CS-301

Computer Networks Lab Report

Assignment - 3

BTech 5th Semester 2021

Department of Computer Science and Engineering,

National Institute of Technology Karnataka, Surathkal

October 2021

Submitted By:
Deepta Devkota - 191CS117
Akanksha More - 191CS106

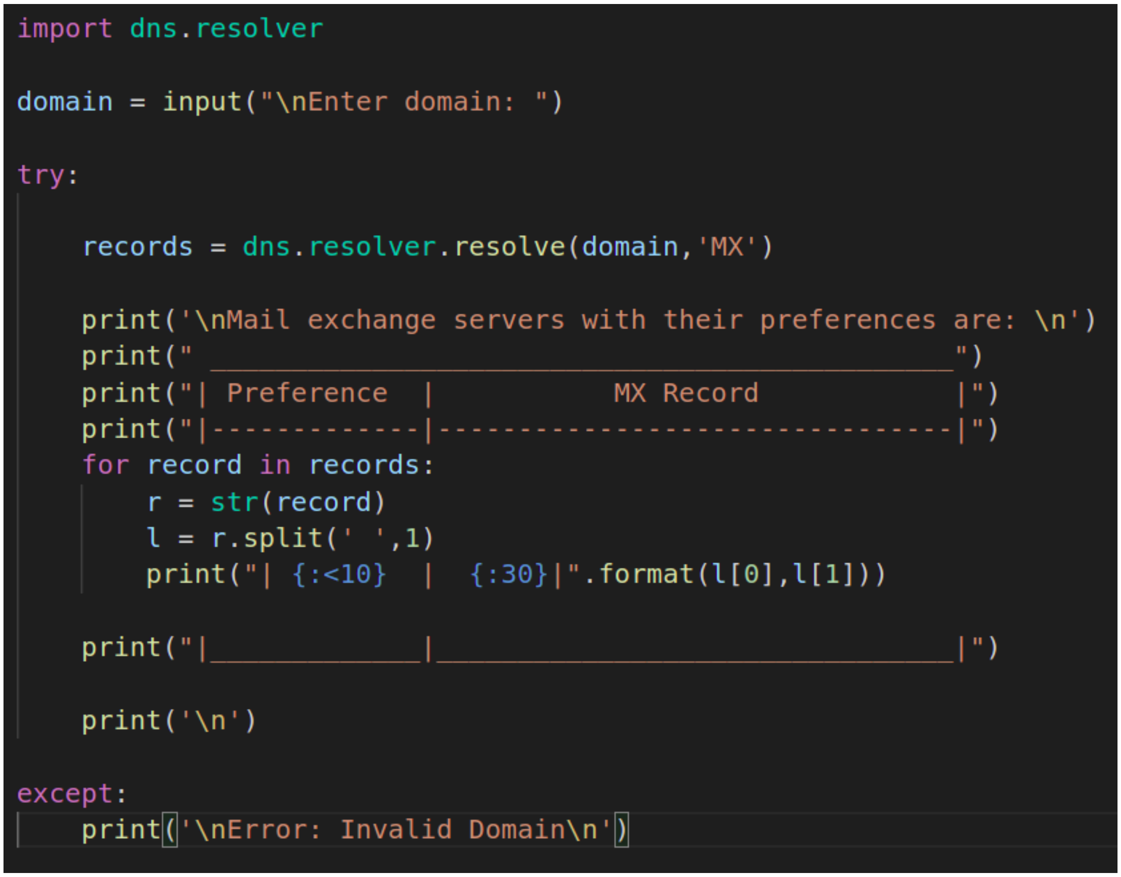
Q1. Develop a program to print the Mail exchange servers of a particular domain with their preferences.

The **MX-record (Mail eXchange-record)** in the Domain Name System (DNS) stores the mail exchange servers of a particular domain. MX records allow the hostnames of mail servers to have simple aliases. By using the MX record, a company can have the same aliased name for its mail server and for one of its other servers as web servers. To obtain the canonical name for the mail server, a DNS client would query for an MX record.

The preference is used when more than one MX record is entered for any single domain name that is using more than one mail server. In this case, the reference number indicates the order in which the mail servers should be used. This enables the use of primary and backup mail servers.

It is a way to set load sharing and priority between multiple mail servers for a domain. The lower the preference number is the higher priority. Two MX records with the same priority will share the workload. The server with the higher preference number will be contacted only if the servers with lower preference numbers are unavailable (this is typically used for backup mail servers).

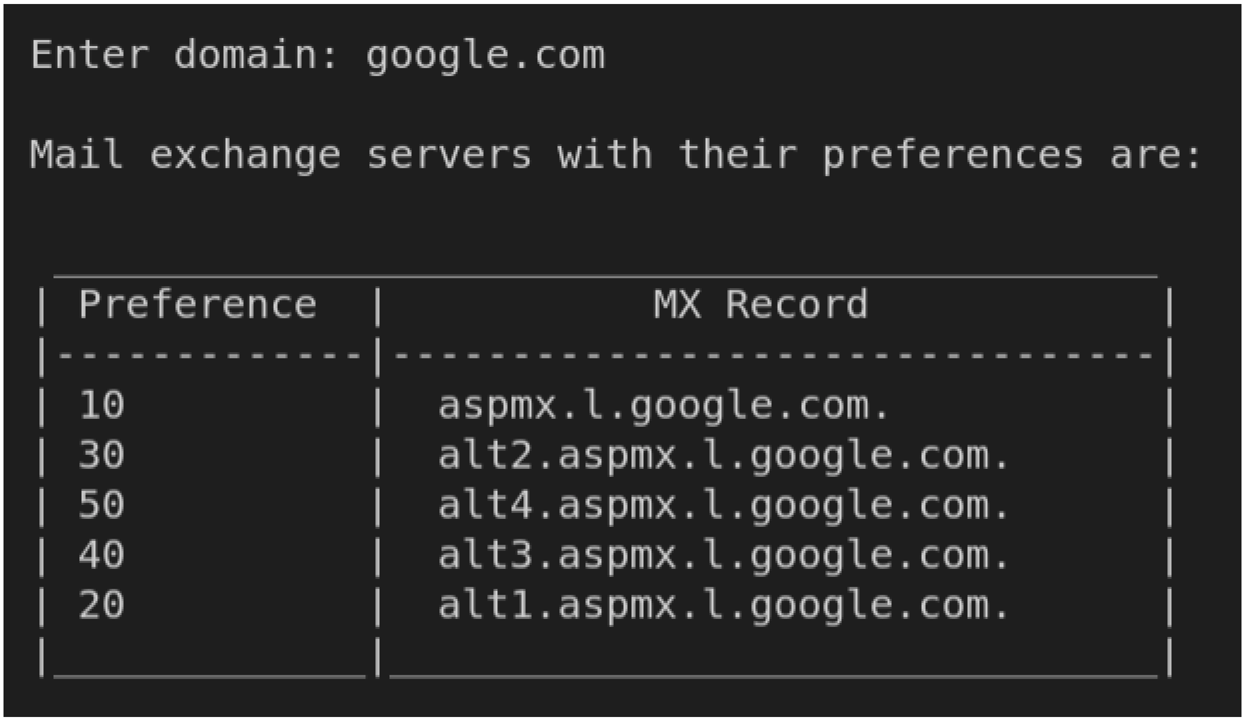
We used the dns.resolver package of python, to resolve the query to get the mail exchange server. After executing the query we get the list of the mail exchange server along with the preference number.



```python
import dns.resolver

domain = input("\nEnter domain: ")

try:

    records = dns.resolver.resolve(domain,'MX')

    print('\nMail exchange servers with their preferences are: \n')
    print(" ______________________________________________")
    print("| Preference  |           MX Record            |")
    print("|-------------|--------------------------------|")
    for record in records:
        r = str(record)
        l = r.split(' ',1)
        print("| {:<10}  |  {:30}|".format(l[0],l[1]))

    print("|_____________|________________________________|")

    print('\n')

except:
    print('\nError: Invalid Domain\n')
```

Fig: Program to get the mail exchange server of a domain

Output:



```
Enter domain: google.com

Mail exchange servers with their preferences are:


  ______________________________________________
 | Preference  |           MX Record            |
 |-------------|--------------------------------|
 | 10          |  aspmx.l.google.com.           |
 | 30          |  alt2.aspmx.l.google.com.      |
 | 50          |  alt4.aspmx.l.google.com.      |
 | 40          |  alt3.aspmx.l.google.com.      |
 | 20          |  alt1.aspmx.l.google.com.      |
 |_____________|________________________________|
```

Q2. Use nslookup and ipconfig commands for finding various network related information.

ipconfig:

The **Internet Protocol Configuration** (ipconfig) command is used to display the current network information such as TCP/IP network configuration, IP and MAC address of the device, IP address of the router, information about DHCP and DNS servers.

