protocol Length
   1 0.000000  EsiExten_fc:f0:de      Spanning-tree-(for-…   STP        6
   2 0.148791  00000004.0001e62225…   00000004.ffffffffff…   IPX SAP   11
   3 0.374081  Cisco_83:e4:54         Broadcast              ARP        6
   4 1.981736  00000004.0001e62cea…   00000004.ffffffffff…   IPX SAP   11
   5 1.999786  EsiExten_fc:f0:de      Spanning-tree-(for-…   STP        6
   6 2.031956  128.238.38.2           224.0.0.2              HSRP       6
   7 2.527474  Cisco_83:e4:54         Broadcast              ARP        6
   8 3.075845  128.238.38.160         128.238.29.23          DNS        7
   9 3.076689  128.238.29.23          128.238.38.160         DNS       10
  10 3.078479  128.238.38.160         132.151.6.75           TCP        6
  11 3.096413  132.151.6.75           128.238.38.160         TCP        6
  12 3.096463  128.238.38.160         132.151.6.75           TCP        5
  13 3.096708  128.238.38.160         132.151.6.75           HTTP      42
  14 3.111678  132.151.6.75           128.238.38.160         TCP        6
```

I type dns in the display filter box to filter the display to filter the display to show the packets captured that used the DNS protocol and press Enter.

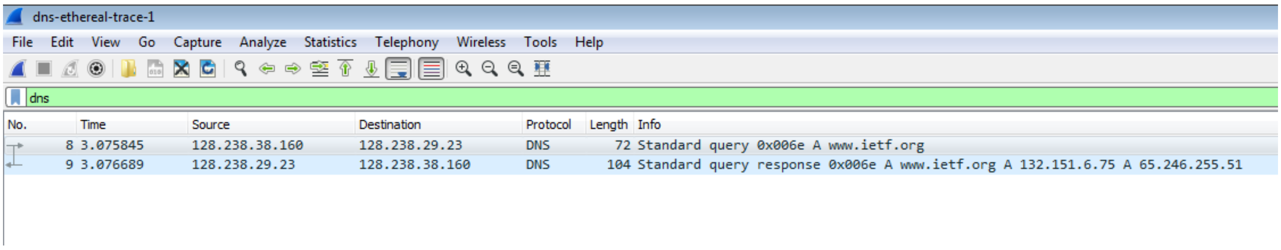

dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
dnsserver

```
No.             Source           Destination      Protocol  Length Info
    8 3.075845  128.238.38.160   128.238.29.23    DNS         72 Standard query 0x006e A www.ietf.org
    9 3.076689  128.238.29.23    128.238.38.160   DNS        104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.
```

```
▷ Frame 8: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
▷ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
▷ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.23
▷ User Datagram Protocol, Src Port: 3163, Dst Port: 53
▷ Domain Name System (query)
```

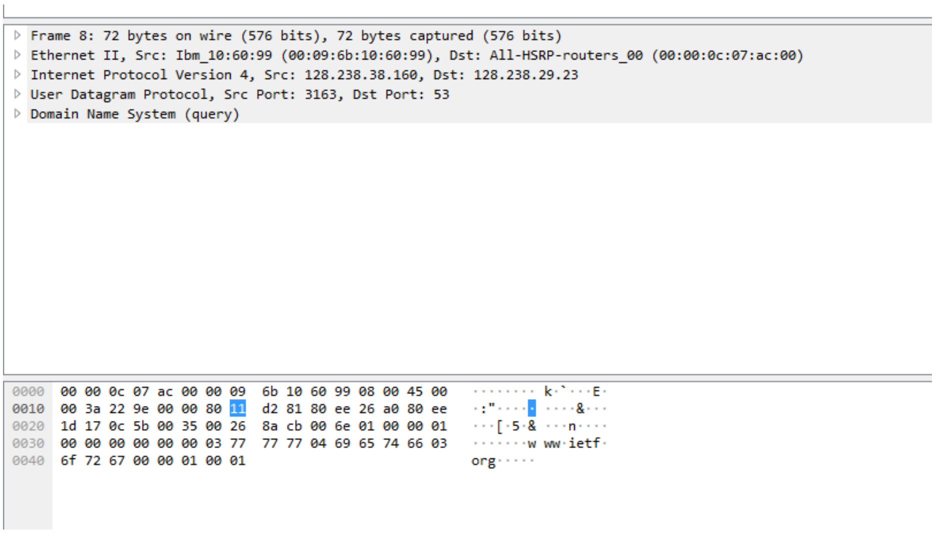
```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

The DNS query and response messages are located here.


dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

```
No.   Time      Source           Destination      Protocol  Length Info
    8 3.075845  128.238.38.160   128.238.29.23    DNS         72 Standard query 0x006e A www.ietf.org
    9 3.076689  128.238.29.23    128.238.38.160   DNS        104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
```

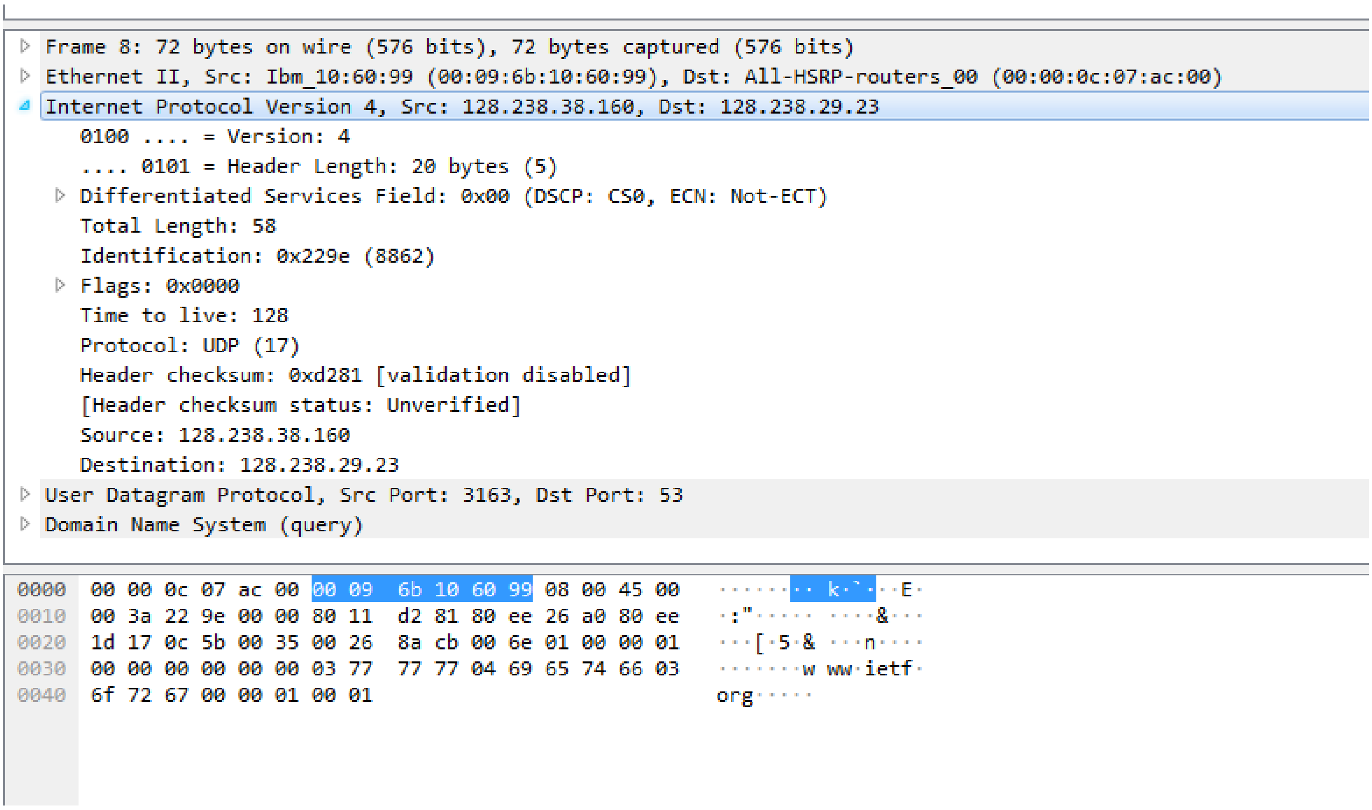
So by clicking on the message sent as a query I can look at the details of the message in the screenshot below.



```
▷ Frame 8: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
▷ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
▷ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.23
▷ User Datagram Protocol, Src Port: 3163, Dst Port: 53
▷ Domain Name System (query)
```

```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

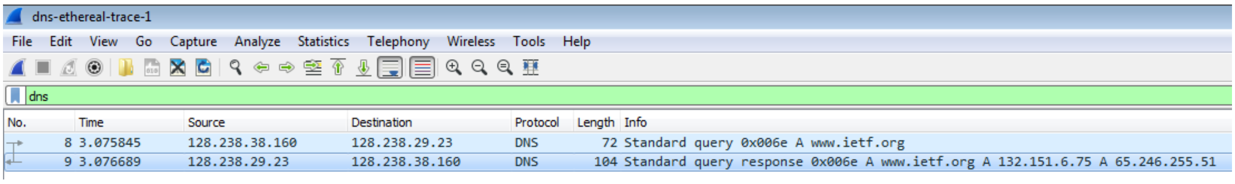
Here we have the Frame, Ethernet Internet Protocol, User Datagram and the DNS Query details to look at which are viewed by clicking on the triangle next to each of them. Clicking on the triangle next to Internet Protocol version 4 expands all the details of the internet protocol used to send the query.



```
▷ Frame 8: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
▷ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
◢ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.23
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 58
     Identification: 0x229e (8862)
   ▷ Flags: 0x0000
     Time to live: 128
     Protocol: UDP (17)
     Header checksum: 0xd281 [validation disabled]
     [Header checksum status: Unverified]
     Source: 128.238.38.160
     Destination: 128.238.29.23
▷ User Datagram Protocol, Src Port: 3163, Dst Port: 53
▷ Domain Name System (query)
```

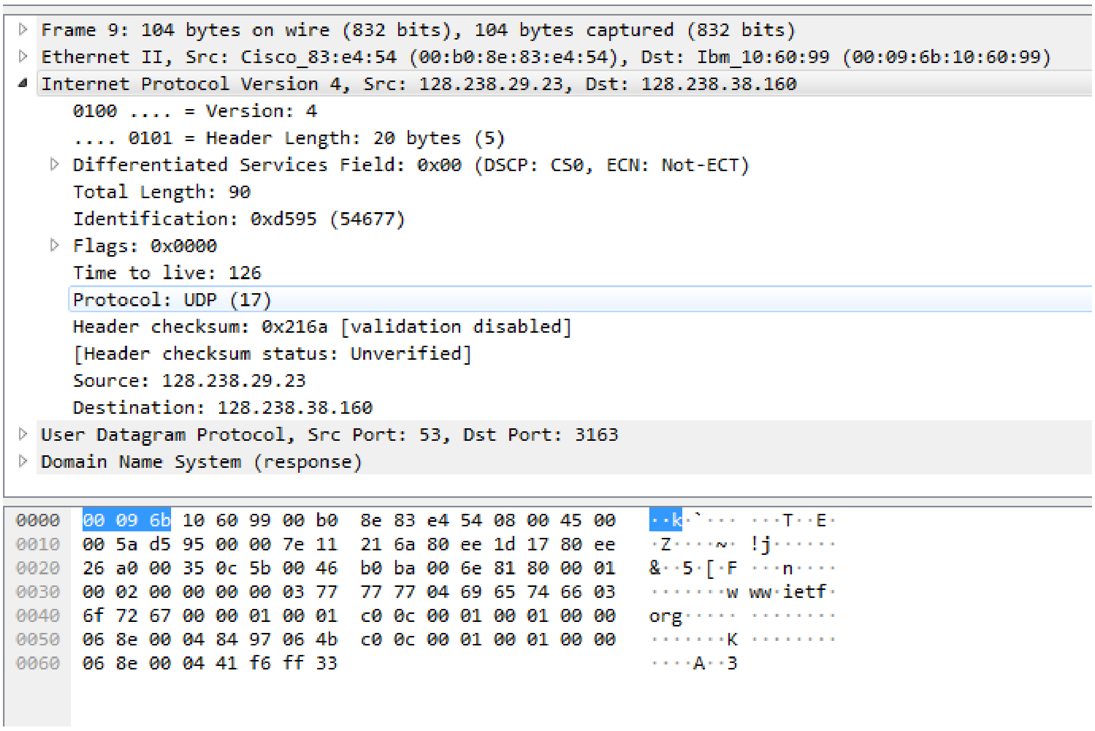
```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ······ ·· k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

I click on the query response message…


dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

```
No.   Time       Source           Destination      Protocol  Length  Info
    8 3.075845   128.238.38.160   128.238.29.23    DNS           72  Standard query 0x006e A www.ietf.org
    9 3.076689   128.238.29.23    128.238.38.160   DNS          104  Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
```

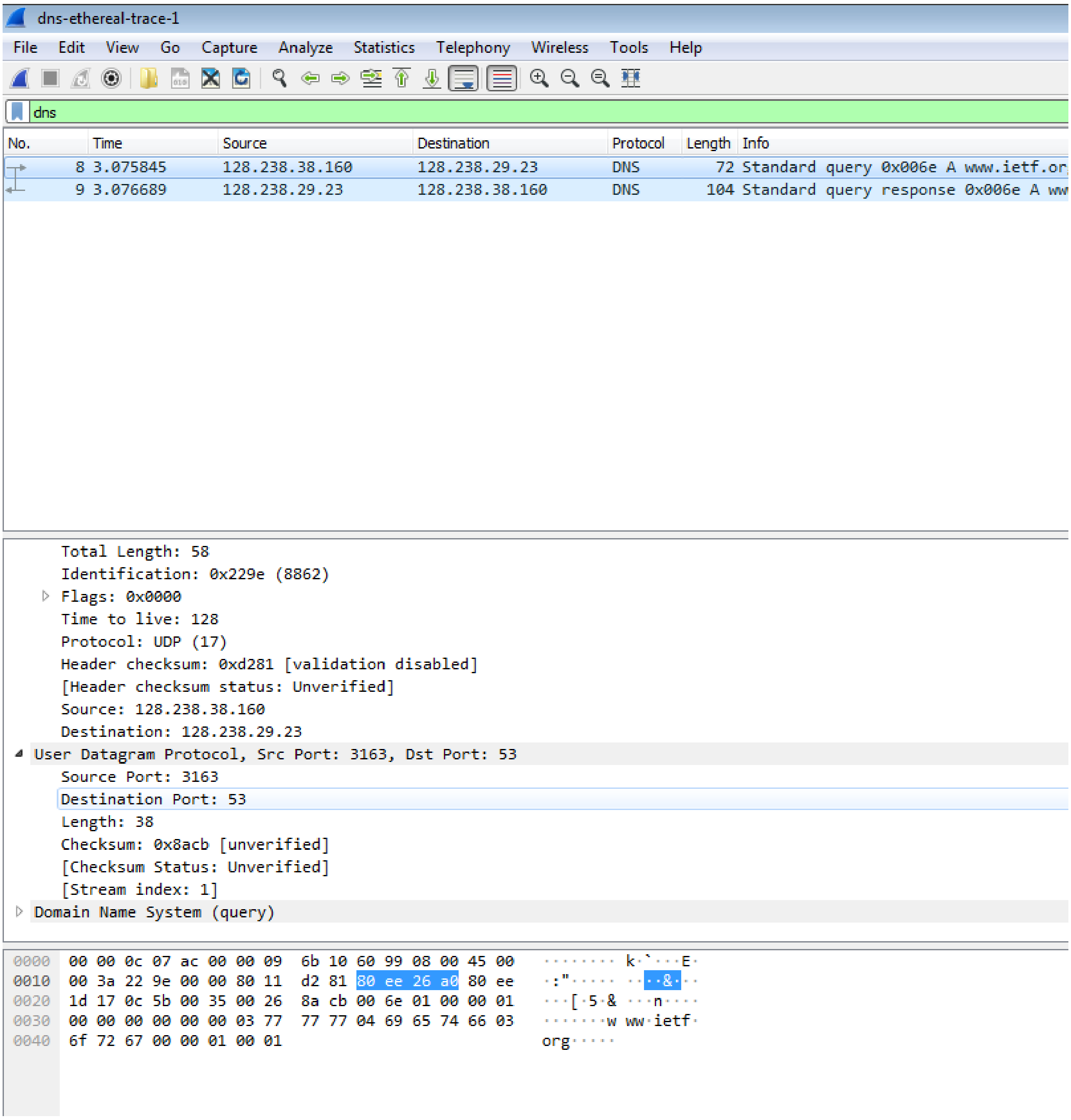
…to get the following details one of which is details of the User Datagram Protocol used to send the messages. The user datagram is sent with the query message from the client to the server.



```
▷ Frame 9: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
▷ Ethernet II, Src: Cisco_83:e4:54 (00:b0:8e:83:e4:54), Dst: Ibm_10:60:99 (00:09:6b:10:60:99)
◢ Internet Protocol Version 4, Src: 128.238.29.23, Dst: 128.238.38.160
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 90
     Identification: 0xd595 (54677)
   ▷ Flags: 0x0000
     Time to live: 126
     Protocol: UDP (17)
     Header checksum: 0x216a [validation disabled]
     [Header checksum status: Unverified]
     Source: 128.238.29.23
     Destination: 128.238.38.160
▷ User Datagram Protocol, Src Port: 53, Dst Port: 3163
▷ Domain Name System (response)

0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~· !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org····· ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```

The DNS query and response messages are sent over UDP(which is the user datagram protocol).

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



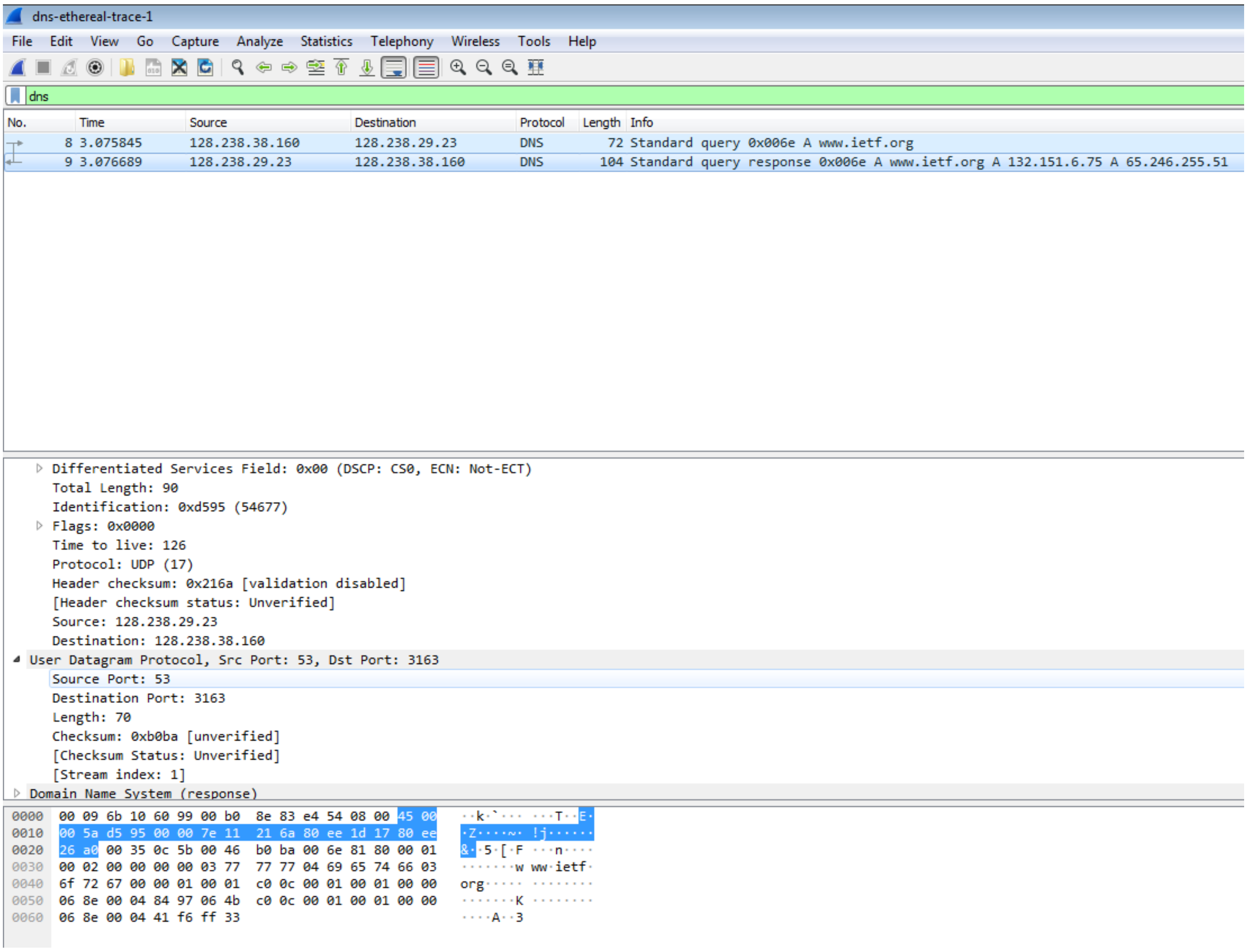
dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

No.	Time	Source	Destination	Protocol	Length	Info
8	3.075845	128.238.38.160	128.238.29.23	DNS	72	Standard query 0x006e A www.ietf.or
9	3.076689	128.238.29.23	128.238.38.160	DNS	104	Standard query response 0x006e A ww

