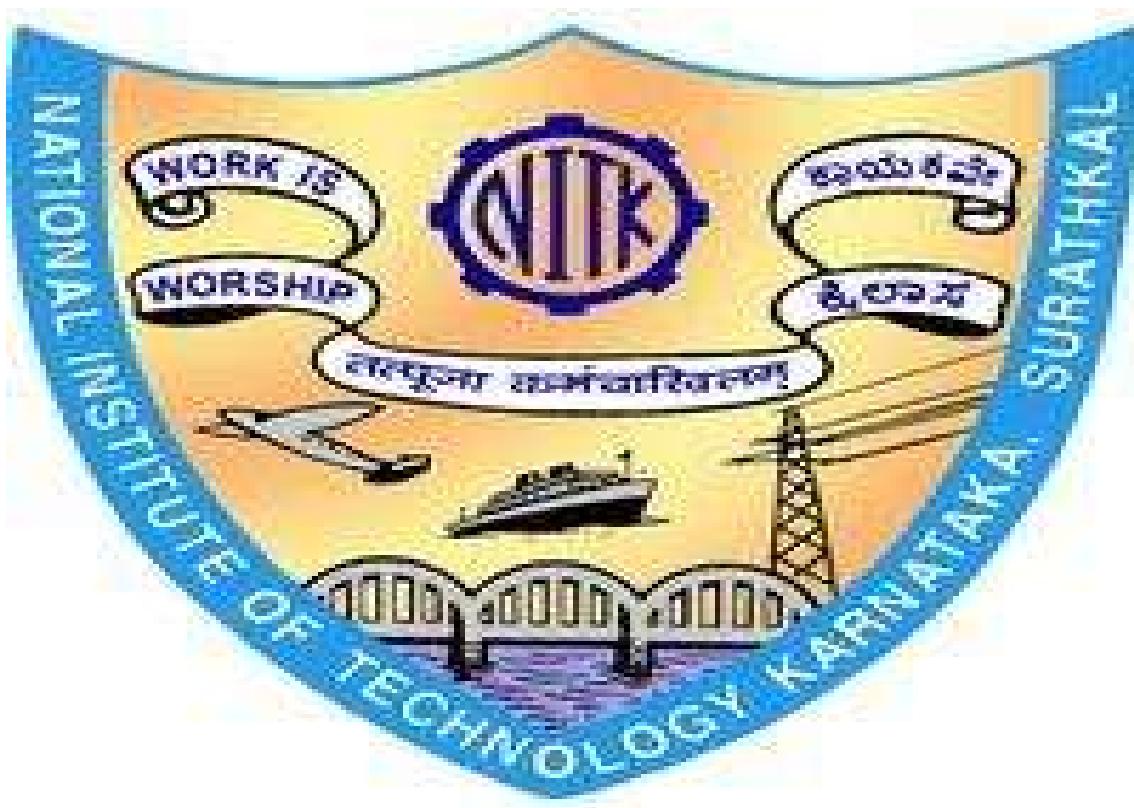


# Computer Networks Lab (CS302)

Report Submission: CN Assignment Lab-4



## Group Member Details:

1. Mahadev M Hatti 191CS133
2. Darshan A V 191CS219

# **1.Develop a basic port scanner to check if particular ports are open or closed for an input remote host.**

**Code:**

```
import socket
import sys
from distlib.compat import raw_input

# Ask for input
remoteServer    = raw_input("Enter a remote host to scan: ")
remoteServerIP  = socket.gethostname(remoteServer)

# Print a nice banner with information on which host we are about to scan
print("-" * 60)
print("Please wait, scanning remote host", remoteServerIP)
print("-" * 60)

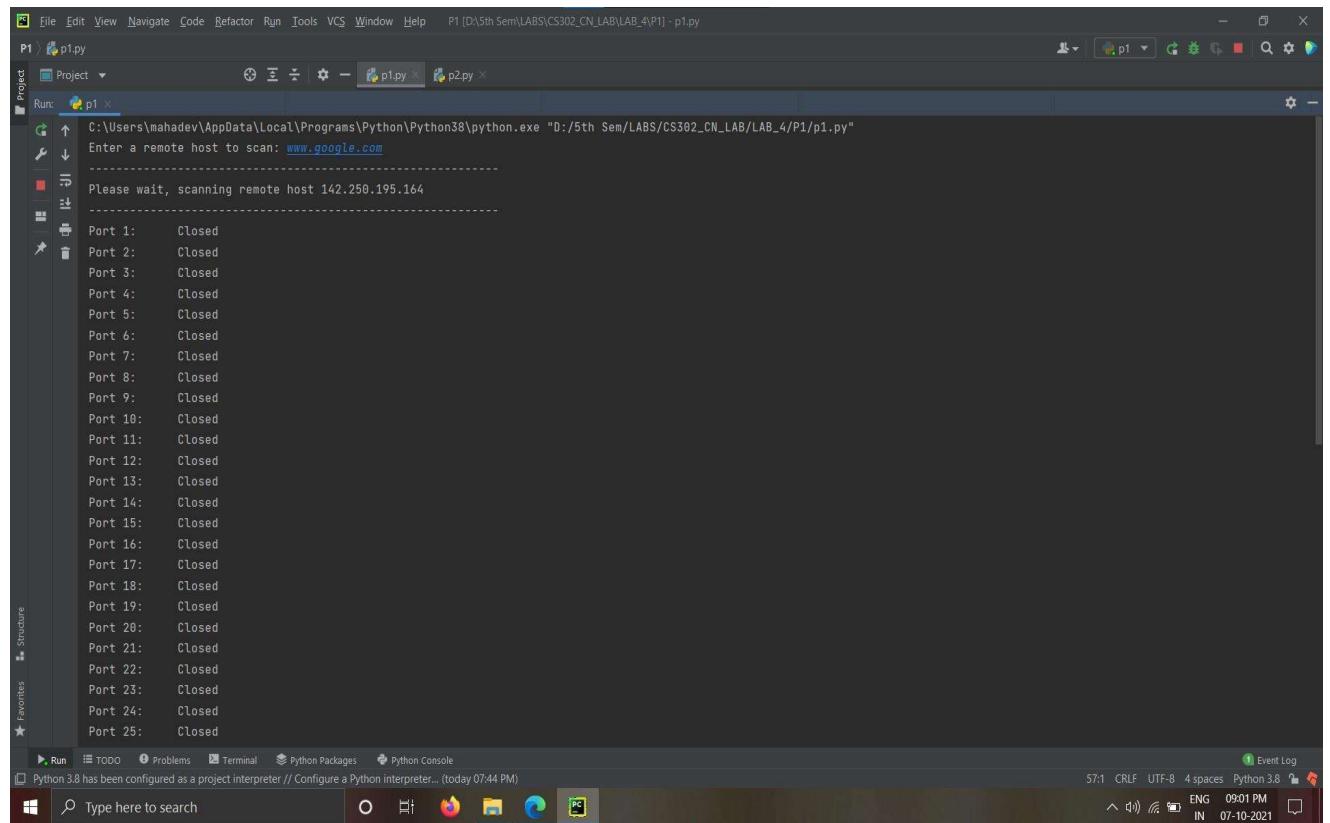
# Using the range function to specify ports (here it will scans all ports between 1
and 1024)

# We also put in some error handling for catching errors

try:
    for port in range(1, 1025):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        result = sock.connect_ex((remoteServerIP, port))
        if result != 0:
            print("Port {}: Closed".format(port))
        else:
            print("Port {}: Open".format(port))
        sock.close()

except socket.gaierror:
    print('Hostname could not be resolved. Exiting')
    sys.exit()

except socket.error:
    print("Couldn't connect to server")
    sys.exit()
```



**2. Develop a threaded port scanner to check if particular ports are open or closed for a remote host. Determine which Port scanner is efficient.**