As in the image, we can observe that the network connectivity is using WiFi. It displays the following information:

IPV6 Address: 128-bit IP address used by the network connection.

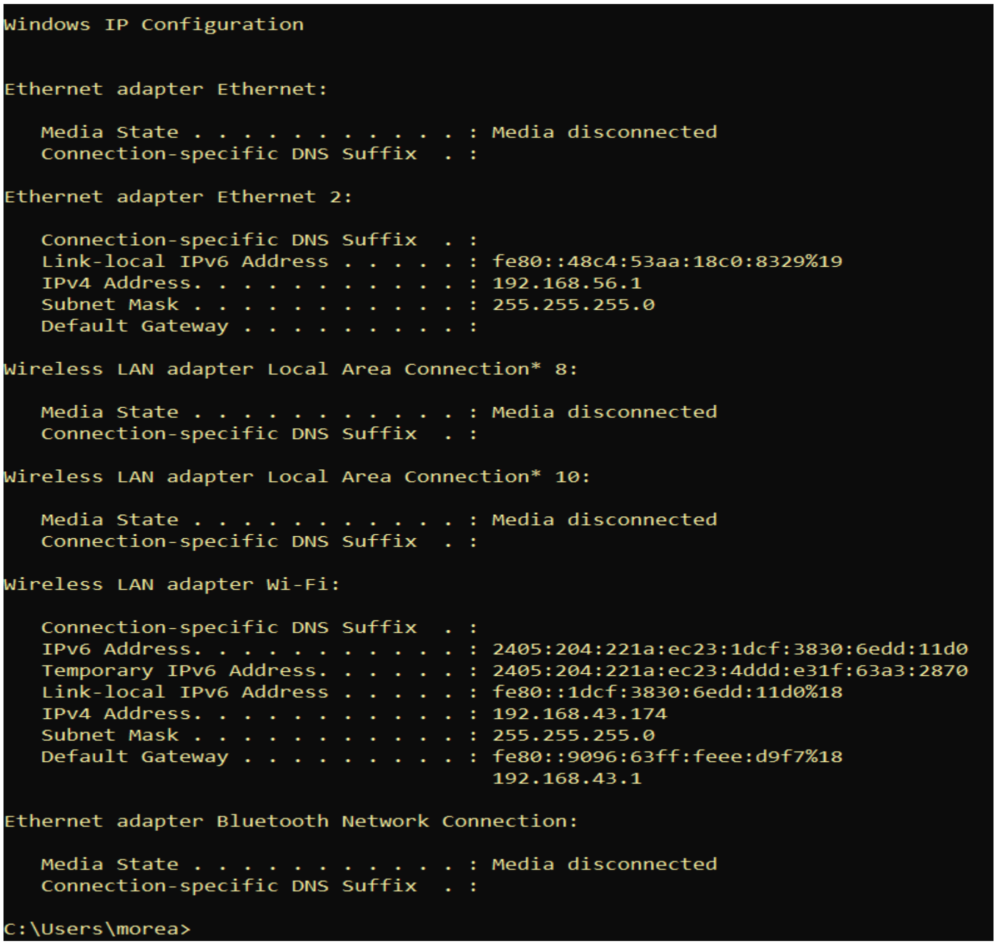
Temporary IPV6 Address: Randomly generated address as the interface ID instead of an interface's MAC address.

Link-local IPV6 Address: Address used to communicate with hosts and routers on an attached link.

IPV4 Address: 128-bit IP address used by the network connection.

Subnet Mask: Divides the IP address into two parts to identify the host and network.

Default Gateway: Router or switch the connection goes through.



```
Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::48c4:53aa:18c0:8329%19
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 8:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2405:204:221a:ec23:1dcf:3830:6edd:11d0
   Temporary IPv6 Address. . . . . . : 2405:204:221a:ec23:4ddd:e31f:63a3:2870
   Link-local IPv6 Address . . . . . : fe80::1dcf:3830:6edd:11d0%18
   IPv4 Address. . . . . . . . . . . : 192.168.43.174
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::9096:63ff:feee:d9f7%18
                                       192.168.43.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\morea>
```

Fig: ipconfig command output

ipconfig/all:

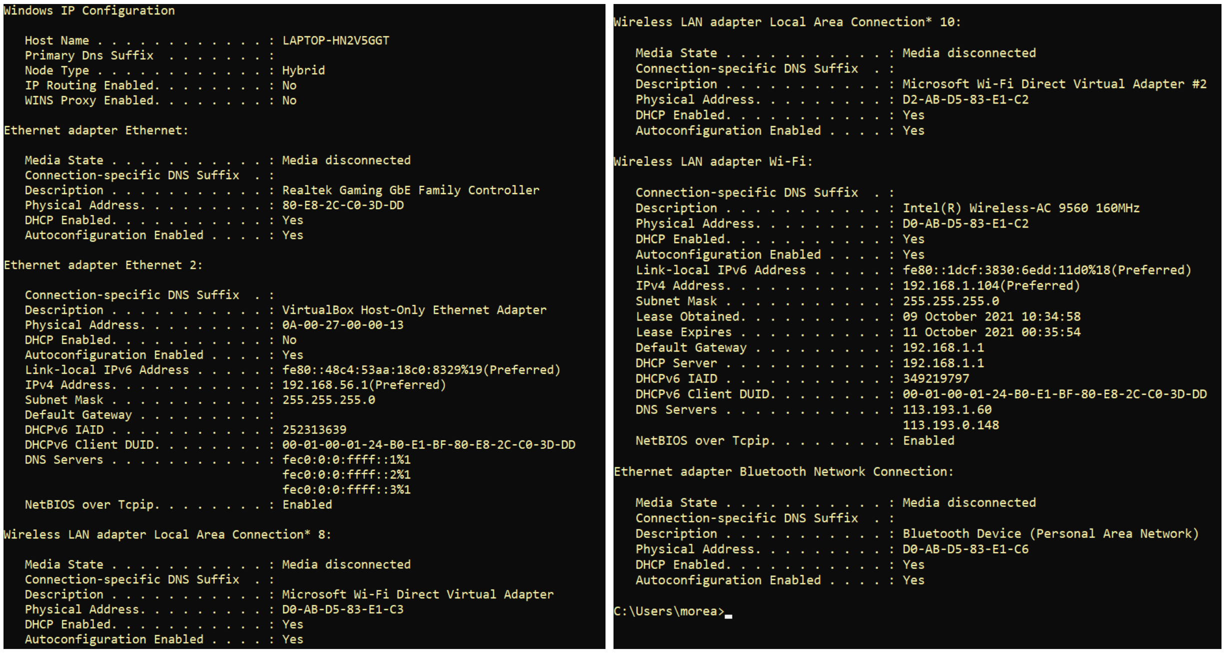
This command is used to display detailed information about TCP/IP configuration for all adapters along with DHCP configuration and DNS servers.

Additional information displayed:

Host Name: Name of the computer on the network.

DHCP server: Server that hands out IP addresses based on a DHCP protocol.

DNS server: Domain name server that translates domain names to IP address.