```
    Total Length: 58
    Identification: 0x229e (8862)
  ▷ Flags: 0x0000
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0xd281 [validation disabled]
    [Header checksum status: Unverified]
    Source: 128.238.38.160
    Destination: 128.238.29.23
◢ User Datagram Protocol, Src Port: 3163, Dst Port: 53
    Source Port: 3163
    Destination Port: 53
    Length: 38
    Checksum: 0x8acb [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
▷ Domain Name System (query)
```

```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

The destination port is 53.

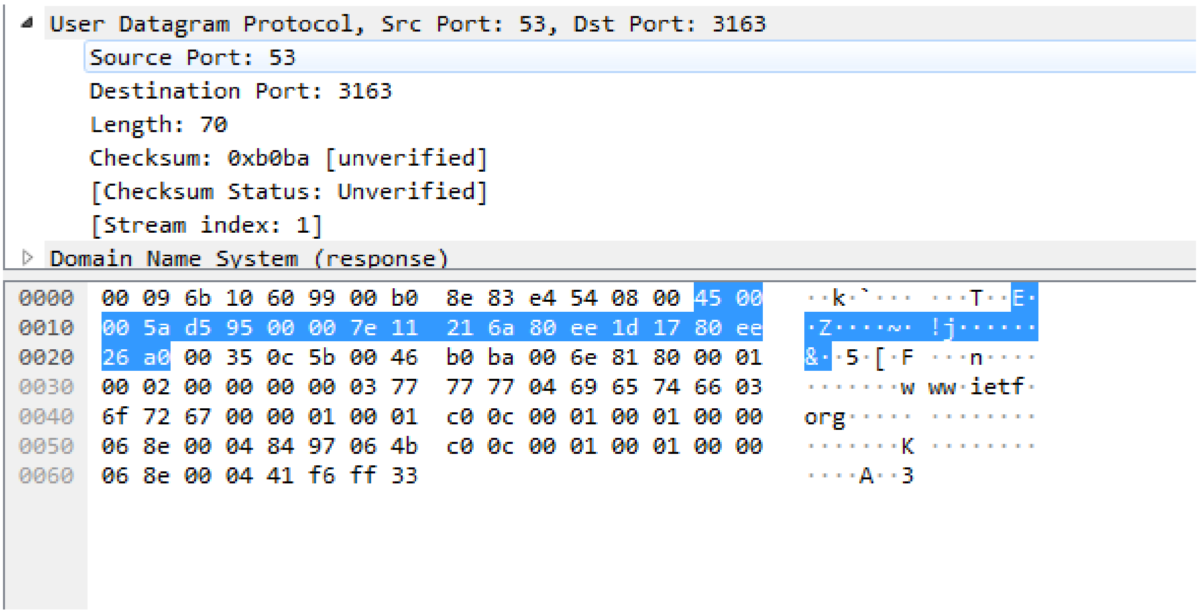


dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

```
No.   Time        Source           Destination      Protocol  Length  Info
  8 3.075845    128.238.38.160   128.238.29.23    DNS          72 Standard query 0x006e A www.ietf.org
  9 3.076689    128.238.29.23    128.238.38.160   DNS         104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
```

```
  ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 90
    Identification: 0xd595 (54677)
  ▷ Flags: 0x0000
    Time to live: 126
    Protocol: UDP (17)
    Header checksum: 0x216a [validation disabled]
    [Header checksum status: Unverified]
    Source: 128.238.29.23
    Destination: 128.238.38.160
◢ User Datagram Protocol, Src Port: 53, Dst Port: 3163
    Source Port: 53
    Destination Port: 3163
    Length: 70
    Checksum: 0xb0ba [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
▷ Domain Name System (response)
```

```
0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~· !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org····· ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```



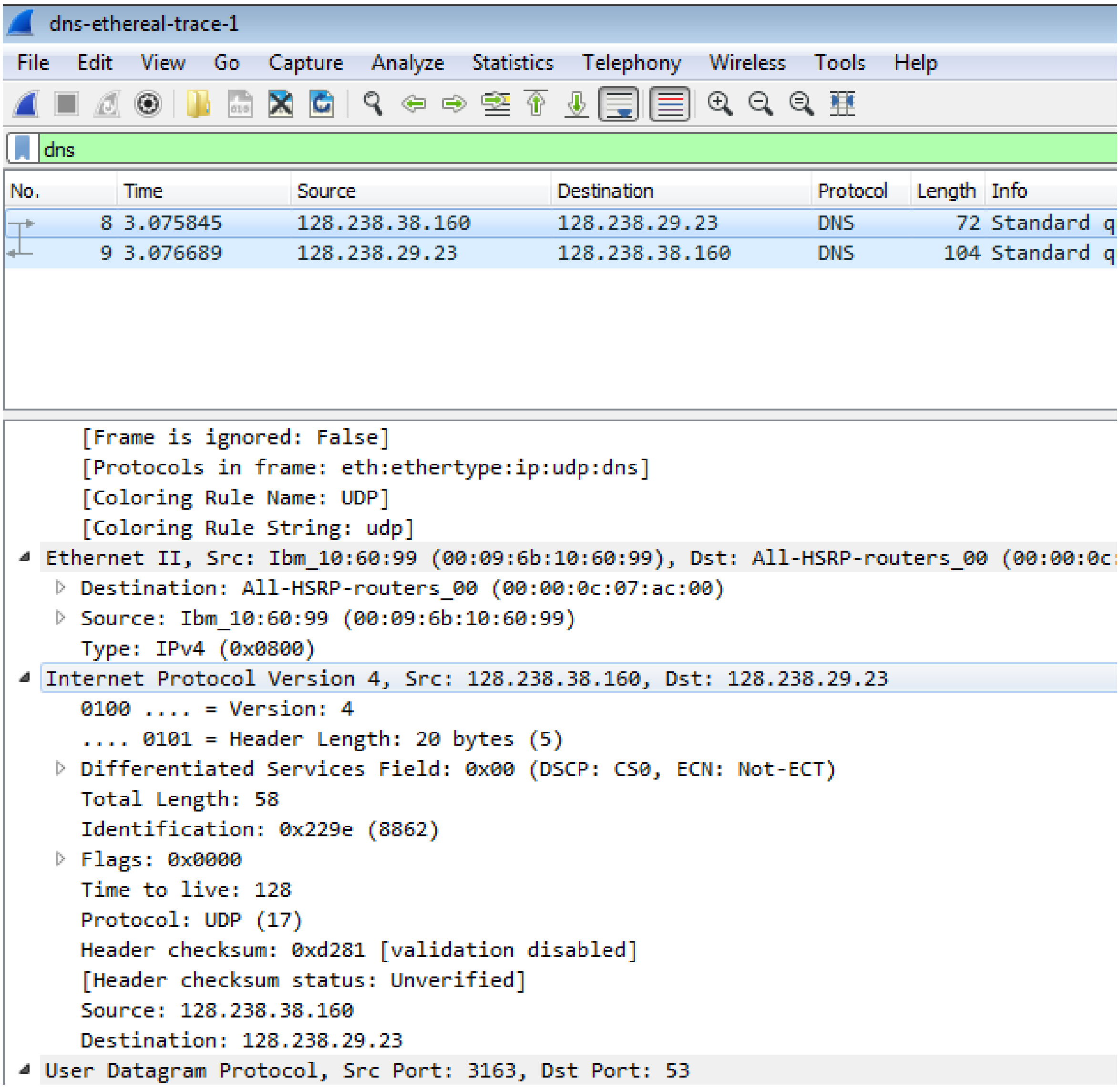
```
◢ User Datagram Protocol, Src Port: 53, Dst Port: 3163
    Source Port: 53
    Destination Port: 3163
    Length: 70
    Checksum: 0xb0ba [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
▷ Domain Name System (response)
```

```
0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~· !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org····· ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```

The user datagram was sent back with the response from the server to the client.

The source port of the DNS response message is 53. The destination and source ports are where the messages go through to the client and the server when they are connected to and communicating with each other.

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?

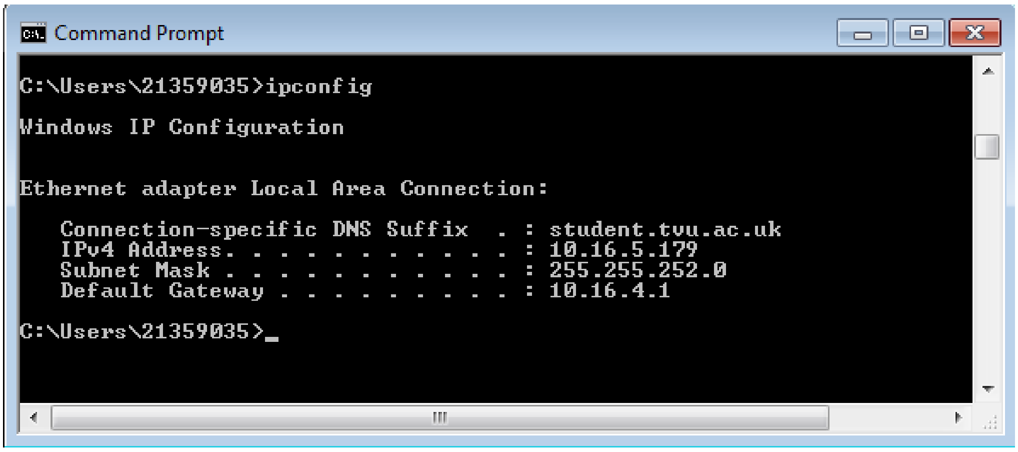

dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

No.	Time	Source	Destination	Protocol	Length	Info
8	3.075845	128.238.38.160	128.238.29.23	DNS	72	Standard q
9	3.076689	128.238.29.23	128.238.38.160	DNS	104	Standard q

```
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:dns]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
◢ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c
  ▷ Destination: All-HSRP-routers_00 (00:00:0c:07:ac:00)
  ▷ Source: Ibm_10:60:99 (00:09:6b:10:60:99)
    Type: IPv4 (0x0800)
◢ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.23
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 58
    Identification: 0x229e (8862)
  ▷ Flags: 0x0000
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0xd281 [validation disabled]
    [Header checksum status: Unverified]
    Source: 128.238.38.160
    Destination: 128.238.29.23
◢ User Datagram Protocol, Src Port: 3163, Dst Port: 53
```

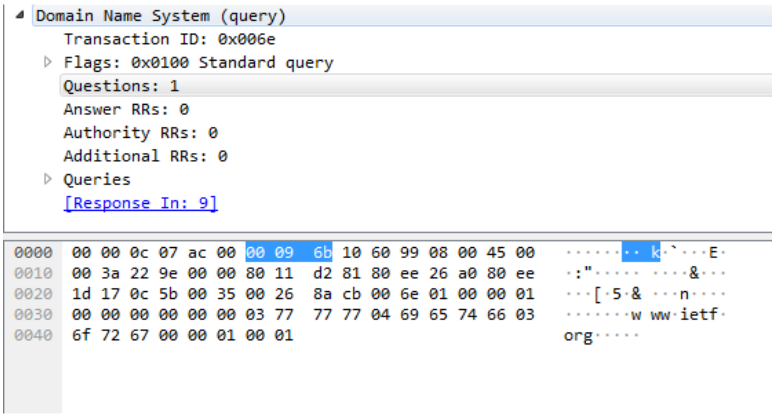
The destination column and the internet protocol details show the destination for the DNS query message as 128.238.29.23.

The IP address the DNS query message is sent to is 128.238.29.23.



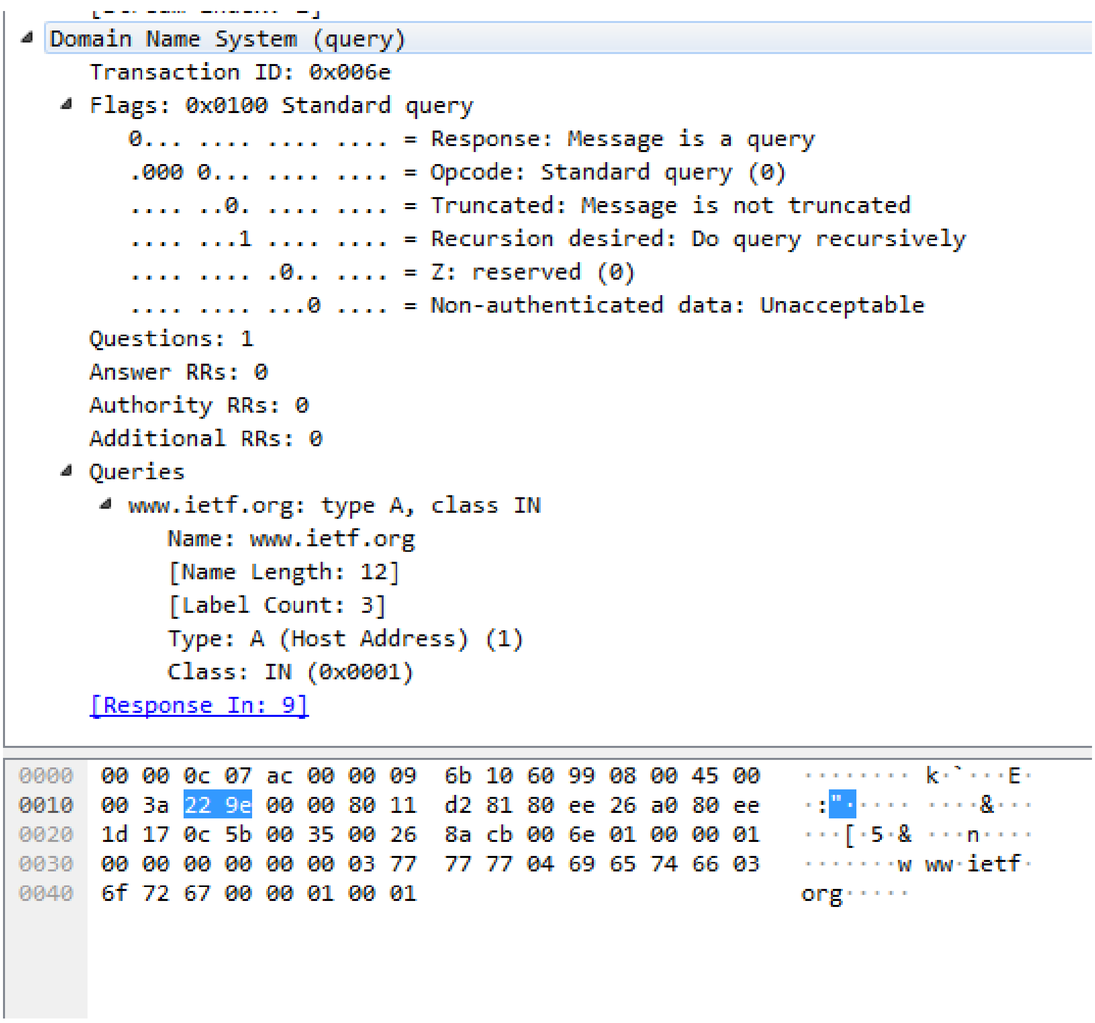
```
C:\Users\21359035>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : student.tvu.ac.uk
   IPv4 Address. . . . . . . . . . . : 10.16.5.179
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 10.16.4.1

C:\Users\21359035>_
```

Since this trace file was captured from the computer at a different location to the one I am using this computer the IP address of our local DNS server is different to the one the query message was sent to but it would be the same if I was using the computer at that location. It would be the same IP address if I was running Wireshark on a live network connection.