**Code:**

```
import argparse
import socket # for connecting
from threading import Thread
from queue import Queue

# number of threads, feel free to tune this parameter as you wish
N_THREADS = 400
# thread queue
q = Queue()

def port_scan(port):
    try:
        s = socket.socket()
        s.connect((host, port))
```

```

except:
    print(f"{host}:{port} is closed\n")
else:
    print(f"{host}:{port} is open\n")
finally:
    s.close()

def scan_thread():
    global q
    while True:
        # get the port number from the queue
        worker = q.get()
        # scan that port number
        port_scan(worker)
        # tells the queue that the scanning for that port
        # is done
        q.task_done()

def main(host, ports):
    global q
    for t in range(N_THREADS):
        # for each thread, start it
        t = Thread(target=scan_thread)
        # when we set daemon to true, that thread will end when the main thread ends
        t.daemon = True
        # start the daemon thread
        t.start()
    for worker in ports:
        # for each port, put that port into the queue
        # to start scanning
        q.put(worker)
    # wait the threads ( port scanners ) to finish, wait until the queue is empty
    # before performing other operations
    q.join()

if __name__ == "__main__":
    # parse some parameters passed
    parser = argparse.ArgumentParser(description="Simple port scanner")
    parser.add_argument("host", help="Host to scan.")
    parser.add_argument("--ports", "-p", dest="port_range", default="1-65535",
                       help="Port range to scan, default is 1-65535 (all ports)")
    args = parser.parse_args()
    host, port_range = args.host, args.port_range

    start_port, end_port = port_range.split("-")
    start_port, end_port = int(start_port), int(end_port)

    ports = [p for p in range(start_port, end_port)]

    main(host, ports)

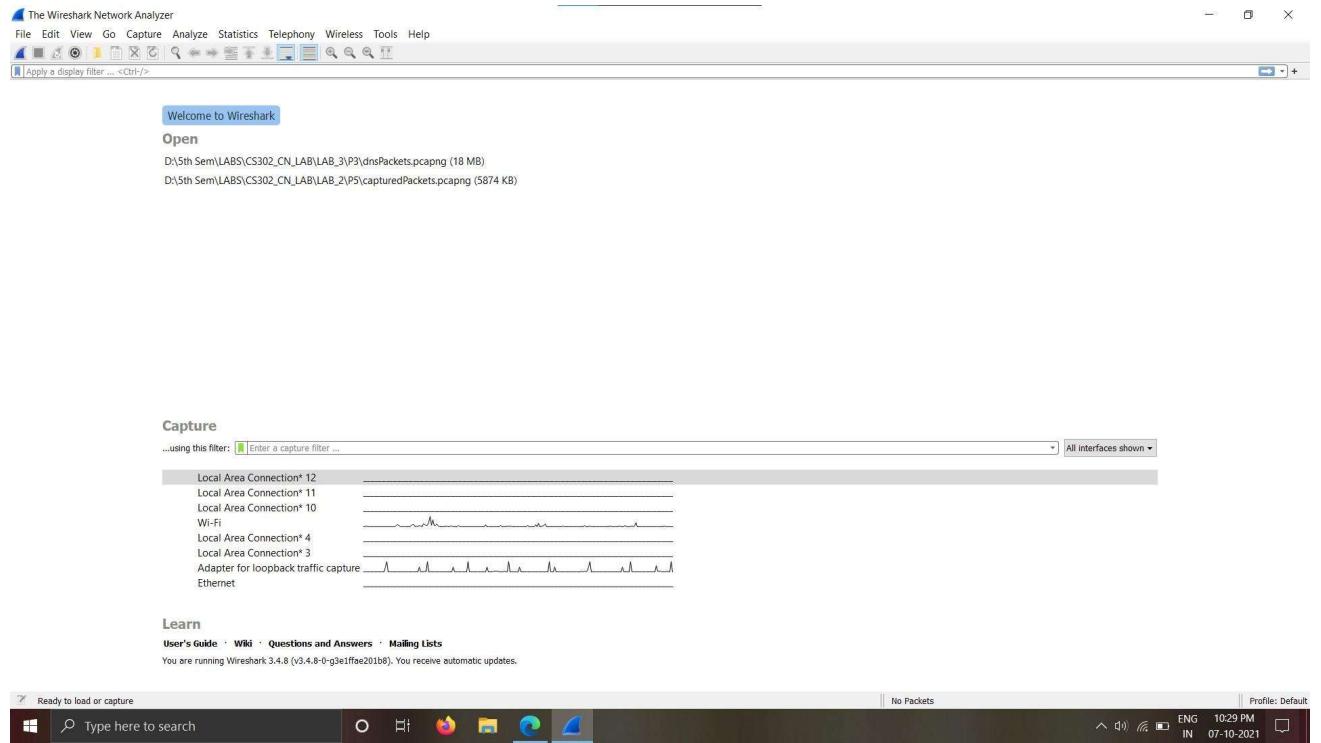
```

```
PS D:\5th Sem\LABS\CS302_CN_LAB\LAB_4\P2> python ./p2.py 192.168.1.1 --ports 1-10
192.168.1.1    :    2 is closed
192.168.1.1    :    9 is closed
192.168.1.1    :    5 is closed
192.168.1.1    :    7 is closed
192.168.1.1    :    1 is closed
192.168.1.1    :    8 is closed
192.168.1.1    :    4 is closed
192.168.1.1    :    6 is closed
```

### 3. Capture UDP packets and with the help of the captured UDP Packets.

#### a. analyse UDP DHCP Packets

1. In the below fig. selects the Wi-Fi option from the Interface list options.



2. In the new window you can see all the current traffic on the network. (Clear cache – Before capturing the traffic, you need to clear your browser’s cache.)

From this Pane you can observe:

- No. – The number of a captured packet.
- Time – This shows you when the packet was captured with regards to when you started capturing.
- Source – This is the origin of a captured packet in the form of an address.
- Destination – The destination address of a captured packet.
- Protocol – The type of a captured packet.
- Length – This shows you the length of a captured packet. This is expressed in bytes.

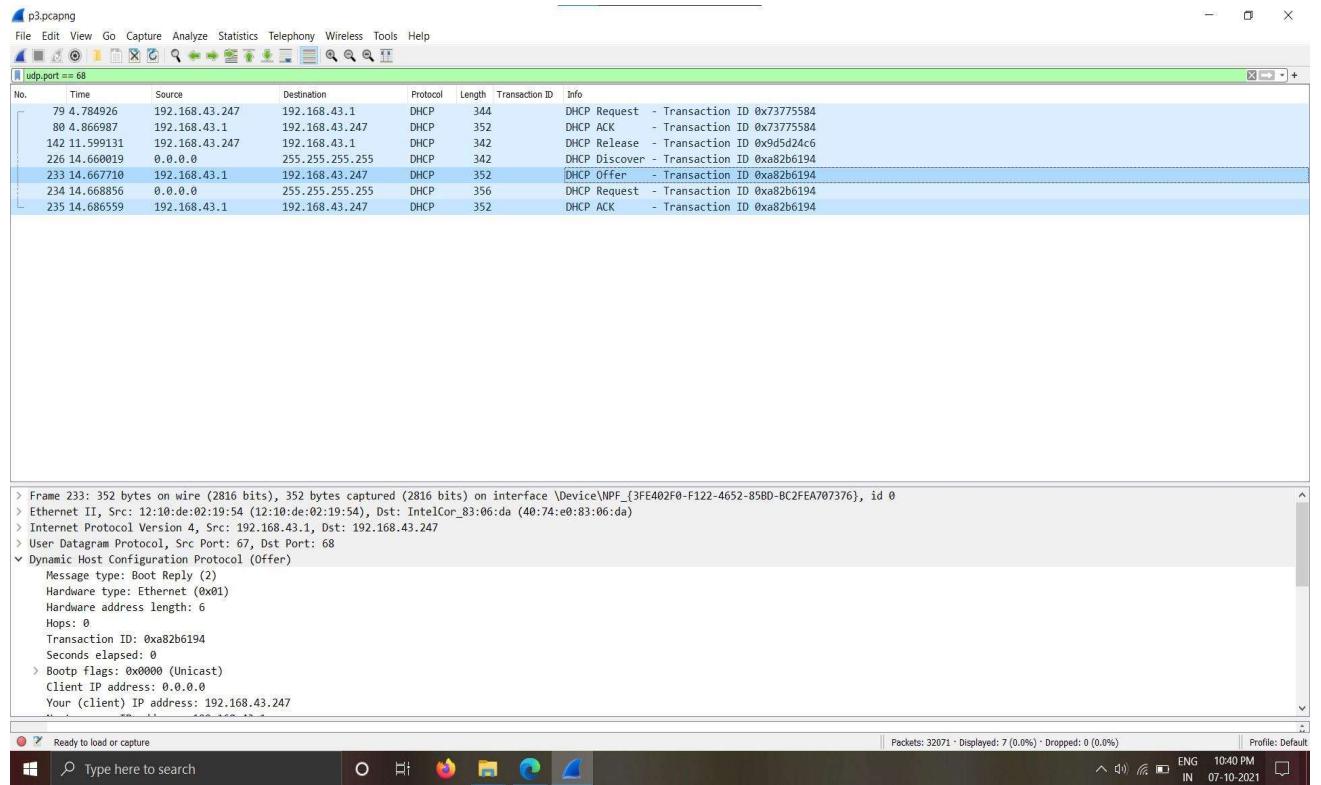
Capturing from Wi-Fi

No.	Time	Source	Destination	Protocol	Length	Transaction ID	Info
1	0.000000	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=1 Ack=1 Win=2071 Len=1370 [TCP segment of a reassembled PDU]	
2	0.000000	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=1371 Ack=1 Win=2071 Len=1370 [TCP segment of a reassembled PDU]	
3	0.000000	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=2741 Ack=1 Win=2071 Len=1370 [TCP segment of a reassembled PDU]	
4	0.000000	2409:4071:2489:7581..	2620:1ec:a92::171	TLSv1..2	435	Application Data	
5	0.0000283	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=4472 Ack=1 Win=2071 Len=1370 [TCP segment of a reassembled PDU]	
6	0.0000283	2409:4071:2489:7581..	2620:1ec:a92::171	TLSv1..2	956	Application Data	
7	0.091254	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 → 61673 [ACK] Seq=1 Ack=1371 Win=2049 Len=0	
8	0.092343	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=1 Ack=2741 Win=2049 Len=0	
9	0.095571	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=1 Ack=4472 Win=2049 Len=0	
10	0.096616	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=1 Ack=5842 Win=2049 Len=0	
11	0.096616	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=1 Ack=6724 Win=2046 Len=0	
12	0.109795	2620:1ec:a92::171	2409:4071:2489:7581..	TLSv1..2	116	Application Data	
13	0.115667	2620:1ec:a92::171	2409:4071:2489:7581..	TLSv1..2	492	Application Data	
14	0.115759	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	74	61673 → 443 [ACK] Seq=6724 Ack=461 Win=2076 Len=0	
15	0.116761	2620:1ec:a92::171	2409:4071:2489:7581..	TLSv1..2	112	Application Data	
16	0.171035	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	74	61673 → 443 [ACK] Seq=6724 Ack=499 Win=2076 Len=0	
17	1.183926	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=6724 Ack=499 Win=2076 Len=1370 [TCP segment of a reassembled PDU]	
18	1.183926	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=8894 Ack=499 Win=2076 Len=1370 [TCP segment of a reassembled PDU]	
19	1.183926	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=9464 Ack=499 Win=2076 Len=1370 [TCP segment of a reassembled PDU]	
20	1.183926	2409:4071:2489:7581..	2620:1ec:a92::171	TLSv1..2	348	Application Data	
21	1.184161	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=11188 Ack=499 Win=2076 Len=1370 [TCP segment of a reassembled PDU]	
22	1.184161	2409:4071:2489:7581..	2620:1ec:a92::171	TLSv1..2	959	Application Data	
23	1.381392	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=499 Ack=9464 Win=2049 Len=0	
24	1.381392	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=499 Ack=11108 Win=2049 Len=0	
25	1.381392	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=499 Ack=12478 Win=2049 Len=0	
26	1.381392	2620:1ec:a92::171	2409:4071:2489:7581..	TCP	74	443 + 61673 [ACK] Seq=499 Ack=13363 Win=2046 Len=0	
27	1.381392	2620:1ec:a92::171	2409:4071:2489:7581..	TLSv1..2	461	Application Data	
28	1.381392	2620:1ec:a92::171	2409:4071:2489:7581..	TLSv1..2	112	Application Data	
29	1.381532	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	74	61673 → 443 [ACK] Seq=13363 Ack=924 Win=2074 Len=0	
30	2.992199	2409:4071:2489:7581..	2620:1ec:a92::171	TCP	1444	61673 → 443 [ACK] Seq=13363 Ack=924 Win=2074 Len=1370 [TCP segment of a reassembled PDU]	