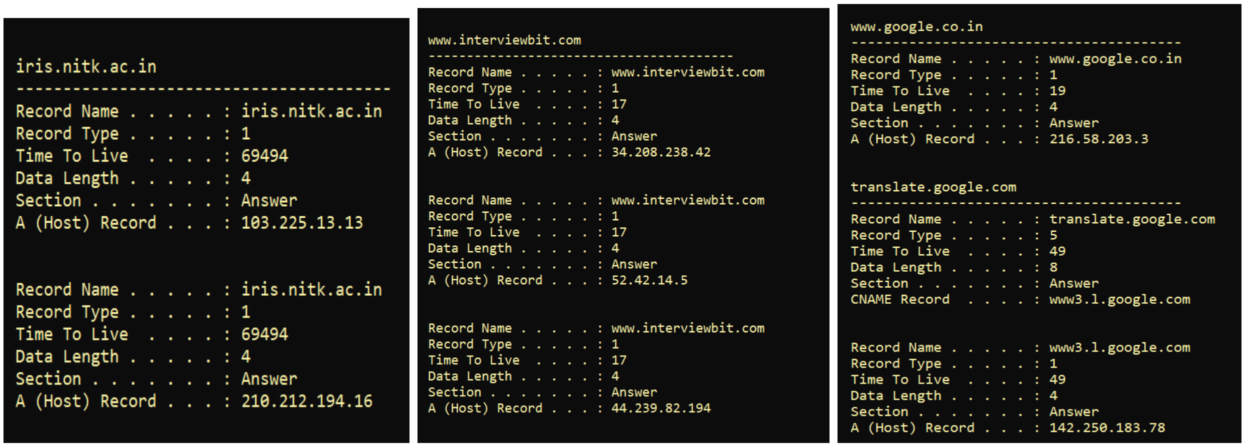


```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : LAPTOP-HN2V5GGT
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek Gaming GbE Family Controller
   Physical Address. . . . . . . . . : 80-E8-2C-C0-3D-DD
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter
   Physical Address. . . . . . . . . : 0A-00-27-00-00-13
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::48c4:53aa:18c0:8329%19(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.56.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 252313639
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-B0-E1-BF-80-E8-2C-C0-3D-DD
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Wireless LAN adapter Local Area Connection* 8:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   Physical Address. . . . . . . . . : D0-AB-D5-83-E1-C3
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
```

```
Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   Physical Address. . . . . . . . . : D2-AB-D5-83-E1-C2
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Wireless-AC 9560 160MHz
   Physical Address. . . . . . . . . : D0-AB-D5-83-E1-C2
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::1dcf:3830:6edd:11d0%18(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.1.104(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 09 October 2021 10:34:58
   Lease Expires . . . . . . . . . . : 11 October 2021 00:35:54
   Default Gateway . . . . . . . . . : 192.168.1.1
   DHCP Server . . . . . . . . . . . : 192.168.1.1
   DHCPv6 IAID . . . . . . . . . . . : 349219797
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-B0-E1-BF-80-E8-2C-C0-3D-DD
   DNS Servers . . . . . . . . . . . : 113.193.1.60
                                       113.193.0.148
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Bluetooth Device (Personal Area Network)
   Physical Address. . . . . . . . . : D0-AB-D5-83-E1-C6
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

C:\Users\morea>
```

ipconfig/displaydns:

This command is used to find all the items in local DNS resolver cache which uses local hosts file as well as recently obtained resource records for name queries resolved by the host.



```
iris.nitk.ac.in
----------------------------------------
Record Name . . . . . : iris.nitk.ac.in
Record Type . . . . . : 1
Time To Live  . . . . : 69494
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 103.225.13.13


Record Name . . . . . : iris.nitk.ac.in
Record Type . . . . . : 1
Time To Live  . . . . : 69494
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 210.212.194.16
```

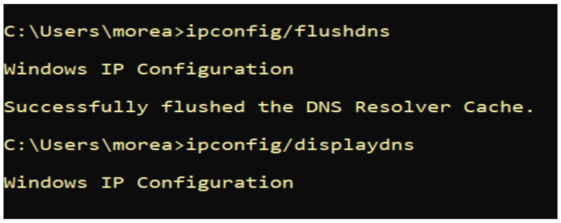
```
www.interviewbit.com
----------------------------------------
Record Name . . . . . : www.interviewbit.com
Record Type . . . . . : 1
Time To Live  . . . . : 17
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 34.208.238.42


Record Name . . . . . : www.interviewbit.com
Record Type . . . . . : 1
Time To Live  . . . . : 17
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 52.42.14.5


Record Name . . . . . : www.interviewbit.com
Record Type . . . . . : 1
Time To Live  . . . . : 17
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 44.239.82.194
```

```
www.google.co.in
----------------------------------------
Record Name . . . . . : www.google.co.in
Record Type . . . . . : 1
Time To Live  . . . . : 19
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 216.58.203.3


translate.google.com
----------------------------------------
Record Name . . . . . : translate.google.com
Record Type . . . . . : 5
Time To Live  . . . . : 49
Data Length . . . . . : 8
Section . . . . . . . : Answer
CNAME Record  . . . . : www3.l.google.com


Record Name . . . . . : www3.l.google.com
Record Type . . . . . : 1
Time To Live  . . . . : 49
Data Length . . . . . : 4
Section . . . . . . . : Answer
A (Host) Record . . . : 142.250.183.78
```

Fig: ipconfig/displaydns command output

ipconfig/flushdns:

This command is used to flush and reset the DNS resolver cache. It helps while troubleshooting by removing any negative cache entries.



```
C:\Users\morea>ipconfig/flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\morea>ipconfig/displaydns

Windows IP Configuration
```

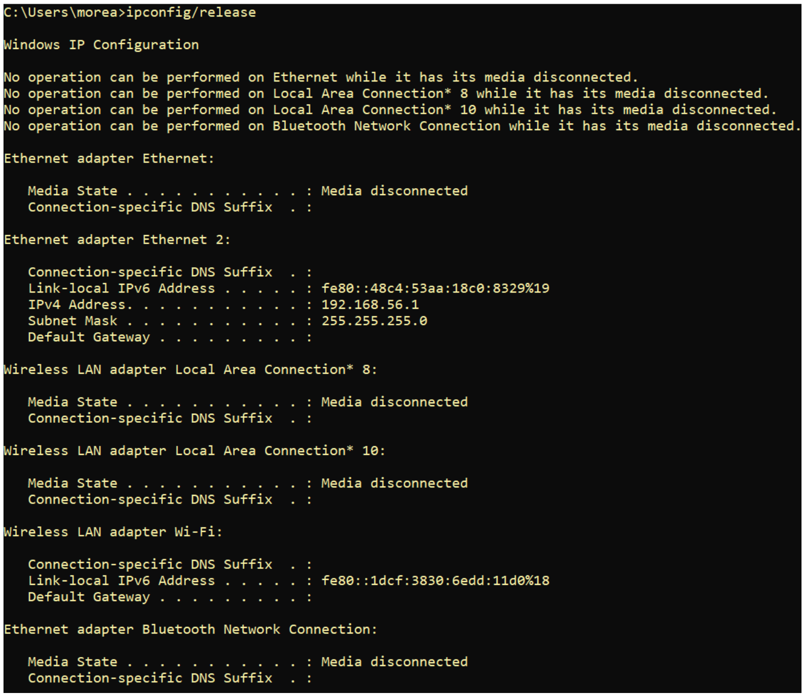
Fig: ipconfig/flushdns command output

ipconfig/registerdns:

This command is used for troubleshooting DNS name registration issues without rebooting the computer.

ipconfig/release:

This command is used to release and renew a dynamically assigned (DHCP) IP address.

