# Lab 5: IP & TCP

**ECSE 308 Introduction to Communication Systems and Networks**

Chenyi Xu 260948311

Yongru Pan 261001758

**Abstract: The lab is divided into three parts. The objective of the first part is to use Wireshark to investigate the Domina Name System(DNS) protocol from the DNS client's standpoint. In the second part of the lab, the focus was to use Wireshark to investigate the UDP structure and protocol properties. Last but not least, the third part concentrate on using Wireshark to collect traces and investigates the different aspects of the HTTP protocol operation.**

## Introduction

This lab explores the core concepts of Domain Name System (DNS), User Datagram Protocol (UDP), and Hypertext Transfer Protocol (HTTP), utilizing tools like nslookup and Wireshark to capture and analyze packet data. By performing a series of queries and analyzing responses, the lab investigates how DNS resolves domain names into IP addresses, the mechanics of UDP as a transport layer protocol for DNS, and the intricacies of HTTP for web communication. Through these exercises, key networking principles such as packet structure, protocol efficiency, and system reliability are examined, providing a comprehensive understanding of how these foundational internet protocols interact to deliver seamless user experiences.

## Analysis

### Part I: Domain Name System (DNS)

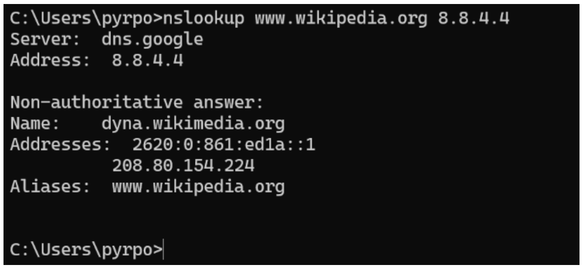
*Q1: Use nslookup to determine the IP address of www.cbc.ca. What is the IP address of this web server?*

The IP address is 184.25.129.124

*Q2: Use nslookup to determine the authoritative DNS servers for McGill University.*

The authoritative DNS is pirns1.mcgill.ca, with the IP address 132.206.44.21.

Q3: Run nslookup to obtain the IP address of www.wikipedia.org by sending a query to 8.8.4.4 which is the IP address of the google public DNS server.

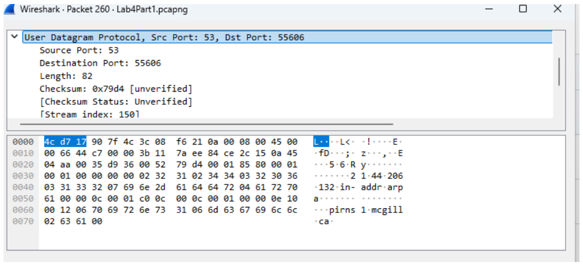


```
C:\Users\pyrpo>nslookup www.wikipedia.org 8.8.4.4
Server:  dns.google
Address:  8.8.4.4

Non-authoritative answer:
Name:    dyna.wikimedia.org
Addresses:  2620:0:861:ed1a::1
          208.80.154.224
Aliases:  www.wikipedia.org


C:\Users\pyrpo>
```

The IPv6 address is 2620:0:861:ed1a::1

The IPv4 address is 208.80.154.224

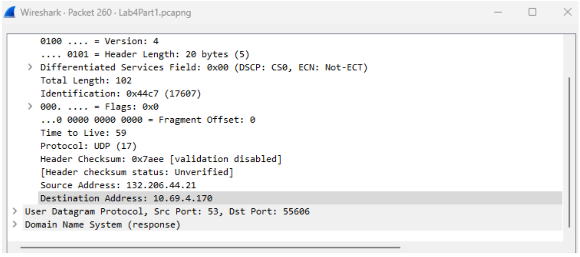
Q4: What are the destination port number for the DNS query message and the source port number of the DNS response message?



```
User Datagram Protocol, Src Port: 53, Dst Port: 55606
    Source Port: 53
    Destination Port: 55606
    Length: 82
    Checksum: 0x79d4 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 150]

0000  4c d7 17 90 7f 4c 3c 08  f6 21 0a 00 08 00 45 00   L···L<· ·!····E·
0010  00 66 44 c7 00 00 3b 11  7a ee 84 ce 2c 15 0a 45   ·fD···;· z···,··E
0020  04 aa 00 35 d9 36 00 52  79 d4 00 01 85 80 00 01   ···5·6·R y·······
0030  00 01 00 00 00 00 02 32  31 02 34 34 03 32 30 36   ·······2 1·44·206
0040  03 31 33 32 07 69 6e 2d  61 64 64 72 04 61 72 70   ·132·in- addr·arp
0050  61 00 00 0c 00 01 c0 0c  00 0c 00 01 00 00 0e 10   a······· ········
0060  00 12 06 70 69 72 6e 73  31 06 6d 63 67 69 6c 6c   ···pirns 1·mcgill
0070  02 63 61 00                                        ·ca·
```

The destination port number is 55606.

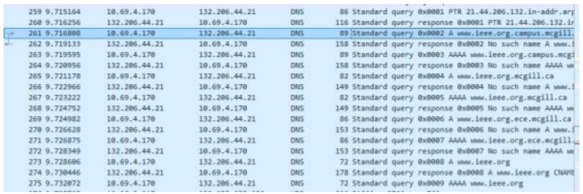
Q5: What is the destination IP address of the DNS query? Is this the IP address of your default local DNS server?



```
Wireshark · Packet 260 · Lab4Part1.pcapng
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 102
    Identification: 0x44c7 (17607)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 59
    Protocol: UDP (17)
    Header Checksum: 0x7aee [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 132.206.44.21
    Destination Address: 10.69.4.170
> User Datagram Protocol, Src Port: 53, Dst Port: 55606
> Domain Name System (response)
```

The destination IP address is 10.69.4.170

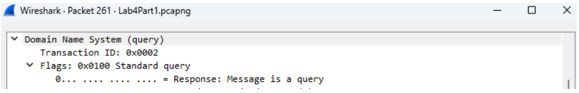
Q6: Examine the DNS query. What is the "Type" of the DNS query? What does this "Type" mean? What are the other values for this field?



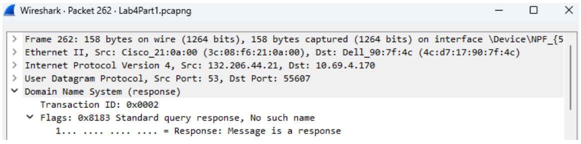
```
259 9.715164  10.69.4.170    132.206.44.21  DNS   86 Standard query 0x0001 PTR 21.44.206.132.in-addr.arp
260 9.716256  132.206.44.21  10.69.4.170    DNS  116 Standard query response 0x0001 PTR 21.44.206.132.ir
261 9.716808  10.69.4.170    132.206.44.21  DNS   89 Standard query 0x0002 A www.ieee.org.campus.mcgill.
262 9.719133  132.206.44.21  10.69.4.170    DNS  158 Standard query response 0x0002 No such name A www.i
263 9.719595  10.69.4.170    132.206.44.21  DNS   89 Standard query 0x0003 AAAA www.ieee.org.campus.mcgi
264 9.720956  132.206.44.21  10.69.4.170    DNS  158 Standard query response 0x0003 No such name AAAA ww
265 9.721178  10.69.4.170    132.206.44.21  DNS   82 Standard query 0x0004 A www.ieee.org.mcgill.ca
266 9.722966  132.206.44.21  10.69.4.170    DNS  149 Standard query response 0x0004 No such name A www.i
267 9.723222  10.69.4.170    132.206.44.21  DNS   82 Standard query 0x0005 AAAA www.ieee.org.mcgill.ca
268 9.724752  132.206.44.21  10.69.4.170    DNS  149 Standard query response 0x0005 No such name AAAA ww
269 9.724982  10.69.4.170    132.206.44.21  DNS   86 Standard query 0x0006 A www.ieee.org.ece.mcgill.ca
270 9.726628  132.206.44.21  10.69.4.170    DNS  153 Standard query response 0x0006 No such name A www.i
271 9.726875  10.69.4.170    132.206.44.21  DNS   86 Standard query 0x0007 AAAA www.ieee.org.ece.mcgill.
272 9.728349  132.206.44.21  10.69.4.170    DNS  153 Standard query response 0x0007 No such name AAAA ww
273 9.728606  10.69.4.170    132.206.44.21  DNS   72 Standard query 0x0008 A www.ieee.org
274 9.730446  132.206.44.21  10.69.4.170    DNS  178 Standard query response 0x0008 A www.ieee.org CNAME
275 9.732072  10.69.4.170    132.206.44.21  DNS   72 Standard query 0x0009 AAAA www.ieee.org
```

The "Type" of the DNS query is A, AAAA, SOA, and CNAME. A signifies the queries for IPv4 address of the host. AAAA signifies the queries for IPv6 address of the host. SOA signifies the query asking for the Start of Authority record. CNAME signifies the query asking for the alias of a domain. Other values include MX, NS, PTR, and TXT.