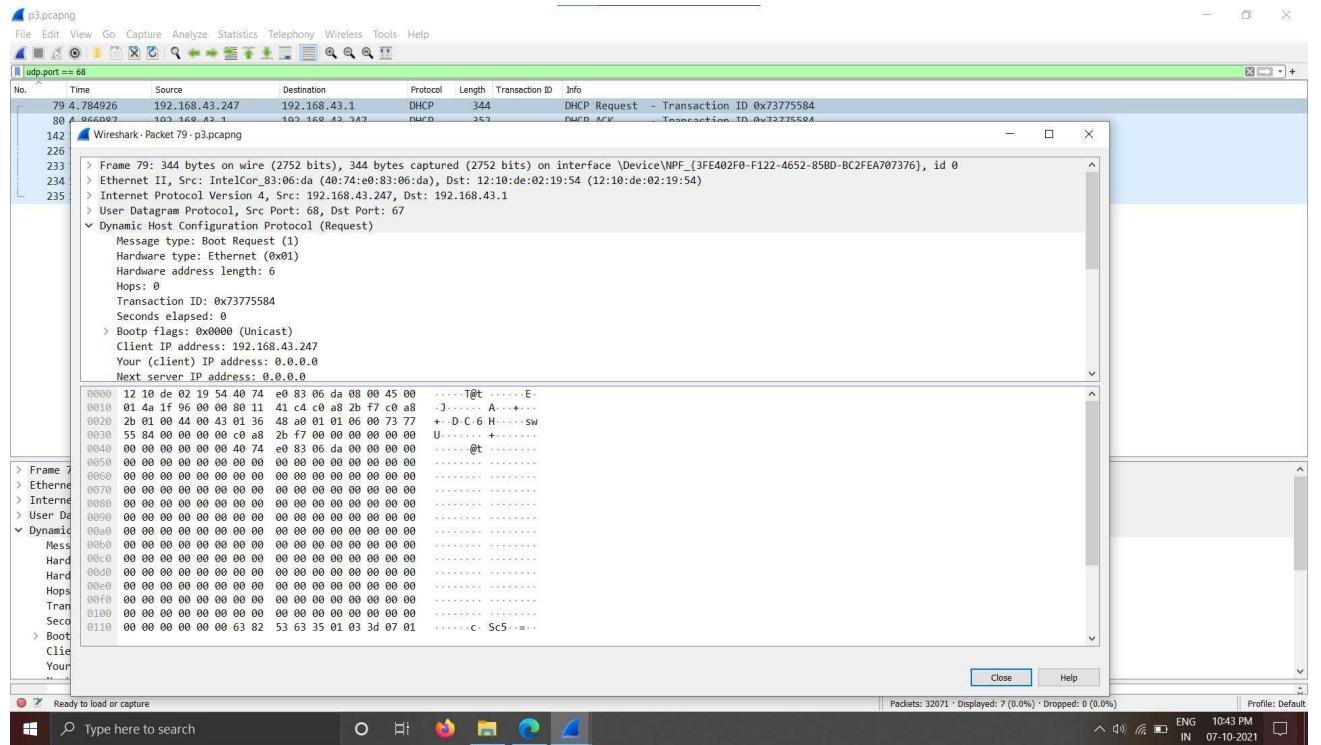
> Frame 1: 1444 bytes wire (1152 bits), 1444 bytes captured (1152 bits) on interface \Device\NPF\_{3FE402F0-F122-4652-85BD-BC2FEA707376}, id 0  
 > Ethernet II, Src: IntelCor\_83:06:da (40:74:e0:83:06:da), Dst: 12:10:de:02:19:54 (12:10:de:02:19:54)  
 > Internet Protocol Version 6, Src: 2409:4071:2489:7581:96:ee71:f20:c156, Dst: 2620:1ec:a92::171  
 > Transmission Control Protocol, Src Port: 61673, Dst Port: 443, Seq: 1, Ack: 1, Len: 1370

### 3. Use filter section to filter out Specific Packets related to dns Server.

- ◆ To view only DHCP traffic, type `udp.port == 68` (lower case) in the Filter box and press Enter.



- **DHCP Request Traffic :**
- Select the first DHCP packet, labeled DHCP Request.



- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
- Expanded Ethernet II to view Ethernet details.

▼ Ethernet II, Src: IntelCor\_83:06:da (40:74:e0:83:06:da), Dst: 12:10:de:02:19:54 (12:10:de:02:19:54)

- Destination: 12:10:de:02:19:54 (12:10:de:02:19:54)
- Source: IntelCor\_83:06:da (40:74:e0:83:06:da)
- Type: IPv4 (0x0800)

- I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
- I observed Internet Protocol Version 4 to view IP details.

```
▼ Internet Protocol Version 4, Src: 192.168.43.247, Dst: 192.168.43.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 330
    Identification: 0x1f96 (8086)
  > Flags: 0x00
    Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x41c4 [validation disabled]
      [Header checksum status: Unverified]
    Source Address: 192.168.43.247
    Destination Address: 192.168.43.1
```

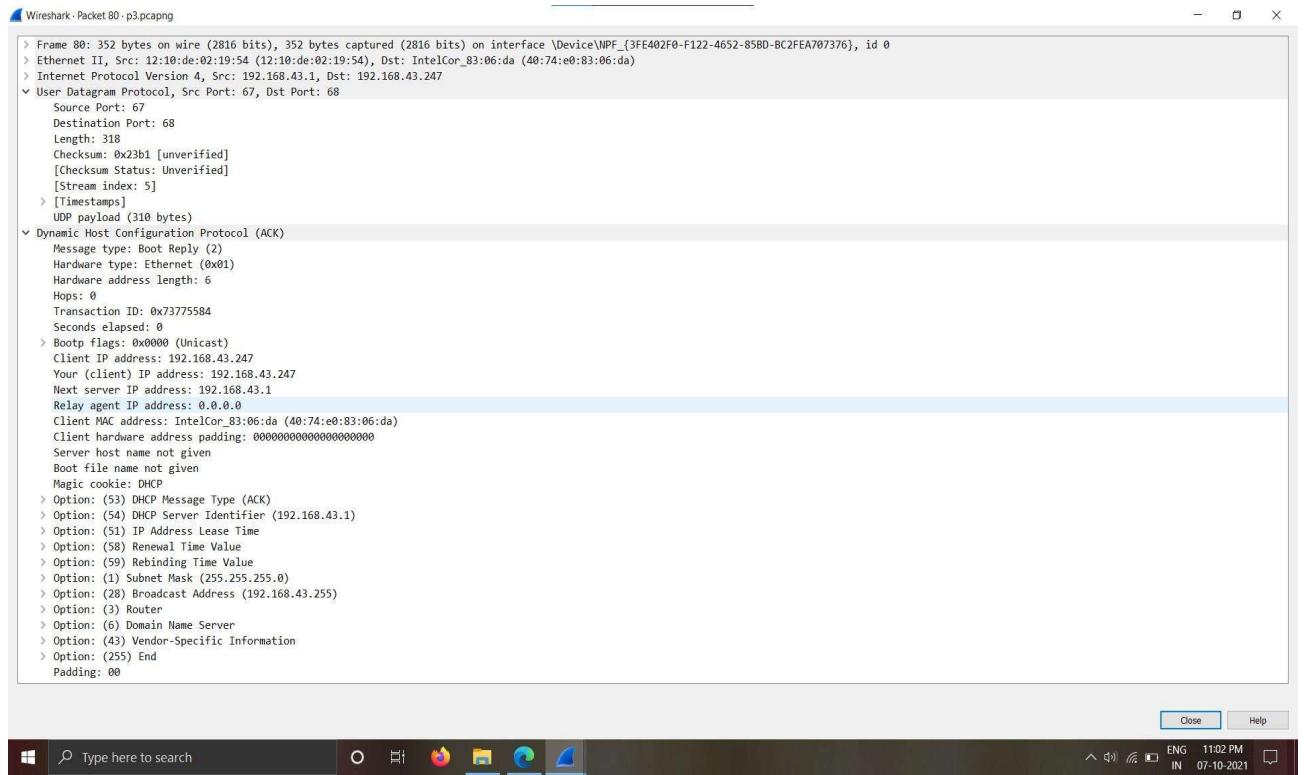
- Noticed that the source address is my IP address
- Noticed that the destination address is the IP address of the DHCP server.
- Expanded User Datagram Protocol to view UDP details.

```
▼ User Datagram Protocol, Src Port: 68, Dst Port: 67
  Source Port: 68
  Destination Port: 67
  Length: 310
  Checksum: 0x48a0 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 5]
  > [Timestamps]
  UDP payload (302 bytes)
```

- Expanded Bootstrap Protocol to view BOOTP details.