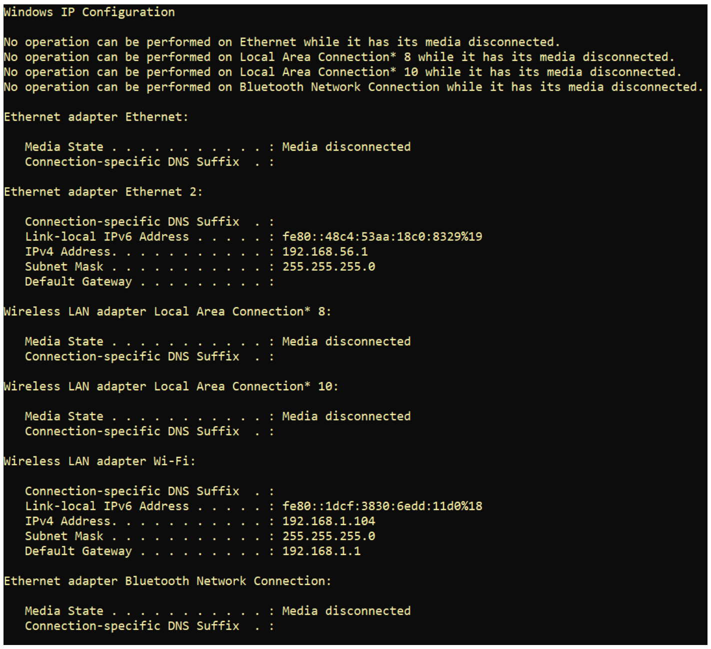


```
C:\Users\morea>ipconfig/release

Windows IP Configuration

No operation can be performed on Ethernet while it has its media disconnected.
No operation can be performed on Local Area Connection* 8 while it has its media disconnected.
No operation can be performed on Local Area Connection* 10 while it has its media disconnected.
No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::48c4:53aa:18c0:8329%19
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 8:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::1dcf:3830:6edd:11d0%18
   Default Gateway . . . . . . . . . :

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Fig: ipconfig/release command output

ipconfig/renew:

This command is used to renew DHCP configuration for all adapters.



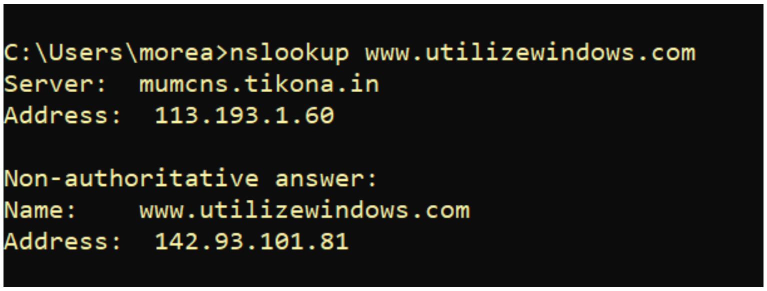
```
Windows IP Configuration

No operation can be performed on Ethernet while it has its media disconnected.
No operation can be performed on Local Area Connection* 8 while it has its media disconnected.
No operation can be performed on Local Area Connection* 10 while it has its media disconnected.
No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::48c4:53aa:18c0:8329%19
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 8:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 10:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::1dcf:3830:6edd:11d0%18
   IPv4 Address. . . . . . . . . . . : 192.168.1.104
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Fig: ipconfig/renew command output

nslookup:

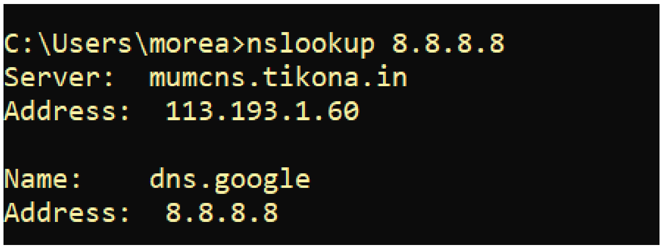
The **Name Server Lookup** (nslookup) command is used for getting information from the DNS server. It allows us to query the **Domain Name System** (DNS) to obtain domain name or IP address mapping. It can also be used for resolving any other DNS related

1)Getting the IP address from the domain name.



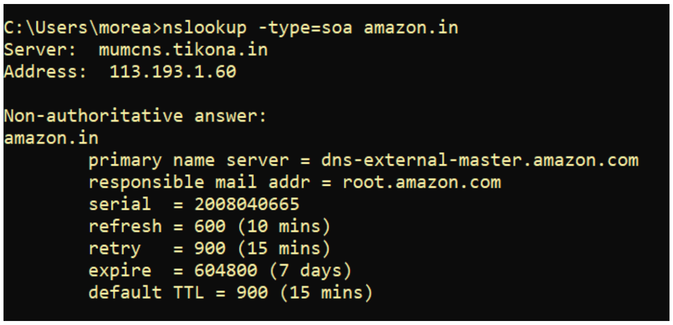
```
C:\Users\morea>nslookup www.utilizewindows.com
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
Name:    www.utilizewindows.com
Address:  142.93.101.81
```

2)Getting the domain name from the IP address.



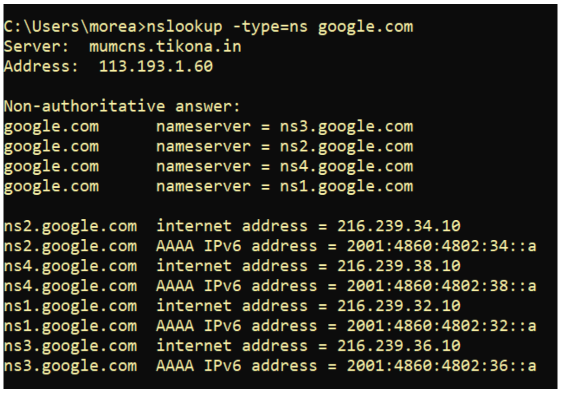
```
C:\Users\morea>nslookup 8.8.8.8
Server:  mumcns.tikona.in
Address:  113.193.1.60

Name:    dns.google
Address:  8.8.8.8
```

3)Getting authoritative information about the domain.



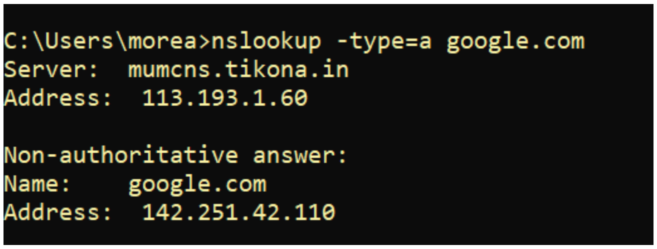
```
C:\Users\morea>nslookup -type=soa amazon.in
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
amazon.in
        primary name server = dns-external-master.amazon.com
        responsible mail addr = root.amazon.com
        serial  = 2008040665
        refresh = 600 (10 mins)
        retry   = 900 (15 mins)
        expire  = 604800 (7 days)
        default TTL = 900 (15 mins)
```

4)Getting the name servers which are associated with the domain.



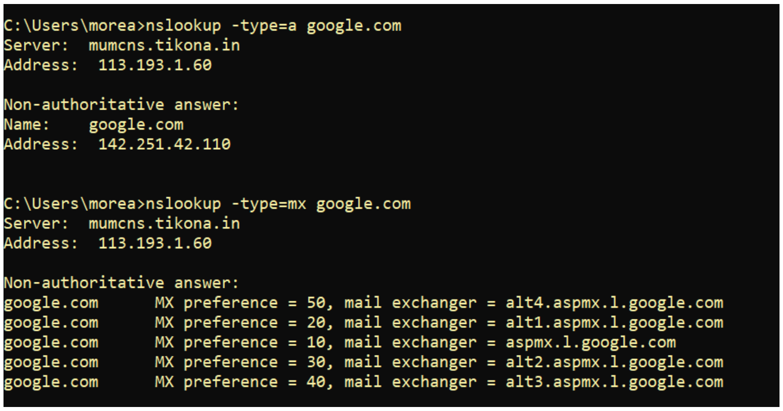
```
C:\Users\morea>nslookup -type=ns google.com
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
google.com      nameserver = ns3.google.com
google.com      nameserver = ns2.google.com
google.com      nameserver = ns4.google.com
google.com      nameserver = ns1.google.com

ns2.google.com  internet address = 216.239.34.10
ns2.google.com  AAAA IPv6 address = 2001:4860:4802:34::a
ns4.google.com  internet address = 216.239.38.10
ns4.google.com  AAAA IPv6 address = 2001:4860:4802:38::a
ns1.google.com  internet address = 216.239.32.10
ns1.google.com  AAAA IPv6 address = 2001:4860:4802:32::a
ns3.google.com  internet address = 216.239.36.10
ns3.google.com  AAAA IPv6 address = 2001:4860:4802:36::a
```

5)Getting all DNS records for a particular record.