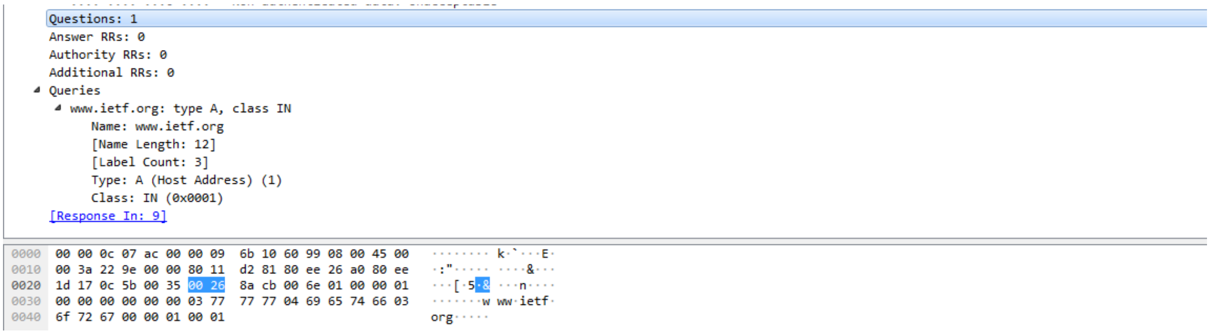
```
◢ Domain Name System (query)
    Transaction ID: 0x006e
  ▷ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
    [Response In: 9]

0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ······ ·· k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

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

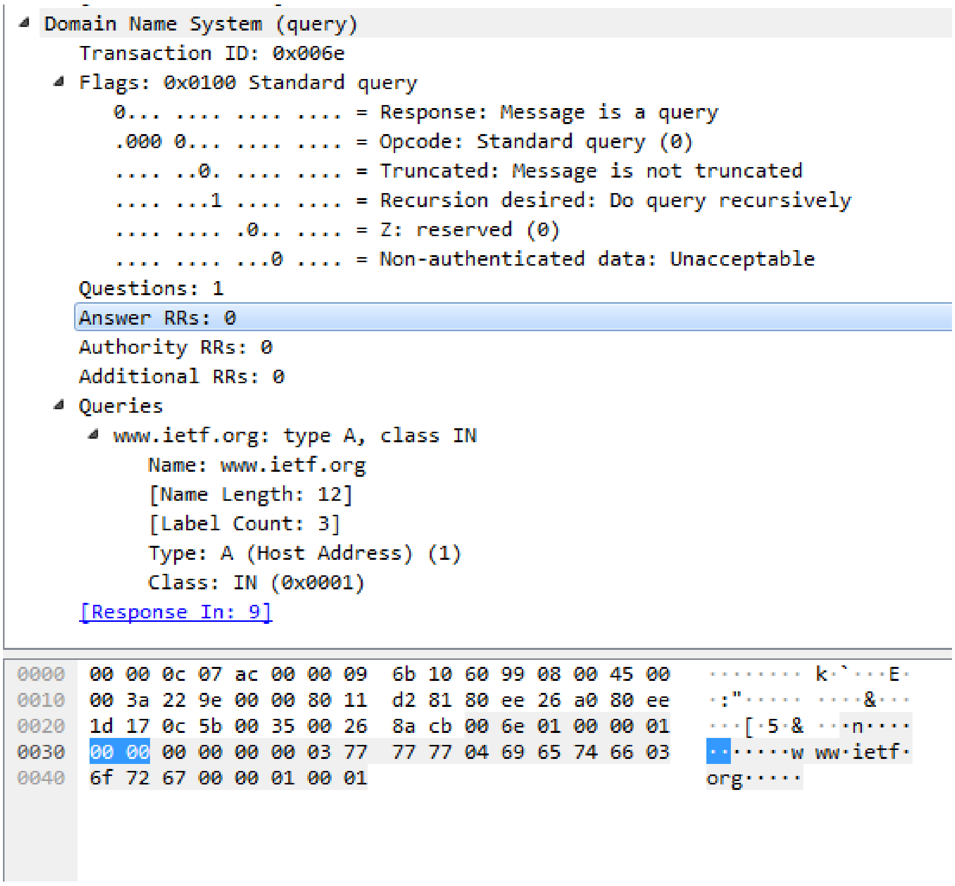


```
Domain Name System (query)
    Transaction ID: 0x006e
  Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  Queries
      www.ietf.org: type A, class IN
          Name: www.ietf.org
          [Name Length: 12]
          [Label Count: 3]
          Type: A (Host Address) (1)
          Class: IN (0x0001)
    [Response In: 9]

0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

Examining the DNS query it is a type A query. This query is not a query for any name server to include the name of any server and their IP addresses.



```
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  Queries
      www.ietf.org: type A, class IN
          Name: www.ietf.org
          [Name Length: 12]
          [Label Count: 3]
          Type: A (Host Address) (1)
          Class: IN (0x0001)
    [Response In: 9]

0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

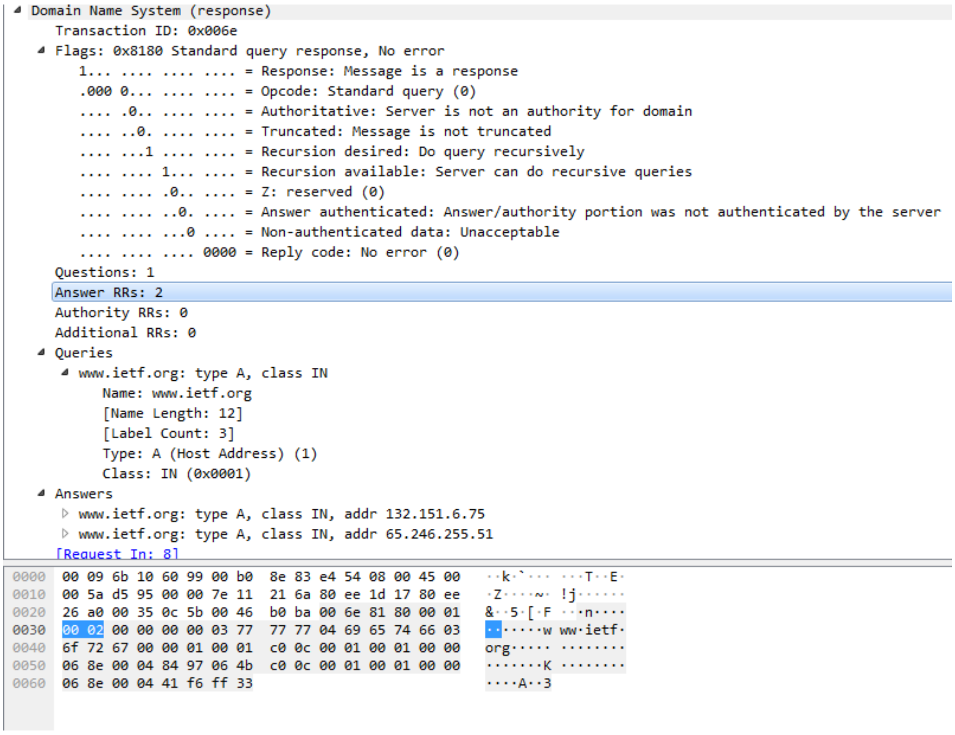


```
◢ Domain Name System (query)
    Transaction ID: 0x006e
  ◢ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ◢ Queries
    ◢ www.ietf.org: type A, class IN
        Name: www.ietf.org
        [Name Length: 12]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 9]
```

```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 3a 22 9e 00 00 80 11  d2 81 80 ee 26 a0 80 ee   ·:"····· ····&···
0020  1d 17 0c 5b 00 35 00 26  8a cb 00 6e 01 00 00 01   ···[·5·& ···n····
0030  00 00 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01                            org·····
```

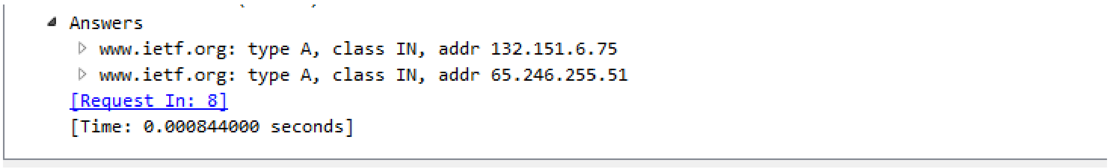
Being a DNS the query is for the web address www.ietf.org and getting its IP.
Before the DNS response message is sent back the message does not have any answers.

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

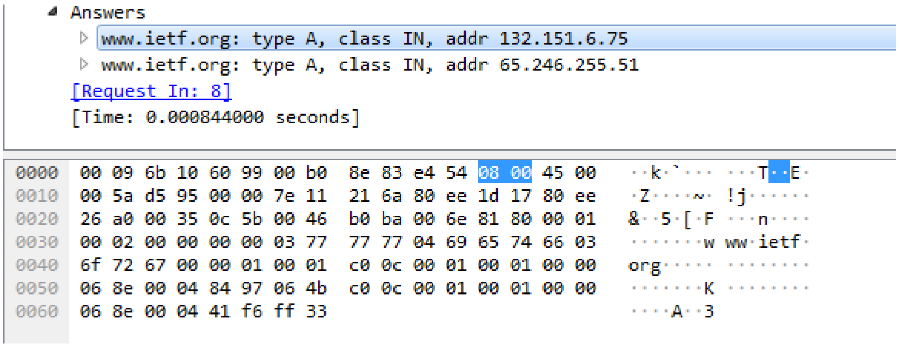


```
◢ Domain Name System (response)
    Transaction ID: 0x006e
  ◢ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for 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)
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ◢ Queries
    ◢ www.ietf.org: type A, class IN
        Name: www.ietf.org
        [Name Length: 12]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ◢ Answers
    ▷ www.ietf.org: type A, class IN, addr 132.151.6.75
    ▷ www.ietf.org: type A, class IN, addr 65.246.255.51
    [Request In: 8]
```

```
0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~· !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org····· ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```

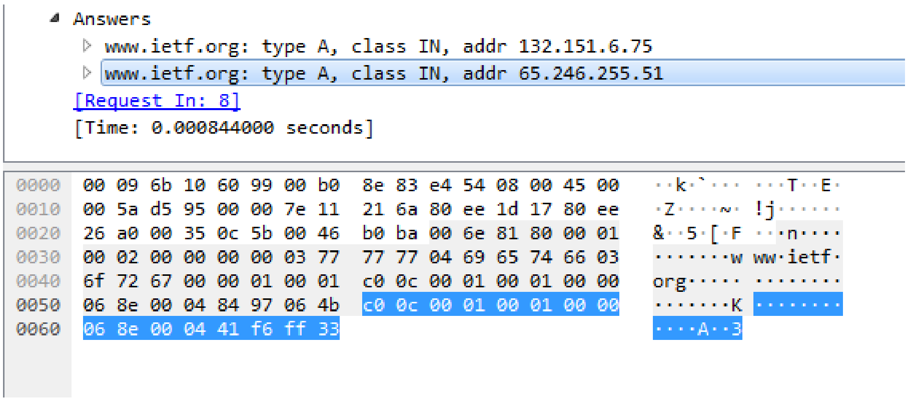


```
  ◢ Answers
    ▷ www.ietf.org: type A, class IN, addr 132.151.6.75
    ▷ www.ietf.org: type A, class IN, addr 65.246.255.51
    [Request In: 8]
    [Time: 0.000844000 seconds]
```



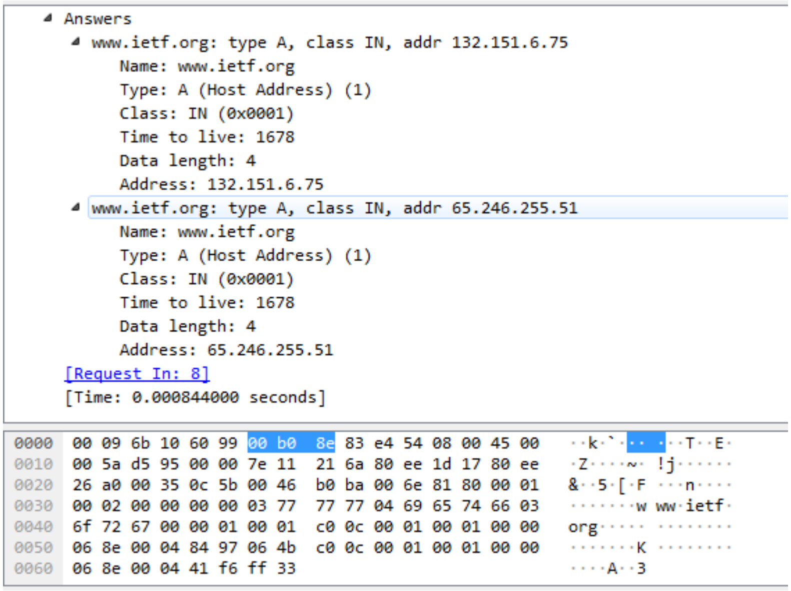
```
  ◢ Answers
    ▷ www.ietf.org: type A, class IN, addr 132.151.6.75
    ▷ www.ietf.org: type A, class IN, addr 65.246.255.51
    [Request In: 8]
    [Time: 0.000844000 seconds]
```

```
0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~· !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org····· ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```



```
  ◢ Answers
    ▷ www.ietf.org: type A, class IN, addr 132.151.6.75
    ▷ www.ietf.org: type A, class IN, addr 65.246.255.51
    [Request In: 8]
    [Time: 0.000844000 seconds]
```

```
0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~· !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org····· ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```

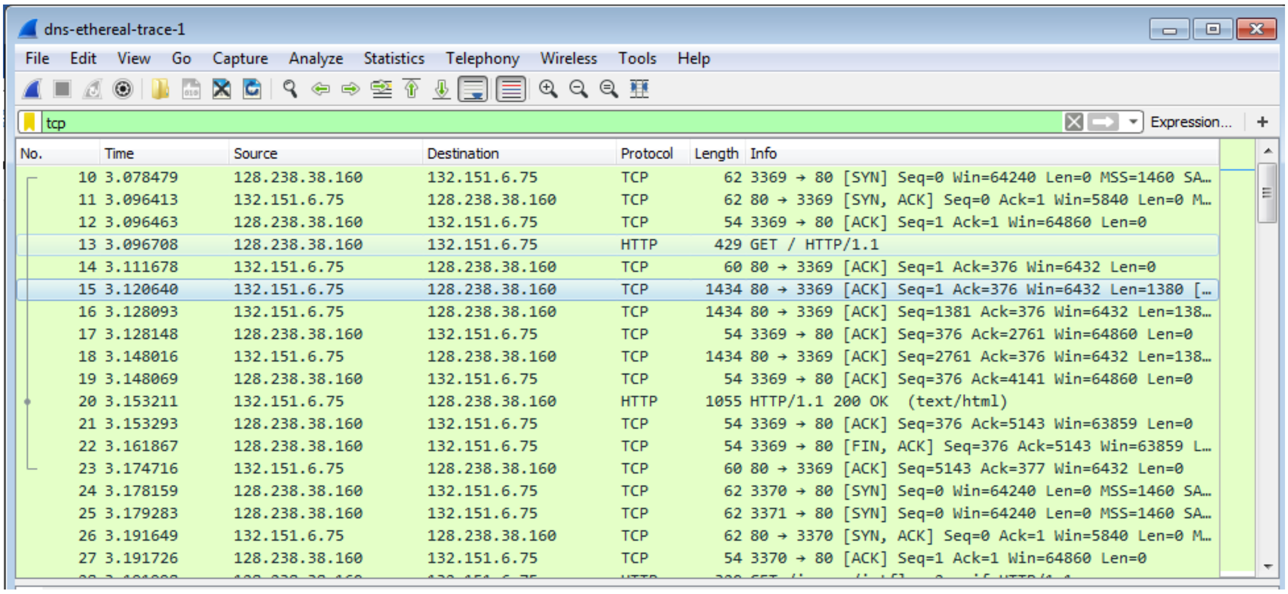


```
◢ Answers
   ◢ www.ietf.org: type A, class IN, addr 132.151.6.75
        Name: www.ietf.org
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 1678
        Data length: 4
        Address: 132.151.6.75
   ◢ www.ietf.org: type A, class IN, addr 65.246.255.51
        Name: www.ietf.org
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 1678
        Data length: 4
        Address: 65.246.255.51
  [Request In: 8]
  [Time: 0.000844000 seconds]
```

```
0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`···  ···T··E·
0010  00 5a d5 95 00 00 7e 11  21 6a 80 ee 1d 17 80 ee   ·Z····~·  !j······
0020  26 a0 00 35 0c 5b 00 46  b0 ba 00 6e 81 80 00 01   &··5·[·F  ···n····
0030  00 02 00 00 00 00 03 77  77 77 04 69 65 74 66 03   ·······w  ww·ietf·
0040  6f 72 67 00 00 01 00 01  c0 0c 00 01 00 01 00 00   org·····  ········
0050  06 8e 00 04 84 97 06 4b  c0 0c 00 01 00 01 00 00   ·······K  ········
0060  06 8e 00 04 41 f6 ff 33                            ····A··3
```

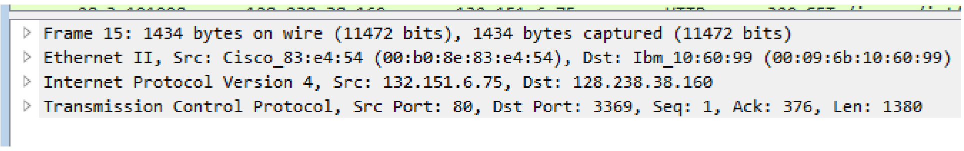
Examining the response message, it is a standard query response as in a response to the query that was sent. It has two answers rather than one. The answers which were sent as the response message each contain the name of the host, an IP address and the type which is A as it is a host address not a query for names and IP addresses of a name server. They also contain the time the message was captured live and the length of the data. The two IP addresses sent as a response were provided using other protocols like HTTP and TCP since they are not the same as the addresses in the DNS query and response messages.