*Q7: Which bit in the "Flags" field indicates that the message is a query or a response?*



```
Wireshark · Packet 261 · Lab4Part1.pcapng
v Domain Name System (query)
    Transaction ID: 0x0002
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
```



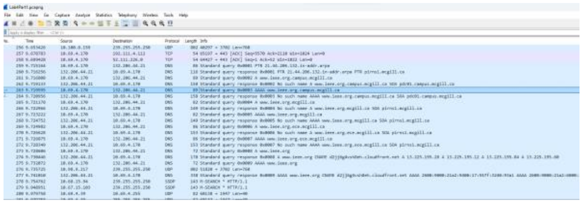
```
Wireshark · Packet 262 · Lab4Part1.pcapng
> Frame 262: 158 bytes on wire (1264 bits), 158 bytes captured (1264 bits) on interface \Device\NPF_{5
> Ethernet II, Src: Cisco_21:0a:00 (3c:08:f6:21:0a:00), Dst: Dell_90:7f:4c (4c:d7:17:90:7f:4c)
> Internet Protocol Version 4, Src: 132.206.44.21, Dst: 10.69.4.170
> User Datagram Protocol, Src Port: 53, Dst Port: 55607
v Domain Name System (response)
    Transaction ID: 0x0002
  v Flags: 0x8183 Standard query response, No such name
      1... .... .... .... = Response: Message is a response
```

The QR bit indicates this. If QR = 0, the message is a query. If QR = 1: indicates the message is a response.

*Q8: Which field of the response message contains the IP address of www.ieee.org?*

In the "Answer" field, you can find the IP address of www.ieee.org.

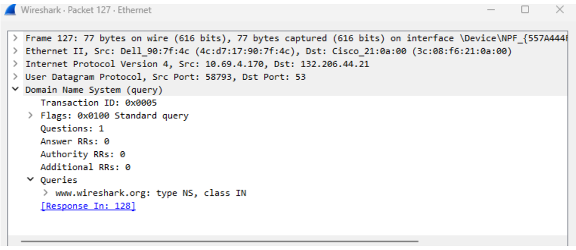
*Q9: Provide a screenshot.*



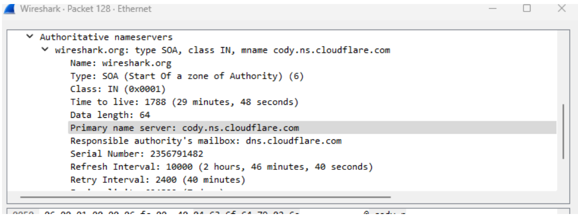
*Q10: What is the destination IP address of the DNS query? What does this address correspond to?*

The destination IP address is "132.206.44.21". This is the DNS server of mcgill.

*Q 11: Determine the "Type" of DNS query. What is the authoritative name server of www.wireshark.org. What is the role of an authoritative name server?*



```
Wireshark · Packet 127 · Ethernet
> Frame 127: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface \Device\NPF_{557A444
> Ethernet II, Src: Dell_90:7f:4c (4c:d7:17:90:7f:4c), Dst: Cisco_21:0a:00 (3c:08:f6:21:0a:00)
> Internet Protocol Version 4, Src: 10.69.4.170, Dst: 132.206.44.21
> User Datagram Protocol, Src Port: 58793, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0005
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    > www.wireshark.org: type NS, class IN
    [Response In: 128]
```



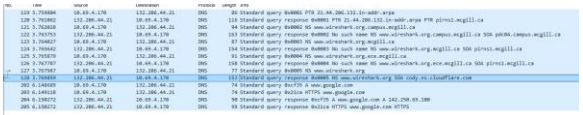
```
Wireshark · Packet 128 · Ethernet
  v Authoritative nameservers
    v wireshark.org: type SOA, class IN, mname cody.ns.cloudflare.com
        Name: wireshark.org
        Type: SOA (Start Of a zone of Authority) (6)
        Class: IN (0x0001)
        Time to live: 1788 (29 minutes, 48 seconds)
        Data length: 64
        Primary name server: cody.ns.cloudflare.com
        Responsible authority's mailbox: dns.cloudflare.com
        Serial Number: 2356791482
        Refresh Interval: 10000 (2 hours, 46 minutes, 40 seconds)
        Retry Interval: 2400 (40 minutes)
```

The type of the DNS query is NS.

The authoritative nameservers is cody.ns.cloudflare.com

Role of an Authoritative Name Server is to hold the authoritative DNS records for this domain and provides definitive answers to DNS queries related to it.

*Q12: Provide a screenshot.*

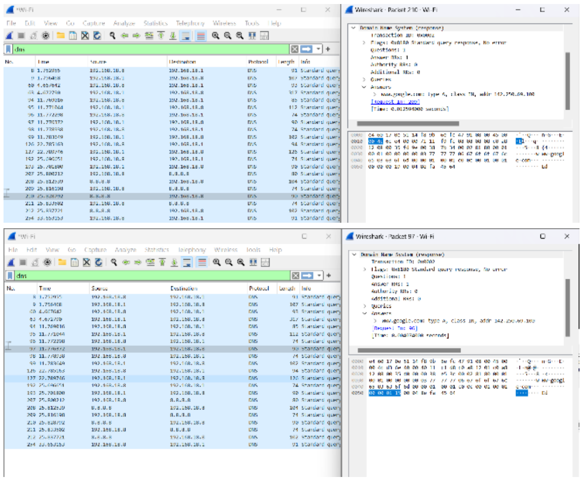


```
No.   Time      Source          Destination     Protocol Length Info
  119 3.759884  10.69.4.170     132.206.44.21   DNS       86 Standard query 0x0001 PTR 21.44.206.132.in-addr.arpa
  120 3.761062  132.206.44.21   10.69.4.170     DNS      116 Standard query response 0x0001 PTR 21.44.206.132.in-addr.arpa PTR pirns1.mcgill.ca
  121 3.762028  10.69.4.170     132.206.44.21   DNS       94 Standard query 0x0002 NS www.wireshark.org.campus.mcgill.ca
  122 3.763753  132.206.44.21   10.69.4.170     DNS      163 Standard query response 0x0002 No such name NS www.wireshark.org.campus.mcgill.ca SOA pdc04.campus.mcgill.ca
  123 3.764027  10.69.4.170     132.206.44.21   DNS       87 Standard query 0x0003 NS www.wireshark.org.mcgill.ca
  124 3.765642  132.206.44.21   10.69.4.170     DNS      154 Standard query response 0x0003 No such name NS www.wireshark.org.mcgill.ca SOA pirns1.mcgill.ca
  125 3.765870  10.69.4.170     132.206.44.21   DNS       91 Standard query 0x0004 NS www.wireshark.org.ece.mcgill.ca
  126 3.767787  132.206.44.21   10.69.4.170     DNS      158 Standard query response 0x0004 No such name NS www.wireshark.org.ece.mcgill.ca SOA pirns1.mcgill.ca
  127 3.767987  10.69.4.170     132.206.44.21   DNS       77 Standard query 0x0005 NS www.wireshark.org
  128 3.768859  132.206.44.21   10.69.4.170     DNS      133 Standard query response 0x0005 NS www.wireshark.org SOA cody.ns.cloudflare.com
  202 6.148689  10.69.4.170     132.206.44.21   DNS       74 Standard query 0xcf35 A www.google.com
  203 6.149118  10.69.4.170     132.206.44.21   DNS       74 Standard query 0x21ca HTTPS www.google.com
  204 6.150272  132.206.44.21   10.69.4.170     DNS       90 Standard query response 0xcf35 A www.google.com A 142.250.69.100
  205 6.150272  132.206.44.21   10.69.4.170     DNS       89 Standard query response 0x21ca HTTPS www.google.com HTTPS
```

*Q13: What are the destination IP addresses for the two DNS queries? What do these IP addresses correspond to?*

It can be seen from the screenshot that the two queries each have the following destination address: 192.168.18.1, which is the local router or DNS server address. The other one is 8.8.8.8, which is Google's public DNS server address.