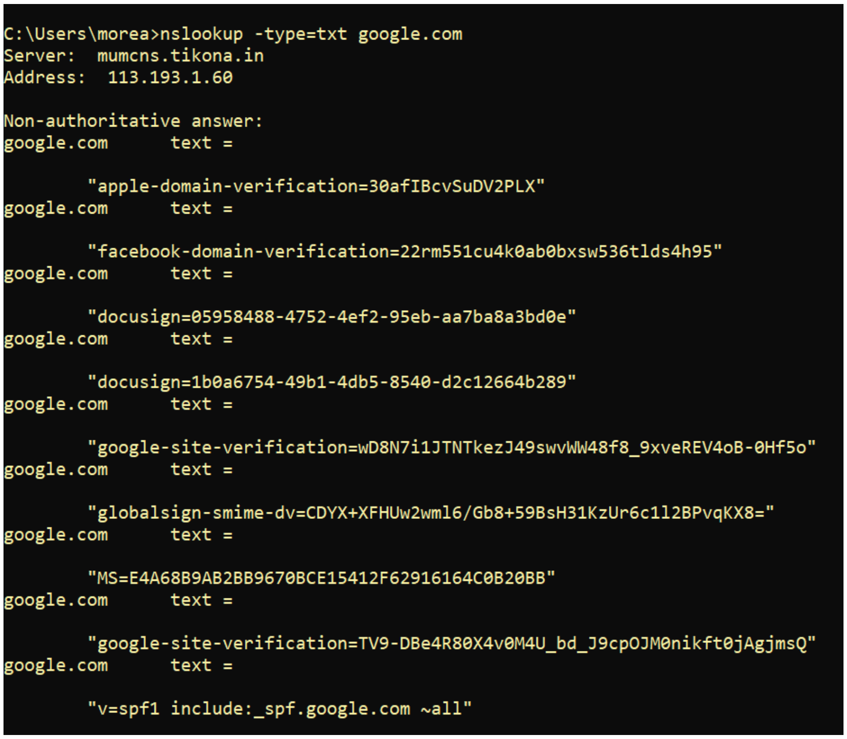
```
C:\Users\morea>nslookup -type=a google.com
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
Name:    google.com
Address:  142.251.42.110
```

6)Getting all main exchange servers for the given domain.



```
C:\Users\morea>nslookup -type=a google.com
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
Name:    google.com
Address:  142.251.42.110


C:\Users\morea>nslookup -type=mx google.com
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
google.com      MX preference = 50, mail exchanger = alt4.aspmx.l.google.com
google.com      MX preference = 20, mail exchanger = alt1.aspmx.l.google.com
google.com      MX preference = 10, mail exchanger = aspmx.l.google.com
google.com      MX preference = 30, mail exchanger = alt2.aspmx.l.google.com
google.com      MX preference = 40, mail exchanger = alt3.aspmx.l.google.com
```

7)Getting all TXT configured records for the given domain.



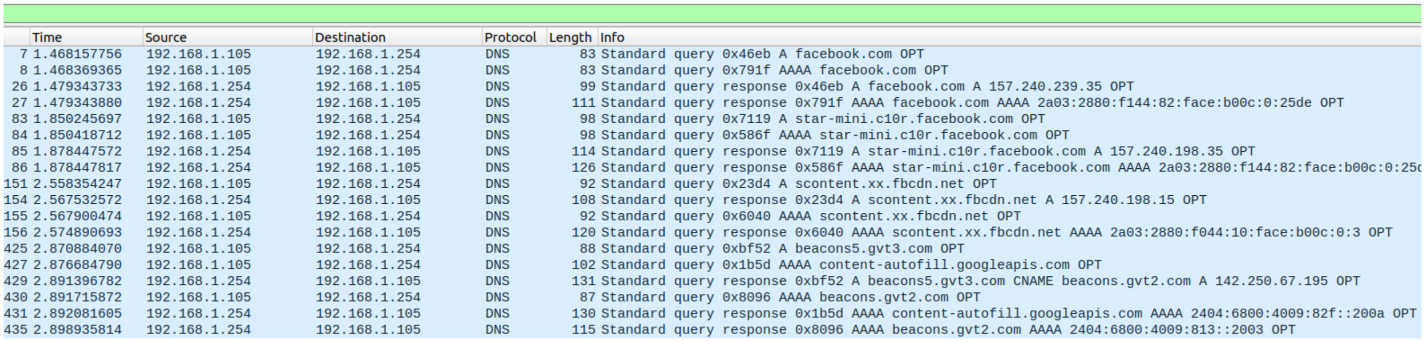
```
C:\Users\morea>nslookup -type=txt google.com
Server:  mumcns.tikona.in
Address:  113.193.1.60

Non-authoritative answer:
google.com      text =

        "apple-domain-verification=30afIBcvSuDV2PLX"
google.com      text =

        "facebook-domain-verification=22rm551cu4k0ab0bxsw536tlds4h95"
google.com      text =

        "docusign=05958488-4752-4ef2-95eb-aa7ba8a3bd0e"
google.com      text =

        "docusign=1b0a6754-49b1-4db5-8540-d2c12664b289"
google.com      text =

        "google-site-verification=wD8N7i1JTNTkezJ49swvWW48f8_9xveREV4oB-0Hf5o"
google.com      text =

        "globalsign-smime-dv=CDYX+XFHUw2wml6/Gb8+59BsH31KzUr6c1l2BPvqKX8="
google.com      text =

        "MS=E4A68B9AB2BB9670BCE15412F62916164C0B20BB"
google.com      text =

        "google-site-verification=TV9-DBe4R80X4v0M4U_bd_J9cpOJM0nikft0jAgjmsQ"
google.com      text =

        "v=spf1 include:_spf.google.com ~all"
```

QN 3. Capture and Analyse DNS Packets using Wireshark.

a. Analyze DNS Query and Response Packets.

After filtering the DNS packets using Wireshark, we get the response and request packets. The DNS queries below are obtained for the domain name ‘www.facebook.com’. Each packet the request and the response have a time, source, and destination address associated with it.