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?



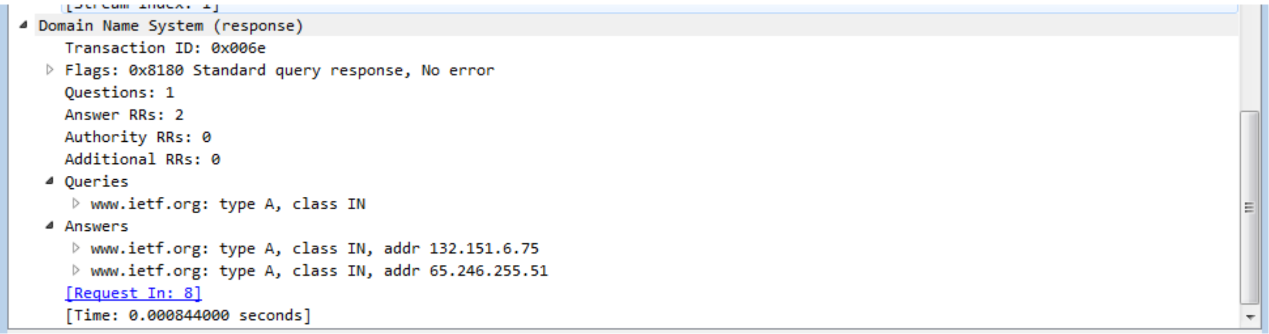
tcp

```
No.     Time        Source            Destination       Protocol  Length  Info
   10 3.078479    128.238.38.160    132.151.6.75      TCP         62 3369 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA…
   11 3.096413    132.151.6.75      128.238.38.160    TCP         62 80 → 3369 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M…
   12 3.096463    128.238.38.160    132.151.6.75      TCP         54 3369 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
   13 3.096708    128.238.38.160    132.151.6.75      HTTP       429 GET / HTTP/1.1
   14 3.111678    132.151.6.75      128.238.38.160    TCP         60 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=0
   15 3.120640    132.151.6.75      128.238.38.160    TCP       1434 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=1380 […
   16 3.128093    132.151.6.75      128.238.38.160    TCP       1434 80 → 3369 [ACK] Seq=1381 Ack=376 Win=6432 Len=138…
   17 3.128148    128.238.38.160    132.151.6.75      TCP         54 3369 → 80 [ACK] Seq=376 Ack=2761 Win=64860 Len=0
   18 3.148016    132.151.6.75      128.238.38.160    TCP       1434 80 → 3369 [ACK] Seq=2761 Ack=376 Win=6432 Len=138…
   19 3.148069    128.238.38.160    132.151.6.75      TCP         54 3369 → 80 [ACK] Seq=376 Ack=4141 Win=64860 Len=0
   20 3.153211    132.151.6.75      128.238.38.160    HTTP      1055 HTTP/1.1 200 OK  (text/html)
   21 3.153293    128.238.38.160    132.151.6.75      TCP         54 3369 → 80 [ACK] Seq=376 Ack=5143 Win=63859 Len=0
   22 3.161867    128.238.38.160    132.151.6.75      TCP         54 3369 → 80 [FIN, ACK] Seq=376 Ack=5143 Win=63859 L…
   23 3.174716    132.151.6.75      128.238.38.160    TCP         60 80 → 3369 [ACK] Seq=5143 Ack=377 Win=6432 Len=0
   24 3.178159    128.238.38.160    132.151.6.75      TCP         62 3370 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA…
   25 3.179283    128.238.38.160    132.151.6.75      TCP         62 3371 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA…
   26 3.191649    132.151.6.75      128.238.38.160    TCP         62 80 → 3370 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M…
   27 3.191726    128.238.38.160    132.151.6.75      TCP         54 3370 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
```

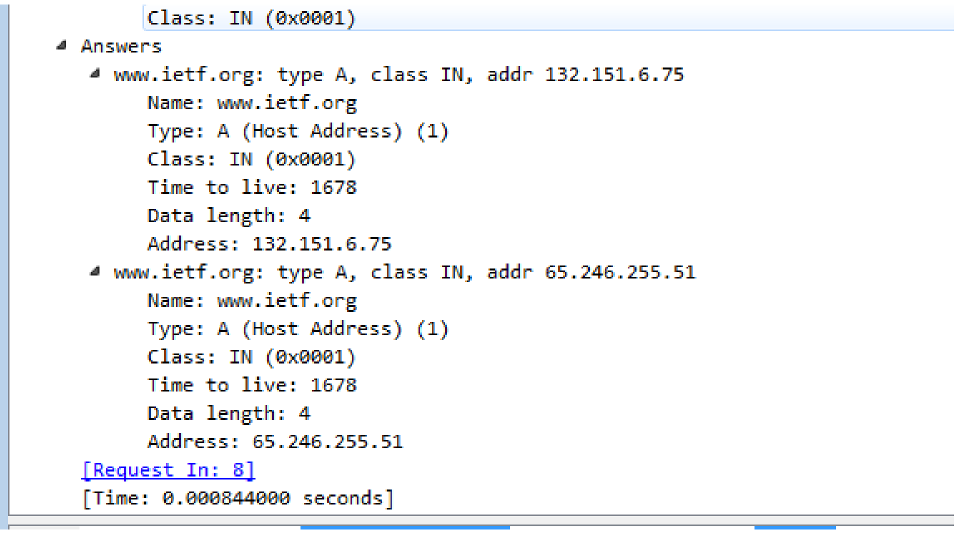


```
▷ Frame 15: 1434 bytes on wire (11472 bits), 1434 bytes captured (11472 bits)
▷ Ethernet II, Src: Cisco_83:e4:54 (00:b0:8e:83:e4:54), Dst: Ibm_10:60:99 (00:09:6b:10:60:99)
▷ Internet Protocol Version 4, Src: 132.151.6.75, Dst: 128.238.38.160
▷ Transmission Control Protocol, Src Port: 80, Dst Port: 3369, Seq: 1, Ack: 376, Len: 1380
```

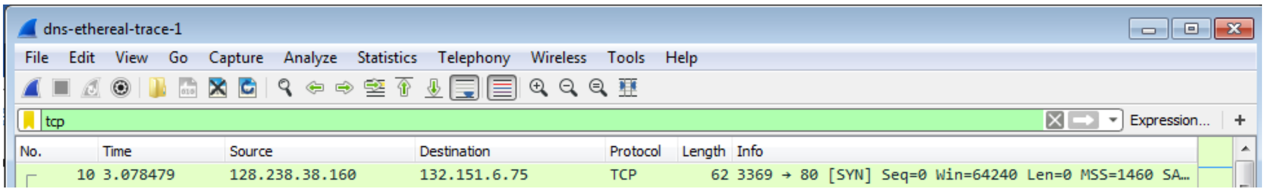
Considering the TCP SYN packet sent by the host here which means has been sent to and back in a sequence using a sequence number as shown in the info column. The protocols used for this are the TCP (Transfer Control Protocol) and HTTP (Hyper Text Transfer Protocol).



```
    [Stream index: 1]
◢ Domain Name System (response)
    Transaction ID: 0x006e
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ◢ Queries
    ▷ www.ietf.org: type A, class IN
  ◢ Answers
    ▷ www.ietf.org: type A, class IN, addr 132.151.6.75
    ▷ www.ietf.org: type A, class IN, addr 65.246.255.51
    [Request In: 8]
    [Time: 0.000844000 seconds]
```



```
          Class: IN (0x0001)
  ◢ Answers
    ◢ www.ietf.org: type A, class IN, addr 132.151.6.75
          Name: www.ietf.org
          Type: A (Host Address) (1)
          Class: IN (0x0001)
          Time to live: 1678
          Data length: 4
          Address: 132.151.6.75
    ◢ www.ietf.org: type A, class IN, addr 65.246.255.51
          Name: www.ietf.org
          Type: A (Host Address) (1)
          Class: IN (0x0001)
          Time to live: 1678
          Data length: 4
          Address: 65.246.255.51
    [Request In: 8]
    [Time: 0.000844000 seconds]
```

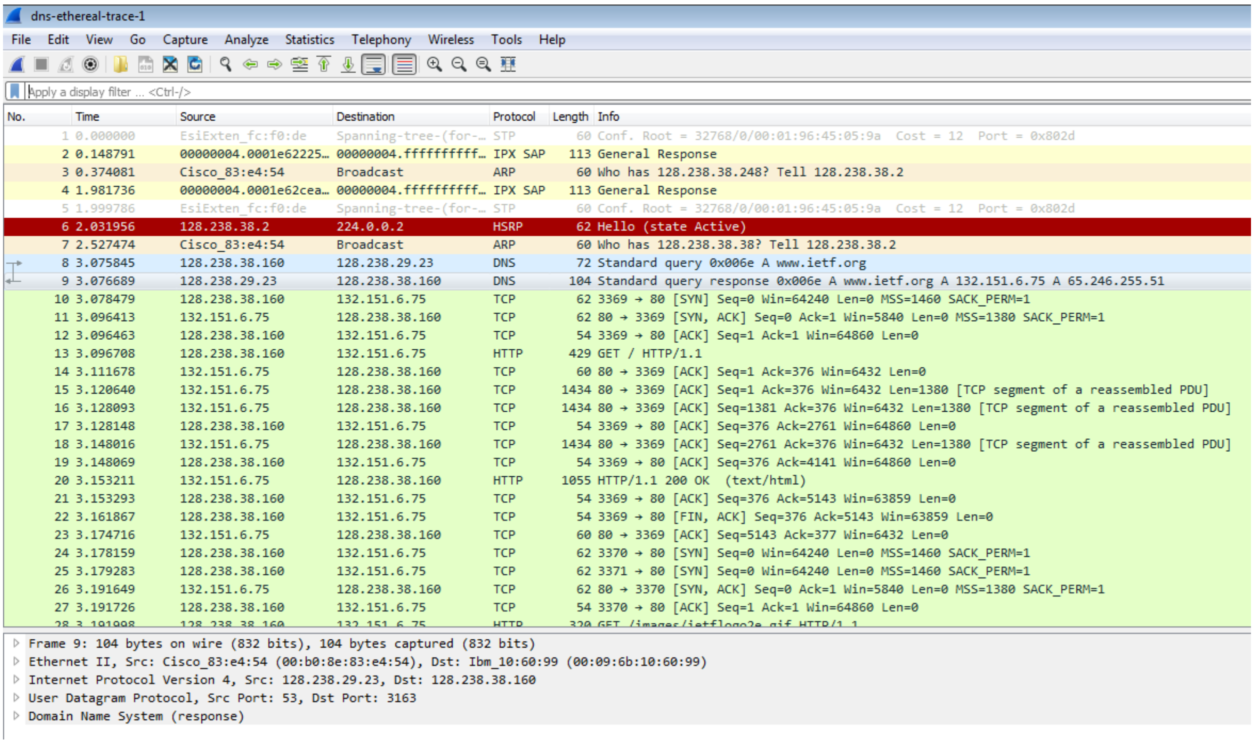


tcp

No.	Time	Source	Destination	Protocol	Length	Info
10	3.078479	128.238.38.160	132.151.6.75	TCP	62	3369 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SA…

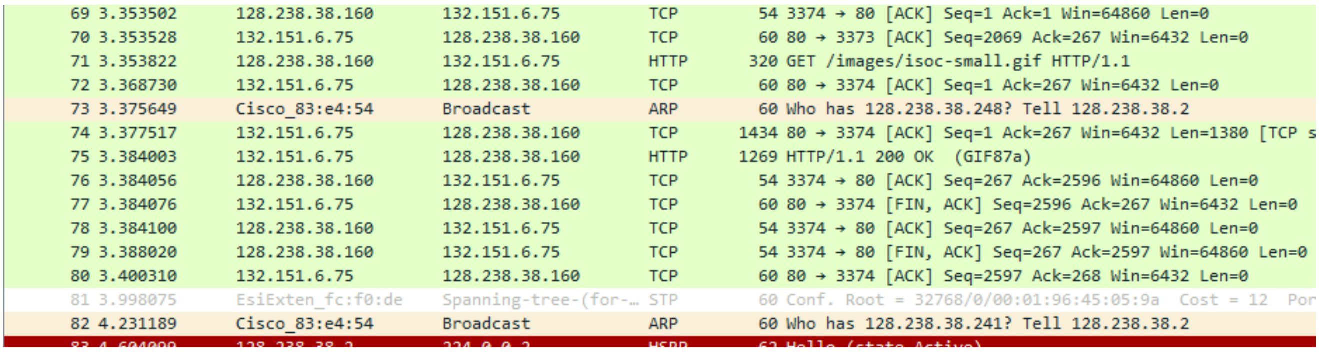
The destination IP address of the first SYN packet sent which is 132.151.6.75 is the same IP address, provided in the DNS response message, of the name server for the webpage address www.ietf.org: type A, class IN, shown above.

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


dns-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time      Source              Destination         Protocol Length Info
   1 0.000000  EsiExten_fc:f0:de   Spanning-tree-(for-… STP       60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Port = 0x802d
   2 0.148791  00000004.0001e62225… 00000004.ffffffffff… IPX SAP  113 General Response
   3 0.374081  Cisco_83:e4:54      Broadcast           ARP       60 Who has 128.238.38.248? Tell 128.238.38.2
   4 1.981736  00000004.0001e62cea… 00000004.ffffffffff… IPX SAP  113 General Response
   5 1.999786  EsiExten_fc:f0:de   Spanning-tree-(for-… STP       60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Port = 0x802d
   6 2.031956  128.238.38.2        224.0.0.2           HSRP      62 Hello (state Active)
   7 2.527474  Cisco_83:e4:54      Broadcast           ARP       60 Who has 128.238.38.38? Tell 128.238.38.2
   8 3.075845  128.238.38.160      128.238.29.23       DNS       72 Standard query 0x006e A www.ietf.org
   9 3.076689  128.238.29.23       128.238.38.160      DNS      104 Standard query response 0x006e A www.ietf.org A 132.151.6.75 A 65.246.255.51
  10 3.078479  128.238.38.160      132.151.6.75        TCP       62 3369 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
  11 3.096413  132.151.6.75        128.238.38.160      TCP       62 80 → 3369 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM=1
  12 3.096463  128.238.38.160      132.151.6.75        TCP       54 3369 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
  13 3.096708  128.238.38.160      132.151.6.75        HTTP     429 GET / HTTP/1.1
  14 3.111678  132.151.6.75        128.238.38.160      TCP       60 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=0
  15 3.120640  132.151.6.75        128.238.38.160      TCP     1434 80 → 3369 [ACK] Seq=1 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
  16 3.128093  132.151.6.75        128.238.38.160      TCP     1434 80 → 3369 [ACK] Seq=1381 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
  17 3.128148  128.238.38.160      132.151.6.75        TCP       54 3369 → 80 [ACK] Seq=376 Ack=2761 Win=64860 Len=0
  18 3.148016  132.151.6.75        128.238.38.160      TCP     1434 80 → 3369 [ACK] Seq=2761 Ack=376 Win=6432 Len=1380 [TCP segment of a reassembled PDU]
  19 3.148069  128.238.38.160      132.151.6.75        TCP       54 3369 → 80 [ACK] Seq=376 Ack=4141 Win=64860 Len=0
  20 3.153211  132.151.6.75        128.238.38.160      HTTP    1055 HTTP/1.1 200 OK  (text/html)
  21 3.153293  128.238.38.160      132.151.6.75        TCP       54 3369 → 80 [ACK] Seq=376 Ack=5143 Win=63859 Len=0
  22 3.161867  128.238.38.160      132.151.6.75        TCP       54 3369 → 80 [FIN, ACK] Seq=376 Ack=5143 Win=63859 Len=0
  23 3.174716  132.151.6.75        128.238.38.160      TCP       60 80 → 3369 [ACK] Seq=5143 Ack=377 Win=6432 Len=0
  24 3.178159  128.238.38.160      132.151.6.75        TCP       62 3370 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
  25 3.179283  128.238.38.160      132.151.6.75        TCP       62 3371 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
  26 3.191649  132.151.6.75        128.238.38.160      TCP       62 80 → 3370 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK_PERM=1
  27 3.191726  128.238.38.160      132.151.6.75        TCP       54 3370 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
  28 3.191998  128.238.38.160      132.151.6.75        HTTP     320 GET /images/ietflogo2e.gif HTTP/1.1