*Q 14: What IP addresses are returned by these two queries? Do they return the same IP addresses for www.google.com? Explain your answer.*



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
Wireshark · Packet 210 · Wi-Fi

```
No.   Time       Source          Destination     Protocol Length Info
    8 1.752055   192.168.18.8    192.168.18.1    DNS       91 Standard query
    9 1.756468   192.168.18.1    192.168.18.8    DNS      107 Standard query
   60 4.667642   192.168.18.8    192.168.18.1    DNS       93 Standard query
   63 4.672790   192.168.18.1    192.168.18.8    DNS      317 Standard query
   94 11.769816  192.168.18.8    192.168.18.1    DNS       85 Standard query
   95 11.771844  192.168.18.1    192.168.18.8    DNS      112 Standard query
   96 11.772298  192.168.18.8    192.168.18.1    DNS       74 Standard query
   97 11.776372  192.168.18.1    192.168.18.8    DNS       90 Standard query
   98 11.778938  192.168.18.8    192.168.18.1    DNS       74 Standard query
   99 11.783640  192.168.18.1    192.168.18.8    DNS      102 Standard query
  126 22.785163  192.168.18.8    192.168.18.1    DNS       94 Standard query
  127 22.789746  192.168.18.1    192.168.18.8    DNS      126 Standard query
  192 25.696651  192.168.18.8    192.168.18.1    DNS       74 Standard query
  193 25.701800  192.168.18.1    192.168.18.8    DNS       90 Standard query
  207 25.808212  192.168.18.8    8.8.8.8         DNS       80 Standard query
  208 25.812539  8.8.8.8         192.168.18.8    DNS      104 Standard query
  209 25.816190  192.168.18.8    8.8.8.8         DNS       74 Standard query
  210 25.828792  8.8.8.8         192.168.18.8    DNS       90 Standard query
  211 25.833502  192.168.18.8    8.8.8.8         DNS       74 Standard query
  212 25.837721  8.8.8.8         192.168.18.8    DNS      102 Standard query
  254 33.653153  192.168.18.8    192.168.18.1    DNS       91 Standard query
```

```
v Domain Name System (response)
    Transaction ID: 0x0002
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  v Answers
    > www.google.com: type A, class IN, addr 142.250.69.100
    [Request In: 209]
    [Time: 0.012594000 seconds]
```

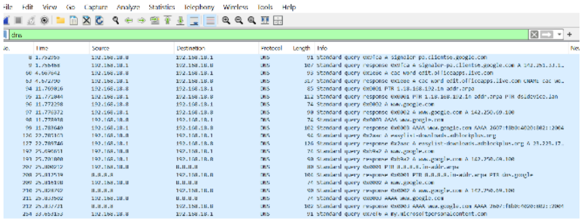
*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
Wireshark · Packet 97 · Wi-Fi

```
No.   Time       Source          Destination     Protocol Length Info
    8 1.752055   192.168.18.8    192.168.18.1    DNS       91 Standard query
    9 1.756468   192.168.18.1    192.168.18.8    DNS      107 Standard query
   60 4.667642   192.168.18.8    192.168.18.1    DNS       93 Standard query
   63 4.672790   192.168.18.1    192.168.18.8    DNS      317 Standard query
   94 11.769816  192.168.18.8    192.168.18.1    DNS       85 Standard query
   95 11.771844  192.168.18.1    192.168.18.8    DNS      112 Standard query
   96 11.772298  192.168.18.8    192.168.18.1    DNS       74 Standard query
   97 11.776372  192.168.18.1    192.168.18.8    DNS       90 Standard query
   98 11.778938  192.168.18.8    192.168.18.1    DNS       74 Standard query
   99 11.783640  192.168.18.1    192.168.18.8    DNS      102 Standard query
  126 22.785163  192.168.18.8    192.168.18.1    DNS       94 Standard query
  127 22.789746  192.168.18.1    192.168.18.8    DNS      126 Standard query
  192 25.696651  192.168.18.8    192.168.18.1    DNS       74 Standard query
  193 25.701800  192.168.18.1    192.168.18.8    DNS       90 Standard query
  207 25.808212  192.168.18.8    8.8.8.8         DNS       80 Standard query
  208 25.812539  8.8.8.8         192.168.18.8    DNS      104 Standard query
  209 25.816190  192.168.18.8    8.8.8.8         DNS       74 Standard query
  210 25.828792  8.8.8.8         192.168.18.8    DNS       90 Standard query
  211 25.833502  192.168.18.8    8.8.8.8         DNS       74 Standard query
  212 25.837721  8.8.8.8         192.168.18.8    DNS      102 Standard query
  254 33.653153  192.168.18.8    192.168.18.1    DNS       91 Standard query
```

```
v Domain Name System (response)
    Transaction ID: 0x0002
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  v Answers
    > www.google.com: type A, class IN, addr 142.250.69.100
    [Request In: 96]
    [Time: 0.004074000 seconds]
```

I have chosen packet 97 and packet 210 because packet 97's source is 192.168.18.1 and packet 210's source is 8.8.8.8, and they end up having the same IP address for www.google.com. The reason behind this can be that they both access the same cached record, or that they have received the same response from Google's authoritative DNS.

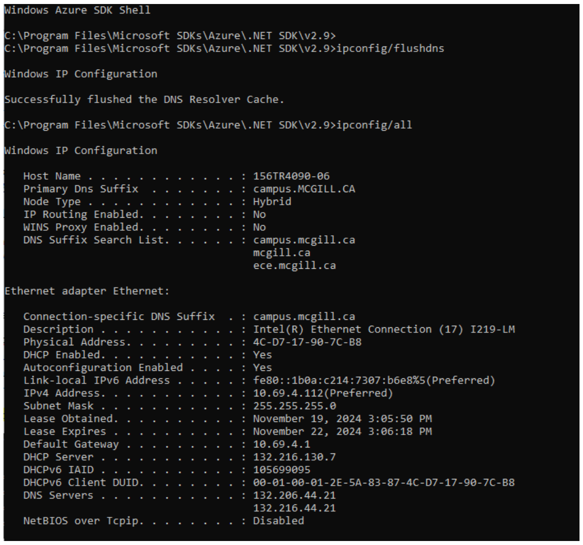
*Q15: Provide a screen shot.*



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns

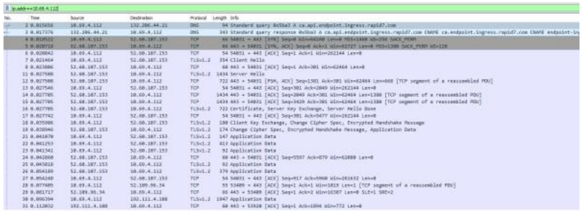
```
No.   Time       Source          Destination     Protocol Length Info
    8 1.752055   192.168.18.8    192.168.18.1    DNS       91 Standard query 0x0fca A signaler-pa.clients6.google.com
    9 1.756468   192.168.18.1    192.168.18.8    DNS      107 Standard query response 0x0fca A signaler-pa.clients6.google.com A 142.251.33.1…
   60 4.667642   192.168.18.8    192.168.18.1    DNS       93 Standard query 0x1e8e A cac word edit.officeapps.live.com
   63 4.672790   192.168.18.1    192.168.18.8    DNS      317 Standard query response 0x1e8e A cac word edit.officeapps.live.com CNAME cac wo…
   94 11.769816  192.168.18.8    192.168.18.1    DNS       85 Standard query 0x0001 PTR 1.18.168.192.in addr.arpa
   95 11.771844  192.168.18.1    192.168.18.8    DNS      112 Standard query response 0x0001 PTR 1.18.168.192.in addr.arpa PTR dsldevice.lan
   96 11.772298  192.168.18.8    192.168.18.1    DNS       74 Standard query 0x0002 A www.google.com
   97 11.776372  192.168.18.1    192.168.18.8    DNS       90 Standard query response 0x0002 A www.google.com A 142.250.69.100
   98 11.778938  192.168.18.8    192.168.18.1    DNS       74 Standard query 0x0003 AAAA www.google.com
   99 11.783640  192.168.18.1    192.168.18.8    DNS      102 Standard query response 0x0003 AAAA www.google.com AAAA 2607:f8b0:4020:802::2004
  126 22.785163  192.168.18.8    192.168.18.1    DNS       94 Standard query 0x2aac A easylist-downloads.adblockplus.org
  127 22.789746  192.168.18.1    192.168.18.8    DNS      126 Standard query response 0x2aac A easylist-downloads.adblockplus.org A 23.223.17…
  192 25.696651  192.168.18.8    192.168.18.1    DNS       74 Standard query 0xb9a2 A www.google.com
  193 25.701800  192.168.18.1    192.168.18.8    DNS       90 Standard query response 0xb9a2 A www.google.com A 142.250.69.100
  207 25.808212  192.168.18.8    8.8.8.8         DNS       80 Standard query 0x0001 PTR 8.8.8.8.in-addr.arpa
  208 25.812539  8.8.8.8         192.168.18.8    DNS      104 Standard query response 0x0001 PTR 8.8.8.8.in-addr.arpa PTR dns.google
  209 25.816190  192.168.18.8    8.8.8.8         DNS       74 Standard query 0x0002 A www.google.com
  210 25.828792  8.8.8.8         192.168.18.8    DNS       90 Standard query response 0x0002 A www.google.com A 142.250.69.100
  211 25.833502  192.168.18.8    8.8.8.8         DNS       74 Standard query 0x0003 AAAA www.google.com
  212 25.837721  8.8.8.8         192.168.18.8    DNS      102 Standard query response 0x0003 AAAA www.google.com AAAA 2607:f8b0:4020:803::2004
  254 33.653153  192.168.18.8    192.168.18.1    DNS       91 Standard query 0x7efe A my.microsoftpersonalcontent.com
```