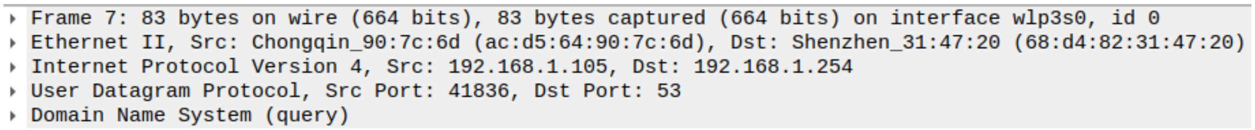
Each query has a code that helps to identify the response message for that query, for example in the very first query the code is 0x46eb and the corresponding response has the same code.



```
    Time         Source          Destination     Protocol Length Info
  7 1.468157756  192.168.1.105   192.168.1.254   DNS        83 Standard query 0x46eb A facebook.com OPT
  8 1.468369365  192.168.1.105   192.168.1.254   DNS        83 Standard query 0x791f AAAA facebook.com OPT
 26 1.479343733  192.168.1.254   192.168.1.105   DNS        99 Standard query response 0x46eb A facebook.com A 157.240.239.35 OPT
 27 1.479343880  192.168.1.254   192.168.1.105   DNS       111 Standard query response 0x791f AAAA facebook.com AAAA 2a03:2880:f144:82:face:b00c:0:25de OPT
 83 1.850245697  192.168.1.105   192.168.1.254   DNS        98 Standard query 0x7119 A star-mini.c10r.facebook.com OPT
 84 1.850418712  192.168.1.105   192.168.1.254   DNS        98 Standard query 0x586f AAAA star-mini.c10r.facebook.com OPT
 85 1.878447572  192.168.1.254   192.168.1.105   DNS       114 Standard query response 0x7119 A star-mini.c10r.facebook.com A 157.240.198.35 OPT
 86 1.878447817  192.168.1.254   192.168.1.105   DNS       126 Standard query response 0x586f AAAA star-mini.c10r.facebook.com AAAA 2a03:2880:f144:82:face:b00c:0:25d
151 2.558354247  192.168.1.105   192.168.1.254   DNS        92 Standard query 0x23d4 A scontent.xx.fbcdn.net OPT
154 2.567532572  192.168.1.254   192.168.1.105   DNS       108 Standard query response 0x23d4 A scontent.xx.fbcdn.net A 157.240.198.15 OPT
155 2.567900474  192.168.1.105   192.168.1.254   DNS        92 Standard query 0x6040 AAAA scontent.xx.fbcdn.net OPT
156 2.574890693  192.168.1.254   192.168.1.105   DNS       120 Standard query response 0x6040 AAAA scontent.xx.fbcdn.net AAAA 2a03:2880:f044:10:face:b00c:0:3 OPT
425 2.870884070  192.168.1.105   192.168.1.254   DNS        88 Standard query 0xbf52 A beacons5.gvt3.com OPT
427 2.876684790  192.168.1.105   192.168.1.254   DNS       102 Standard query 0x1b5d AAAA content-autofill.googleapis.com OPT
429 2.891396782  192.168.1.254   192.168.1.105   DNS       131 Standard query response 0xbf52 A beacons5.gvt3.com CNAME beacons.gvt2.com A 142.250.67.195 OPT
430 2.891715872  192.168.1.105   192.168.1.254   DNS        87 Standard query 0x8096 AAAA beacons.gvt2.com OPT
431 2.892081605  192.168.1.254   192.168.1.105   DNS       130 Standard query response 0x1b5d AAAA content-autofill.googleapis.com AAAA 2404:6800:4009:82f::200a OPT
435 2.898935814  192.168.1.254   192.168.1.105   DNS       115 Standard query response 0x8096 AAAA beacons.gvt2.com AAAA 2404:6800:4009:813::2003 OPT
```

Analyzing the DNS query packet

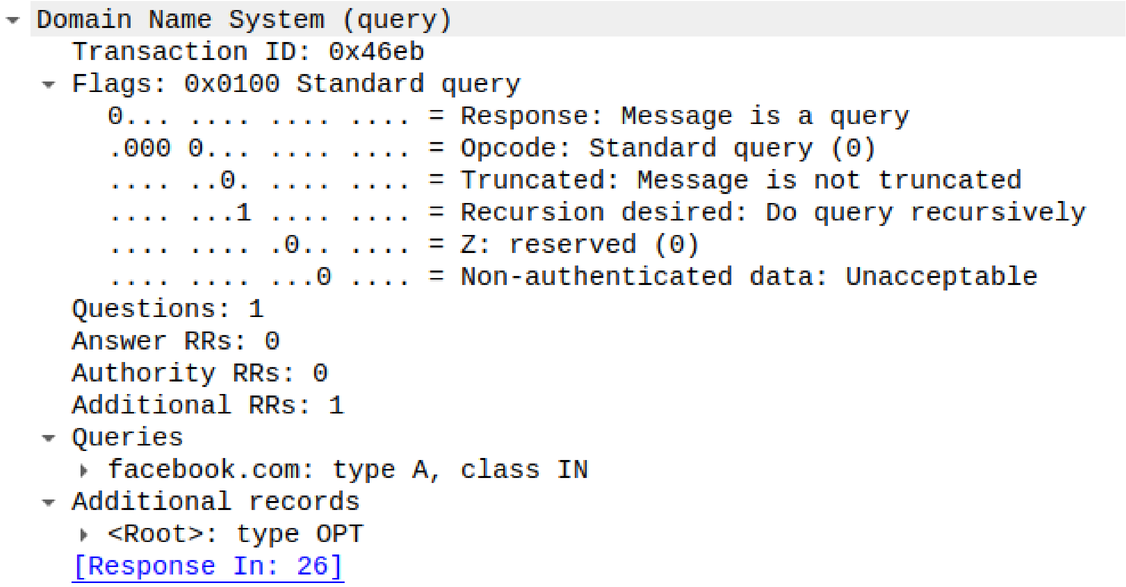
The DNS request packet has the following fields:



```
▸ Frame 7: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface wlp3s0, id 0
▸ Ethernet II, Src: Chongqin_90:7c:6d (ac:d5:64:90:7c:6d), Dst: Shenzhen_31:47:20 (68:d4:82:31:47:20)
▸ Internet Protocol Version 4, Src: 192.168.1.105, Dst: 192.168.1.254
▸ User Datagram Protocol, Src Port: 41836, Dst Port: 53
▸ Domain Name System (query)
```

As DNS uses UDP in the transport layer we can observe the UDP as one of the fields.

The attributes in the DNS field of the DNS request packet:



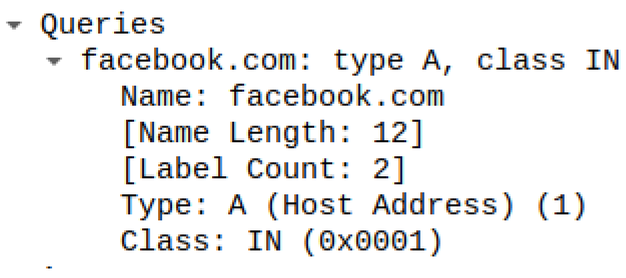
```
▾ Domain Name System (query)
    Transaction ID: 0x46eb
  ▾ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ facebook.com: type A, class IN
  ▾ Additional records
    ▸ <Root>: type OPT
    [Response In: 26]
```

The **Transaction ID** is there to uniquely identify a query.

The **Flags** are there to have additional information about the type of query.

- It helps to know whether the packet is a **response or a query**
- The **opcode** is used to know the type of query (standard or reverse)
- **Truncated** implies whether the packet has been truncated to multiple packets. A packet is truncated because of its large size.
- The **recursion desired** signifies whether the query can be done recursively.

As the packet, we are analyzing is a query message the
Questions: 1, implies that it is a single query message.
The
Answer RRs field is set to zero as this is not a response message.