▷ Frame 9: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
▷ Ethernet II, Src: Cisco_83:e4:54 (00:b0:8e:83:e4:54), Dst: Ibm_10:60:99 (00:09:6b:10:60:99)
▷ Internet Protocol Version 4, Src: 128.238.29.23, Dst: 128.238.38.160
▷ User Datagram Protocol, Src Port: 53, Dst Port: 3163
▷ Domain Name System (response)
```

The DNS query and response messages are listed above the messages that use the TCP and HTTP protocols to get the images. So yes the host does issue new DNS queries before retrieving each image.



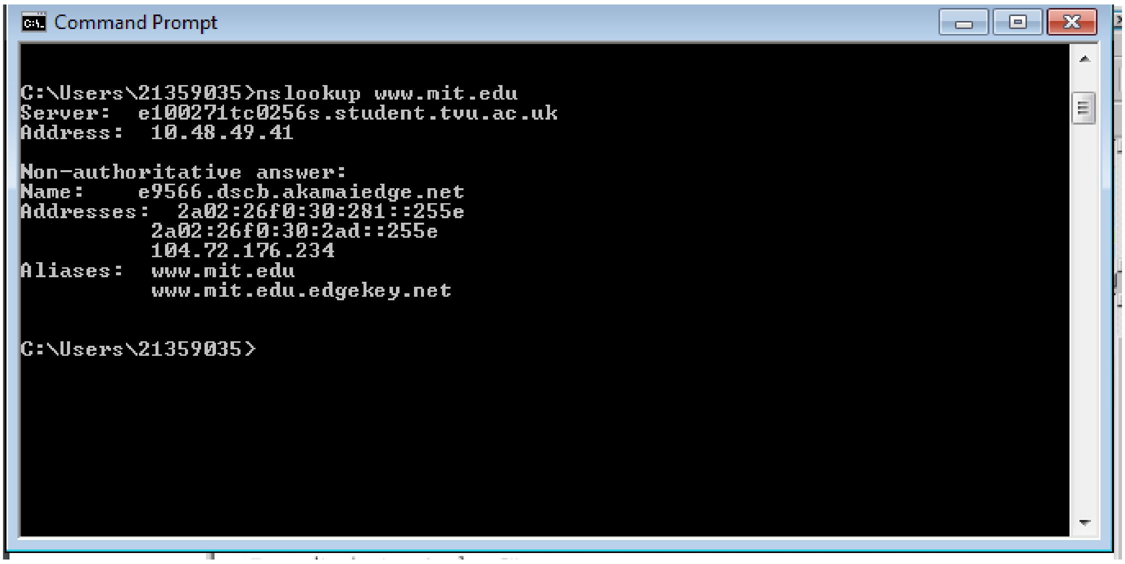
```
  69 3.353502  128.238.38.160      132.151.6.75        TCP       54 3374 → 80 [ACK] Seq=1 Ack=1 Win=64860 Len=0
  70 3.353528  132.151.6.75        128.238.38.160      TCP       60 80 → 3373 [ACK] Seq=2069 Ack=267 Win=6432 Len=0
  71 3.353822  128.238.38.160      132.151.6.75        HTTP     320 GET /images/isoc-small.gif HTTP/1.1
  72 3.368730  132.151.6.75        128.238.38.160      TCP       60 80 → 3374 [ACK] Seq=1 Ack=267 Win=6432 Len=0
  73 3.375649  Cisco_83:e4:54      Broadcast           ARP       60 Who has 128.238.38.248? Tell 128.238.38.2
  74 3.377517  132.151.6.75        128.238.38.160      TCP     1434 80 → 3374 [ACK] Seq=1 Ack=267 Win=6432 Len=1380 [TCP s
  75 3.384003  132.151.6.75        128.238.38.160      HTTP    1269 HTTP/1.1 200 OK  (GIF87a)
  76 3.384056  128.238.38.160      132.151.6.75        TCP       54 3374 → 80 [ACK] Seq=267 Ack=2596 Win=64860 Len=0
  77 3.384076  132.151.6.75        128.238.38.160      TCP       60 80 → 3374 [FIN, ACK] Seq=2596 Ack=267 Win=6432 Len=0
  78 3.384100  128.238.38.160      132.151.6.75        TCP       54 3374 → 80 [ACK] Seq=267 Ack=2597 Win=64860 Len=0
  79 3.388020  128.238.38.160      132.151.6.75        TCP       54 3374 → 80 [FIN, ACK] Seq=267 Ack=2597 Win=64860 Len=0
  80 3.400310  132.151.6.75        128.238.38.160      TCP       60 80 → 3374 [ACK] Seq=2597 Ack=268 Win=6432 Len=0
  81 3.998075  EsiExten_fc:f0:de   Spanning-tree-(for-… STP       60 Conf. Root = 32768/0/00:01:96:45:05:9a  Cost = 12  Por
  82 4.231189  Cisco_83:e4:54      Broadcast           ARP       60 Who has 128.238.38.241? Tell 128.238.38.2
  83 4.604099  128.238.38.2        224.0.0.2           HSRP      62 Hello (state Active)
```

You can also see above the word get in capitals, under the Info column, which is the method used to get an image loaded onto the webpage.

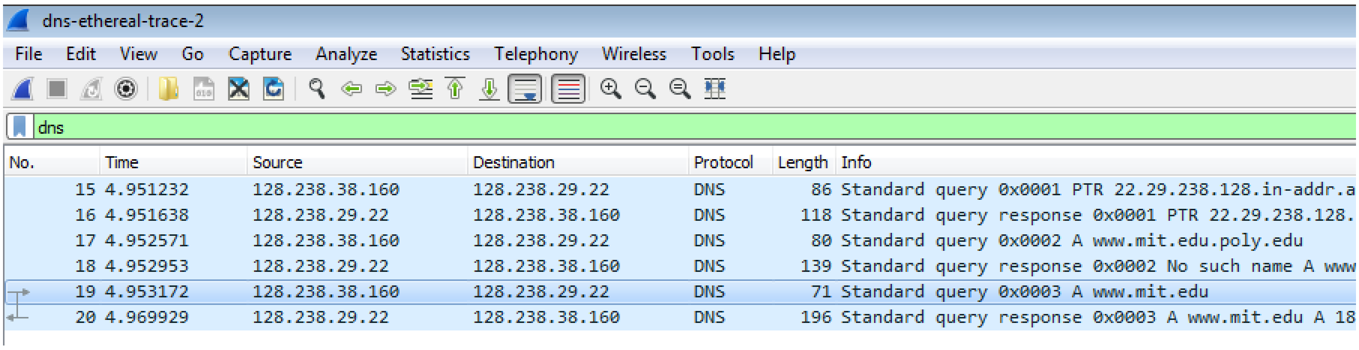
Now I play with nslookup:

Doing an nslookup on www.mit.edu

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

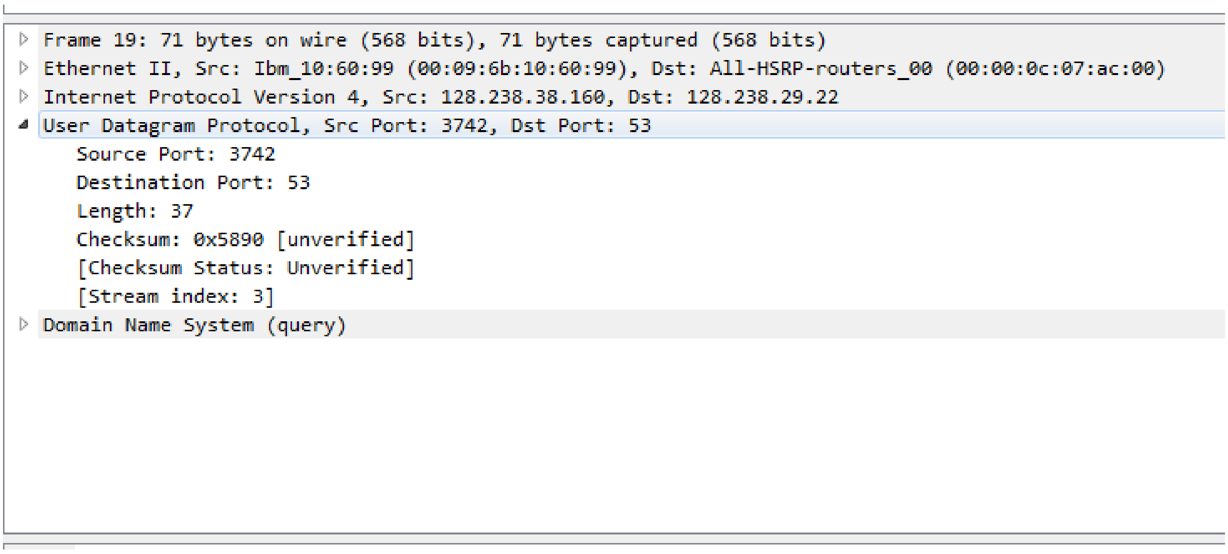


```
C:\Users\21359035>nslookup www.mit.edu
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
Name:    e9566.dscb.akamaiedge.net
Addresses:  2a02:26f0:30:281::255e
          2a02:26f0:30:2ad::255e
          104.72.176.234
Aliases:  www.mit.edu
          www.mit.edu.edgekey.net


C:\Users\21359035>
```



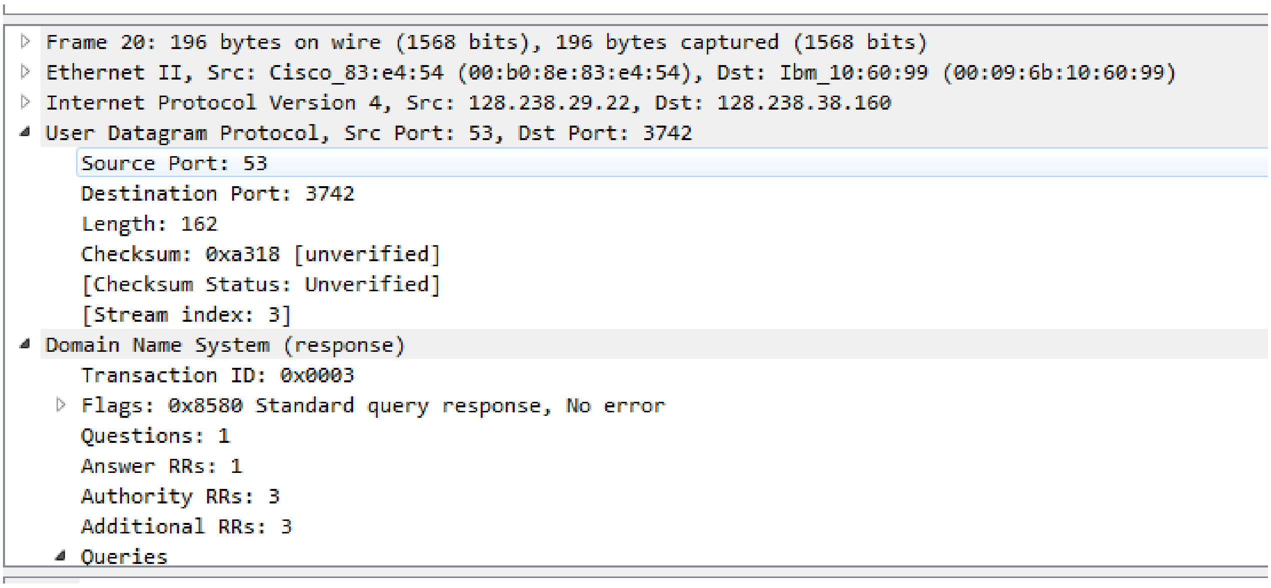
dns

```
No.  Time      Source          Destination     Protocol Length Info
  15 4.951232  128.238.38.160  128.238.29.22   DNS       86 Standard query 0x0001 PTR 22.29.238.128.in-addr.a
  16 4.951638  128.238.29.22   128.238.38.160  DNS      118 Standard query response 0x0001 PTR 22.29.238.128.
  17 4.952571  128.238.38.160  128.238.29.22   DNS       80 Standard query 0x0002 A www.mit.edu.poly.edu
  18 4.952953  128.238.29.22   128.238.38.160  DNS      139 Standard query response 0x0002 No such name A www
  19 4.953172  128.238.38.160  128.238.29.22   DNS       71 Standard query 0x0003 A www.mit.edu
  20 4.969929  128.238.29.22   128.238.38.160  DNS      196 Standard query response 0x0003 A www.mit.edu A 18
```



```
Frame 19: 71 bytes on wire (568 bits), 71 bytes captured (568 bits)
Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.22
User Datagram Protocol, Src Port: 3742, Dst Port: 53
    Source Port: 3742
    Destination Port: 53
    Length: 37
    Checksum: 0x5890 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
Domain Name System (query)
```

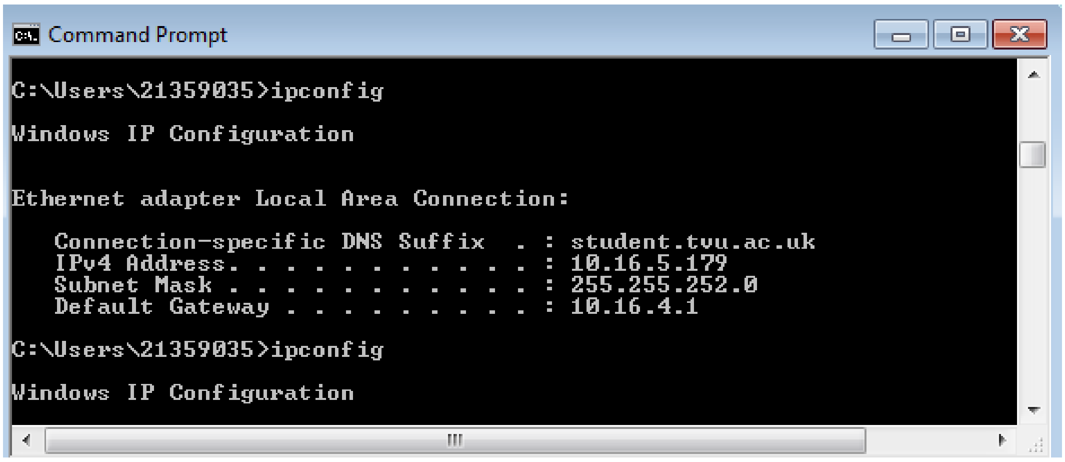
The destination port for the query message is 53.



```
Frame 20: 196 bytes on wire (1568 bits), 196 bytes captured (1568 bits)
Ethernet II, Src: Cisco_83:e4:54 (00:b0:8e:83:e4:54), Dst: Ibm_10:60:99 (00:09:6b:10:60:99)
Internet Protocol Version 4, Src: 128.238.29.22, Dst: 128.238.38.160
User Datagram Protocol, Src Port: 53, Dst Port: 3742
    Source Port: 53
    Destination Port: 3742
    Length: 162
    Checksum: 0xa318 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 3]
Domain Name System (response)
    Transaction ID: 0x0003
  Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 3
  Queries
```

The source port for the response message is 53.

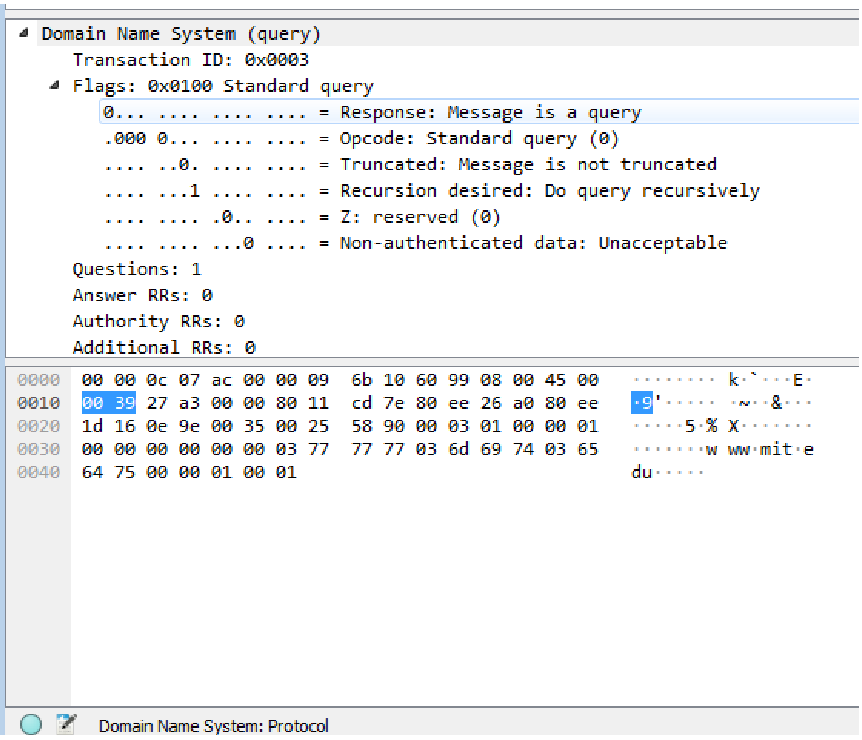
12. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?