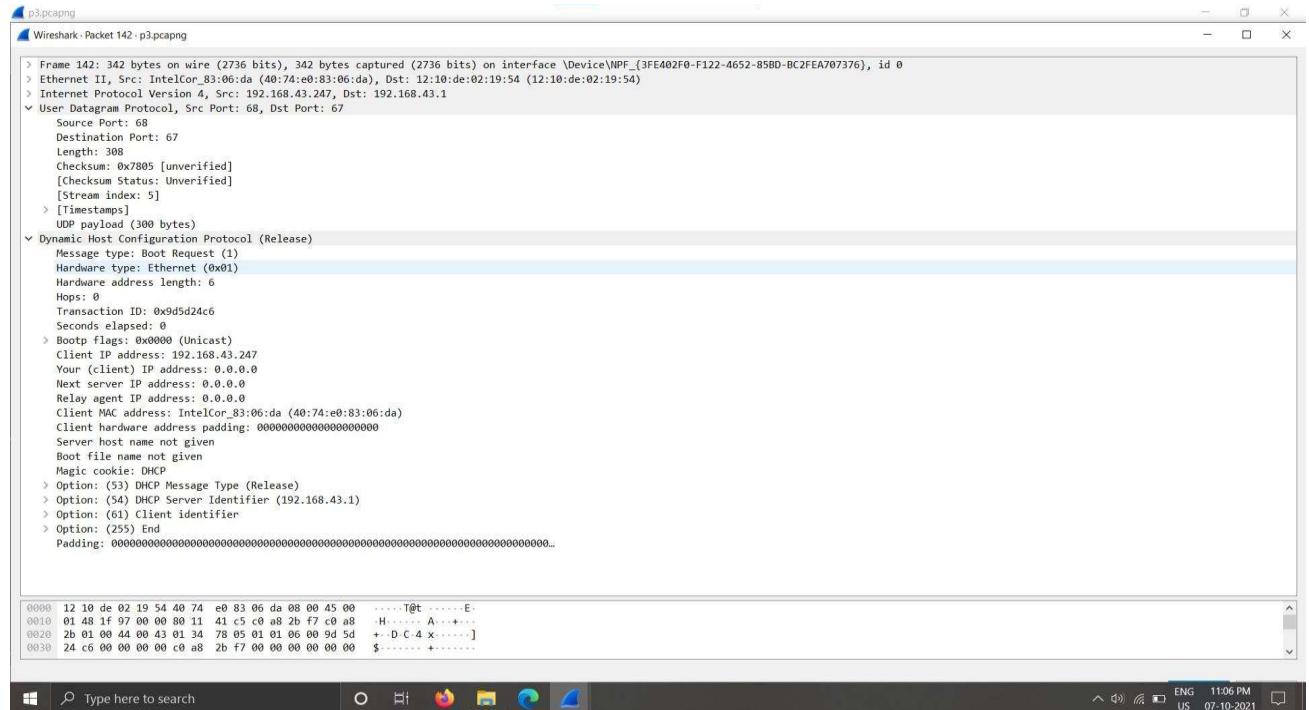
```
001 payload (302 bytes)
└ Dynamic Host Configuration Protocol (Request)
  Message type: Boot Request (1)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x73775584
  Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
  Client IP address: 192.168.43.247
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: IntelCor_83:06:da (40:74:e0:83:06:da)
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Request)
  > Option: (61) Client identifier
  > Option: (12) Host Name
  > Option: (81) Client Fully Qualified Domain Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  > Option: (255) End
```

- **DHCP ACK Traffic:**
- Selected the second DHCP packet, labeled DHCP ACK



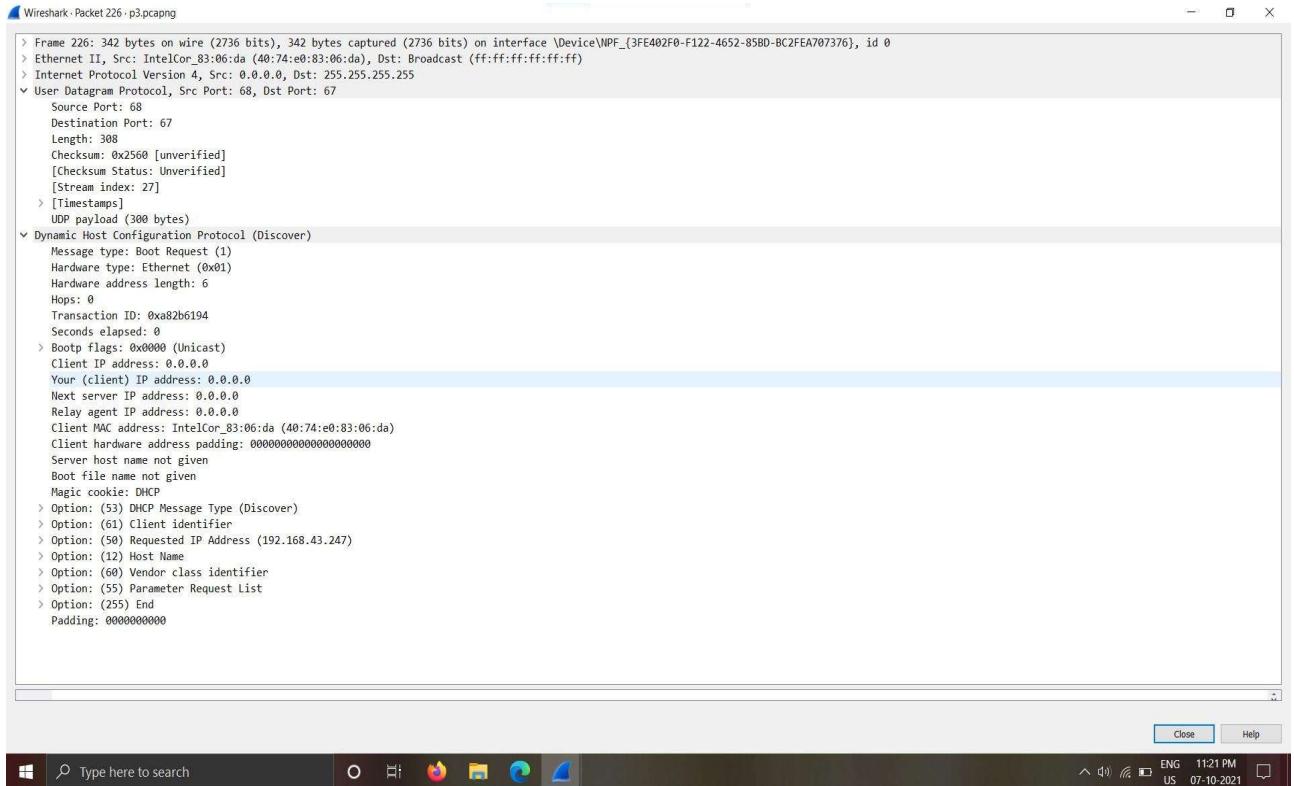
- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
- Expanded Ethernet II to view Ethernet details.
- I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
- I observed Internet Protocol Version 4 to view IP details.
- Noticed that the source address is my IP address
- Noticed that the destination address is the IP address of the DHCP server.
- Expanded User Datagram Protocol to view UDP details.
- Expanded Bootstrap Protocol to view BOOTP details.

- **DHCP Release Traffic:**
- Selected the third DHCP packet, labeled DHCP Release.