## Part II: User Datagram Protocol (UDP)



```
Windows Azure SDK Shell

C:\Program Files\Microsoft SDKs\Azure\.NET SDK\v2.9>
C:\Program Files\Microsoft SDKs\Azure\.NET SDK\v2.9>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Program Files\Microsoft SDKs\Azure\.NET SDK\v2.9>ipconfig/all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : 156TR4090-06
   Primary Dns Suffix  . . . . . . . : campus.MCGILL.CA
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : campus.mcgill.ca
                                       mcgill.ca
                                       ece.mcgill.ca

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : campus.mcgill.ca
   Description . . . . . . . . . . . : Intel(R) Ethernet Connection (17) I219-LM
   Physical Address. . . . . . . . . : 4C-D7-17-90-7C-B8
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::1b0a:c214:7307:b6e8%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.69.4.112(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : November 19, 2024 3:05:50 PM
   Lease Expires . . . . . . . . . . : November 22, 2024 3:06:18 PM
   Default Gateway . . . . . . . . . : 10.69.4.1
   DHCP Server . . . . . . . . . . . : 132.216.130.7
   DHCPv6 IAID . . . . . . . . . . . : 105699095
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2E-5A-83-87-4C-D7-17-90-7C-B8
   DNS Servers . . . . . . . . . . . : 132.206.44.21
                                       132.216.44.21
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

Ip address = 10.69.4.112

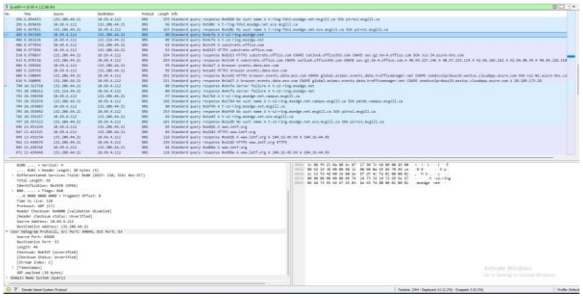


```
No.  Time      Source          Destination     Protocol Length Info
   2 0.015656  10.69.4.112     132.206.44.21   DNS       94 Standard query 0x5ba3 A ca.api.endpoint.ingress.rapid7.com
   3 0.017376  132.206.44.21   10.69.4.112     DNS      343 Standard query response 0x5ba3 A ca.api.endpoint.ingress.rapid7.com CNAME ca.endpoint.ingress.rapid7.com CNAME endpoint-in…
   4 0.018522  10.69.4.112     52.60.187.153   TCP       66 54031 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
   5 0.020718  52.60.187.153   10.69.4.112     TCP       66 443 → 54031 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=1380 SACK_PERM WS=128
   6 0.020842  10.69.4.112     52.60.187.153   TCP       54 54031 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
   7 0.021464  10.69.4.112     52.60.187.153   TLSv1.2  354 Client Hello
   8 0.023886  52.60.187.153   10.69.4.112     TCP       60 443 → 54031 [ACK] Seq=1 Ack=301 Win=62464 Len=0
  11 0.027508  52.60.187.153   10.69.4.112     TLSv1.2 1434 Server Hello
  12 0.027508  52.60.187.153   10.69.4.112     TCP      722 443 → 54031 [PSH, ACK] Seq=1381 Ack=301 Win=62464 Len=668 [TCP segment of a reassembled PDU]
  13 0.027546  10.69.4.112     52.60.187.153   TCP       54 54031 → 443 [ACK] Seq=301 Ack=2049 Win=262144 Len=0
  14 0.027705  52.60.187.153   10.69.4.112     TCP     1434 443 → 54031 [ACK] Seq=2049 Ack=301 Win=62464 Len=1380 [TCP segment of a reassembled PDU]
  15 0.027705  52.60.187.153   10.69.4.112     TCP     1434 443 → 54031 [ACK] Seq=3429 Ack=301 Win=62464 Len=1380 [TCP segment of a reassembled PDU]
  16 0.027705  52.60.187.153   10.69.4.112     TLSv1.2  722 Certificate, Server Key Exchange, Server Hello Done
  17 0.027742  10.69.4.112     52.60.187.153   TCP       54 54031 → 443 [ACK] Seq=301 Ack=5477 Win=262144 Len=0
  18 0.035986  10.69.4.112     52.60.187.153   TLSv1.2  180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
  19 0.038046  52.60.187.153   10.69.4.112     TLSv1.2  174 Change Cipher Spec, Encrypted Handshake Message, Application Data
  21 0.041070  10.69.4.112     52.60.187.153   TLSv1.2  147 Application Data
  22 0.041253  10.69.4.112     52.60.187.153   TLSv1.2  413 Application Data
  23 0.041341  10.69.4.112     52.60.187.153   TLSv1.2   92 Application Data
  24 0.042860  52.60.187.153   10.69.4.112     TCP       60 443 → 54031 [ACK] Seq=5597 Ack=879 Win=62080 Len=0
  25 0.043018  52.60.187.153   10.69.4.112     TLSv1.2   92 Application Data
  26 0.054189  52.60.187.153   10.69.4.112     TLSv1.2  379 Application Data
  27 0.054248  10.69.4.112     52.60.187.153   TCP       54 54031 → 443 [ACK] Seq=917 Ack=5960 Win=261632 Len=0
  28 0.077405  10.69.4.112     52.109.96.34    TCP       55 53409 → 443 [ACK] Seq=1 Ack=1 Win=1019 Len=1 [TCP segment of a reassembled PDU]
  29 0.081717  52.109.96.34    10.69.4.112     TCP       66 443 → 53409 [ACK] Seq=1 Ack=2 Win=16387 Len=0 SLE=1 SRE=2
  30 0.096394  10.69.4.112     192.111.4.108   TLSv1.2 1047 Application Data
  31 0.112032  192.111.4.108   10.69.4.112     TCP       60 443 → 53020 [ACK] Seq=1 Ack=1094 Win=772 Len=0
```

Q16: What transport layer protocol is used to transfer the DNS query and the response message?

UDP

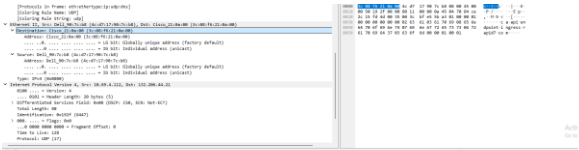
Q17: To set up the connection, how many UDP datagrams are exchanged between your computer and the server? Explain your answer.



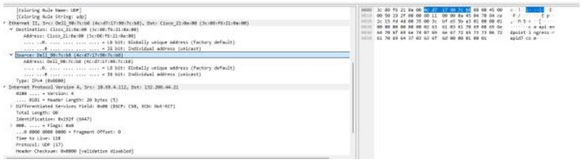
61 UDP datagrams are exchanged between the computer and the server. A filter is used to filter out all the DNS. According to Wireshark, 61 UDP datagrams are exchanged.

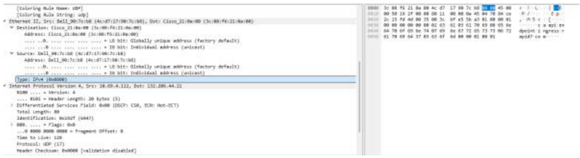
Q18: Select the first DNS packet in your trace. From this packet, determine the header fields of UDP.

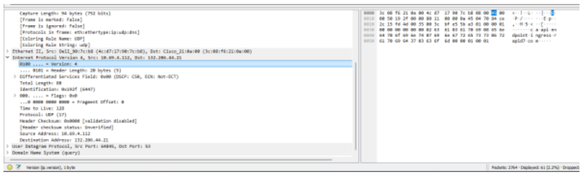
Destination, Source, Types, Version and header length, Differentiated Service Fields, Total length, Identification , Flags, Fragment offset, Time to live, Protocol, Header Checksum, Source address, Destination address, Source port, destination port, length, checksum, UDP payload

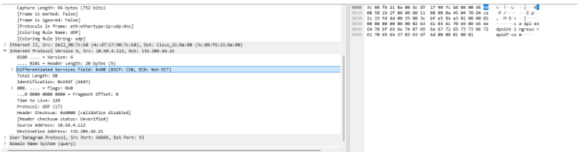


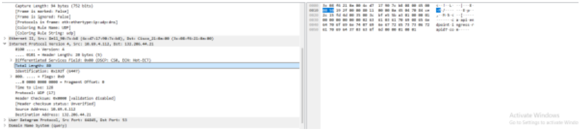
```
    [Protocols in frame: eth:ethertype:ip:udp:dns]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
v Ethernet II, Src: Dell_90:7c:b8 (4c:d7:17:90:7c:b8), Dst: Cisco_21:0a:00 (3c:08:f6:21:0a:00)
  v Destination: Cisco_21:0a:00 (3c:08:f6:21:0a:00)
      Address: Cisco_21:0a:00 (3c:08:f6:21:0a:00)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  v Source: Dell_90:7c:b8 (4c:d7:17:90:7c:b8)
      Address: Dell_90:7c:b8 (4c:d7:17:90:7c:b8)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 10.69.4.112, Dst: 132.206.44.21
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 88
    Identification: 0x192f (6447)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
```

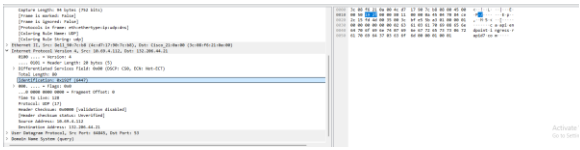


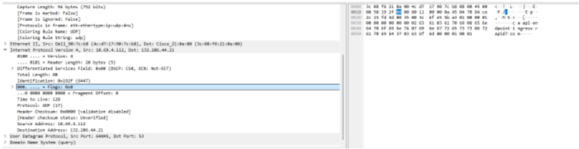


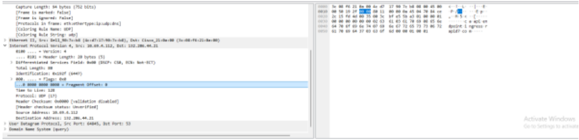


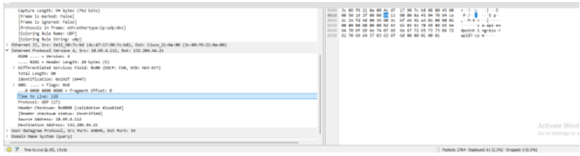


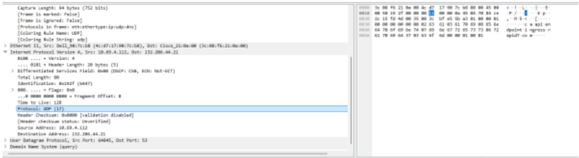


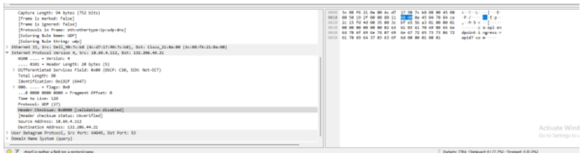


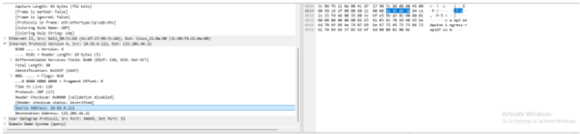


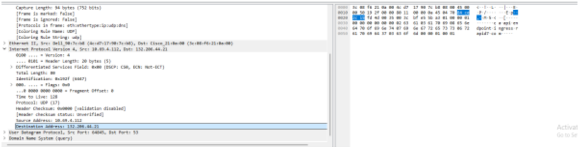


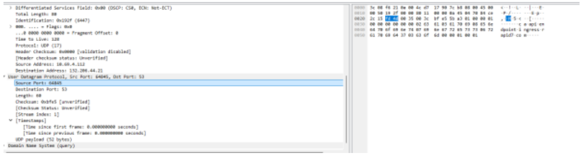






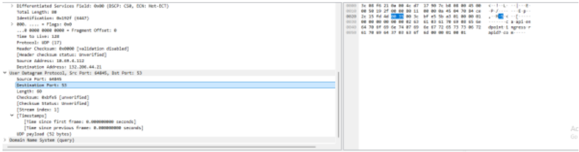






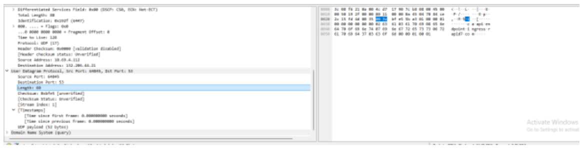
```
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 80
  Identification: 0x192f (6447)
> 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: UDP (17)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.69.4.112
  Destination Address: 132.206.44.21
v User Datagram Protocol, Src Port: 64845, Dst Port: 53
    Source Port: 64845
    Destination Port: 53
    Length: 60
    Checksum: 0xbfe5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (52 bytes)
> Domain Name System (query)
```