```
C:\Users\21359035>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : student.tvu.ac.uk
   IPv4 Address. . . . . . . . . . . : 10.16.5.179
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 10.16.4.1

C:\Users\21359035>ipconfig

Windows IP Configuration
```

The DNS query message is sent to IP address 128.238.29.22. Not the IP address of my local default DNS server since this is only a trace file that was captured using a computer at a different location to the one I am currently using this computer in.

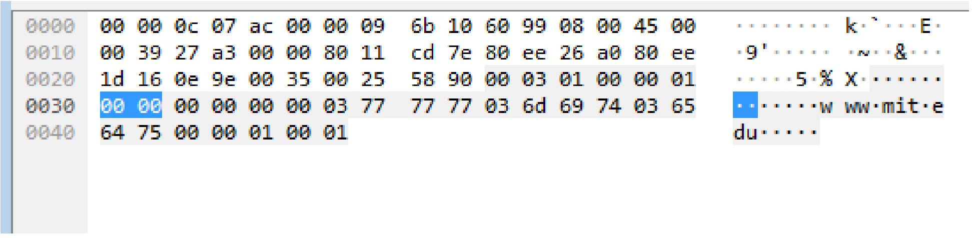
13. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?



```
◢ Domain Name System (query)
    Transaction ID: 0x0003
  ◢ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0

0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 39 27 a3 00 00 80 11  cd 7e 80 ee 26 a0 80 ee   ·9'····· ·~··&···
0020  1d 16 0e 9e 00 35 00 25  58 90 00 03 01 00 00 01   ·····5·% X·······
0030  00 00 00 00 00 00 03 77  77 77 03 6d 69 74 03 65   ·······w ww·mit·e
0040  64 75 00 00 01 00 01                               du·····
```

Domain Name System: Protocol

It is a standard type AAA query. It has no answers since it is the query before the response message is sent.



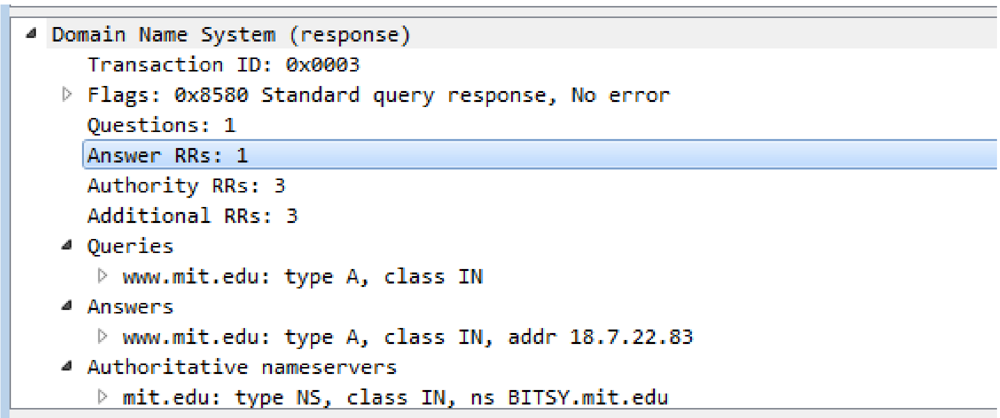
```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 39 27 a3 00 00 80 11  cd 7e 80 ee 26 a0 80 ee   ·9'····· ·~··&···
0020  1d 16 0e 9e 00 35 00 25  58 90 00 03 01 00 00 01   ·····5·% X·······
0030  00 00 00 00 00 00 03 77  77 77 03 6d 69 74 03 65   ·······w ww·mit·e
0040  64 75 00 00 01 00 01                               du·····
```

Shown above are details of the packet in hexadecimal and ASCII. No answers mean zeroes are shown.

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

One answer is provided. It contains details of the name server for www.mit.edu. This includes (which I forgot to show) the name of the website, time taken to capture the packet during the live capture and the data length.

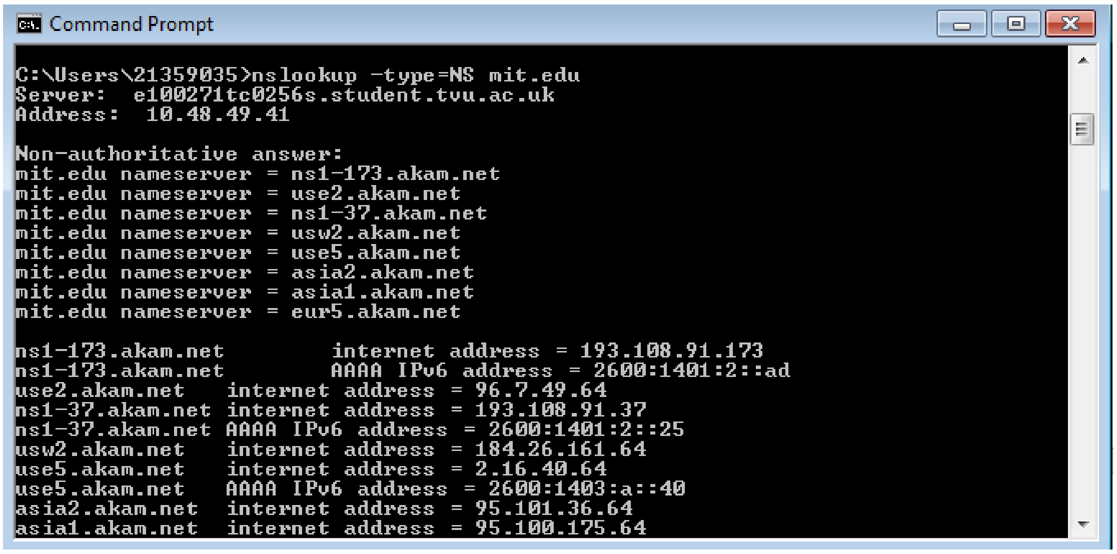
15. Screenshots



```
◢ Domain Name System (response)
    Transaction ID: 0x0003
  ▷ Flags: 0x8580 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 3
    Additional RRs: 3
  ◢ Queries
    ▷ www.mit.edu: type A, class IN
  ◢ Answers
    ▷ www.mit.edu: type A, class IN, addr 18.7.22.83
  ◢ Authoritative nameservers
    ▷ mit.edu: type NS, class IN, ns BITSY.mit.edu
```

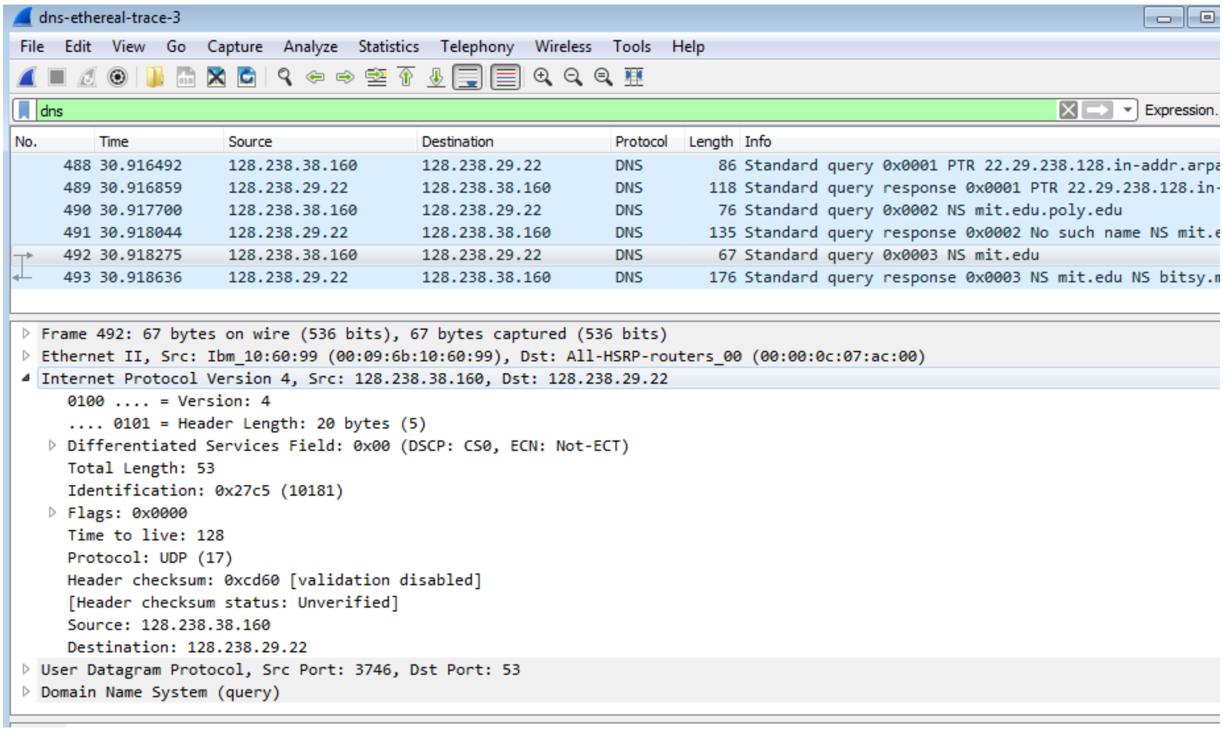
I now run this command:

nslookup –type=NS mit.edu



```
C:\Users\21359035>nslookup -type=NS mit.edu
Server:  e100271tc0256s.student.tvu.ac.uk
Address:  10.48.49.41

Non-authoritative answer:
mit.edu nameserver = ns1-173.akam.net
mit.edu nameserver = use2.akam.net
mit.edu nameserver = ns1-37.akam.net
mit.edu nameserver = usw2.akam.net
mit.edu nameserver = use5.akam.net
mit.edu nameserver = asia2.akam.net
mit.edu nameserver = asia1.akam.net
mit.edu nameserver = eur5.akam.net

ns1-173.akam.net        internet address = 193.108.91.173
ns1-173.akam.net        AAAA IPv6 address = 2600:1401:2::ad
use2.akam.net   internet address = 96.7.49.64
ns1-37.akam.net internet address = 193.108.91.37
ns1-37.akam.net AAAA IPv6 address = 2600:1401:2::25
usw2.akam.net   internet address = 184.26.161.64
use5.akam.net   internet address = 2.16.40.64
use5.akam.net   AAAA IPv6 address = 2600:1403:a::40
asia2.akam.net  internet address = 95.101.36.64
asia1.akam.net  internet address = 95.100.175.64
```

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

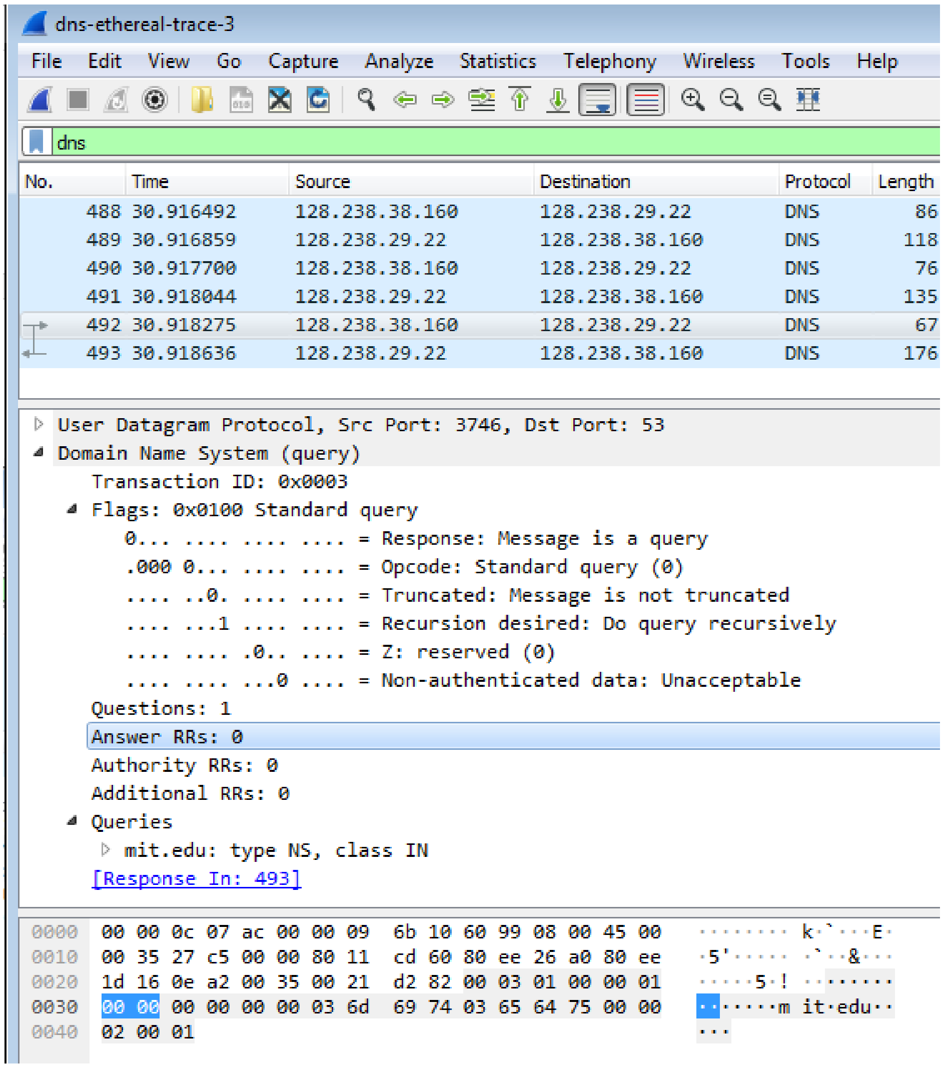

dns-ethereal-trace-3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
Expression.

```
No.     Time       Source           Destination      Protocol  Length Info
    488 30.916492  128.238.38.160   128.238.29.22    DNS          86 Standard query 0x0001 PTR 22.29.238.128.in-addr.arpa
    489 30.916859  128.238.29.22    128.238.38.160   DNS         118 Standard query response 0x0001 PTR 22.29.238.128.in-
    490 30.917700  128.238.38.160   128.238.29.22    DNS          76 Standard query 0x0002 NS mit.edu.poly.edu
    491 30.918044  128.238.29.22    128.238.38.160   DNS         135 Standard query response 0x0002 No such name NS mit.e
    492 30.918275  128.238.38.160   128.238.29.22    DNS          67 Standard query 0x0003 NS mit.edu
    493 30.918636  128.238.29.22    128.238.38.160   DNS         176 Standard query response 0x0003 NS mit.edu NS bitsy.m
```

```
▷ Frame 492: 67 bytes on wire (536 bits), 67 bytes captured (536 bits)
▷ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
◢ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 128.238.29.22
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 53
     Identification: 0x27c5 (10181)
   ▷ Flags: 0x0000
     Time to live: 128
     Protocol: UDP (17)
     Header checksum: 0xcd60 [validation disabled]
     [Header checksum status: Unverified]
     Source: 128.238.38.160
     Destination: 128.238.29.22
▷ User Datagram Protocol, Src Port: 3746, Dst Port: 53
▷ Domain Name System (query)
```

It is sent to IP address 128.238.29.22. Not the IP address of my default local DNS server for the same reasons as mentioned in answer to question 6 and 12.

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


dns-ethereal-trace-3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

```
No.    Time        Source           Destination      Protocol  Length
  488  30.916492   128.238.38.160   128.238.29.22    DNS           86
  489  30.916859   128.238.29.22    128.238.38.160   DNS          118
  490  30.917700   128.238.38.160   128.238.29.22    DNS           76
  491  30.918044   128.238.29.22    128.238.38.160   DNS          135
  492  30.918275   128.238.38.160   128.238.29.22    DNS           67
  493  30.918636   128.238.29.22    128.238.38.160   DNS          176
```