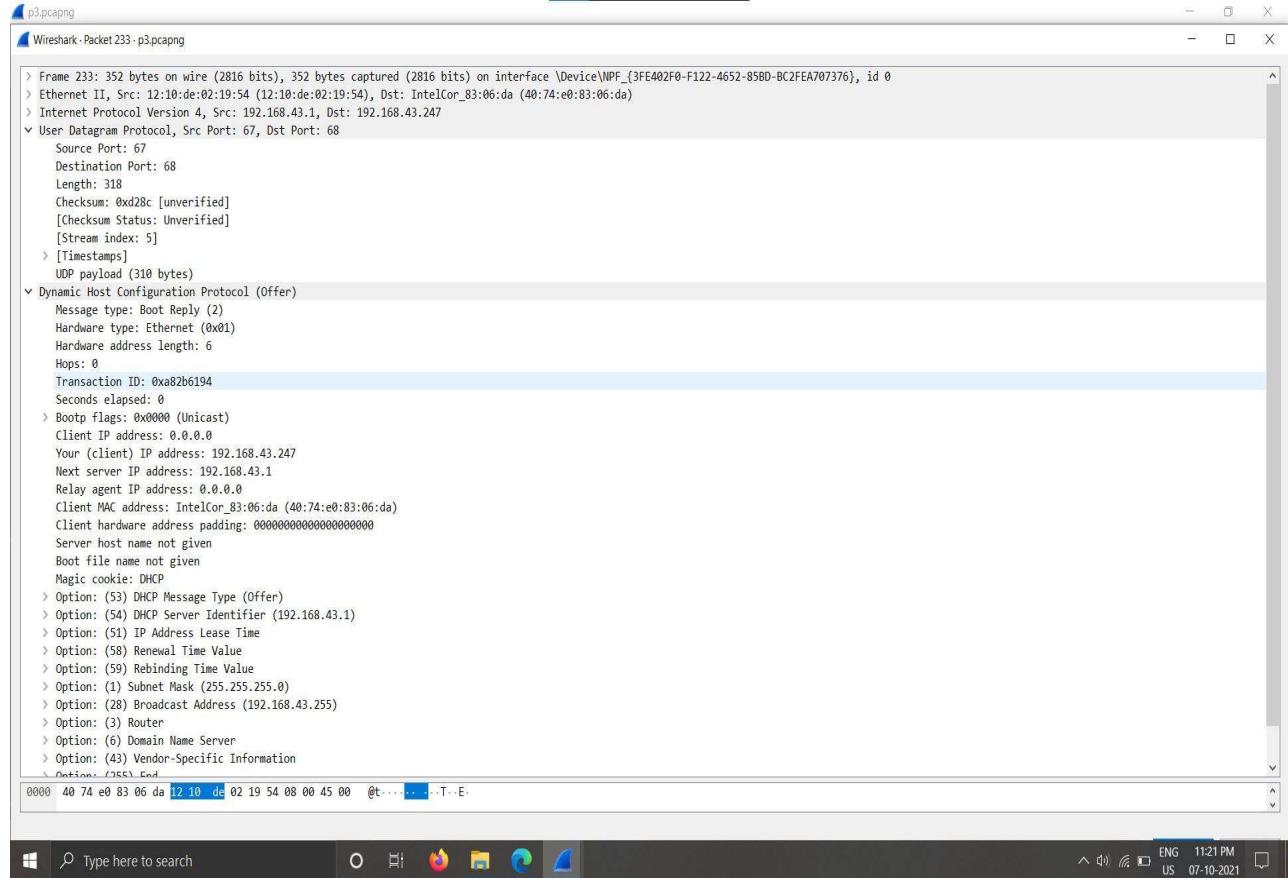
- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
- Expanded Ethernet II to view Ethernet details.
- I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
- I observed Internet Protocol Version 4 to view IP details.
- Noticed that the source address is my IP address
- Noticed that the destination address is the IP address of the DHCP server.
- Expanded User Datagram Protocol to view UDP details.
- Expanded Bootstrap Protocol to view BOOTP details.

- **DHCP Discover Traffic:**
- Selected the fourth DHCP packet, labeled DHCP Discover.



- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
- Expanded Ethernet II to view Ethernet details.
- I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
- I observed Internet Protocol Version 4 to view IP details.
- Noticed that the source address is my IP address
- Noticed that the destination address is the IP address of the DHCP server.
- Expanded User Datagram Protocol to view UDP details.
- Expanded Bootstrap Protocol to view BOOTP details.

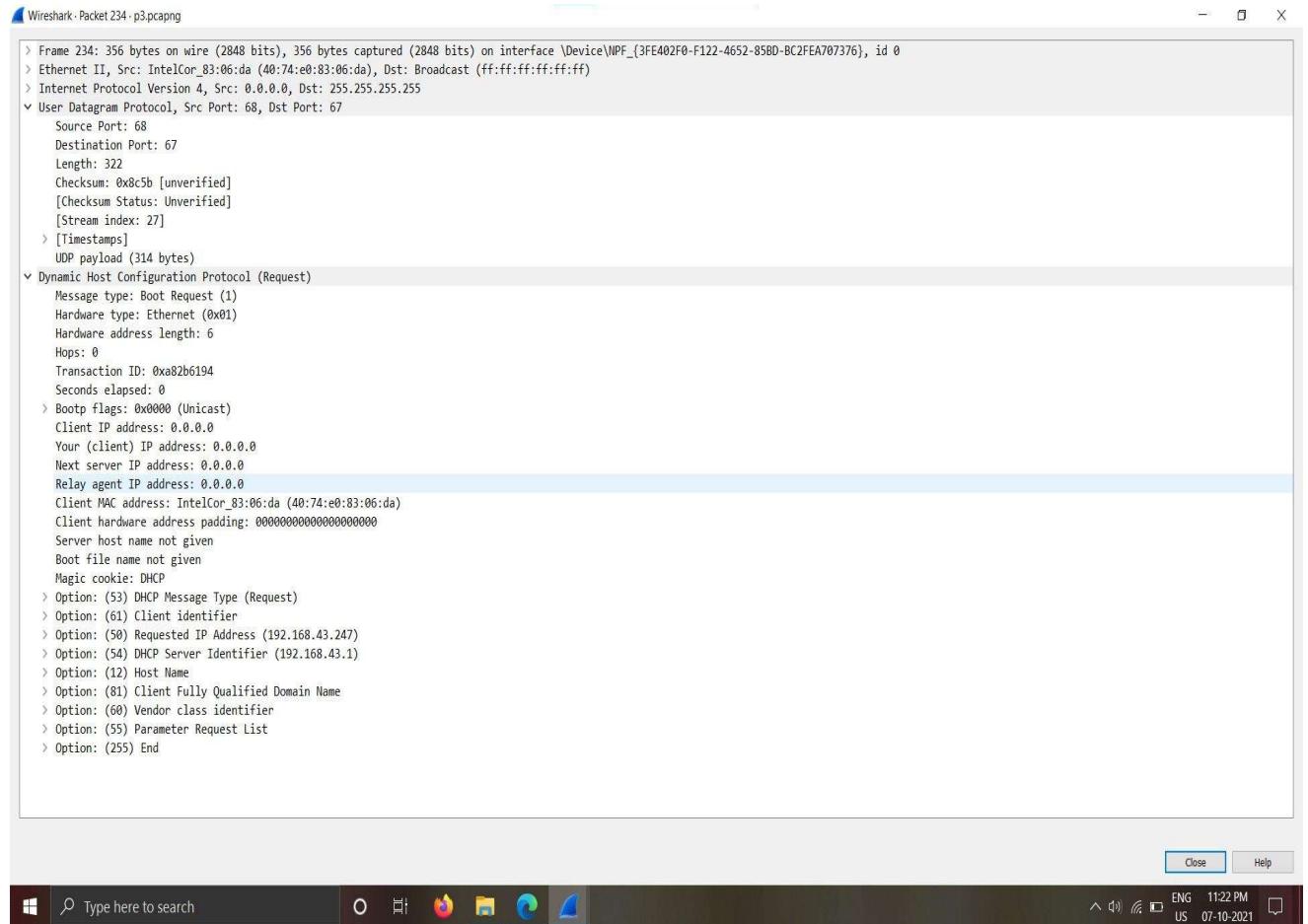
- **DHCP Offer Traffic:**
- Selected the fifth DHCP packet, labeled DHCP Offer.



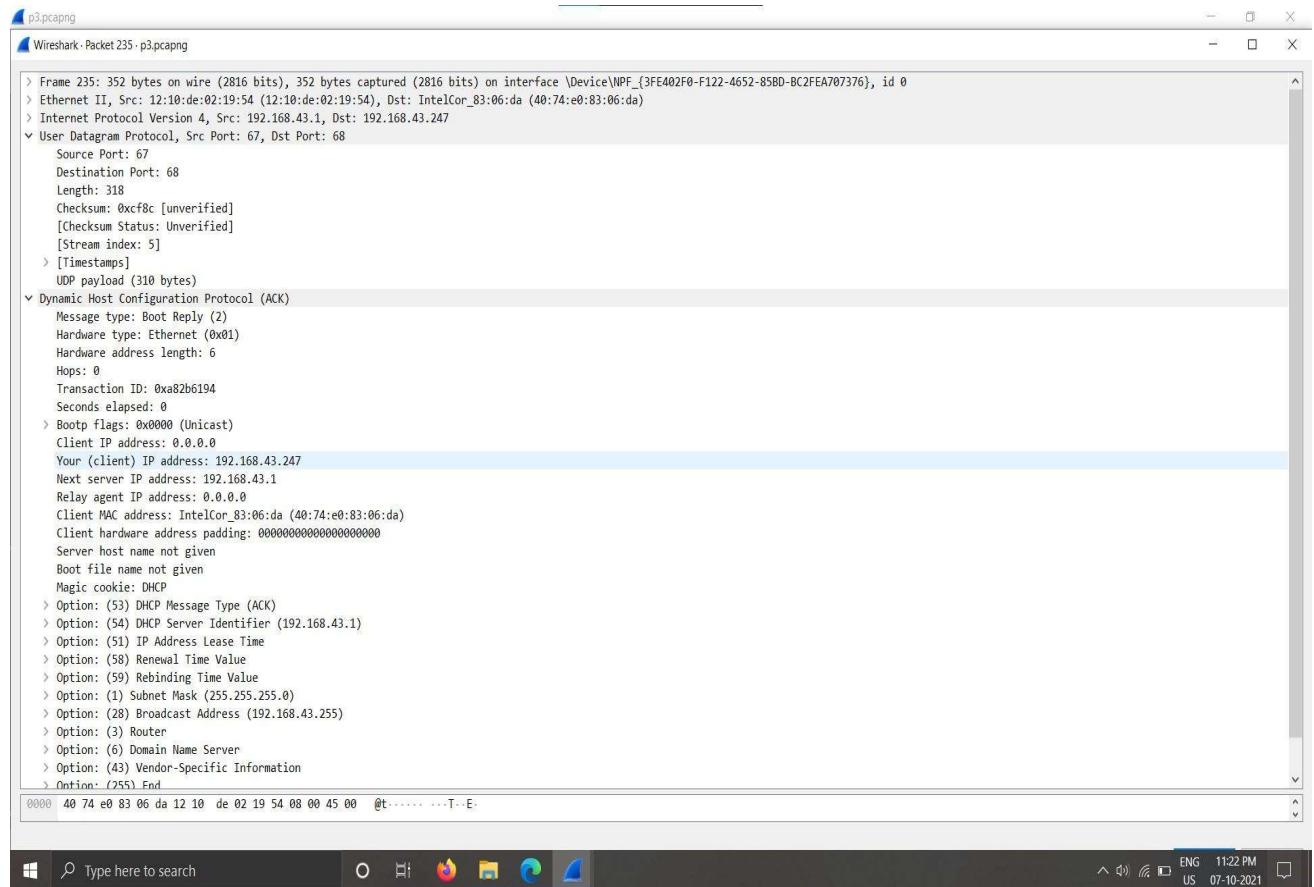
- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
- Expanded Ethernet II to view Ethernet details.
- I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
- I observed Internet Protocol Version 4 to view IP details.