```
0000  3c 08 f6 21 0a 00 4c d7  17 90 7c b8 08 00 45 00   <··!··L· ··|···E·
0010  00 50 19 2f 00 00 80 11  00 00 0a 45 04 70 84 ce   ·P·/···· ···E·p··
0020  2c 15 fd 4d 00 35 00 3c  bf e5 5b a3 01 00 00 01   ,··M·5·< ··[·····
0030  00 00 00 00 00 00 02 63  61 03 61 70 69 08 65 6e   ·······c a·api·en
0040  64 70 6f 69 6e 74 07 69  6e 67 72 65 73 73 06 72   dpoint·i ngress·r
0050  61 70 69 64 37 03 63 6f  6d 00 00 01 00 01         apid7·co m·····
```



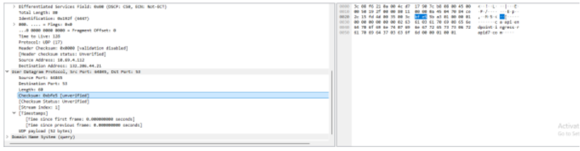
```
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 80
  Identification: 0x192f (6447)
> 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: UDP (17)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.69.4.112
  Destination Address: 132.206.44.21
v User Datagram Protocol, Src Port: 64845, Dst Port: 53
    Source Port: 64845
    Destination Port: 53
    Length: 60
    Checksum: 0xbfe5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (52 bytes)
> Domain Name System (query)
```

```
0000  3c 08 f6 21 0a 00 4c d7  17 90 7c b8 08 00 45 00   <··!··L· ··|···E·
0010  00 50 19 2f 00 00 80 11  00 00 0a 45 04 70 84 ce   ·P·/···· ···E·p··
0020  2c 15 fd 4d 00 35 00 3c  bf e5 5b a3 01 00 00 01   ,··M·5·< ··[·····
0030  00 00 00 00 00 00 02 63  61 03 61 70 69 08 65 6e   ·······c a·api·en
0040  64 70 6f 69 6e 74 07 69  6e 67 72 65 73 73 06 72   dpoint·i ngress·r
0050  61 70 69 64 37 03 63 6f  6d 00 00 01 00 01         apid7·co m·····
```



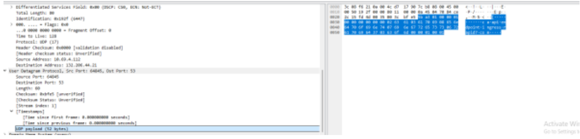
```
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 80
  Identification: 0x192f (6447)
> 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: UDP (17)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.69.4.112
  Destination Address: 132.206.44.21
v User Datagram Protocol, Src Port: 64845, Dst Port: 53
    Source Port: 64845
    Destination Port: 53
    Length: 60
    Checksum: 0xbfe5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (52 bytes)
> Domain Name System (query)
```

```
0000  3c 08 f6 21 0a 00 4c d7  17 90 7c b8 08 00 45 00   <··!··L· ··|···E·
0010  00 50 19 2f 00 00 80 11  00 00 0a 45 04 70 84 ce   ·P·/···· ···E·p··
0020  2c 15 fd 4d 00 35 00 3c  bf e5 5b a3 01 00 00 01   ,··M·5·< ··[·····
0030  00 00 00 00 00 00 02 63  61 03 61 70 69 08 65 6e   ·······c a·api·en
0040  64 70 6f 69 6e 74 07 69  6e 67 72 65 73 73 06 72   dpoint·i ngress·r
0050  61 70 69 64 37 03 63 6f  6d 00 00 01 00 01         apid7·co m·····
```



```
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 80
  Identification: 0x192f (6447)
> 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: UDP (17)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.69.4.112
  Destination Address: 132.206.44.21
v User Datagram Protocol, Src Port: 64845, Dst Port: 53
    Source Port: 64845
    Destination Port: 53
    Length: 60
    Checksum: 0xbfe5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (52 bytes)
> Domain Name System (query)
```