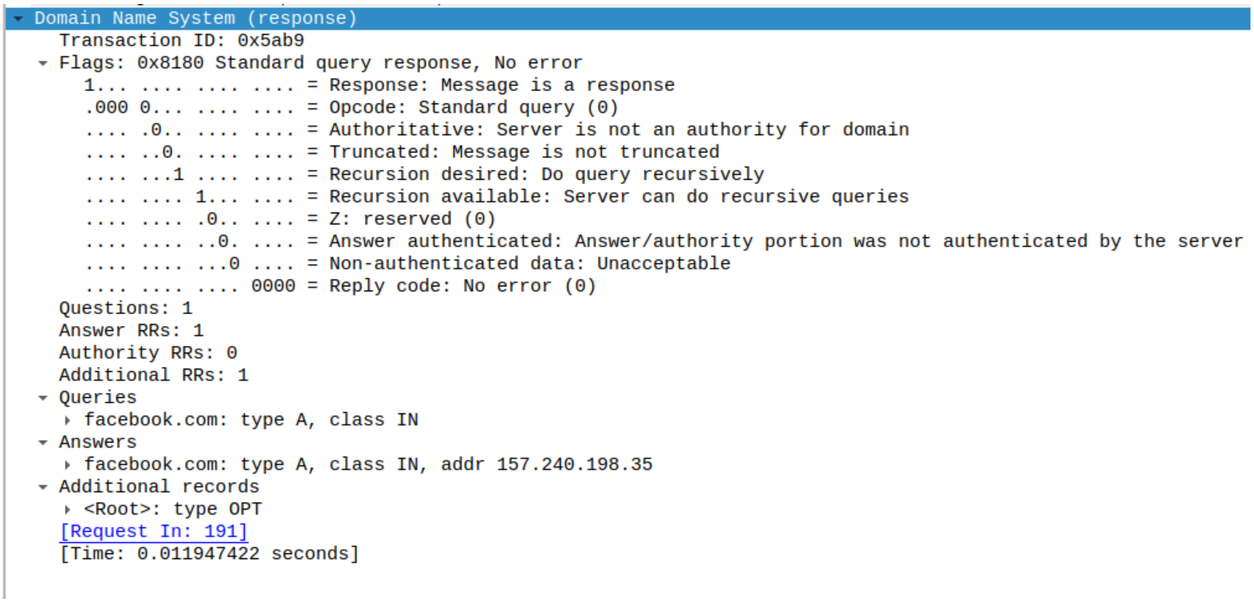


```
▾ Queries
  ▾ facebook.com: type A, class IN
      Name: facebook.com
      [Name Length: 12]
      [Label Count: 2]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
```

The **Queries** field has the query message:
It has the following format:
Query name, **query type**, and the **query class**.
In the above query, our query name is **facebook.com**
The **query type is A**, which is a standard hostname-to-IP address mapping
And the **query class IN** stands for INTERNET

Analyzing the DNS response packet

Additional RRs field implies that the number of additional responses.

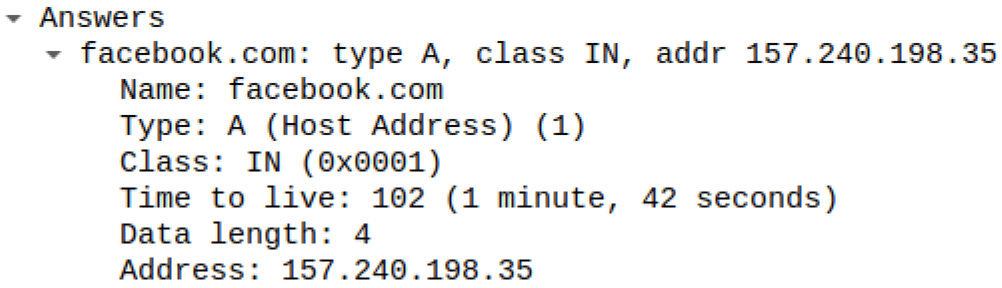


```
▾ Domain Name System (response)
    Transaction ID: 0x5ab9
  ▾ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for 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: 1
    Authority RRs: 0
    Additional RRs: 1
  ▾ Queries
    ▸ facebook.com: type A, class IN
  ▾ Answers
    ▸ facebook.com: type A, class IN, addr 157.240.198.35
  ▾ Additional records
    ▸ <Root>: type OPT
    [Request In: 191]
    [Time: 0.011947422 seconds]
```

As discussed above in the query packet, we have flags in the response packet as well. The **Flags** are there to have additional information about the type of response.

- It helps to know whether the packet is a **response or a query**
- The **opcode** is used to know the type of query (standard or reverse)
- **Authoritative** helps to know whether the server is authoritative for the given domain.
- **Truncated** implies whether the packet has been truncated to multiple packets. A packet is truncated because of its large size.
- The **recursion desired** signifies whether the query is can be done
- The **recursion available** signifies whether the server can perform recursive queries.
- **Answer authenticated** implies whether the answer was authenticated by the server.
- **Reply code: No error**, implies that the DNS query was successfully completed

Answer RRs field is set to one as this packet contains a single response message.
Additional RRs field implies that the number of additional responses.



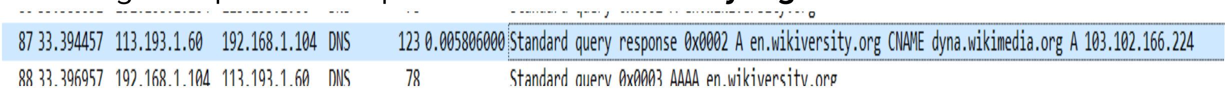
```
▾ Answers
  ▾ facebook.com: type A, class IN, addr 157.240.198.35
      Name: facebook.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 102 (1 minute, 42 seconds)
      Data length: 4
      Address: 157.240.198.35
```

The **Answers** field has the response message:
It has the following format:
Query name, **query type**, **query class,** and the **response**
In the above query, our query name is **facebook.com**
The **query type is A**, which is a standard hostname-to-IP address mapping
And the **query class IN** stands for INTERNET
As the query type is A the **response message** is the corresponding **IP address** for the domain name facebook.com, here the response is **addr : 157.240.198.35**

Q3. Capture and Analyse DNS Packets using Wireshark.

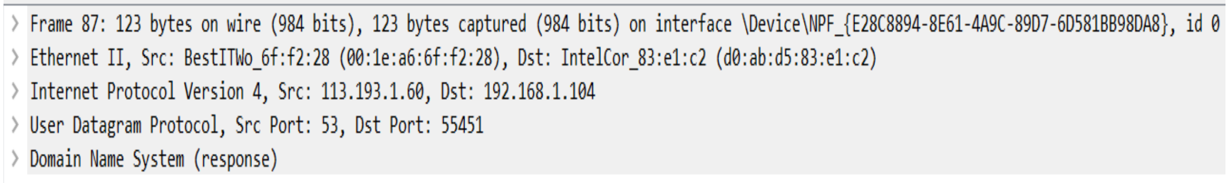
b. By using the captured packets identify the source and destination ports query and response messages.

Following DNS packet is captured for **“en.wikiversity.org”**:



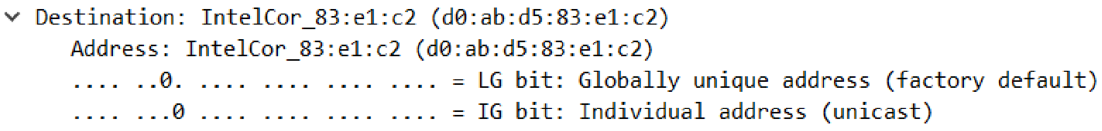
```
87 33.394457  113.193.1.60   192.168.1.104  DNS   123 0.005806000 Standard query response 0x0002 A en.wikiversity.org CNAME dyna.wikimedia.org A 103.102.166.224
88 33.396957  192.168.1.104  113.193.1.60   DNS    78             Standard query 0x0003 AAAA en.wikiversity.org
```

We can observe the following fields for the packet:



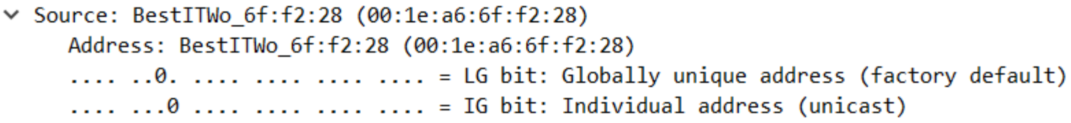
```
> Frame 87: 123 bytes on wire (984 bits), 123 bytes captured (984 bits) on interface \Device\NPF_{E28C8894-8E61-4A9C-89D7-6D581BB98DA8}, id 0
> Ethernet II, Src: BestITWo_6f:f2:28 (00:1e:a6:6f:f2:28), Dst: IntelCor_83:e1:c2 (d0:ab:d5:83:e1:c2)
> Internet Protocol Version 4, Src: 113.193.1.60, Dst: 192.168.1.104
> User Datagram Protocol, Src Port: 53, Dst Port: 55451
> Domain Name System (response)
```

The **destination address** is the MAC address of the device.