- Noticed that the source address is my IP address
- Noticed that the destination address is the IP address of the DHCP server.
- Expanded User Datagram Protocol to view UDP details.
- Expanded Bootstrap Protocol to view BOOTP details.

- **DHCP Request Traffic:**
- Selected the sixth DHCP packet, labeled DHCP Request.



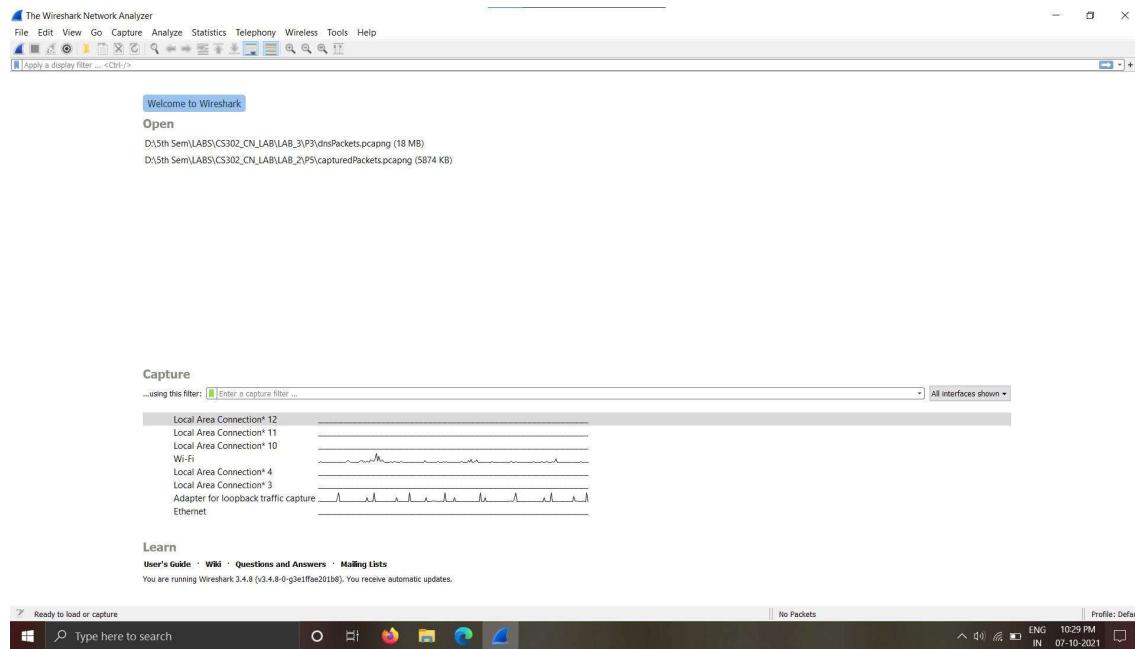
- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
  - Expanded Ethernet II to view Ethernet details.
  - I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
  - I observed Internet Protocol Version 4 to view IP details.
  - Noticed that the source address is my IP address
  - Noticed that the destination address is the IP address of the DHCP server.
  - Expanded User Datagram Protocol to view UDP details.
  - Expanded Bootstrap Protocol to view BOOTP details.
- 
- **DHCP ACK Traffic:**
  - Selected the seventh DHCP packet, labeled DHCP ACK.



- I Noticed that there is an Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Bootstrap Protocol frame.
- Expanded Ethernet II to view Ethernet details.
- I observed that the destination is my DHCP server's MAC address and the source is my MAC address.
- I observed Internet Protocol Version 4 to view IP details.
- Noticed that the source address is my IP address
- Noticed that the destination address is the IP address of the DHCP server.
- Expanded User Datagram Protocol to view UDP details.
- Expanded Bootstrap Protocol to view BOOTP details.

## b. Analyse UDP DNS Packet

1.In the below fig. selects the Wi-Fi option from the Interface list options.

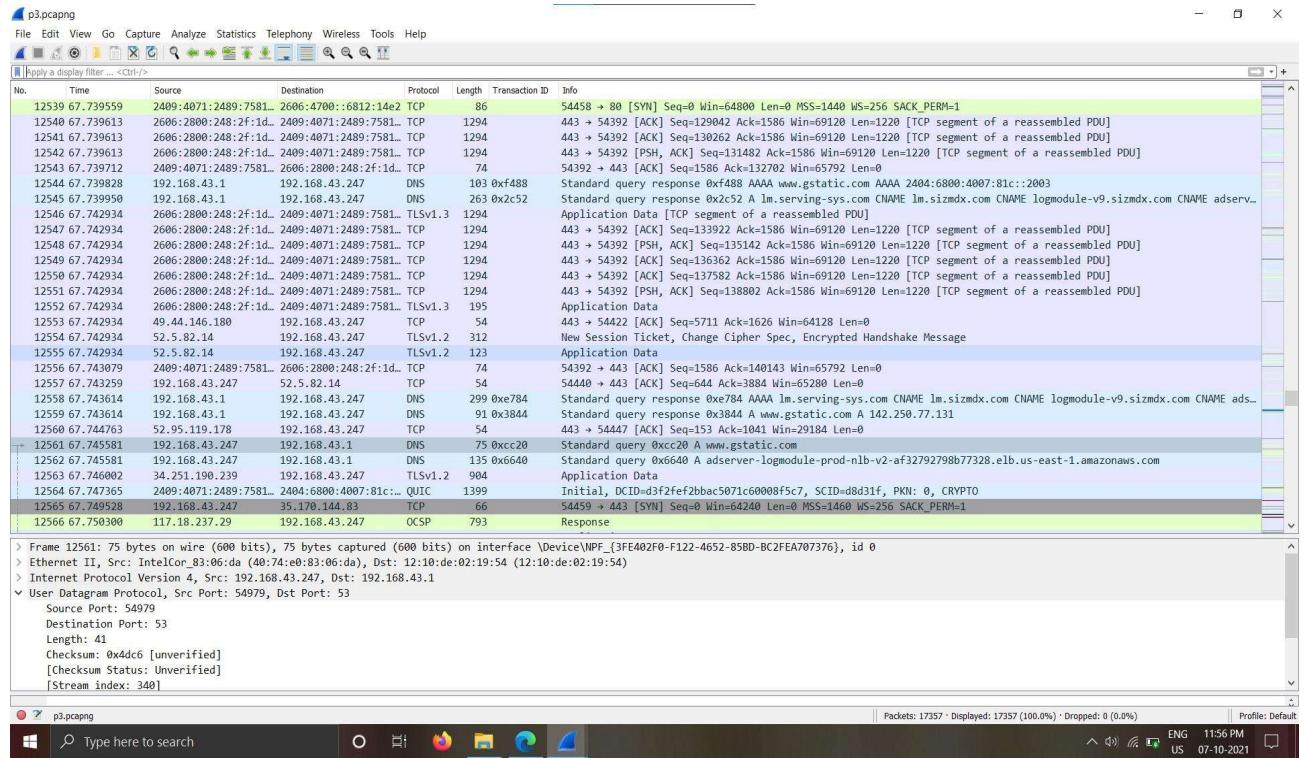


2.In the new window you can see all the current traffic on the network. (Clear cache – Before capturing the traffic, you need to clear your browser's cache.)

From this Pane you can observe:

- No. – The number of a captured packet.
- Time – This shows you when the packet was captured with regards to when you started capturing.
- Source – This is the origin of a captured packet in the form of an address.
- Destination – The destination address of a captured packet.
- Protocol – The type of a captured packet.

- Length – This shows you the length of a captured packet. This is expressed in bytes.



### 3. Use filter section to filter out Specific Packets related to dns Server.

- ◆ To view only DNS traffic, type `udp.port == 53` (lower case) in the Filter box and press Enter.