```
0000  3c 08 f6 21 0a 00 4c d7  17 90 7c b8 08 00 45 00   <··!··L· ··|···E·
0010  00 50 19 2f 00 00 80 11  00 00 0a 45 04 70 84 ce   ·P·/···· ···E·p··
0020  2c 15 fd 4d 00 35 00 3c  bf e5 5b a3 01 00 00 01   ,··M·5·< ··[·····
0030  00 00 00 00 00 00 02 63  61 03 61 70 69 08 65 6e   ·······c a·api·en
0040  64 70 6f 69 6e 74 07 69  6e 67 72 65 73 73 06 72   dpoint·i ngress·r
0050  61 70 69 64 37 03 63 6f  6d 00 00 01 00 01         apid7·co m·····
```



```
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 80
  Identification: 0x192f (6447)
> 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: UDP (17)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 10.69.4.112
  Destination Address: 132.206.44.21
v User Datagram Protocol, Src Port: 64845, Dst Port: 53
    Source Port: 64845
    Destination Port: 53
    Length: 60
    Checksum: 0xbfe5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
  v [Timestamps]
      [Time since first frame: 0.000000000 seconds]
      [Time since previous frame: 0.000000000 seconds]
    UDP payload (52 bytes)
> Domain Name System (query)
```

```
0000  3c 08 f6 21 0a 00 4c d7  17 90 7c b8 08 00 45 00   <··!··L· ··|···E·
0010  00 50 19 2f 00 00 80 11  00 00 0a 45 04 70 84 ce   ·P·/···· ···E·p··
0020  2c 15 fd 4d 00 35 00 3c  bf e5 5b a3 01 00 00 01   ,··M·5·< ··[·····
0030  00 00 00 00 00 00 02 63  61 03 61 70 69 08 65 6e   ·······c a·api·en
0040  64 70 6f 69 6e 74 07 69  6e 67 72 65 73 73 06 72   dpoint·i ngress·r
0050  61 70 69 64 37 03 63 6f  6d 00 00 01 00 01         apid7·co m·····
```

Q19: By consulting the displayed information in Wireshark's packet content field for the first DNS message, determine the length(in bytes) of each of the UDP header fields.

Destination  6 bytes

Source 6 bytes

Types: 2 bytes

IP Version and header length: 1 byte

Differentiated Service Fields: 1 byte

Total length: 2 bytes

Identification: 2 bytes

Fragment offset: 2 byte

Flags: 1 byte

Time to live: 1 byte

Protocol: 1 byte

Header Checksum: 2 bytes

Source address: 4 bytes

Destination adresse: 4 bytes

Source port: 2 bytes

Destination port: 2 bytes

Length: 2 bytes

Checksum: 2 bytes

UDP payload: 52 bytes

Q20: The value in the Length field indicates the length of what? Verify your claim with your captured UDP packet.

It specifies the size of the entire UDP datagram. The length field here shows a value of 60 bytes. 8 bytes are used for the UDP header(source, destination, length, checksum) and 52 bytes are used for the UDP payload.

Q21: What is the maximum number of bytes that can be included in a UDP payload? (Hint: The answer to this question can be determined by your previous answer).

Since length occupies 2 bytes. The largest possible number is ff ff which is 65535. 8 bytes are used for the header field. So the maximum number of bytes that can be included in a UDP pauload is ff ff -

Q22: What is the largest possible source port number?

Source port occupies 2 bytes. The largest possible source port number would be ff ff which is 65535

Q23: Determine whether a checksum is provided for the first DNS message or not. What is the usage of this field?

Yes, it is provided. It ensures that the data in the UDP header and payload arrives without corruption.

Q24: Determine the destination port number for the DNS query message and the source port number of the DNS response. What is the relationship between the two? Which port number is a well-known port number?

Destination port number: 53

Source port number: 64845

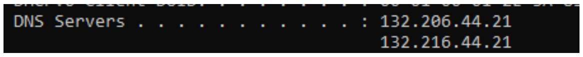
Port 53 is a well-known port number, as it is reserved for the DNS protocol. It is used by DNS servers to listen for incoming queries from clients.

The client initiates the query, using the dynamic source port (64845 in this case) and sends the request to port 53 on the DNS server. The server replies using port 53 as its source port (matching the client's query destination port) and sends the response to the dynamic port (64845) that was originally used as the client's source port.

Q25: List two other well-known port numbers used by UDP

Port 67, Port 123

Q26: Determine the IP address of your local DNS sever (use ipcongif). Is it the same as destination IP address of the DNS query?

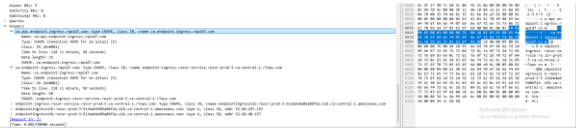


```
DNS Servers . . . . . . . . . . . : 132.206.44.21
                                    132.216.44.21
```

Destination address of the DNS query: 132.206.44.21. It is the same as the destination IP address of the local DNS server.

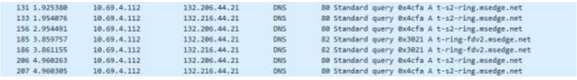
Q27: Examine the DNS response message. How many answers are provided in this message? What do each of these answers contain?

5 answers are provided in the message. Each of those answers contain Name, Type, Class, Time to live, Data Length, CNAME



```
Answer RRs: 5
Authority RRs: 0
Additional RRs: 0
> Queries
v Answers
  v ca.api.endpoint.ingress.rapid7.com: type CNAME, class IN, cname ca.endpoint.ingress.rapid7.com
      Name: ca.api.endpoint.ingress.rapid7.com
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 118 (1 minute, 58 seconds)
      Data length: 32
      CNAME: ca.endpoint.ingress.rapid7.com
  v ca.endpoint.ingress.rapid7.com: type CNAME, class IN, cname endpoint-ingress.razor-service.razor-prod-3.ca-central-1.r7ops.com
      Name: ca.endpoint.ingress.rapid7.com
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 118 (1 minute, 58 seconds)
      Data length: 68
      CNAME: endpoint-ingress.razor-service.razor-prod-3.ca-central-1.r7ops.com
  > endpoint-ingress.razor-service.razor-prod-3.ca-central-1.r7ops.com: type CNAME, class IN, cname endpointingresslb-razor-prod-3-513aeb4e05a0bf2a.elb.ca-central-1.amazonaws.com
  > endpointingresslb-razor-prod-3-513aeb4e05a0bf2a.elb.ca-central-1.amazonaws.com: type A, class IN, addr 52.60.187.133
  > endpointingresslb-razor-prod-3-513aeb4e05a0bf2a.elb.ca-central-1.amazonaws.com: type A, class IN, addr 52.60.48.157
[Request In: 2]
[Time: 0.001720000 seconds]
```

Q28: By checking the trace, determine whether UDP is a reliable protocol or not. Explain your answer.



```
131 1.925380   10.69.4.112   132.206.44.21   DNS   80 Standard query 0x4cfa A t-s2-ring.msedge.net
133 1.954876   10.69.4.112   132.216.44.21   DNS   80 Standard query 0x4cfa A t-s2-ring.msedge.net
156 2.954491   10.69.4.112   132.206.44.21   DNS   80 Standard query 0x4cfa A t-s2-ring.msedge.net
185 3.859757   10.69.4.112   132.206.44.21   DNS   82 Standard query 0x3021 A t-ring-fdv2.msedge.net
186 3.861155   10.69.4.112   132.216.44.21   DNS   82 Standard query 0x3021 A t-ring-fdv2.msedge.net
206 4.960263   10.69.4.112   132.206.44.21   DNS   80 Standard query 0x4cfa A t-s2-ring.msedge.net
207 4.960305   10.69.4.112   132.216.44.21   DNS   80 Standard query 0x4cfa A t-s2-ring.msedge.net
```

The trace demonstrates that UDP is unreliable because it does not guarantee packet delivery or handle retransmissions, as evidenced by the repeated DNS queries. It is the best trying mechanism. The DNS queries was sent repeatedly indicating the transmission failed. Hence not reliable.

Q29: Why does DNS use UDP services?

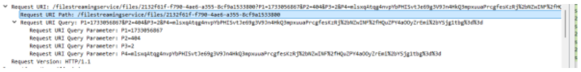
It uses UDP due to its speed, efficiency, and suitability. It is a connectionless protocol. No need to establish connection which reduces latency, allowing DNS to resolve queries more quickly. UDP has low overhead. It is ideal for DNS, where most queries and responses are small and don’t need the reliability mechanism of TCP.

## Part III: Hyper-Text Transfer Protocol (HTTP)

*Q30. What HTTP request method is used to retrieve the HTML file?*

The HTTP request method used to retrieve the HTML file is **GET.**

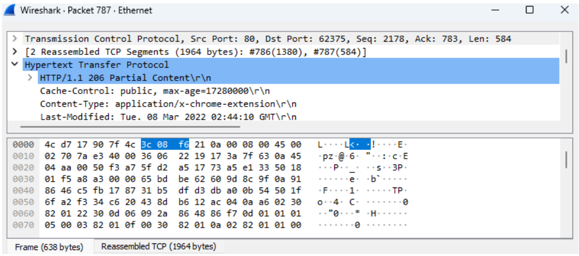
*Q31. What is the URI of the requested file?*