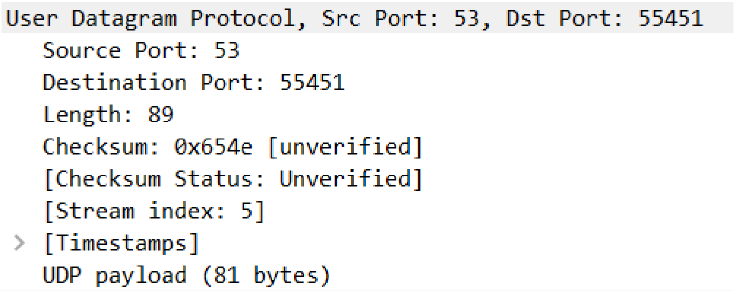
```
v Destination: IntelCor_83:e1:c2 (d0:ab:d5:83:e1:c2)
    Address: IntelCor_83:e1:c2 (d0:ab:d5:83:e1:c2)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

The **source address** is the MAC address of our local DNS server or the default gateway’s MAC address.



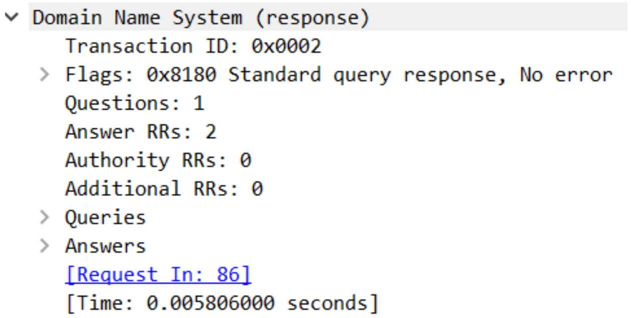
```
v Source: BestITWo_6f:f2:28 (00:1e:a6:6f:f2:28)
    Address: BestITWo_6f:f2:28 (00:1e:a6:6f:f2:28)
    .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

In the **User Datagram Protocol** field, the **source port is the DNS server port** whereas the **destination port is the dynamic port used to make DNS query** in the first packet. Here **Source port is 53** and **destination port is 55451**



```
User Datagram Protocol, Src Port: 53, Dst Port: 55451
   Source Port: 53
   Destination Port: 55451
   Length: 89
   Checksum: 0x654e [unverified]
   [Checksum Status: Unverified]
   [Stream index: 5]
 > [Timestamps]
   UDP payload (81 bytes)
```

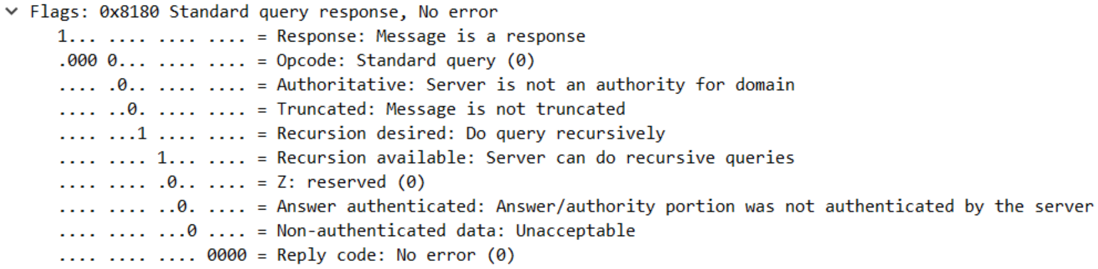
The **Domain Name System** field consists of the following:



```
⌄ Domain Name System (response)
    Transaction ID: 0x0002
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  > Answers
    [Request In: 86]
    [Time: 0.005806000 seconds]
```

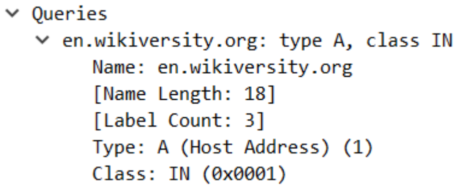
On observing the **Flags**:

1. Bit 1 is set hence its response message.
2. Bits for **Recursion desired** and **Recursion available** are set, hence our query is resolved recursively.



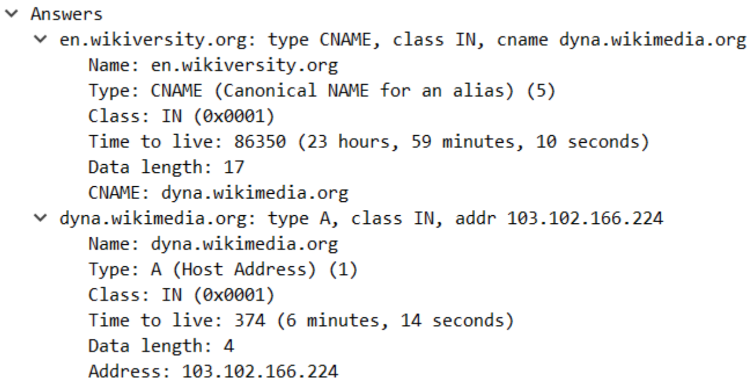
```
⌄ Flags: 0x8180 Standard query response, No error
    1... .... .... .... = Response: Message is a response
    .000 0... .... .... = Opcode: Standard query (0)
    .... .0.. .... .... = Authoritative: Server is not an authority for domain
    .... ..0. .... .... = Truncated: Message is not truncated
    .... ...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)
```

In the **query** section, we can observe the query details corresponding to the response.



```
⌄ Queries
  ⌄ en.wikiversity.org: type A, class IN
      Name: en.wikiversity.org
      [Name Length: 18]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
```

In the **answers** section, we can observe that there are two answers corresponding to the above query. It displays the type of record, time to live (TTL), name and the data length. One answer is of type CNAME which maps to canonical name record while other is of type A which maps to IP address.

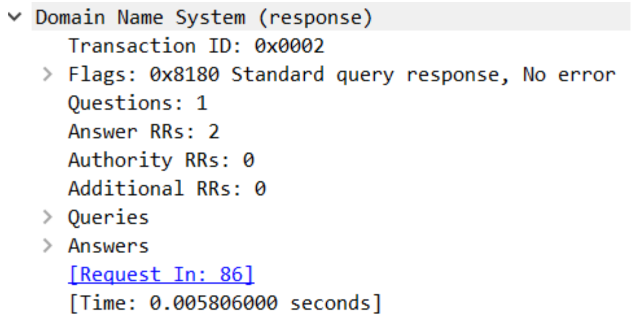


```
⌄ Answers
  ⌄ en.wikiversity.org: type CNAME, class IN, cname dyna.wikimedia.org
      Name: en.wikiversity.org
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 86350 (23 hours, 59 minutes, 10 seconds)
      Data length: 17
      CNAME: dyna.wikimedia.org
  ⌄ dyna.wikimedia.org: type A, class IN, addr 103.102.166.224
      Name: dyna.wikimedia.org
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 374 (6 minutes, 14 seconds)
      Data length: 4
      Address: 103.102.166.224
```

Q3. Capture and Analyse DNS Packets using Wireshark.

c. Check whether a DNS request receives multiple responses, if so, determine the reason for this

For the above query we observed that for a single query we receive two answers:



```
v Domain Name System (response)
    Transaction ID: 0x0002
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  > Answers
    [Request In: 86]
    [Time: 0.005806000 seconds]
```

Why multiple answers to a single DNS query?

1. The number of answers for a DNS query depends on how the domain is set up. It may be set up using multiple nameservers, hence in that case we receive multiple answers. Thus, for the above query, we receive two answers which means the domain consists of two configured nameservers. However, the values of fields differ for both the answers depending on their type.
2. Providing multiple answers in a single response, allows authoritative name servers to assist full-service resolvers in pre-populating their cache before stub resolvers or other clients ask for the subsequent queries. Apart from decreasing the latency for end-users, this also decreases the total number of queries that full-service resolvers need to send and authoritative servers need to answer.