Wireshark for Link layer

Submitted By,
SHRUSTI
CS22MTECH11017

Answer the following questions using the attached Wireshark trace files. Place the inline screenshot of the Wireshark along with your answers where you think it is appropriate to do so.

1. **Download /git clone the dhcp-client from** https://github.com/samueldotj/dhcp-client.git**. Execute the following steps**
cd dhcp-client/make

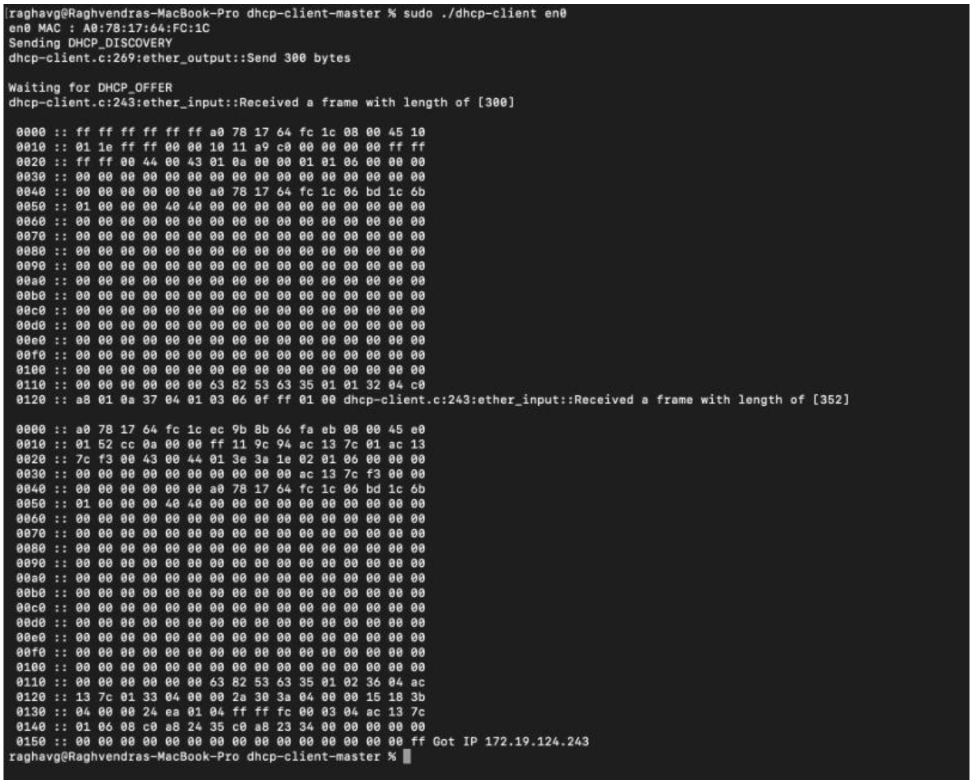
Confirm the successful build of dhcp-client binary/program in the same folder.
Start Wireshark and start capturing the packets.
Now try sudo ./dhcp-client <network interface name> (E.g wlp2s0, eno2, etc)
Stop capturing now.

Observe the output of this program on the terminal. Using the suitable filter on Wireshark, and output prints on the terminal, explain what is happening on this DHCP communication. List the source and destination IP and MAC addresses of all the associated packets seen and comment about how IP addresses are being mapped using MAC addresses. If all the packets required for the successful DHCP communication aren't seen, comment on what is expected to complete this DHCP communication successfully with important fields in the respective DHCP message.

Solution :

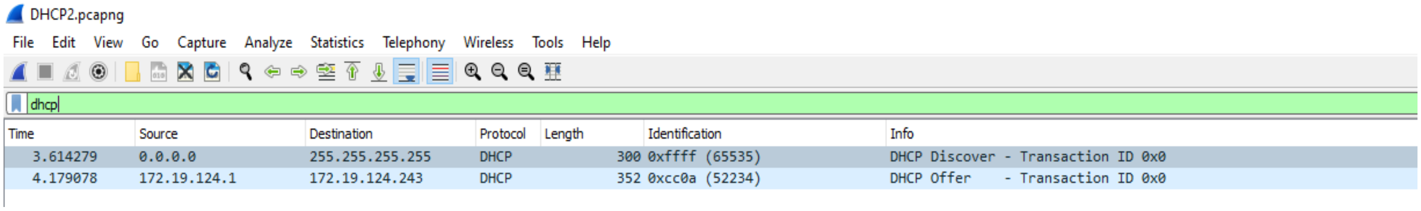
NOTE: *I was unable to capture packets in Wireshark due to a non-compatible system, hence by seeking Sir's permission, I have captured the packets from my friend's system and analysis is as follows.*

When a device wants access to a network that's using DHCP, it sends a request for an IP address that is picked up by a DHCP server. The server responds by delivering an IP address to the device, then monitors the use of the address and takes it back after a specified time or when the device shuts down. The IP address is then returned to the pool of addresses managed by the DHCP server to be reassigned to another device as it seeks access to the network.



```
raghavg@Raghvendras-MacBook-Pro dhcp-client-master % sudo ./dhcp-client en0
en0 MAC : A0:78:17:64:FC:1C
Sending DHCP_DISCOVERY
dhcp-client.c:269:ether_output::Send 300 bytes

Waiting for DHCP_OFFER
dhcp-client.c:243:ether_input::Received a frame with length of [300]

 0000 :: ff ff ff ff ff ff a0 78 17 64 fc 1c 08 00 45 10
 0010 :: 01 1e ff ff 00 00 10 11 a9 c0 00 00 00 00 ff ff
 0020 :: ff ff 00 44 00 43 01 0a 00 00 01 01 06 00 00 00
 0030 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0040 :: 00 00 00 00 00 00 a0 78 17 64 fc 1c 06 bd 1c 6b
 0050 :: 01 00 00 00 40 40 00 00 00 00 00 00 00 00 00 00
 0060 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0070 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0080 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0090 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00a0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00b0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00c0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00d0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00e0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00f0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0100 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0110 :: 00 00 00 00 00 00 63 82 53 63 35 01 01 32 04 c0
 0120 :: a8 01 0a 37 04 01 03 06 0f ff 01 00 dhcp-client.c:243:ether_input::Received a frame with length of [352]

 0000 :: a0 78 17 64 fc 1c ec 9b 8b 66 fa eb 08 00 45 e0
 0010 :: 01 52 cc 0a 00 00 ff 11 9c 94 ac 13 7c 01 ac 13
 0020 :: 7c f3 00 43 00 44 01 3e 3a 1e 02 01 06 00 00 00
 0030 :: 00 00 00 00 00 00 00 00 00 00 ac 13 7c f3 00 00
 0040 :: 00 00 00 00 00 00 a0 78 17 64 fc 1c 06 bd 1c 6b
 0050 :: 01 00 00 00 40 40 00 00 00 00 00 00 00 00 00 00
 0060 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0070 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0080 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0090 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00a0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00b0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00c0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00d0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00e0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 00f0 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0100 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
 0110 :: 00 00 00 00 00 00 63 82 53 63 35 01 02 36 04 ac
 0120 :: 13 7c 01 33 04 00 00 2a 30 3a 04 00 00 15 18 3b
 0130 :: 04 00 00 24 ea 01 04 ff ff fc 00 03 04 ac 13 7c
 0140 :: 01 06 08 c0 a8 24 35 c0 a8 23 34 00 00 00 00 00
 0150 :: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ff Got IP 172.19.124.243
raghavg@Raghvendras-MacBook-Pro dhcp-client-master % 
```

Applying “dhcp” filter:

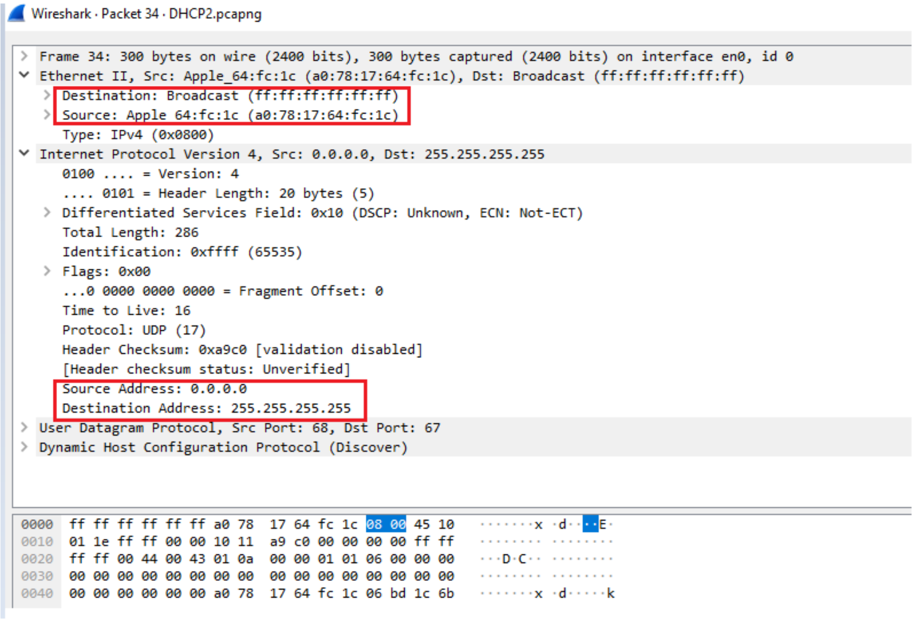


dhcp

Time	Source	Destination	Protocol	Length	Identification	Info
3.614279	0.0.0.0	255.255.255.255	DHCP	300	0xffff (65535)	DHCP Discover - Transaction ID 0x0
4.179078	172.19.124.1	172.19.124.243	DHCP	352	0xcc0a (52234)	DHCP Offer - Transaction ID 0x0

Dynamic Host Configuration Protocol is a network management protocol that is used to dynamically assign the IP address to each host on the network so that they can communicate efficiently. DHCP automates and centrally manages the assignment of IP addresses, easing the work of network administrators. In addition to the IP address, the DHCP also assigns the subnet masks, default gateway and domain name server(DNS) address and other configuration to the host.

DHCP Discovery:



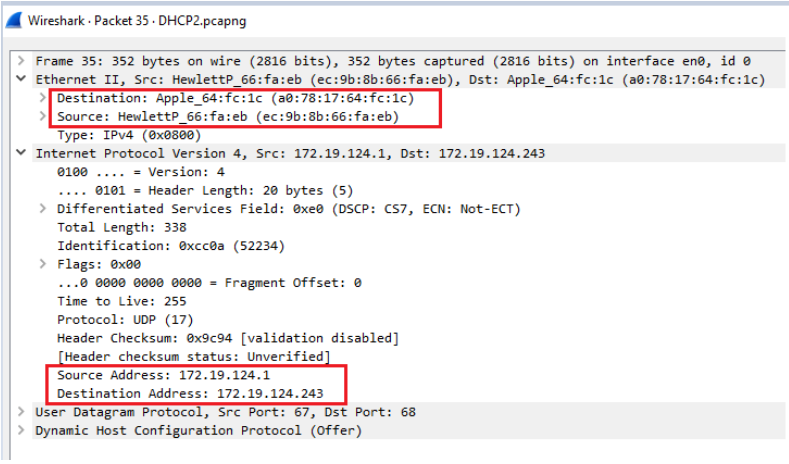
Wireshark · Packet 34 · DHCP2.pcapng

```
> Frame 34: 300 bytes on wire (2400 bits), 300 bytes captured (2400 bits) on interface en0, id 0
v Ethernet II, Src: Apple_64:fc:1c (a0:78:17:64:fc:1c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x10 (DSCP: Unknown, ECN: Not-ECT)
    Total Length: 286
    Identification: 0xffff (65535)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 16
    Protocol: UDP (17)
    Header Checksum: 0xa9c0 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 0.0.0.0
    Destination Address: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Dynamic Host Configuration Protocol (Discover)

0000  ff ff ff ff ff ff a0 78  17 64 fc 1c 08 00 45 10   ·······x ·d····E·
0010  01 1e ff ff 00 00 10 11  a9 c0 00 00 00 00 ff ff   ········ ········
0020  ff ff 00 44 00 43 01 0a  00 00 01 01 06 00 00 00   ···D·C·· ········
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0040  00 00 00 00 00 00 a0 78  17 64 fc 1c 06 bd 1c 6b   ·······x ·d·····k
```

Source IP address : **0.0.0.0**
SOurce MAC address: **a0:78:17:64:fc:1c**
Destination IP address : **255.255.255.255**
Destination MAC address : **ff:ff:ff:ff:ff:ff**

The DHCP client broadcasts messages to discover the DHCP servers. The client computer will set the destination MAC address as **ff:ff:ff:ff:ff:ff,** source MAC address as **a0:78:17:64:fc:1c**, source IP address as **0.0.0.0** and send a packet with the default broadcast destination of **255.255.255.255** or the specific subnet broadcast address if any configured.

DHCP Offer :



Wireshark · Packet 35 · DHCP2.pcapng

```
> Frame 35: 352 bytes on wire (2816 bits), 352 bytes captured (2816 bits) on interface en0, id 0
v Ethernet II, Src: HewlettP_66:fa:eb (ec:9b:8b:66:fa:eb), Dst: Apple_64:fc:1c (a0:78:17:64:fc:1c)
  > Destination: Apple_64:fc:1c (a0:78:17:64:fc:1c)
  > Source: HewlettP_66:fa:eb (ec:9b:8b:66:fa:eb)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 172.19.124.1, Dst: 172.19.124.243
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xe0 (DSCP: CS7, ECN: Not-ECT)
    Total Length: 338
    Identification: 0xcc0a (52234)
  > Flags: 0x00
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 255
    Protocol: UDP (17)
    Header Checksum: 0x9c94 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.19.124.1
    Destination Address: 172.19.124.243
> User Datagram Protocol, Src Port: 67, Dst Port: 68
> Dynamic Host Configuration Protocol (Offer)
```

Source IP address **: 172.19.124.1**
Source MAC address**: ec:9b:8b:66:fa:eb**
Destination IP address **: 172.19.126.143**
Destination MAC address **: a0:78:17:64:fc:1c**

When the DHCP server receives the DHCP Discover message then it suggests or offers an IP address(from IP address pool) to the client by sending a DHCP offer message to the client. This DHCP offer message contains the proposed IP address for DHCP client, IP address of the server, MAC address of the client, subnet mask, default gateway, DNS address, and lease information.

Mapping of IP addresses using MAC addresses:

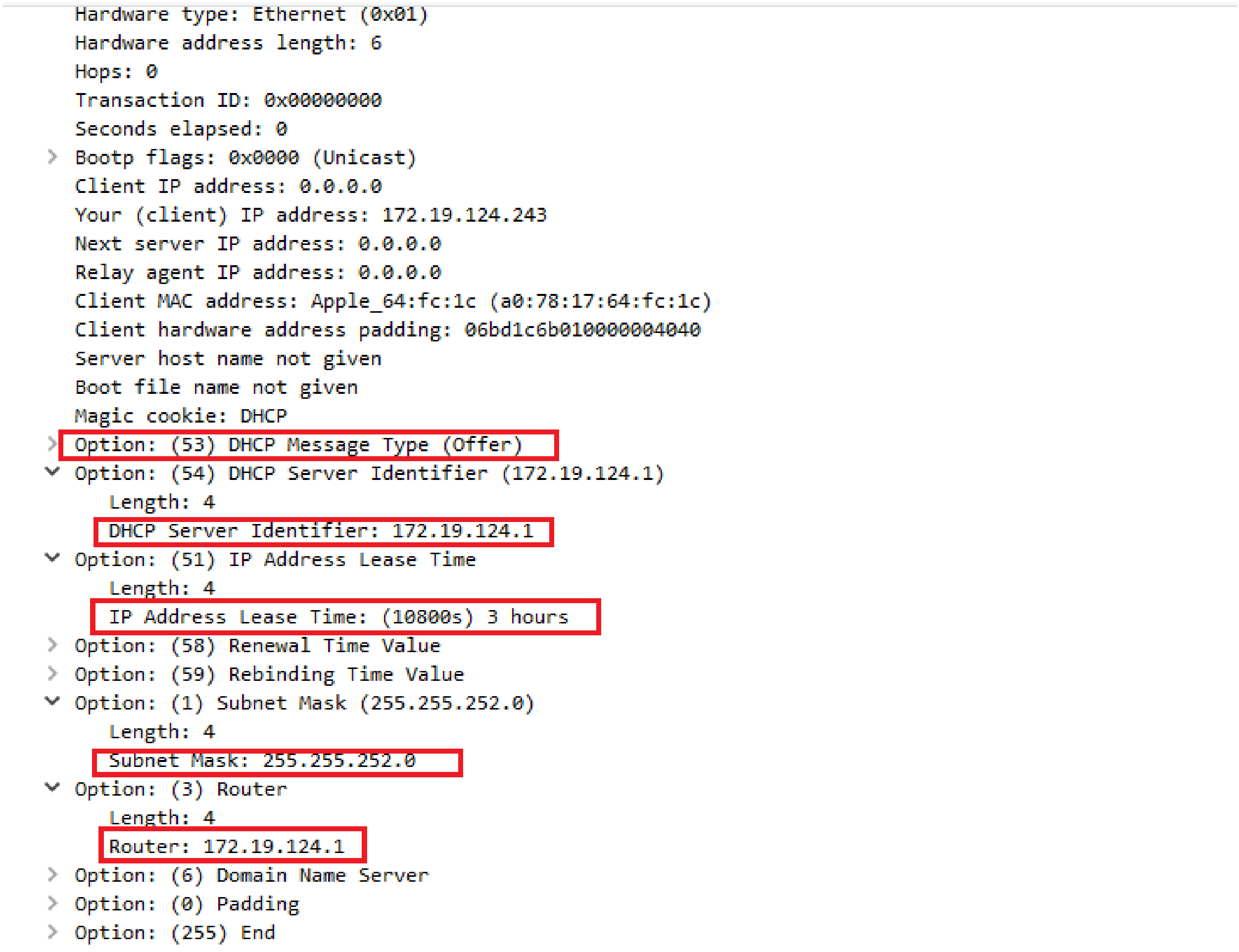
Step1 : client broadcasts the DHCP DISCOVER message over the network channel to establish a network connection with the DHCP server. This message indicates that the client device wants to connect to the internet through the DHCP server.

Step 2: when the DHCP server receives the DHCP DISCOVER message. According to the message, the DHCP server reserves an IP address for the connecting client and other network configuration settings, including subnet-mask default gateway, preferred DNS server, and shares it with the client device through the DHCP OFFER message.

Step 3: the client responds to the DHCP server's DHCP OFFER through a DHCPREQUEST message requesting the offered IP address and relevant network configuration sent by the DHCP server for the system.

Step 4: the server acknowledges the DHCP REQUEST broadcast from the client device and sends the DHCP ACK packet to the DHCP client, which comprises the required network configuration for the client device.

Other DHCP Components are :



```
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x00000000
    Seconds elapsed: 0
  > Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 172.19.124.243
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Client hardware address padding: 06bd1c6b010000004040
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Offer)
  v Option: (54) DHCP Server Identifier (172.19.124.1)
       Length: 4
       DHCP Server Identifier: 172.19.124.1
  v Option: (51) IP Address Lease Time
       Length: 4
       IP Address Lease Time: (10800s) 3 hours
  > Option: (58) Renewal Time Value
  > Option: (59) Rebinding Time Value
  v Option: (1) Subnet Mask (255.255.252.0)
       Length: 4
       Subnet Mask: 255.255.252.0
  v Option: (3) Router
       Length: 4
       Router: 172.19.124.1
  > Option: (6) Domain Name Server
  > Option: (0) Padding
  > Option: (255) End
```

DHCP Server Identifier : This is a networked device running the DHCP service that holds IP addresses and related configuration information.

IP Address Lease Time : The length of time for which a DHCP client holds the IP address information is known as the lease. When a lease expires, the client must renew it. Here, we can see the time is **3 hours.**

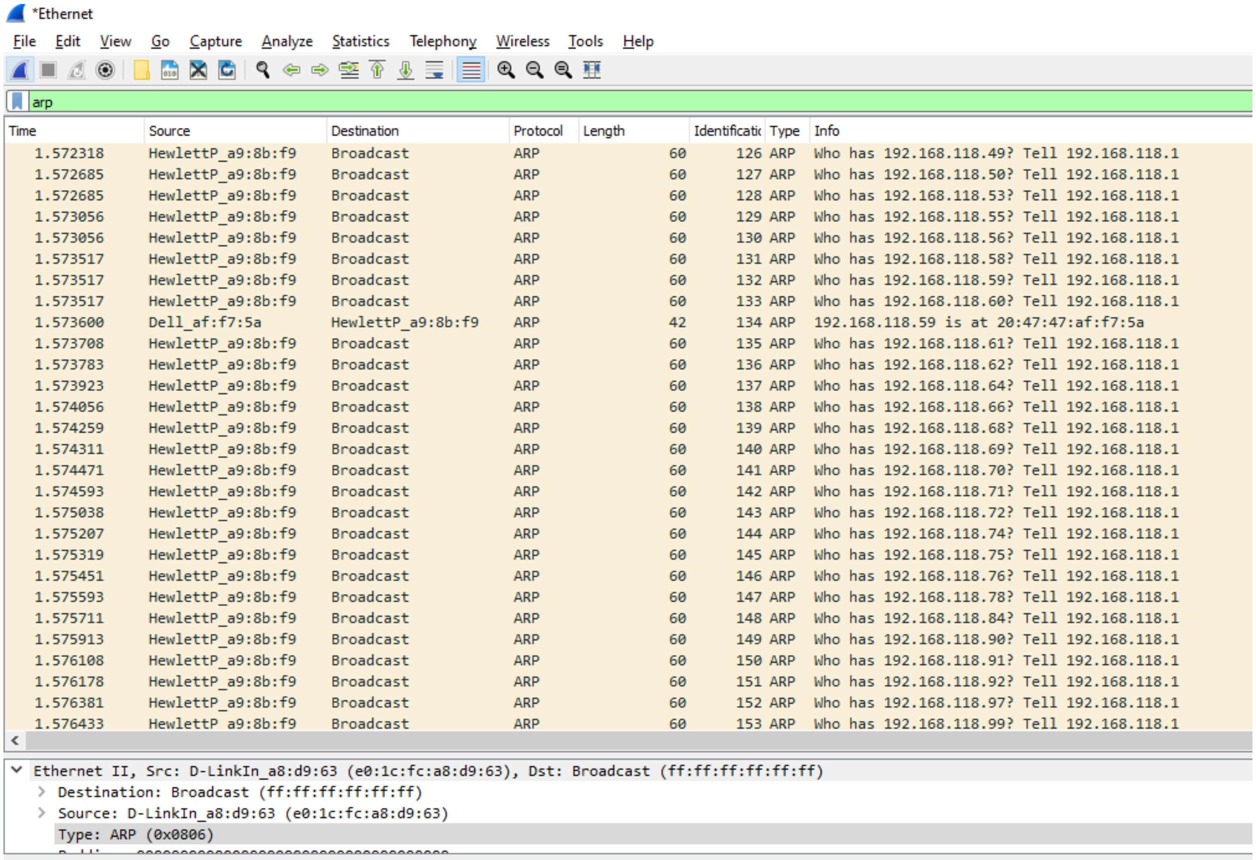
Subnet Mask : Tells on which subnet client is present.

Router : Gives the IP address of the first hop router in the communication path, here it is **172.19.124.1**

2. Start the Wireshark and start capturing the packets on any active network interface (like any, wlp2s0, eno2, etc) for a maximum of 30 seconds. Stop capturing now. Using the captured file, answer the following questions.

Note: 5 ARP replies were not obtained by capturing for 30sec hence captured the packets for nearly 60sec .