```
v Request URI: /filestreamingservice/files/2132f61f-f790-4ae6-a355-8cf9a1533800?P1=1733056867&P2=404&P3=2&P4=mlsxqAtqg4nvpYbPHISvtJe69g3V9Jn4HkQ3mpxuuaPrcgfesKzRj%2bNZwINF%2fHQ
    Request URI Path: /filestreamingservice/files/2132f61f-f790-4ae6-a355-8cf9a1533800
  v Request URI Query: P1=1733056867&P2=404&P3=2&P4=mlsxqAtqg4nvpYbPHISvtJe69g3V9Jn4HkQ3mpxuuaPrcgfesKzRj%2bNZwINF%2fHQuZPY4a0OyZr6mi%2bYSjg1tbg%3d%3d
      Request URI Query Parameter: P1=1733056867
      Request URI Query Parameter: P2=404
      Request URI Query Parameter: P3=2
      Request URI Query Parameter: P4=mlsxqAtqg4nvpYbPHISvtJe69g3V9Jn4HkQ3mpxuuaPrcgfesKzRj%2bNZwINF%2fHQuZPY4a0OyZr6mi%2bYSjg1tbg%3d%3d
  Request Version: HTTP/1.1
```

The URI is as shown in the screenshot

*Q32. What HTTP version is your browser running? What are the other versions of HTTP?*



Wireshark · Packet 787 · Ethernet

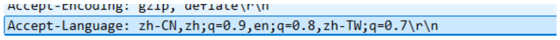
```
> Transmission Control Protocol, Src Port: 80, Dst Port: 62375, Seq: 2178, Ack: 783, Len: 584
> [2 Reassembled TCP Segments (1964 bytes): #786(1380), #787(584)]
v Hypertext Transfer Protocol
  > HTTP/1.1 206 Partial Content\r\n
    Cache-Control: public, max-age=17280000\r\n
    Content-Type: application/x-chrome-extension\r\n
    Last-Modified: Tue. 08 Mar 2022 02:44:10 GMT\r\n

0000  4c d7 17 90 7f 4c 3c 08  f6 21 0a 00 08 00 45 00   L····L<··!····E·
0010  02 70 7a e3 40 00 36 06  22 19 17 3a 7f 63 0a 45   ·pz·@·6·"··:·c·E
0020  04 aa 00 50 f3 a7 5f d2  a5 17 73 a5 e1 33 50 18   ···P··_···s··3P·
0030  01 f5 a8 a3 00 00 65 bd  be 62 60 9d 8c 9f 0a 91   ······e··b`·····
0040  86 46 c5 fb 17 87 31 b5  df d3 db a0 0b 54 50 1f   ·F····1······TP·
0050  6f a2 f3 34 c6 20 43 8d  b6 12 ac 04 0a a6 02 30   o··4· C········0
0060  82 01 22 30 0d 06 09 2a  86 48 86 f7 0d 01 01 01   ··"0···*·H······
0070  05 00 03 82 01 0f 00 30  82 01 0a 02 82 01 01 00   ·······0········
```

Frame (638 bytes) | Reassembled TCP (1964 bytes)

The HTTP version is HTTP/1.1

The other versions of HTTP include HTTP/0.9, HTTP/1.0, HTTP/1.1, HTTP/2, and HTTP/3.

*Q33. What languages does your browser accept for response?*



```
Accept-Encoding: gzip, deflate\r\n
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,zh-TW;q=0.7\r\n
```

These are the accepted languages.

*Q 34. What is the IP address of your computer?*

The IP of my computer is 10.69.4.170

*Q35. What is the server's IP address?*

The IP address of the server is 34.223.124.45

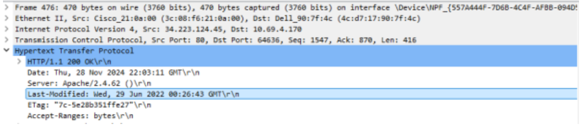
*36. What is the relationship between source and destination IP addresses of the first GET and the source and destination IP addresses of the first Response?*

The first GET's source and destination IP addresses is the first response's destination and source IP addresses.

*37. What is the status code of the first response message? What does this code indicate? What code is returned if the requested file cannot be found on the server?*

The status code is 200. It indicates that the request is successful, and that the content has delivered. If the requested file cannot be found the code will be 404.

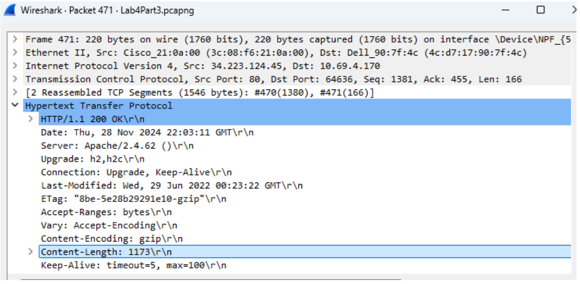
*38. When was the last time that the received HTML file was modified at the server?*



```
> Frame 476: 470 bytes on wire (3760 bits), 470 bytes captured (3760 bits) on interface \Device\NPF_{557A444F-7D6B-4C4F-AFBB-094D
> Ethernet II, Src: Cisco_21:0a:00 (3c:08:f6:21:0a:00), Dst: Dell_90:7f:4c (4c:d7:17:90:7f:4c)
> Internet Protocol Version 4, Src: 34.223.124.45, Dst: 10.69.4.170
> Transmission Control Protocol, Src Port: 80, Dst Port: 64636, Seq: 1547, Ack: 870, Len: 416
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Thu, 28 Nov 2024 22:03:11 GMT\r\n
    Server: Apache/2.4.62 ()\r\n
    Last-Modified: Wed, 29 Jun 2022 00:26:43 GMT\r\n
    ETag: "7c-5e28b351ffe27"\r\n
    Accept-Ranges: bytes\r\n
```

Last-Modified: Wed, 29 Jun 2022 00:26:43 GMT\r\n

*39. What is the size of the content that is returned to your browser?*



Wireshark · Packet 471 · Lab4Part3.pcapng

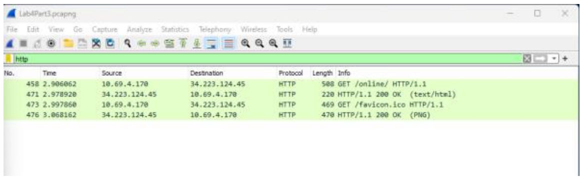
```
> Frame 471: 220 bytes on wire (1760 bits), 220 bytes captured (1760 bits) on interface \Device\NPF_{5
> Ethernet II, Src: Cisco_21:0a:00 (3c:08:f6:21:0a:00), Dst: Dell_90:7f:4c (4c:d7:17:90:7f:4c)
> Internet Protocol Version 4, Src: 34.223.124.45, Dst: 10.69.4.170
> Transmission Control Protocol, Src Port: 80, Dst Port: 64636, Seq: 1381, Ack: 455, Len: 166
> [2 Reassembled TCP Segments (1546 bytes): #470(1380), #471(166)]
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Date: Thu, 28 Nov 2024 22:03:11 GMT\r\n
    Server: Apache/2.4.62 ()\r\n
    Upgrade: h2,h2c\r\n
    Connection: Upgrade, Keep-Alive\r\n
    Last-Modified: Wed, 29 Jun 2022 00:23:22 GMT\r\n
    ETag: "8be-5e28b29291e10-gzip"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
  > Content-Length: 1173\r\n
    Keep-Alive: timeout=5, max=100\r\n
```

The size is 1173 Bytes.

*Q40: How many HTTP GET request messages are sent by your web browser?*

Two GET are sent by my browser.

*41. By inspecting the entire trace, determine the number of packets that contain HTTP header. Explain your answer.*

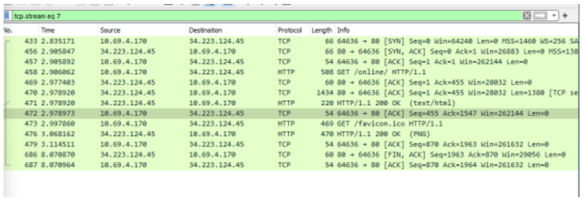

Lab4Part3.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http
```

```
No.     Time        Source           Destination      Protocol  Length Info
    458 2.906062    10.69.4.170      34.223.124.45    HTTP       508 GET /online/ HTTP/1.1
    471 2.978920    34.223.124.45    10.69.4.170      HTTP       220 HTTP/1.1 200 OK  (text/html)
    473 2.997860    10.69.4.170      34.223.124.45    HTTP       469 GET /favicon.ico HTTP/1.1
    476 3.068162    34.223.124.45    10.69.4.170      HTTP       470 HTTP/1.1 200 OK  (PNG)
```

Four packets contain the HTTP header because it is being filtered out by http.