p3.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 53

No.	Time	Source	Destination	Protocol	Length	Transaction ID	Info
12279	67.521119	192.168.43.1	192.168.43.247	DNS	161	0x42e9	Standard query response 0x42e9 AAAA www.amazon.in CNAME tp.c95e7e602-frontier.amazon.in CNAME d1elgm1ww0d6wo.cloud...
12280	67.525415	192.168.43.247	192.168.43.1	DNS	78	0xd9e4	Standard query 0xd9e4 A www.hackerrank.com
12282	67.526033	192.168.43.247	192.168.43.1	DNS	78	0xdf63	Standard query 0xdf63 AAAA www.hackerrank.com
12316	67.559725	192.168.43.247	192.168.43.1	DNS	79	0xf562	Standard query 0xf562 A ocsp.globalsign.com
12317	67.560283	192.168.43.247	192.168.43.1	DNS	79	0x2eff	Standard query 0x2eff AAAA ocsp.globalsign.com
12337	67.571257	192.168.43.247	192.168.43.247	DNS	170	0xd9e4	Standard query response 0xd9e4 A www.hackerrank.com CNAME hackerrank.com.edgekey.net CNAME e8937.dsrb.akamaiedge.n...
12350	67.586817	192.168.43.1	192.168.43.247	DNS	126	0x6656	Standard query response 0x6656 AAAA www.nitk.ac.in SOA a.ns.nitk.ac.in
12351	67.586958	192.168.43.1	192.168.43.247	DNS	127	0xc231	Standard query response 0xc231 AAAA iris.nitk.ac.in SOA a.ns.nitk.ac.in
12352	67.588468	192.168.43.247	192.168.43.1	DNS	75	0xa48b	Standard query 0xa48b AAAA iris.nitk.ac.in
12353	67.589511	192.168.43.247	192.168.43.1	DNS	74	0x6de6	Standard query response 0x6de6 AAAA www.nitk.ac.in
12354	67.593312	192.168.43.1	192.168.43.247	DNS	75	0xa48b	Standard query response 0xa48b AAAA iris.nitk.ac.in
12355	67.593312	192.168.43.1	192.168.43.247	DNS	74	0x6de6	Standard query response 0x6de6 AAAA www.nitk.ac.in
12361	67.596643	192.168.43.1	192.168.43.247	DNS	204	0x4f63	Standard query response 0x4f63 AAAA www.hackerrank.com CNAME hackerrank.com.edgekey.net CNAME e8937.dsrb.akamaiedge.n...
12362	67.598571	192.168.43.1	192.168.43.247	DNS	194	0xf562	Standard query response 0xf562 A ocsp.globalsign.com CNAME global.prdcdn.globalsign.com CNAME cdn.globalsigncdn.c...
12364	67.601918	192.168.43.1	192.168.43.247	DNS	218	0x2eff	Standard query response 0x2eff AAAA ocsp.globalsign.com CNAME global.prdcdn.globalsign.com CNAME cdn.globalsign...
12368	67.604468	192.168.43.247	192.168.43.1	DNS	100	0x9f15	Standard query 0x9f15 A cdn.globalsigncdn.com.cdn.cloudflare.net
12401	67.656790	192.168.43.1	192.168.43.247	DNS	132	0x9f15	Standard query response 0x9f15 A cdn.globalsigncdn.com.cdn.cloudflare.net A 104.18.20.226 A 104.18.21.226
12402	67.657831	192.168.43.247	192.168.43.1	DNS	100	0xe939	Standard query 0xe939 AAAA cdn.globalsigncdn.com.cdn.cloudflare.net
12410	67.672763	192.168.43.247	192.168.43.1	DNS	78	0x2c52	Standard query 0x2c52 A lm.serving-sys.com
12411	67.673376	192.168.43.247	192.168.43.1	DNS	78	0x784	Standard query 0x784 AAAA lm.serving-sys.com
12486	67.694853	192.168.43.247	192.168.43.1	DNS	75	0x3844	Standard query 0x3844 A www.gstatic.com
12488	67.695493	192.168.43.247	192.168.43.1	DNS	75	0xf488	Standard query 0xf488 AAAA www.gstatic.com
12497	67.706591	192.168.43.1	192.168.43.247	DNS	156	0xe939	Standard query response 0xe939 AAAA cdn.globalsigncdn.com.cdn.cloudflare.net AAAA 2606:4700::6812:15e2 AAAA 2606:4...
12544	67.739828	192.168.43.1	192.168.43.247	DNS	103	0xf488	Standard query response 0xf488 AAAA www.gstatic.com AAAA 2404:6800:4007:81::2003
12545	67.739950	192.168.43.1	192.168.43.247	DNS	263	0x2c52	Standard query response 0x2c52 A lm.serving-sys.com CNAME lm.sizmdx.com CNAME logmodule-v9.sizmdx.com CNAME adserv...
12558	67.743614	192.168.43.1	192.168.43.247	DNS	299	0x784	Standard query response 0x784 AAAA lm.serving-sys.com CNAME lm.sizmdx.com CNAME logmodule-v9.sizmdx.com CNAME ads...
12559	67.743614	192.168.43.1	192.168.43.247	DNS	91	0x3844	Standard query response 0x3844 A www.gstatic.com A 142.250.77.131
12561	67.745581	192.168.43.247	192.168.43.1	DNS	75	0xcc20	Standard query 0xcc20 A www.gstatic.com

> Frame 12561: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface \Device\NPF\_{3FE402F0-F122-4652-85BD-BC2FEA707376}, id 0

> Ethernet II, Src: IntelCor\_83:06:da (48:74:e0:83:06:da), Dst: 12:10:de:02:19:54 (12:10:de:02:19:54)

> Internet Protocol Version 4, Src: 192.168.43.247, Dst: 192.168.43.1

✓ User Datagram Protocol, Src Port: 53

    Source Port: 54979  
     Destination Port: 53  
     Length: 41  
     Checksum: 0x4dc0 [unverified]  
     [Checksum Status: Unverified]  
     [Stream index: 340]

Packets: 17357 - Displayed: 832 (4.8%) - Dropped: 0 (0.0%)

ENG 11:57 PM US 07-10-2021

## ■ DNS Query Traffic:

- Selected the DNS packet labeled Standard query A [www.gstatic.com](http://www.gstatic.com).

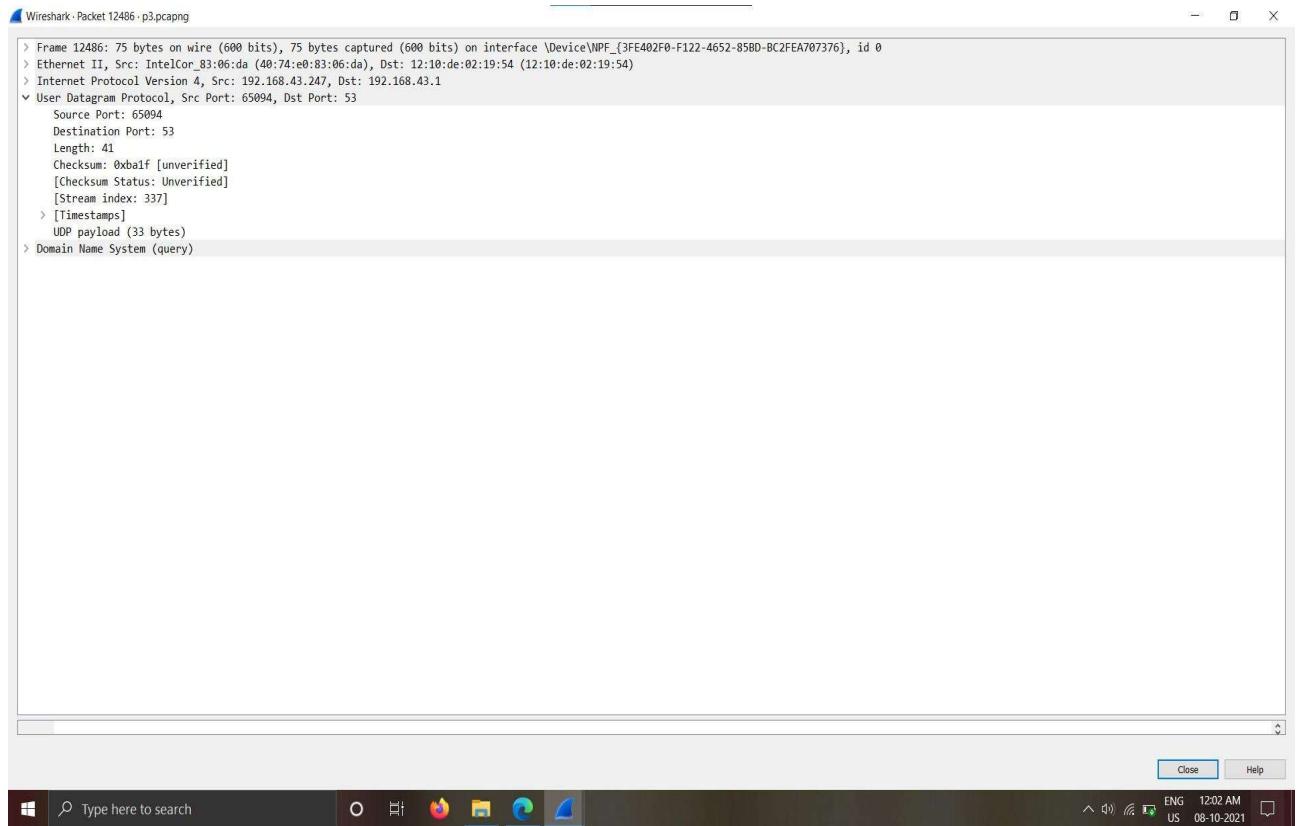
p3.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns.id == 0x3844

No.	Time	Source	Destination	Protocol	Length	Transaction ID	Info
12486	67.694853	192.168.43.247	192.168.43.1	DNS	75	0x3844	Standard query 0x3844 A www.gstatic.com
12559	67.743614	192.168.43.1	192.168.43.247	DNS	91	0x3844	Standard query response 0x3844 A www.gstatic.com A 142.250.77.131

- Noticed the Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Domain Name System (query) frame.



- Expanded Ethernet II to view Ethernet details.



- I observed Internet Protocol Version 4 to view IP details.