```
▷ User Datagram Protocol, Src Port: 3746, Dst Port: 53
◢ Domain Name System (query)
    Transaction ID: 0x0003
  ◢ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ◢ Queries
    ▷ mit.edu: type NS, class IN
    [Response In: 493]
```

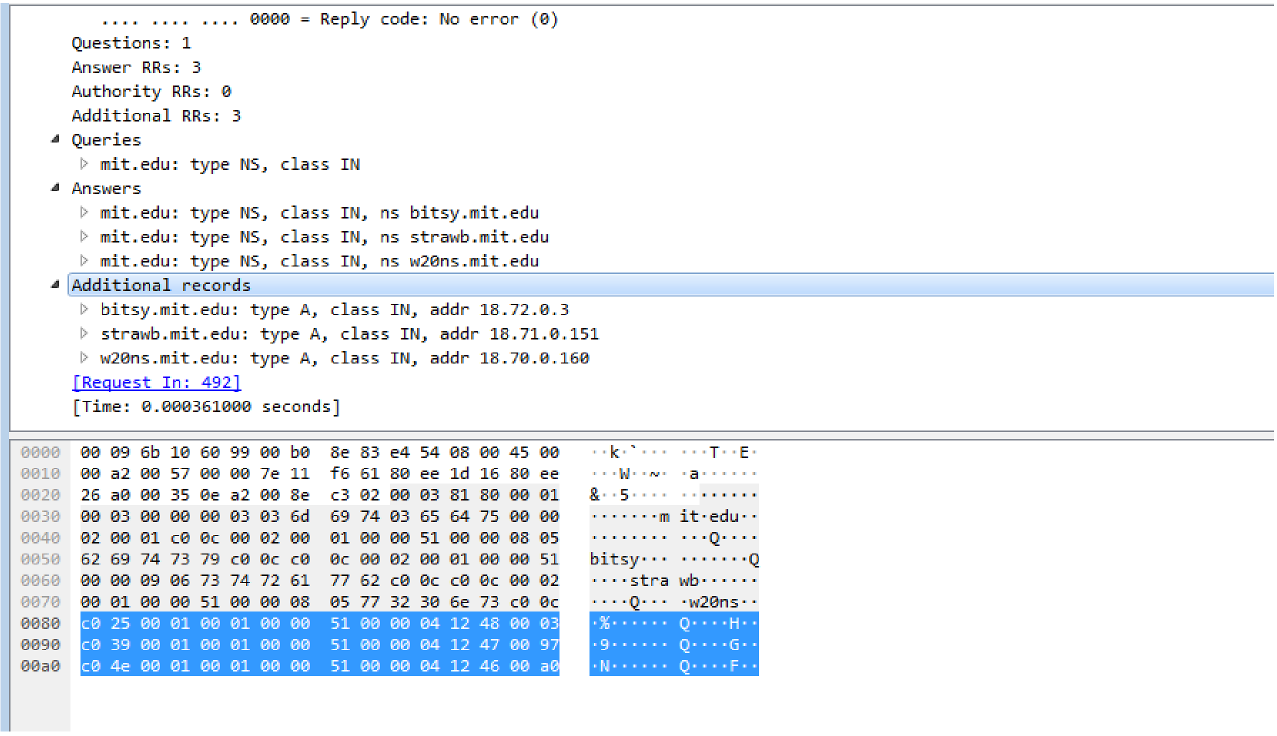
```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········ k·`···E·
0010  00 35 27 c5 00 00 80 11  cd 60 80 ee 26 a0 80 ee   ·5'····· ·`··&···
0020  1d 16 0e a2 00 35 00 21  d2 82 00 03 01 00 00 01   ·····5·! ········
0030  00 00 00 00 00 00 03 6d  69 74 03 65 64 75 00 00   ·······m it·edu··
0040  02 00 01                                            ···
```

The type of DNS query is an NS type (query sent to the server for records of name servers), with no answers.

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

It provides nameservers bitsy.mit.edu(address 18.72.0.3), strawb.mit.edu(address 18.71.0.151) and w20ns.mit.edu(address 18.70.0.160). This response message provides the IP addresses of the MIT name servers as a list of additional records.

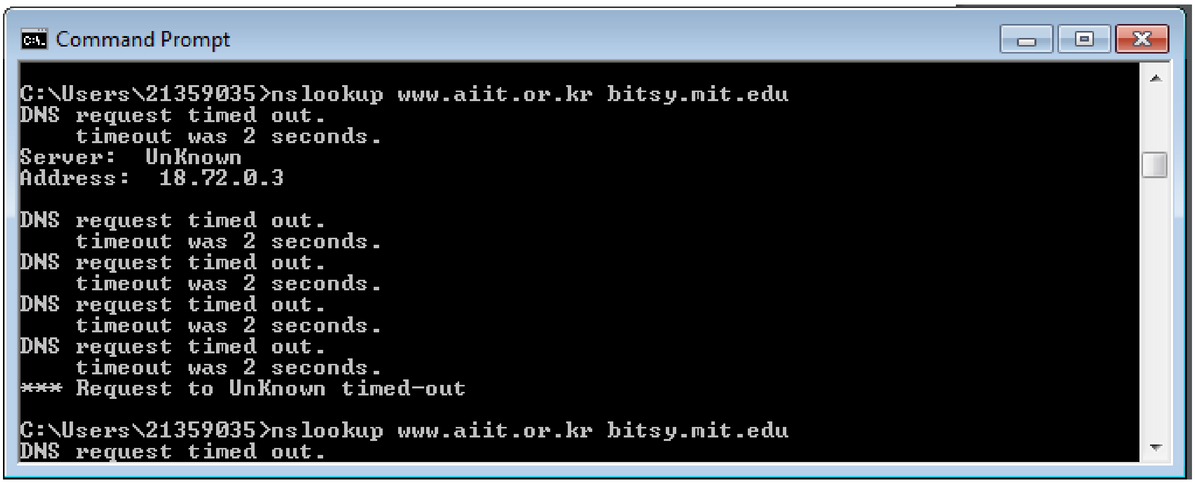
19.Screenshot provided below



```
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 3
    Authority RRs: 0
    Additional RRs: 3
  ◢ Queries
    ▷ mit.edu: type NS, class IN
  ◢ Answers
    ▷ mit.edu: type NS, class IN, ns bitsy.mit.edu
    ▷ mit.edu: type NS, class IN, ns strawb.mit.edu
    ▷ mit.edu: type NS, class IN, ns w20ns.mit.edu
  ◢ Additional records
    ▷ bitsy.mit.edu: type A, class IN, addr 18.72.0.3
    ▷ strawb.mit.edu: type A, class IN, addr 18.71.0.151
    ▷ w20ns.mit.edu: type A, class IN, addr 18.70.0.160
    [Request In: 492]
    [Time: 0.000361000 seconds]

0000  00 09 6b 10 60 99 00 b0  8e 83 e4 54 08 00 45 00   ··k·`··· ···T··E·
0010  00 a2 00 57 00 00 7e 11  f6 61 80 ee 1d 16 80 ee   ···W··~· ·a······
0020  26 a0 00 35 0e a2 00 8e  c3 02 00 03 81 80 00 01   &··5···· ········
0030  00 03 00 00 00 03 03 6d  69 74 03 65 64 75 00 00   ·······m it·edu··
0040  02 00 01 c0 0c 00 02 00  01 00 00 51 00 00 08 05   ········ ···Q····
0050  62 69 74 73 79 c0 0c c0  0c 00 02 00 01 00 00 51   bitsy··· ·······Q
0060  00 00 09 06 73 74 72 61  77 62 c0 0c c0 0c 00 02   ····stra wb······
0070  00 01 00 00 51 00 00 08  05 77 32 30 6e 73 c0 0c   ····Q··· ·w20ns··
0080  c0 25 00 01 00 01 00 00  51 00 00 04 12 48 00 03   ·%······ Q····H··
0090  c0 39 00 01 00 01 00 00  51 00 00 04 12 47 00 97   ·9······ Q····G··
00a0  c0 4e 00 01 00 01 00 00  51 00 00 04 12 46 00 a0   ·N······ Q····F··
```

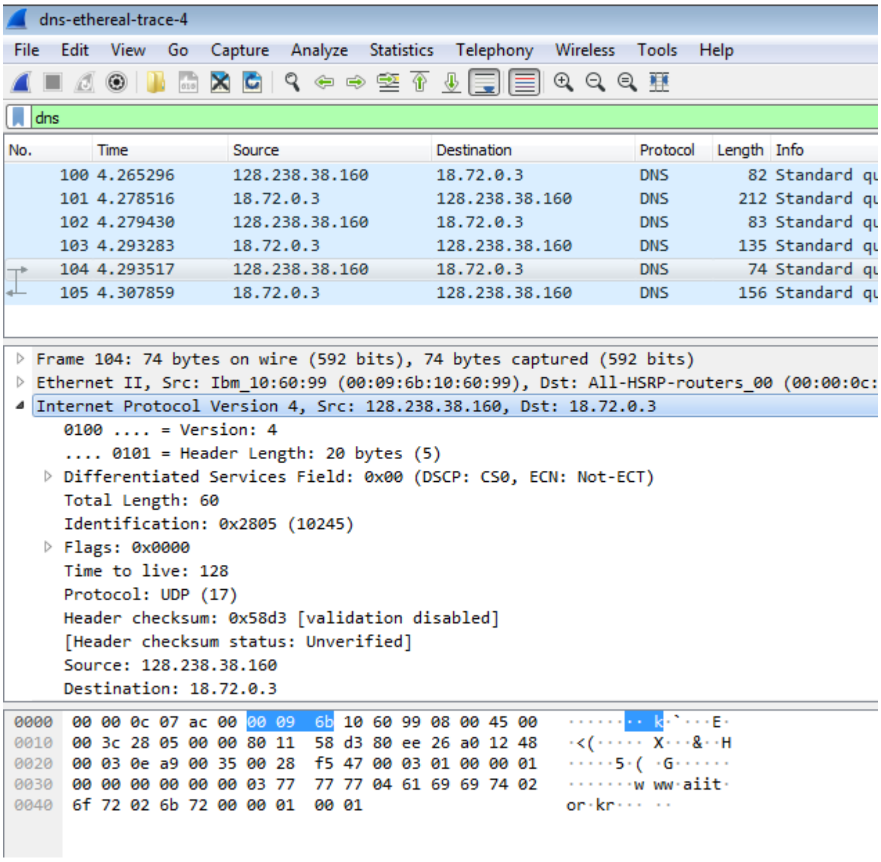
I repeat the previous experiment but with this command:

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



```
C:\Users\21359035>nslookup www.aiit.or.kr bitsy.mit.edu
DNS request timed out.
    timeout was 2 seconds.
Server:  UnKnown
Address:  18.72.0.3

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
*** Request to UnKnown timed-out

C:\Users\21359035>nslookup www.aiit.or.kr bitsy.mit.edu
DNS request timed out.
```

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?



dns-ethereal-trace-4
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

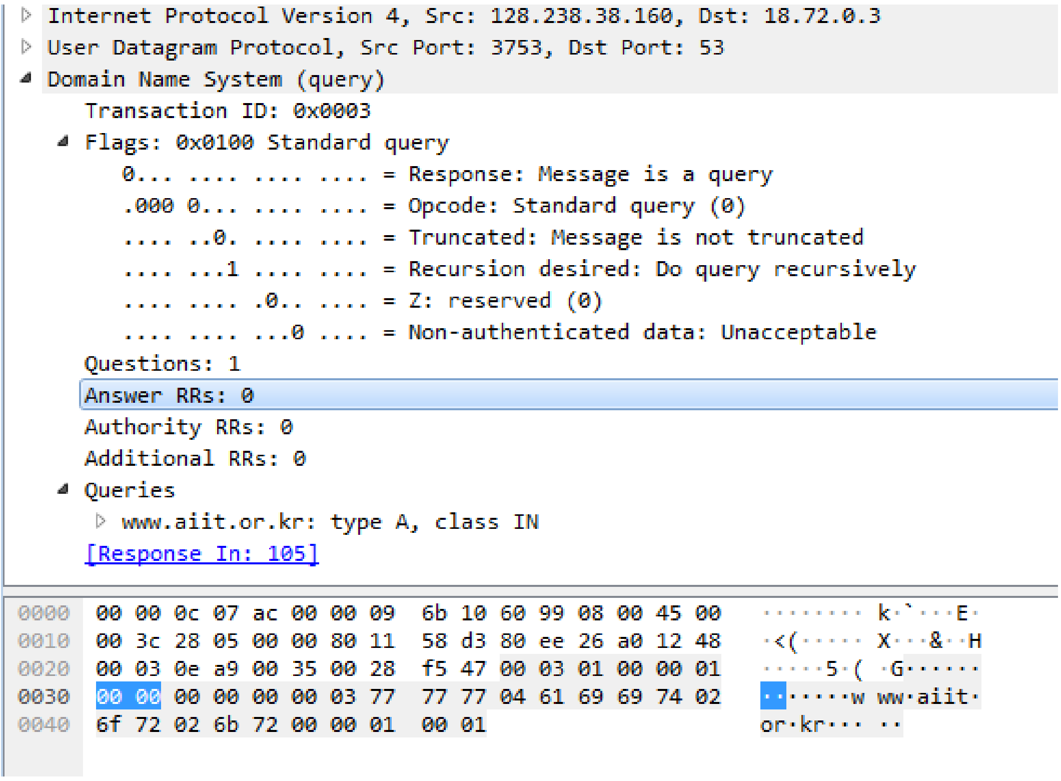
```
No.     Time        Source            Destination       Protocol  Length  Info
    100 4.265296    128.238.38.160    18.72.0.3         DNS           82 Standard qu
    101 4.278516    18.72.0.3         128.238.38.160    DNS          212 Standard qu
    102 4.279430    128.238.38.160    18.72.0.3         DNS           83 Standard qu
    103 4.293283    18.72.0.3         128.238.38.160    DNS          135 Standard qu
    104 4.293517    128.238.38.160    18.72.0.3         DNS           74 Standard qu
    105 4.307859    18.72.0.3         128.238.38.160    DNS          156 Standard qu
```

```
▷ Frame 104: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▷ Ethernet II, Src: Ibm_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:
◢ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 18.72.0.3
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x2805 (10245)
  ▷ Flags: 0x0000
    Time to live: 128
    Protocol: UDP (17)
    Header checksum: 0x58d3 [validation disabled]
    [Header checksum status: Unverified]
    Source: 128.238.38.160
    Destination: 18.72.0.3
```

```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········k·`···E·
0010  00 3c 28 05 00 00 80 11  58 d3 80 ee 26 a0 12 48   ·<(·····X···&··H
0020  00 03 0e a9 00 35 00 28  f5 47 00 03 01 00 00 01   ·····5·(·G······
0030  00 00 00 00 00 00 03 77  77 77 04 61 69 69 74 02   ·······w ww·aiit·
0040  6f 72 02 6b 72 00 00 01  00 01                     or·kr··· ··
```

The DNS query message is sent to IP address 18.72.0.3. This is not the IP address of my local default DNS server. It is the same as name server at mit.edu which is bitsy.mit.edu.

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



```
▷ Internet Protocol Version 4, Src: 128.238.38.160, Dst: 18.72.0.3
▷ User Datagram Protocol, Src Port: 3753, Dst Port: 53
◢ Domain Name System (query)
    Transaction ID: 0x0003
  ◢ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ◢ Queries
    ▷ www.aiit.or.kr: type A, class IN
    [Response In: 105]
```

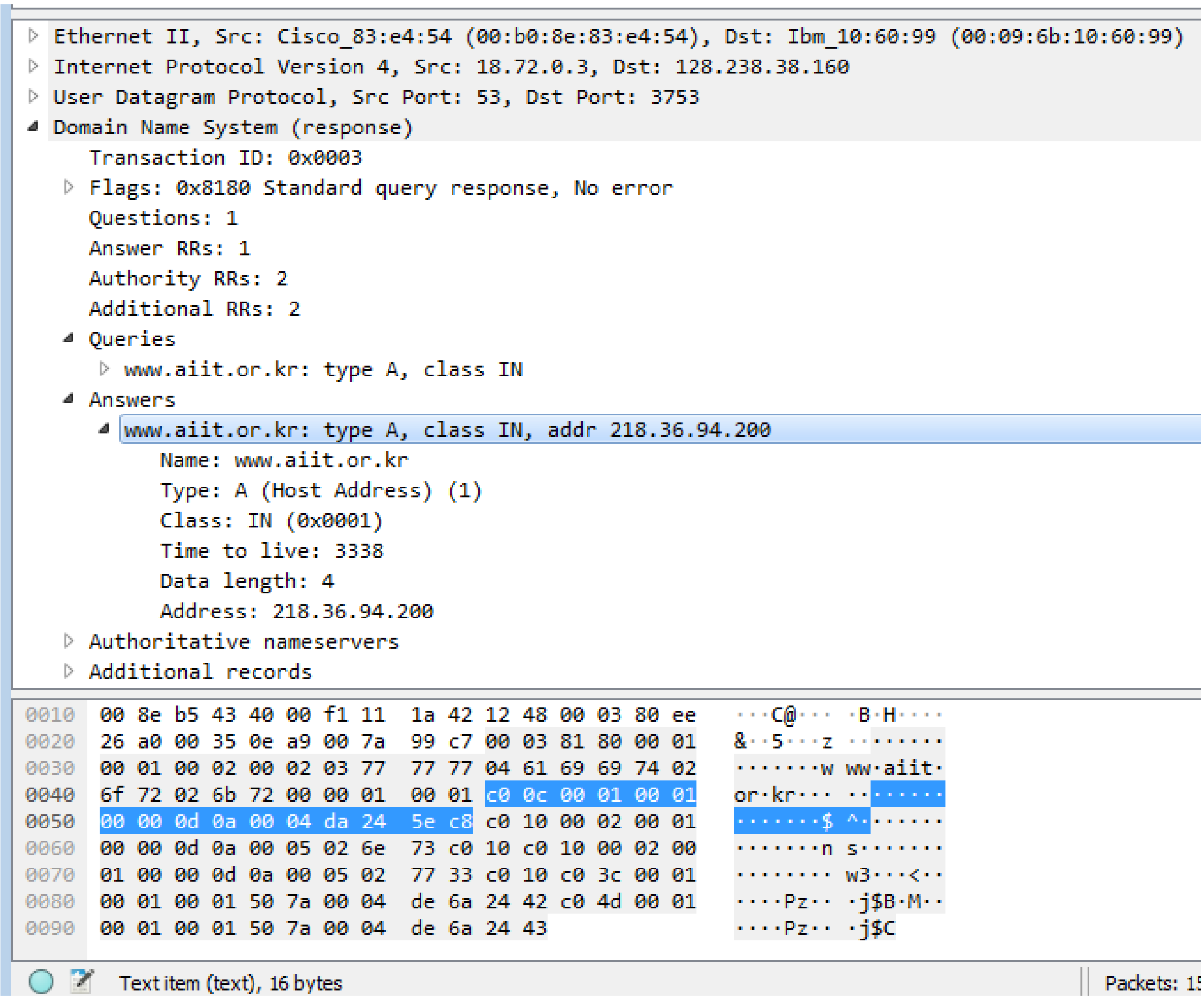
```
0000  00 00 0c 07 ac 00 00 09  6b 10 60 99 08 00 45 00   ········k·`···E·
0010  00 3c 28 05 00 00 80 11  58 d3 80 ee 26 a0 12 48   ·<(·····X···&··H
0020  00 03 0e a9 00 35 00 28  f5 47 00 03 01 00 00 01   ·····5·(·G······
0030  00 00 00 00 00 00 03 77  77 77 04 61 69 69 74 02   ·······w ww·aiit·
0040  6f 72 02 6b 72 00 00 01  00 01                     or·kr··· ··
```

It is a type A query. It does not contain any answers.

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

One answer is provided which contains the details of the host including server name, address, the type, the time taken to capture it live and the length of the data.

23. Screenshot