*Q42. How many TCP segments are transmitted to your computer? Why multiple segments are required to retrieve this single HTML file?*



```
tcp.stream eq 7
```

```
No.     Time        Source           Destination      Protocol  Length Info
    433 2.835171    10.69.4.170      34.223.124.45    TCP          66 64636 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SA
    456 2.905847    34.223.124.45    10.69.4.170      TCP          66 80 → 64636 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=138
    457 2.905892    10.69.4.170      34.223.124.45    TCP          54 64636 → 80 [ACK] Seq=1 Ack=1 Win=262144 Len=0
    458 2.906062    10.69.4.170      34.223.124.45    HTTP        508 GET /online/ HTTP/1.1
    469 2.977403    34.223.124.45    10.69.4.170      TCP          60 80 → 64636 [ACK] Seq=1 Ack=455 Win=28032 Len=0
    470 2.978920    34.223.124.45    10.69.4.170      TCP        1434 80 → 64636 [ACK] Seq=1 Ack=455 Win=28032 Len=1380 [TCP se
    471 2.978920    34.223.124.45    10.69.4.170      HTTP        220 HTTP/1.1 200 OK  (text/html)
    472 2.978973    10.69.4.170      34.223.124.45    TCP          54 64636 → 80 [ACK] Seq=455 Ack=1547 Win=262144 Len=0
    473 2.997860    10.69.4.170      34.223.124.45    HTTP        469 GET /favicon.ico HTTP/1.1
    476 3.068162    34.223.124.45    10.69.4.170      HTTP        470 HTTP/1.1 200 OK  (PNG)
    479 3.114511    10.69.4.170      34.223.124.45    TCP          54 64636 → 80 [ACK] Seq=870 Ack=1963 Win=261632 Len=0
    686 8.070870    34.223.124.45    10.69.4.170      TCP          60 80 → 64636 [FIN, ACK] Seq=1963 Ack=870 Win=29056 Len=0
    687 8.070964    10.69.4.170      34.223.124.45    TCP          54 64636 → 80 [ACK] Seq=870 Ack=1964 Win=261632 Len=0
```

9 TCP segments are transmitted to my computer.

The reason behind needing multiple segments is that the TCP handshake and connection management (e.g., SYN, ACK, FIN packets) and segmentation of HTTP headers and body based on the Maximum Segment Size (MSS).

*43. Determine the length of these TCP segments. Do they have the same size? Explain your answer.*

The lengths are as follows:

66 bytes for packets 433 and 456

60 bytes for packets 468 and 486

1434 bytes for packets 470

54 bytes for packets 472, 479 and 687

They do not have the same size. The size of the packets depends on the purpose of the packet. If they are control packets, they tend to be small like those with a length of 54, 60, and 66. If they are data packets, they tend to be bigger like the one with the 1434 bytes.

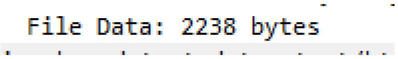
*Q44. Which message and what field in that message indicate that the server was able to process the request successfully?*

The 200 ok in the status code field tells us that indicates the server was able to process the request successfully.

*Q45. What is the status code of the first response message?*

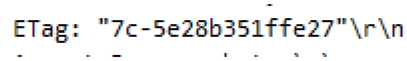
200

*Q46. What is the value of the content size of the first response message?*



```
File Data: 2238 bytes
```

*Q47. What is the etag of the first response message?*

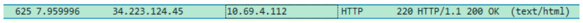


```
ETag: "7c-5e28b351ffe27"\r\n
```

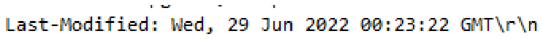
*Q48. What is the application of etag in conditional HTTP request? Which line in the second response contains the etag value of the first response?*

ETags optimize HTTP interactions by enabling conditional requests that validate cached resources, prevent unnecessary updates, and avoid redundant data transfers. This helps improve web performance and ensures data consistency during interactions between the client and server. It is found under Hypertext transfer protocol section.

*Q49. Which HTTP Get contains the IF-MODIFIED_SINCE line? What is the usage of this field?*



```
625 7.959996     34.223.124.45     10.69.4.112     HTTP     220 HTTP/1.1 200 OK  (text/html)
```



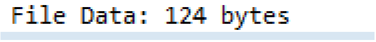
```
Last-Modified: Wed, 29 Jun 2022 00:23:22 GMT\r\n
```

It is used in HTTP GET requests to optimize the retrieval of resources by checking if they have been modified since a specific date and time.

*Q50. What is the status code of the second response message? What does this code mean?*

200, it is a three digit number that is included in the server's response to a client's request. It indicates the result of the request and provides information about how client should handle the request. 200 represent that the request was successful, and the server is returning the requested resource.

*Q51. What is the content length of the second response? Explain.*



```
File Data: 124 bytes
```

This information does not contain any header. It's the smallest and content only.

*Q52. How many HTTP GET Requests are sent by your web browser?*

2 request

*Q53. What is the content type of each response message?*

Text/html, and PNG

*Q54. Did your browser download the two images serially or in parallel? Explain. What are the pros and cons of each approach.*

There is only 1 image in this web. The browser can download image both serially and in parallel.

Serial:

Pros: Lower resource usage, images downloaded in a predictable order, reduce the risk of overwhelming the network

Cons: Slower total completion time, inefficient use of resources

Parallel:

Pros: faster overall download time, better resource utilization, resilient to slow connections

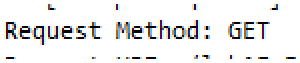
Cons: higher resource usage, potential server overload, complex error handling, image may not download in the desired order.

*Q55. Has the HTTP used persistent or non-persistent connection? Explain your answer.*

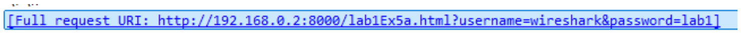
HTTP use persistent connection. TCP connection reused for multiple requests. It is reflected through source and destination ports on Wireshark.

*Q56. What is the requested URL in the frame #101? What HTTP field contains the username and password information? What are the submitted values for the username and password?*

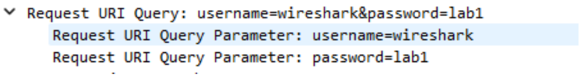
*Request URI Query contains the information*



```
Request Method: GET
```

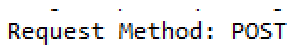


```
[Full request URI: http://192.168.0.2:8000/lab1Ex5a.html?username=wireshark&password=lab1]
```

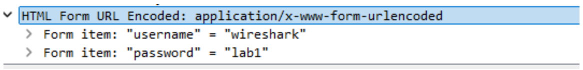


```
˅ Request URI Query: username=wireshark&password=lab1
     Request URI Query Parameter: username=wireshark
     Request URI Query Parameter: password=lab1
```

*Q57. What HTTP request method is used in the frame #172? What HTTP field contains the username and password information? Explain the difference between this request method and the GET method.*



```
Request Method: POST
```

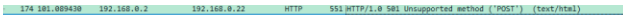


```
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
   > Form item: "username" = "wireshark"
   > Form item: "password" = "lab1"
```

GET fetch data from the server. Data located in URL query string. It is safe. Responses can be cached. Data is visible in URL. Data size is limited by URL length.

POST send data to the server. Data locates in HTTP request body. It is not safe. Responses typically not cached. Data is hidden in request body. There is no practical data size limit.

*Q58. What is the status code of the frame #174? What is the description of this code?*



```
174 101.089430   192.168.0.2   192.168.0.22   HTTP   551 HTTP/1.0 501 Unsupported method ('POST')  (text/html)
```

501 Not Implemented indicates that the server lacks the ability to perform the requested operation because the required functionality is not implemented.

## Conclusion

This lab effectively demonstrated the functionality and interaction of DNS, UDP, and HTTP protocols within network communication. Through hands-on analysis, we observed how DNS resolves domain names to IP addresses, facilitated by UDP for its efficiency and speed. Wireshark packet traces highlighted the headers, fields, and payloads involved in these exchanges, reinforcing the concepts of protocol design and reliability. In the HTTP section, we explored GET and POST methods, status codes, and resource transfer efficiency. This lab underscored the significance of these protocols in enabling reliable and efficient internet communication while exposing their respective strengths, limitations, and trade-offs in real-world applications.