Apply the filter **arp** to observe the captured ARP packets.



arp

```
Time        Source              Destination         Protocol  Length  Identificati Type  Info
  1.572318  HewlettP_a9:8b:f9   Broadcast           ARP           60     126 ARP  Who has 192.168.118.49? Tell 192.168.118.1
  1.572685  HewlettP_a9:8b:f9   Broadcast           ARP           60     127 ARP  Who has 192.168.118.50? Tell 192.168.118.1
  1.572685  HewlettP_a9:8b:f9   Broadcast           ARP           60     128 ARP  Who has 192.168.118.53? Tell 192.168.118.1
  1.573056  HewlettP_a9:8b:f9   Broadcast           ARP           60     129 ARP  Who has 192.168.118.55? Tell 192.168.118.1
  1.573056  HewlettP_a9:8b:f9   Broadcast           ARP           60     130 ARP  Who has 192.168.118.56? Tell 192.168.118.1
  1.573517  HewlettP_a9:8b:f9   Broadcast           ARP           60     131 ARP  Who has 192.168.118.58? Tell 192.168.118.1
  1.573517  HewlettP_a9:8b:f9   Broadcast           ARP           60     132 ARP  Who has 192.168.118.59? Tell 192.168.118.1
  1.573517  HewlettP_a9:8b:f9   Broadcast           ARP           60     133 ARP  Who has 192.168.118.60? Tell 192.168.118.1
  1.573600  Dell_af:f7:5a       HewlettP_a9:8b:f9   ARP           42     134 ARP  192.168.118.59 is at 20:47:47:af:f7:5a
  1.573708  HewlettP_a9:8b:f9   Broadcast           ARP           60     135 ARP  Who has 192.168.118.61? Tell 192.168.118.1
  1.573783  HewlettP_a9:8b:f9   Broadcast           ARP           60     136 ARP  Who has 192.168.118.62? Tell 192.168.118.1
  1.573923  HewlettP_a9:8b:f9   Broadcast           ARP           60     137 ARP  Who has 192.168.118.64? Tell 192.168.118.1
  1.574056  HewlettP_a9:8b:f9   Broadcast           ARP           60     138 ARP  Who has 192.168.118.66? Tell 192.168.118.1
  1.574259  HewlettP_a9:8b:f9   Broadcast           ARP           60     139 ARP  Who has 192.168.118.68? Tell 192.168.118.1
  1.574311  HewlettP_a9:8b:f9   Broadcast           ARP           60     140 ARP  Who has 192.168.118.69? Tell 192.168.118.1
  1.574471  HewlettP_a9:8b:f9   Broadcast           ARP           60     141 ARP  Who has 192.168.118.70? Tell 192.168.118.1
  1.574593  HewlettP_a9:8b:f9   Broadcast           ARP           60     142 ARP  Who has 192.168.118.71? Tell 192.168.118.1
  1.575038  HewlettP_a9:8b:f9   Broadcast           ARP           60     143 ARP  Who has 192.168.118.72? Tell 192.168.118.1
  1.575207  HewlettP_a9:8b:f9   Broadcast           ARP           60     144 ARP  Who has 192.168.118.74? Tell 192.168.118.1
  1.575319  HewlettP_a9:8b:f9   Broadcast           ARP           60     145 ARP  Who has 192.168.118.75? Tell 192.168.118.1
  1.575451  HewlettP_a9:8b:f9   Broadcast           ARP           60     146 ARP  Who has 192.168.118.76? Tell 192.168.118.1
  1.575593  HewlettP_a9:8b:f9   Broadcast           ARP           60     147 ARP  Who has 192.168.118.78? Tell 192.168.118.1
  1.575711  HewlettP_a9:8b:f9   Broadcast           ARP           60     148 ARP  Who has 192.168.118.84? Tell 192.168.118.1
  1.575913  HewlettP_a9:8b:f9   Broadcast           ARP           60     149 ARP  Who has 192.168.118.90? Tell 192.168.118.1
  1.576108  HewlettP_a9:8b:f9   Broadcast           ARP           60     150 ARP  Who has 192.168.118.91? Tell 192.168.118.1
  1.576178  HewlettP_a9:8b:f9   Broadcast           ARP           60     151 ARP  Who has 192.168.118.92? Tell 192.168.118.1
  1.576381  HewlettP_a9:8b:f9   Broadcast           ARP           60     152 ARP  Who has 192.168.118.97? Tell 192.168.118.1
  1.576433  HewlettP_a9:8b:f9   Broadcast           ARP           60     153 ARP  Who has 192.168.118.99? Tell 192.168.118.1
```

```
v Ethernet II, Src: D-LinkIn_a8:d9:63 (e0:1c:fc:a8:d9:63), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: D-LinkIn_a8:d9:63 (e0:1c:fc:a8:d9:63)
    Type: ARP (0x0806)
```

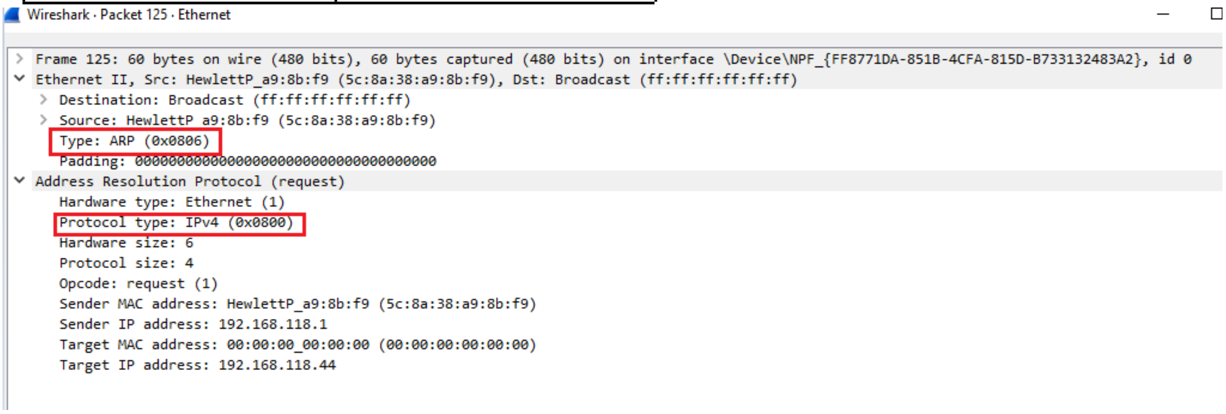
a. In the captured file, how many different types of ethernet payloads are present? Give the count for each type of the payload seen.

Solution:

Three different types of ethernet payload are present, they are shown below.

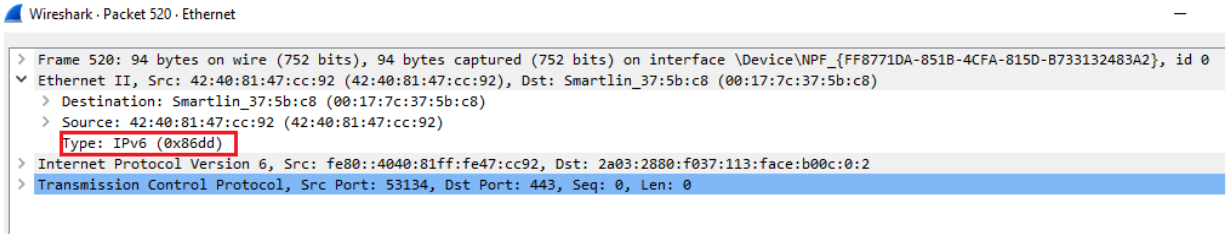
Type	**Count**
ARP (0x0806)	685
IPV4 (0x0800)	4801
IPv6 (0x86dd)	44

Total	**= 5618**



Wireshark · Packet 125 · Ethernet

```
> Frame 125: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Sender IP address: 192.168.118.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.118.44
```