```
▷ Ethernet II, Src: Cisco_83:e4:54 (00:b0:8e:83:e4:54), Dst: Ibm_10:60:99 (00:09:6b:10:60:99)
▷ Internet Protocol Version 4, Src: 18.72.0.3, Dst: 128.238.38.160
▷ User Datagram Protocol, Src Port: 53, Dst Port: 3753
◢ Domain Name System (response)
    Transaction ID: 0x0003
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 2
    Additional RRs: 2
  ◢ Queries
    ▷ www.aiit.or.kr: type A, class IN
  ◢ Answers
    ◢ www.aiit.or.kr: type A, class IN, addr 218.36.94.200
        Name: www.aiit.or.kr
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 3338
        Data length: 4
        Address: 218.36.94.200
  ▷ Authoritative nameservers
  ▷ Additional records

0010  00 8e b5 43 40 00 f1 11  1a 42 12 48 00 03 80 ee   ···C@··· ·B·H····
0020  26 a0 00 35 0e a9 00 7a  99 c7 00 03 81 80 00 01   &··5···z ········
0030  00 01 00 02 00 02 03 77  77 77 04 61 69 69 74 02   ·······w ww·aiit·
0040  6f 72 02 6b 72 00 00 01  00 01 c0 0c 00 01 00 01   or·kr··· ········
0050  00 00 0d 0a 00 04 da 24  5e c8 c0 10 00 02 00 01   ·······$ ^·······
0060  00 00 0d 0a 00 05 02 6e  73 c0 10 c0 10 00 02 00   ·······n s·······
0070  01 00 00 0d 0a 00 05 02  77 33 c0 10 c0 3c 00 01   ········ w3···<··
0080  00 01 00 01 50 7a 00 04  de 6a 24 42 c0 4d 00 01   ····Pz·· ·j$B·M··
0090  00 01 00 01 50 7a 00 04  de 6a 24 43               ····Pz·· ·j$C
```

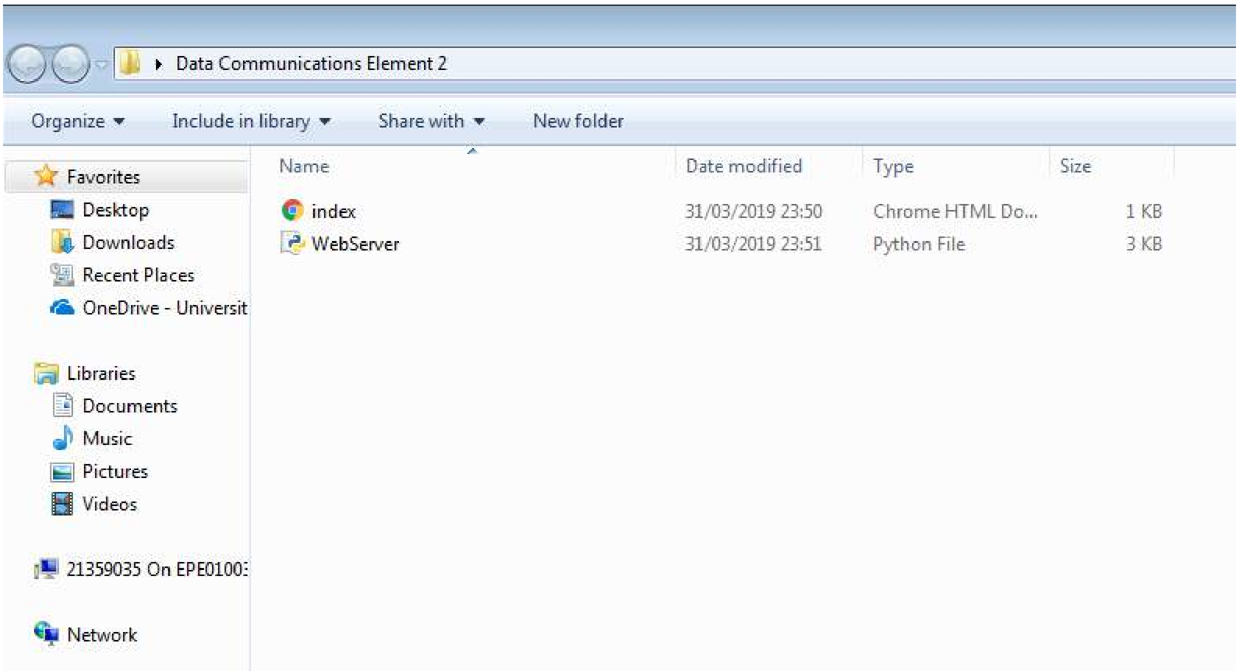
Text item (text), 16 bytes

Packets: 1

Element 2

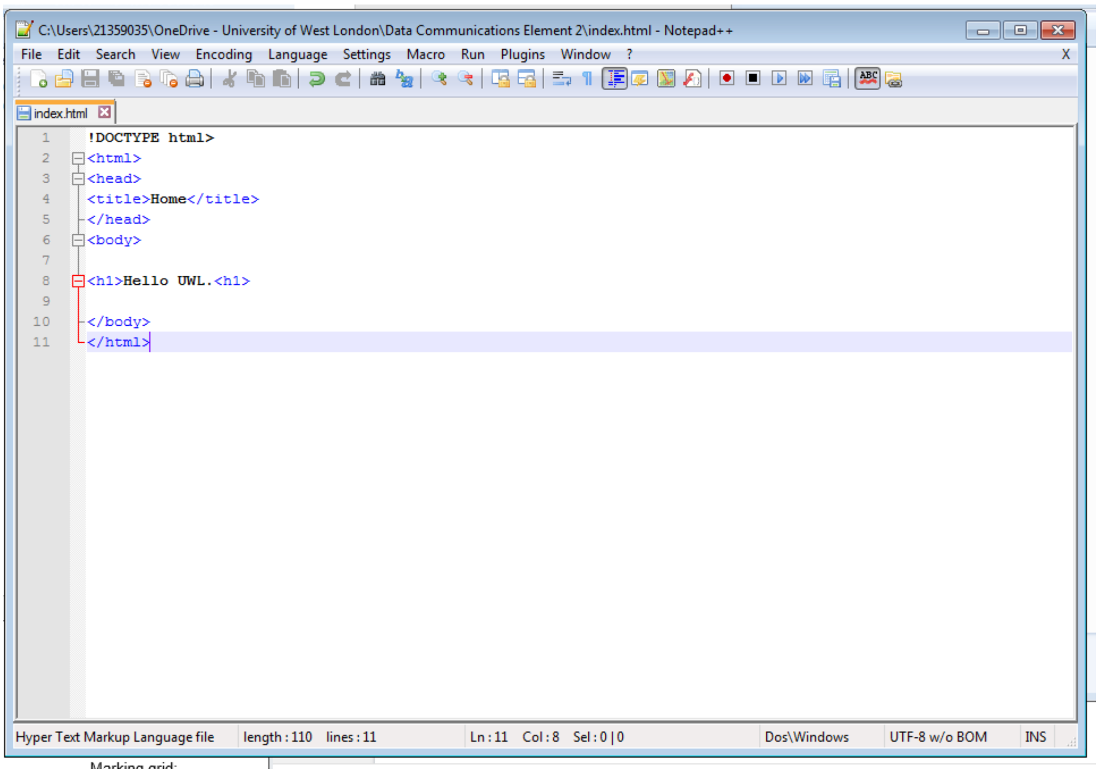
This second part of this assignment focuses on communication between the server and the client. A network is devices connected to each other such as the computer connected to a server to request a webpage from it. The server is the computer which holds a live copy of the webpage and the client is the computer that has requested a copy of the webpage from the server. When the client requests the webpage from the server, the server receives the request and then sends the page to the client.

I am going to test the connection between the server and the client by creating a webpage using language known as html and store that file in the same folder as the code for the web server written using programming code in a language known as Python. I have to store them both in the same folder in order for them to connect to each other. So the web browser that will load webpage is the client and the web server file written in programming language is the server.



Name	Date modified	Type	Size
index	31/03/2019 23:50	Chrome HTML Do...	1 KB
WebServer	31/03/2019 23:51	Python File	3 KB

Here is the html file of the webpage in the same folder as the web server which is the server code in a programming language.


C:\Users\21359035\OneDrive - University of West London\Data Communications Element 2\index.html - Notepad++
File
Edit
Search
View
Encoding
Language
Settings
Macro
Run
Plugins
Window
?
index.html

```
!DOCTYPE html>
<html>
<head>
<title>Home</title>
</head>
<body>

<h1>Hello UWL.<h1>

</body>
</html>
```

Hyper Text Markup Language file length : 110 lines : 11 Ln : 11 Col : 8 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS

A screenshot of the html code for the web page.

Below is the actual programming code of the web server written in the programminglanguage known as Python.

```
#import socket module from socket import *

import sys # In order to terminate the program

 serverSocket = socket(AF_INET, SOCK_STREAM)

#Prepare a sever socket and a port number for the server.

serverPort = 6789

# Bind the server socket to the server port

serverSocket.bind((“”, serverPort))

# Server socket starts listening which means it is ready to receive a connection which is in a queue as many as 1

serverSocket.listen(1)

#Fill in end

while True:

#Establish the connection

print('Ready to serve...')

# Create a connection to the client through a connection socket

connectionSocket, addr = serverSocket.accept()

 try:

# Read the request sent by the client at the connection socket

 message = connectionSocket.recv(1024)

# Split the message received at the connection socket and decode it

 filename = message.split()[1]

f = open(filename[1:]).decode())

outputdata = f. read()

# Start sending a reply to the clients request

connectionSocket.send("HTTP/1.1 200 OK\r\n\r\n".encode())

#Send the requested file to the connection socket

for i in range(0, len(outputdata)):

            connectionSocket.send(outputdata[i].encode())

connectionSocket.send("\r\n".encode())

#Close the connection socket to the client
```

```
connectionSocket.close()

except IOError:

#Send a response message if the file is not held by the server

connectionSocket.send("HTTP/1.1 404 Not Found\r\n\r\n".encode())

connectionSocket.send("<html><head></head><body><h1>404 Not Found</h1></body></html>\r\n".encode())

#Close the connection socket to the client

connectionSocket.close()

serverSocket.close()

sys.exit() #Terminate the program after sending the corresponding data
```

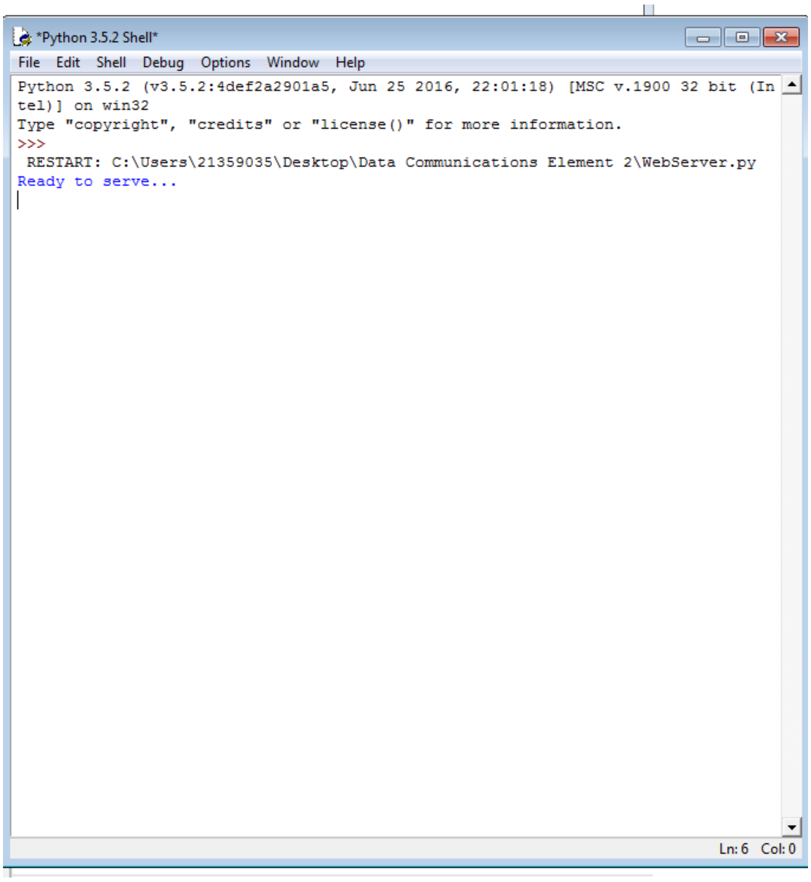
What the code above does

The code above is the code for the server. It makes the server receive the request from the client and send a message to client. To do that it first needs a connection socket created called the server socket and then a port number assigned to it using a variable called serverPort. The port number is binded to the server to create a connection to the client. The server starts listening to a request from the client. These number of client requests are in a queue as many as one. Once the server hears a request from the client a connection is ready and a connection to the client is made through a connection socket. The server prints a message ‘Ready to serve…’ when the server program code is being run with while being true.

A connection socket is created to connect to the client using the accept() method with the server socket. In the next part of the code starting with try the request sent by the client is read from the connection socket. A variable called message is then created with the clients port number 1024 being a parameter of the receive method .recv().

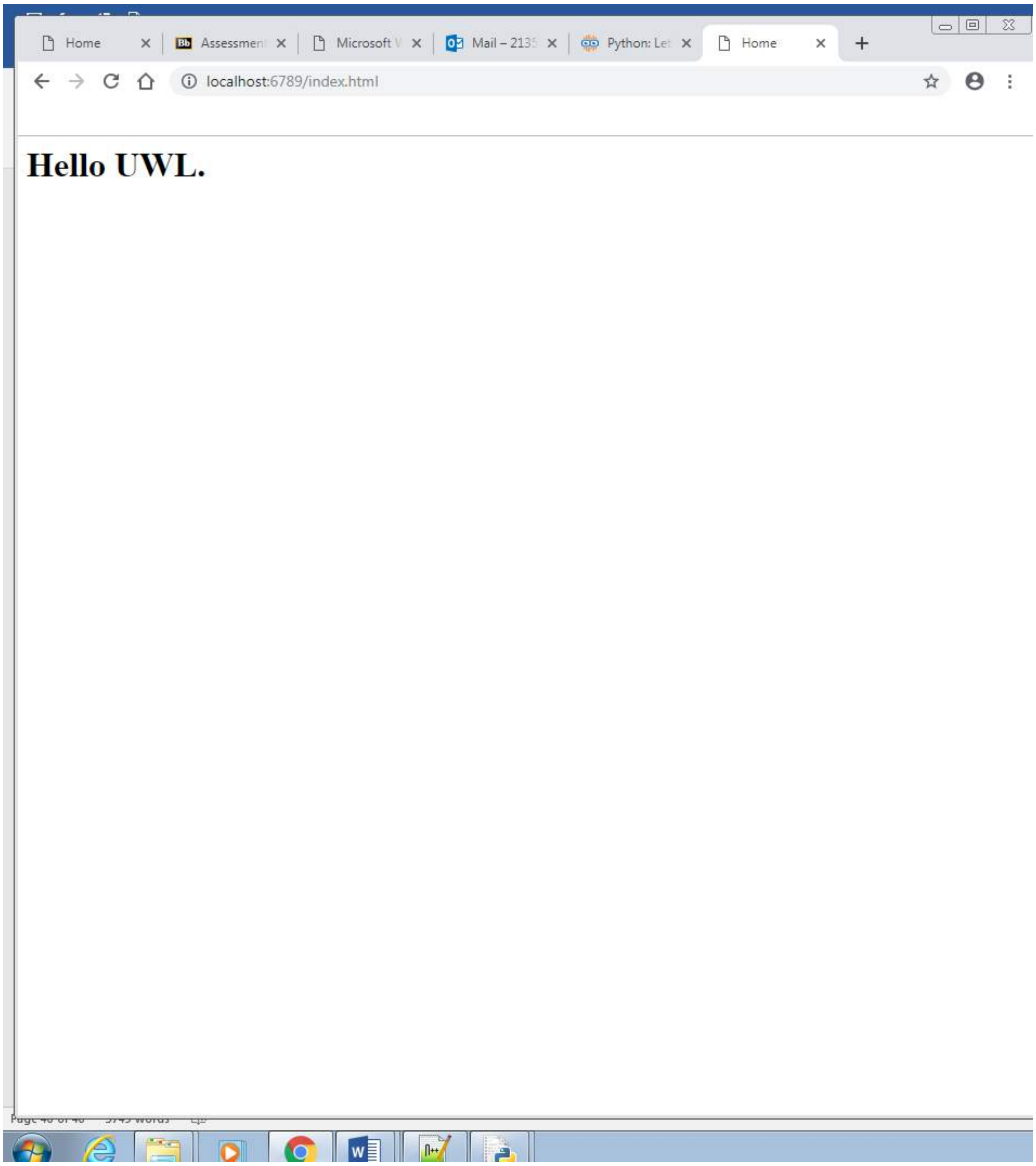
Before sending a response to the client the message, at the connection socket, is split and then decoded. It is then stored as output. A for loop is used to encode the response to the client and then send it through the connection socket to the client. If the requested web page file is not held by the server an exception error is sent displaying the message file Not Found.

The python web server code running. This screenshot indicates the server will be ready to serve…



```
Python 3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 
 RESTART: C:\Users\21359035\Desktop\Data Communications Element 2\WebServer.py
Ready to serve...
```

And here is the webpage running having used localhost followed by the port number and the name of the html file.



Hello UWL.