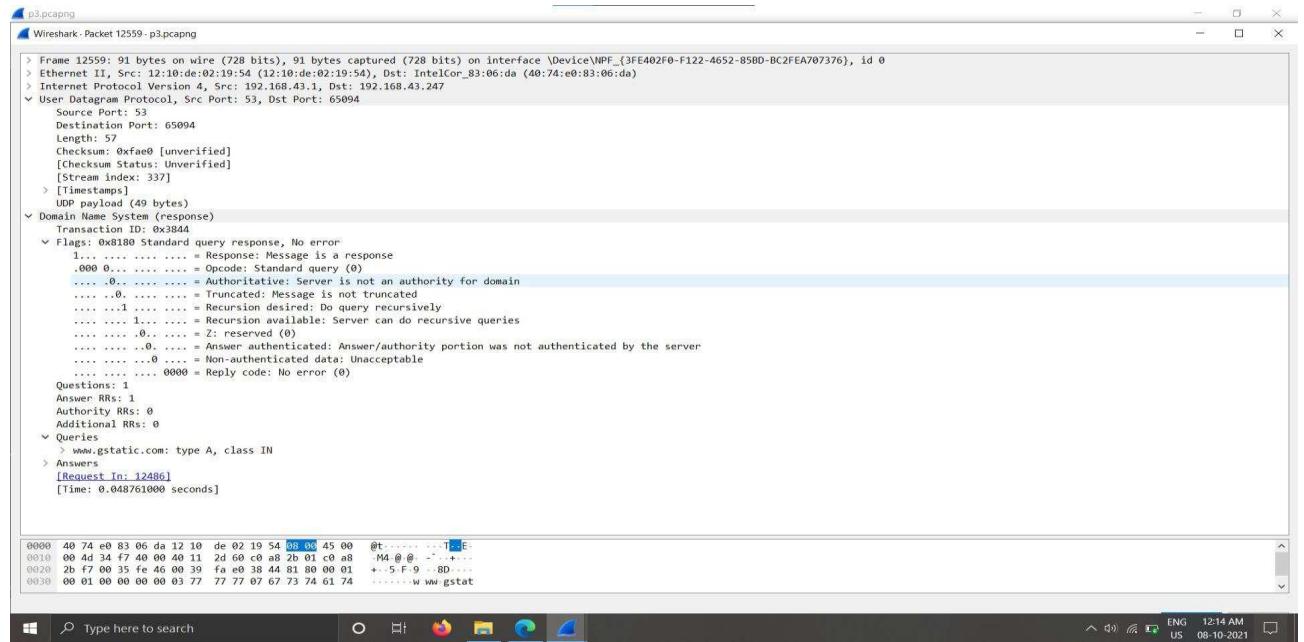
```
└─ Internet Protocol Version 4, Src: 192.168.43.247, Dst: 192.168.43.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        Total Length: 61
        Identification: 0x3220 (12832)
    > Flags: 0x00
        Fragment Offset: 0
        Time to Live: 128
        Protocol: UDP (17)
        Header Checksum: 0x3047 [validation disabled]
        [Header checksum status: Unverified]
        Source Address: 192.168.43.247
        Destination Address: 192.168.43.1
```

- Noticed that the source address is my IP address
  - Noticed that the destination address is the IP address of the DNS server.
- 
- Expanded User Datagram Protocol to view UDP details.
  - Expanded Domain Name System (query) to view DNS details.
  - Expanded Flags to view flags details.
  - Expanded Queries to view query details.
  - Observed the query for [www.gstatic.com](http://www.gstatic.com).

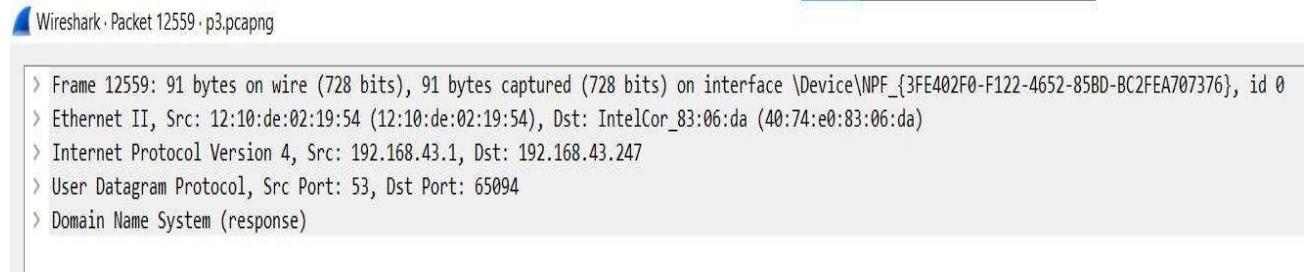
```
✓ User Datagram Protocol, Src Port: 65094, Dst Port: 53
    Source Port: 65094
    Destination Port: 53
    Length: 41
    Checksum: 0xbaf [unverified]
    [Checksum Status: Unverified]
    [Stream index: 337]
    > [Timestamps]
    UDP payload (33 bytes)
▼ Domain Name System (query)
    Transaction ID: 0x3844
    ▼ Flags: 0x0100 Standard query
        0... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ..1 .... .... = Recursion desired: Do query recursively
        .... .... .0... .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data: Unacceptable
        Questions: 1
        Answer RRs: 0
        Authority RRs: 0
        Additional RRs: 0
    ▼ Queries
        > www.gstatic.com: type A, class IN
        [Response In: 12559]
```

## ■ DNS Response Traffic:

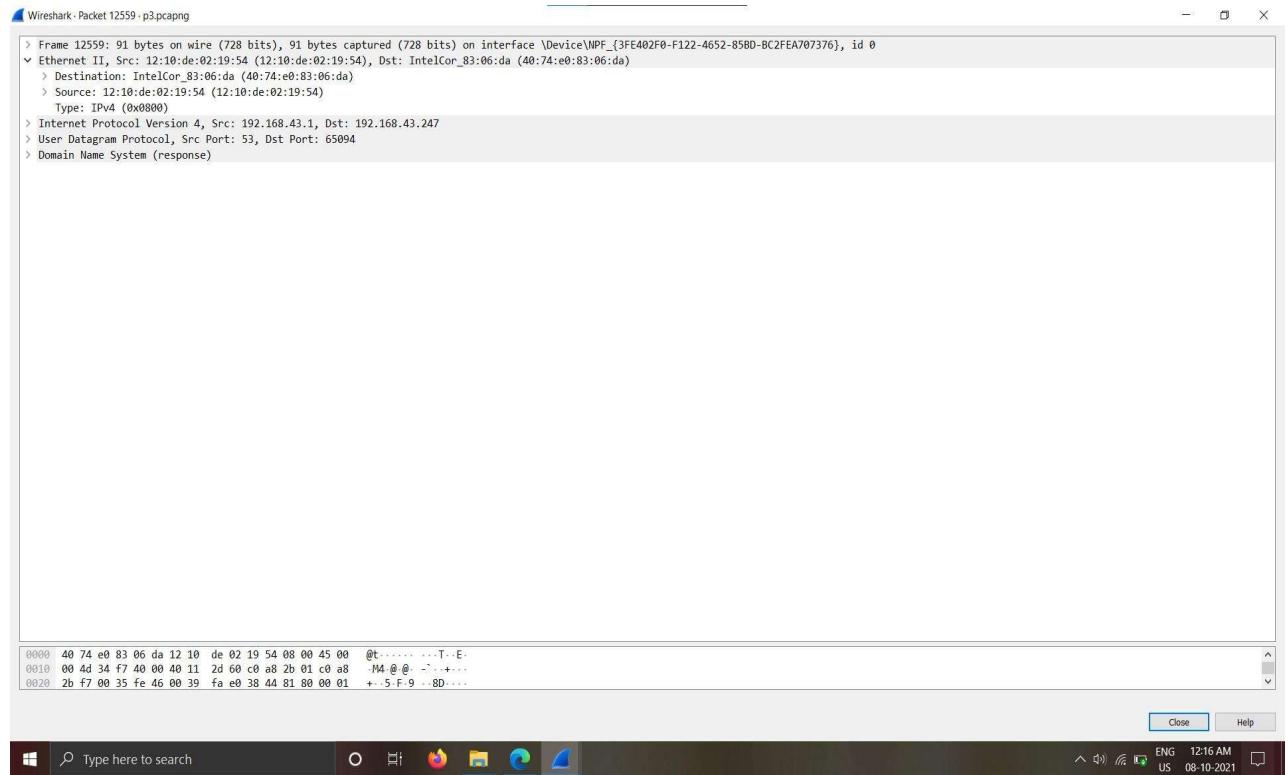
- Selected the next DNS packet, labeled Standard query response 0x3844 A www.gstatic.com A 142.250.77.131



- Noticed the Ethernet II / Internet Protocol Version 4 / User Datagram Protocol / Domain Name System (query) frame.



- Expanded Ethernet II to view Ethernet details.



- I observed Internet Protocol Version 4 to view IP details.

▼ Internet Protocol Version 4, Src: 192.168.43.1, Dst: 192.168.43.247

- 0100 .... = Version: 4
- .... 0101 = Header Length: 20 bytes (5)
- Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 77
- Identification: 0x34f7 (13559)
- Flags: 0x40, Don't fragment
- Fragment Offset: 0
- Time to Live: 64
- Protocol: UDP (17)
- Header Checksum: 0x2d60 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.43.1
- Destination Address: 192.168.43.247

- Noticed that the source address is my IP address

- Noticed that the destination address is the IP address of the DNS server.
- Expanded User Datagram Protocol to view UDP details.
- Expanded Domain Name System (query) to view DNS details.
- Expanded Flags to view flags details.
- Expanded Queries to view query details.
- Observed the query for [www.gstatic.com](http://www.gstatic.com).
- Expanded Answers to view answer details.

```

User Datagram Protocol, Src Port: 53, Dst Port: 65094
  Source Port: 53
  Destination Port: 65094
  Length: 57
  Checksum: 0xfae0 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 337]
  ▾ [Timestamps]
    [Time since first frame: 0.048761000 seconds]
    [Time since previous frame: 0.048761000 seconds]
  UDP payload (49 bytes)
Domain Name System (response)
  Transaction ID: 0x3844
  ▾ 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: 0
  ▾ Queries
    > www.gstatic.com: type A, class IN
  > Answers
    [Request In: 12486]
    [Time: 0.048761000 seconds]

```