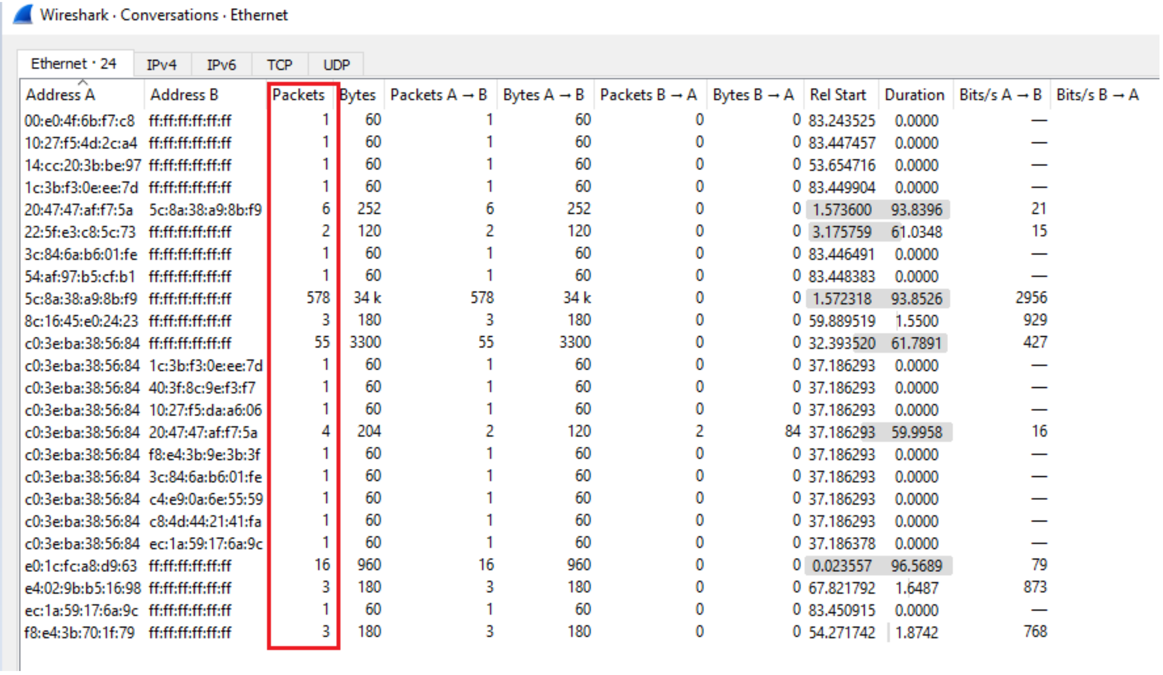


Wireshark · Packet 520 · Ethernet

```
> Frame 520: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: 42:40:81:47:cc:92 (42:40:81:47:cc:92), Dst: Smartlin_37:5b:c8 (00:17:7c:37:5b:c8)
  > Destination: Smartlin_37:5b:c8 (00:17:7c:37:5b:c8)
  > Source: 42:40:81:47:cc:92 (42:40:81:47:cc:92)
    Type: IPv6 (0x86dd)
> Internet Protocol Version 6, Src: fe80::4040:81ff:fe47:cc92, Dst: 2a03:2880:f037:113:face:b00c:0:2
> Transmission Control Protocol, Src Port: 53134, Dst Port: 443, Seq: 0, Len: 0
```

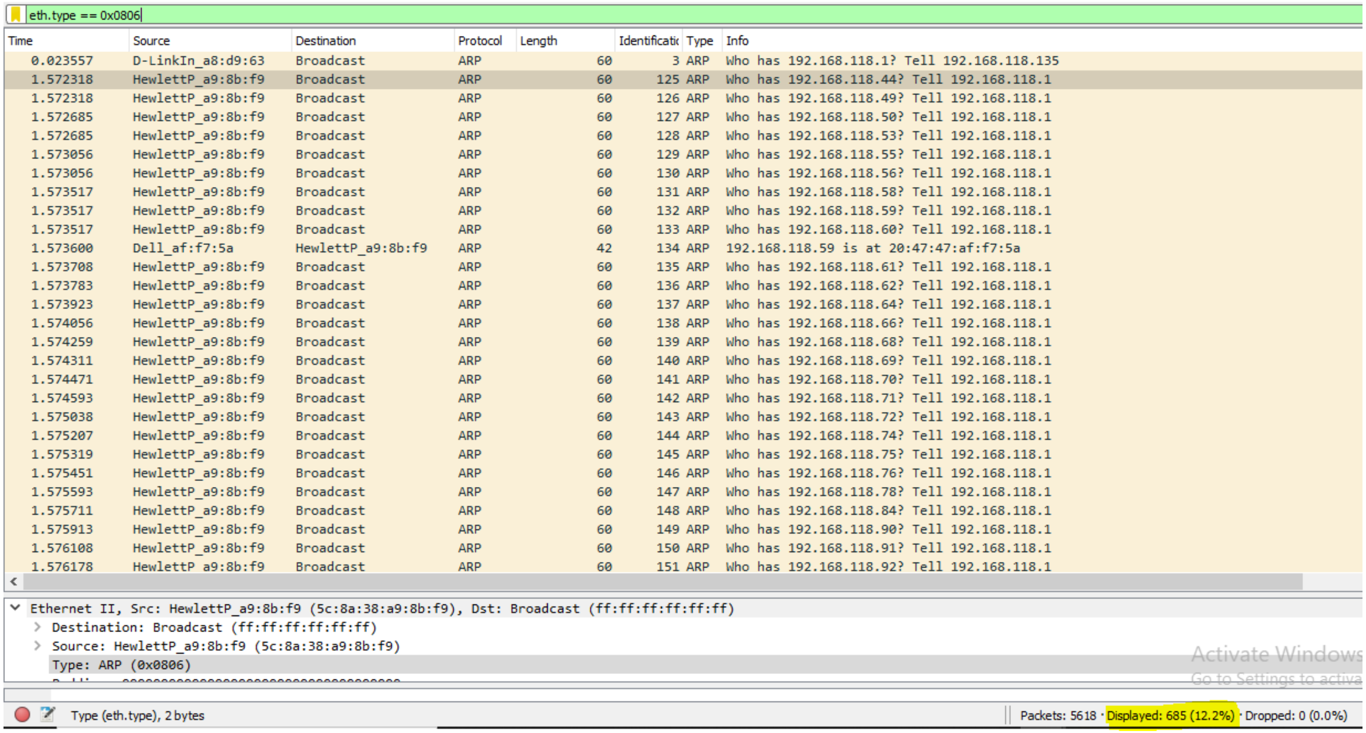
Number of packets in ARP:

Apply the filter **eth.type == 0x0806** and in Statistics → Conversations → limit to display filter observe the number of packets captured.



Wireshark · Conversations · Ethernet

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
00:e0:4f:6b:f7:c8	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.243525	0.0000	—	
10:27:f5:4d:2c:a4	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.447457	0.0000	—	
14:cc:20:3b:be:97	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	53.654716	0.0000	—	
1c:3b:f3:0e:ee:7d	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.449904	0.0000	—	
20:47:47:af:f7:5a	5c:8a:38:a9:8b:f9	6	252	6	252	0	0	1.573600	93.8396	21	
22:5f:e3:c8:5c:73	ff:ff:ff:ff:ff:ff	2	120	2	120	0	0	3.175759	61.0348	15	
3c:84:6a:b6:01:fe	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.446491	0.0000	—	
54:af:97:b5:cf:b1	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.448383	0.0000	—	
5c:8a:38:a9:8b:f9	ff:ff:ff:ff:ff:ff	578	34 k	578	34 k	0	0	1.572318	93.8526	2956	
8c:16:45:e0:24:23	ff:ff:ff:ff:ff:ff	3	180	3	180	0	0	59.889519	1.5500	929	
c0:3e:ba:38:56:84	ff:ff:ff:ff:ff:ff	55	3300	55	3300	0	0	32.393520	61.7891	427	
c0:3e:ba:38:56:84	1c:3b:f3:0e:ee:7d	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	40:3f:8c:9e:f3:f7	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	10:27:f5:da:a6:06	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	20:47:47:af:f7:5a	4	204	2	120	2	84	37.186293	59.9958	16	
c0:3e:ba:38:56:84	f8:e4:3b:9e:3b:3f	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	3c:84:6a:b6:01:fe	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	c4:e9:0a:6e:55:59	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	c8:4d:44:21:41:fa	1	60	1	60	0	0	37.186293	0.0000	—	
c0:3e:ba:38:56:84	ec:1a:59:17:6a:9c	1	60	1	60	0	0	37.186378	0.0000	—	
e0:1c:fc:a8:d9:63	ff:ff:ff:ff:ff:ff	16	960	16	960	0	0	0.023557	96.5689	79	
e4:02:9b:b5:16:98	ff:ff:ff:ff:ff:ff	3	180	3	180	0	0	67.821792	1.6487	873	
ec:1a:59:17:6a:9c	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.450915	0.0000	—	
f8:e4:3b:70:1f:79	ff:ff:ff:ff:ff:ff	3	180	3	180	0	0	54.271742	1.8742	768	



eth.type == 0x0806

Time	Source	Destination	Protocol	Length	Identificatic	Type	Info
0.023557	D-LinkIn_a8:d9:63	Broadcast	ARP	60	3	ARP	Who has 192.168.118.1? Tell 192.168.118.135
1.572318	HewlettP_a9:8b:f9	Broadcast	ARP	60	125	ARP	Who has 192.168.118.44? Tell 192.168.118.1
1.572318	HewlettP_a9:8b:f9	Broadcast	ARP	60	126	ARP	Who has 192.168.118.49? Tell 192.168.118.1
1.572685	HewlettP_a9:8b:f9	Broadcast	ARP	60	127	ARP	Who has 192.168.118.50? Tell 192.168.118.1
1.572685	HewlettP_a9:8b:f9	Broadcast	ARP	60	128	ARP	Who has 192.168.118.53? Tell 192.168.118.1
1.573056	HewlettP_a9:8b:f9	Broadcast	ARP	60	129	ARP	Who has 192.168.118.55? Tell 192.168.118.1
1.573056	HewlettP_a9:8b:f9	Broadcast	ARP	60	130	ARP	Who has 192.168.118.56? Tell 192.168.118.1
1.573517	HewlettP_a9:8b:f9	Broadcast	ARP	60	131	ARP	Who has 192.168.118.58? Tell 192.168.118.1
1.573517	HewlettP_a9:8b:f9	Broadcast	ARP	60	132	ARP	Who has 192.168.118.59? Tell 192.168.118.1
1.573517	HewlettP_a9:8b:f9	Broadcast	ARP	60	133	ARP	Who has 192.168.118.60? Tell 192.168.118.1
1.573600	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	134	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
1.573708	HewlettP_a9:8b:f9	Broadcast	ARP	60	135	ARP	Who has 192.168.118.61? Tell 192.168.118.1
1.573783	HewlettP_a9:8b:f9	Broadcast	ARP	60	136	ARP	Who has 192.168.118.62? Tell 192.168.118.1
1.573923	HewlettP_a9:8b:f9	Broadcast	ARP	60	137	ARP	Who has 192.168.118.64? Tell 192.168.118.1
1.574056	HewlettP_a9:8b:f9	Broadcast	ARP	60	138	ARP	Who has 192.168.118.66? Tell 192.168.118.1
1.574259	HewlettP_a9:8b:f9	Broadcast	ARP	60	139	ARP	Who has 192.168.118.68? Tell 192.168.118.1
1.574311	HewlettP_a9:8b:f9	Broadcast	ARP	60	140	ARP	Who has 192.168.118.69? Tell 192.168.118.1
1.574471	HewlettP_a9:8b:f9	Broadcast	ARP	60	141	ARP	Who has 192.168.118.70? Tell 192.168.118.1
1.574593	HewlettP_a9:8b:f9	Broadcast	ARP	60	142	ARP	Who has 192.168.118.71? Tell 192.168.118.1
1.575038	HewlettP_a9:8b:f9	Broadcast	ARP	60	143	ARP	Who has 192.168.118.72? Tell 192.168.118.1
1.575207	HewlettP_a9:8b:f9	Broadcast	ARP	60	144	ARP	Who has 192.168.118.74? Tell 192.168.118.1
1.575319	HewlettP_a9:8b:f9	Broadcast	ARP	60	145	ARP	Who has 192.168.118.75? Tell 192.168.118.1
1.575451	HewlettP_a9:8b:f9	Broadcast	ARP	60	146	ARP	Who has 192.168.118.76? Tell 192.168.118.1
1.575593	HewlettP_a9:8b:f9	Broadcast	ARP	60	147	ARP	Who has 192.168.118.78? Tell 192.168.118.1
1.575711	HewlettP_a9:8b:f9	Broadcast	ARP	60	148	ARP	Who has 192.168.118.84? Tell 192.168.118.1
1.575913	HewlettP_a9:8b:f9	Broadcast	ARP	60	149	ARP	Who has 192.168.118.90? Tell 192.168.118.1
1.576108	HewlettP_a9:8b:f9	Broadcast	ARP	60	150	ARP	Who has 192.168.118.91? Tell 192.168.118.1
1.576178	HewlettP_a9:8b:f9	Broadcast	ARP	60	151	ARP	Who has 192.168.118.92? Tell 192.168.118.1

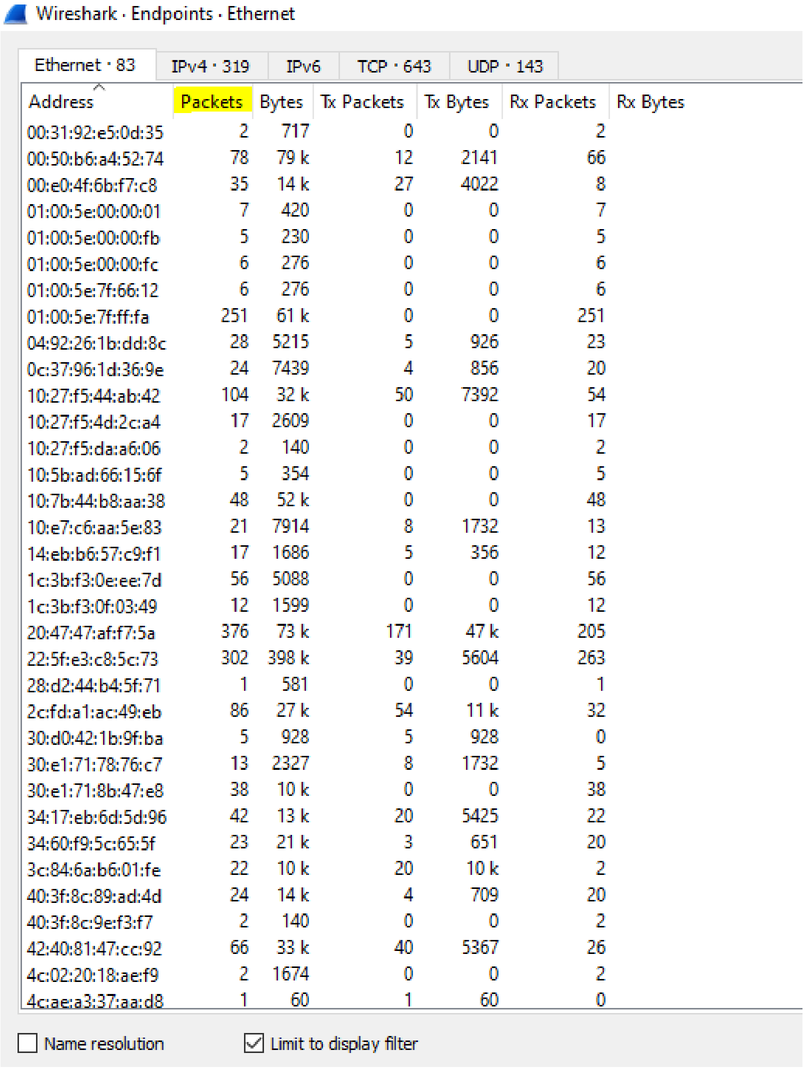
```
Ethernet II, Src: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
  Type: ARP (0x0806)
```

Type (eth.type), 2 bytes

Packets: 5618 · Displayed: 685 (12.2%) · Dropped: 0 (0.0%)

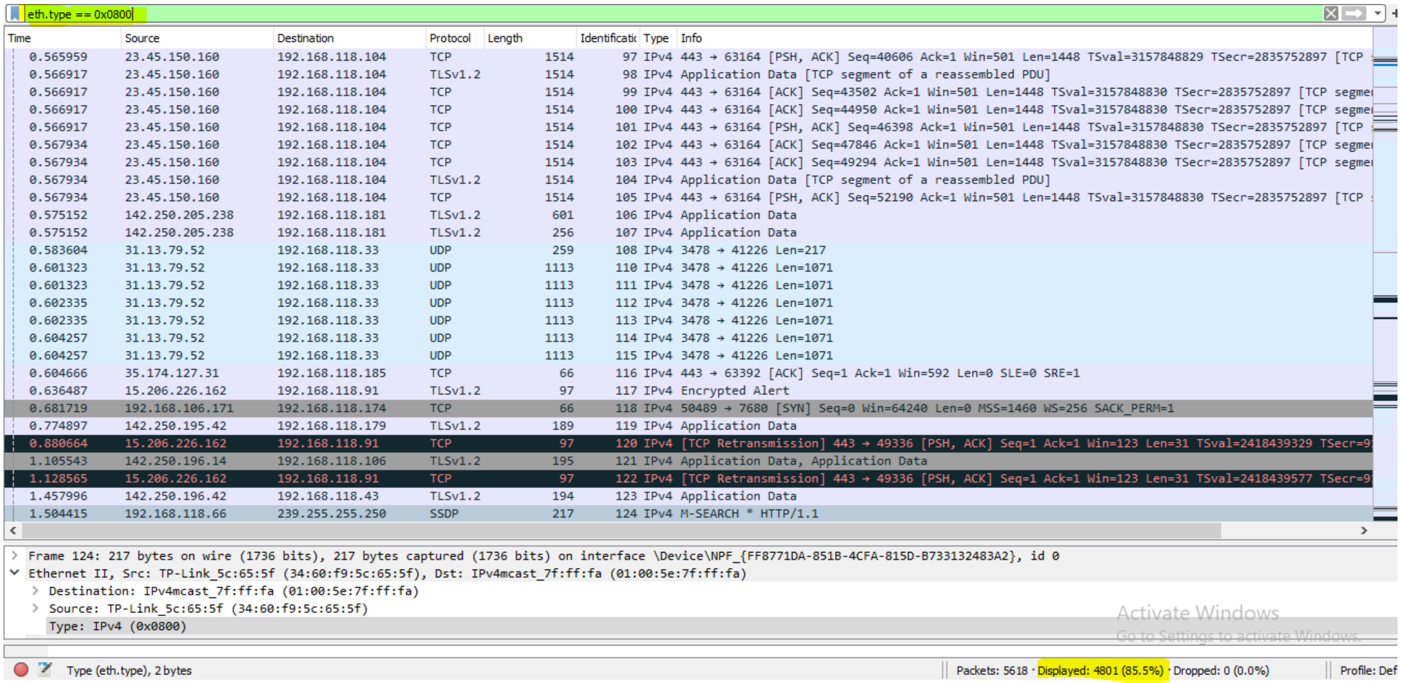
Number of packets in IPV4:

Apply the filter **eth.type == 0x0800** and in Statistics → Conversations → limit to display filter observe the number of packets captured.



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:31:92:e5:0d:35	2	717	0	0	2	
00:50:b6:a4:52:74	78	79 k	12	2141	66	
00:e0:4f:6b:f7:c8	35	14 k	27	4022	8	
01:00:5e:00:00:01	7	420	0	0	7	
01:00:5e:00:00:fb	5	230	0	0	5	
01:00:5e:00:00:fc	6	276	0	0	6	
01:00:5e:7f:66:12	6	276	0	0	6	
01:00:5e:7f:ff:fa	251	61 k	0	0	251	
04:92:26:1b:dd:8c	28	5215	5	926	23	
0c:37:96:1d:36:9e	24	7439	4	856	20	
10:27:f5:44:ab:42	104	32 k	50	7392	54	
10:27:f5:4d:2c:a4	17	2609	0	0	17	
10:27:f5:da:a6:06	2	140	0	0	2	
10:5b:ad:66:15:6f	5	354	0	0	5	
10:7b:44:b8:aa:38	48	52 k	0	0	48	
10:e7:c6:aa:5e:83	21	7914	8	1732	13	
14:eb:b6:57:c9:f1	17	1686	5	356	12	
1c:3b:f3:0e:ee:7d	56	5088	0	0	56	
1c:3b:f3:0f:03:49	12	1599	0	0	12	
20:47:47:af:f7:5a	376	73 k	171	47 k	205	
22:5f:e3:c8:5c:73	302	398 k	39	5604	263	
28:d2:44:b4:5f:71	1	581	0	0	1	
2c:fd:a1:ac:49:eb	86	27 k	54	11 k	32	
30:d0:42:1b:9f:ba	5	928	5	928	0	
30:e1:71:78:76:c7	13	2327	8	1732	5	
30:e1:71:8b:47:e8	38	10 k	0	0	38	
34:17:eb:6d:5d:96	42	13 k	20	5425	22	
34:60:f9:5c:65:5f	23	21 k	3	651	20	
3c:84:6a:b6:01:fe	22	10 k	20	10 k	2	
40:3f:8c:89:ad:4d	24	14 k	4	709	20	
40:3f:8c:9e:f3:f7	2	140	0	0	2	
42:40:81:47:cc:92	66	33 k	40	5367	26	
4c:02:20:18:ae:f9	2	1674	0	0	2	
4c:ae:a3:37:aa:d8	1	60	1	60	0	

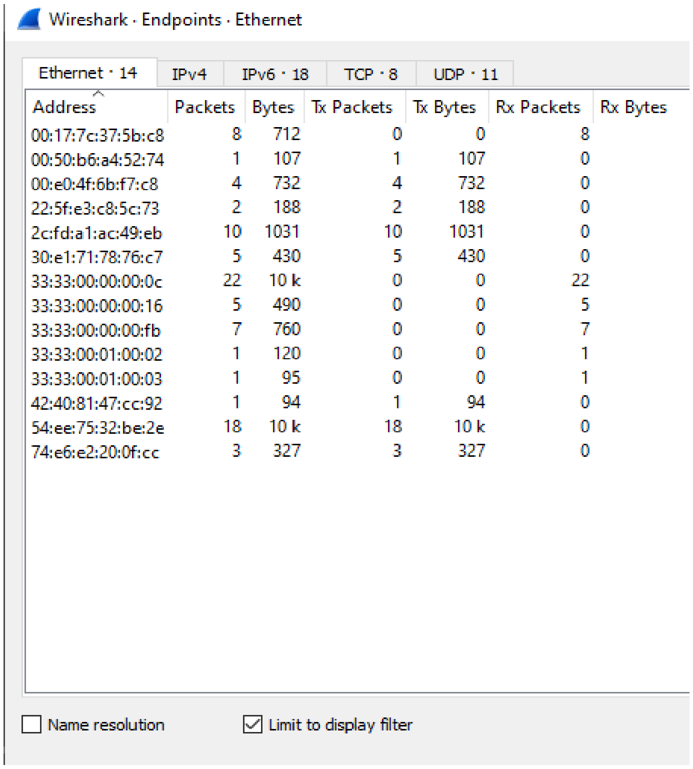
☐ Name resolution ☑ Limit to display filter



```
eth.type == 0x0800

Time       Source            Destination       Protocol  Length  Identificati Type Info
0.565959   23.45.150.160     192.168.118.104   TCP       1514      97 IPv4 443 → 63164 [PSH, ACK] Seq=40606 Ack=1 Win=501 Len=1448 TSval=3157848829 TSecr=2835752897 [TCP
0.566917   23.45.150.160     192.168.118.104   TLSv1.2   1514      98 IPv4 Application Data [TCP segment of a reassembled PDU]
0.566917   23.45.150.160     192.168.118.104   TCP       1514      99 IPv4 443 → 63164 [ACK] Seq=43502 Ack=1 Win=501 Len=1448 TSval=3157848830 TSecr=2835752897 [TCP segme
0.566917   23.45.150.160     192.168.118.104   TCP       1514     100 IPv4 443 → 63164 [ACK] Seq=44950 Ack=1 Win=501 Len=1448 TSval=3157848830 TSecr=2835752897 [TCP segme
0.566917   23.45.150.160     192.168.118.104   TCP       1514     101 IPv4 443 → 63164 [PSH, ACK] Seq=46398 Ack=1 Win=501 Len=1448 TSval=3157848830 TSecr=2835752897 [TCP
0.567934   23.45.150.160     192.168.118.104   TCP       1514     102 IPv4 443 → 63164 [ACK] Seq=47846 Ack=1 Win=501 Len=1448 TSval=3157848830 TSecr=2835752897 [TCP segme
0.567934   23.45.150.160     192.168.118.104   TCP       1514     103 IPv4 443 → 63164 [ACK] Seq=49294 Ack=1 Win=501 Len=1448 TSval=3157848830 TSecr=2835752897 [TCP segme
0.567934   23.45.150.160     192.168.118.104   TLSv1.2   1514     104 IPv4 Application Data [TCP segment of a reassembled PDU]
0.567934   23.45.150.160     192.168.118.104   TCP       1514     105 IPv4 443 → 63164 [PSH, ACK] Seq=52190 Ack=1 Win=501 Len=1448 TSval=3157848830 TSecr=2835752897 [TCP
0.575152   142.250.205.238   192.168.118.181   TLSv1.2    601     106 IPv4 Application Data
0.575152   142.250.205.238   192.168.118.181   TLSv1.2    256     107 IPv4 Application Data
0.583604   31.13.79.52       192.168.118.33    UDP        259     108 IPv4 3478 → 41226 Len=217
0.601323   31.13.79.52       192.168.118.33    UDP       1113     110 IPv4 3478 → 41226 Len=1071
0.601323   31.13.79.52       192.168.118.33    UDP       1113     111 IPv4 3478 → 41226 Len=1071
0.602335   31.13.79.52       192.168.118.33    UDP       1113     112 IPv4 3478 → 41226 Len=1071
0.602335   31.13.79.52       192.168.118.33    UDP       1113     113 IPv4 3478 → 41226 Len=1071
0.604257   31.13.79.52       192.168.118.33    UDP       1113     114 IPv4 3478 → 41226 Len=1071
0.604257   31.13.79.52       192.168.118.33    UDP       1113     115 IPv4 3478 → 41226 Len=1071
0.604666   35.174.127.31     192.168.118.185   TCP         66     116 IPv4 443 → 63392 [ACK] Seq=1 Ack=1 Win=592 Len=0 SLE=0 SRE=1
0.636487   15.206.226.162    192.168.118.91    TLSv1.2     97     117 IPv4 Encrypted Alert
0.681719   192.168.106.171   192.168.118.174   TCP         66     118 IPv4 50489 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
0.774897   142.250.195.42    192.168.118.179   TLSv1.2    189     119 IPv4 Application Data
0.880664   15.206.226.162    192.168.118.91    TCP         97     120 IPv4 [TCP Retransmission] 443 → 49336 [PSH, ACK] Seq=1 Ack=1 Win=123 Len=31 TSval=2418439329 TSecr=9
1.105543   142.250.196.14    192.168.118.106   TLSv1.2    195     121 IPv4 Application Data, Application Data
1.128565   15.206.226.162    192.168.118.91    TCP         97     122 IPv4 [TCP Retransmission] 443 → 49336 [PSH, ACK] Seq=1 Ack=1 Win=123 Len=31 TSval=2418439577 TSecr=9
1.457996   142.250.196.42    192.168.118.43    TLSv1.2    194     123 IPv4 Application Data
1.504415   192.168.118.66    239.255.255.250   SSDP       217     124 IPv4 M-SEARCH * HTTP/1.1

> Frame 124: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: TP-Link_5c:65:5f (34:60:f9:5c:65:5f), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Source: TP-Link_5c:65:5f (34:60:f9:5c:65:5f)
    Type: IPv4 (0x0800)

Type (eth.type), 2 bytes          Packets: 5618 · Displayed: 4801 (85.5%) · Dropped: 0 (0.0%)          Profile: Def
```

Number of packets in IPV6:

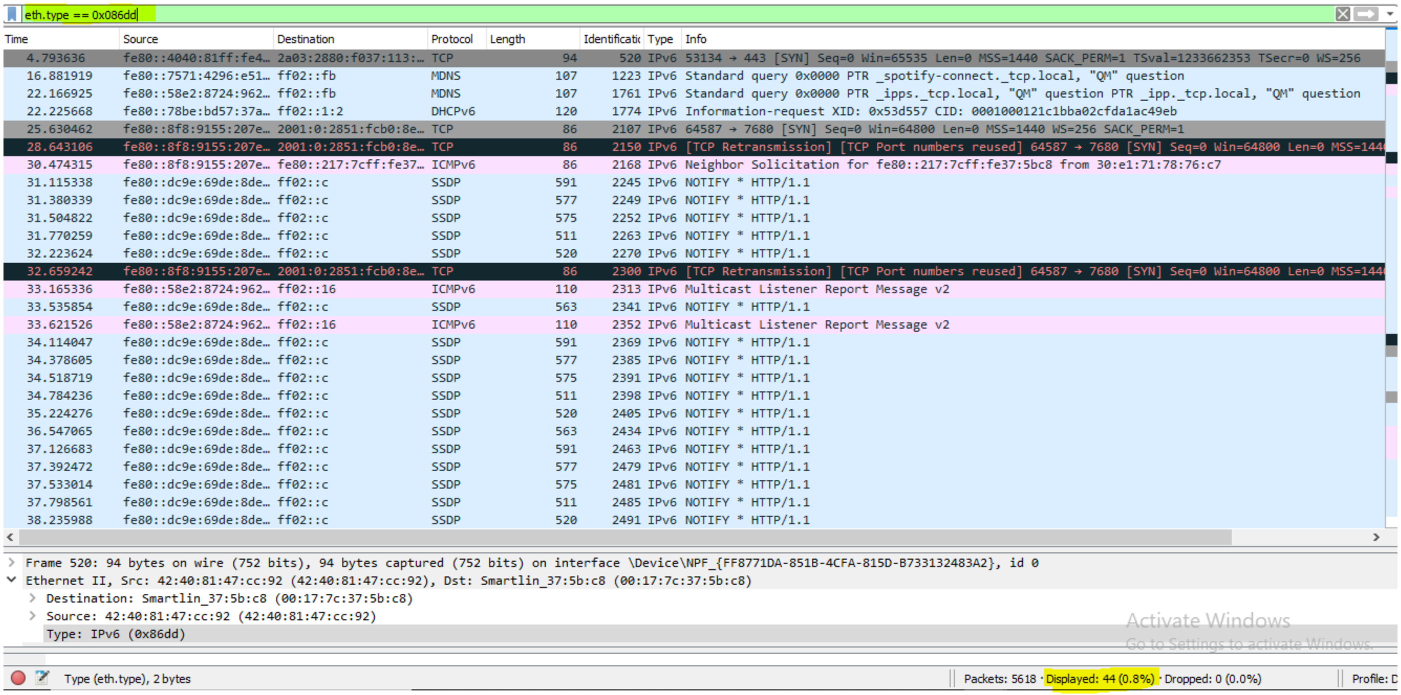
Apply the filter **eth.type == 0x086dd** and in Statistics → Conversations → limit to display filter observe the number of packets captured.



Wireshark · Endpoints · Ethernet

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:17:7c:37:5b:c8	8	712	0	0	8	
00:50:b6:a4:52:74	1	107	1	107	0	
00:e0:4f:6b:f7:c8	4	732	4	732	0	
22:5f:e3:c8:5c:73	2	188	2	188	0	
2c:fd:a1:ac:49:eb	10	1031	10	1031	0	
30:e1:71:78:76:c7	5	430	5	430	0	
33:33:00:00:00:0c	22	10 k	0	0	22	
33:33:00:00:00:16	5	490	0	0	5	
33:33:00:00:00:fb	7	760	0	0	7	
33:33:00:01:00:02	1	120	0	0	1	
33:33:00:01:00:03	1	95	0	0	1	
42:40:81:47:cc:92	1	94	1	94	0	
54:ee:75:32:be:2e	18	10 k	18	10 k	0	
74:e6:e2:20:0f:cc	3	327	3	327	0	

□ Name resolution ☑ Limit to display filter



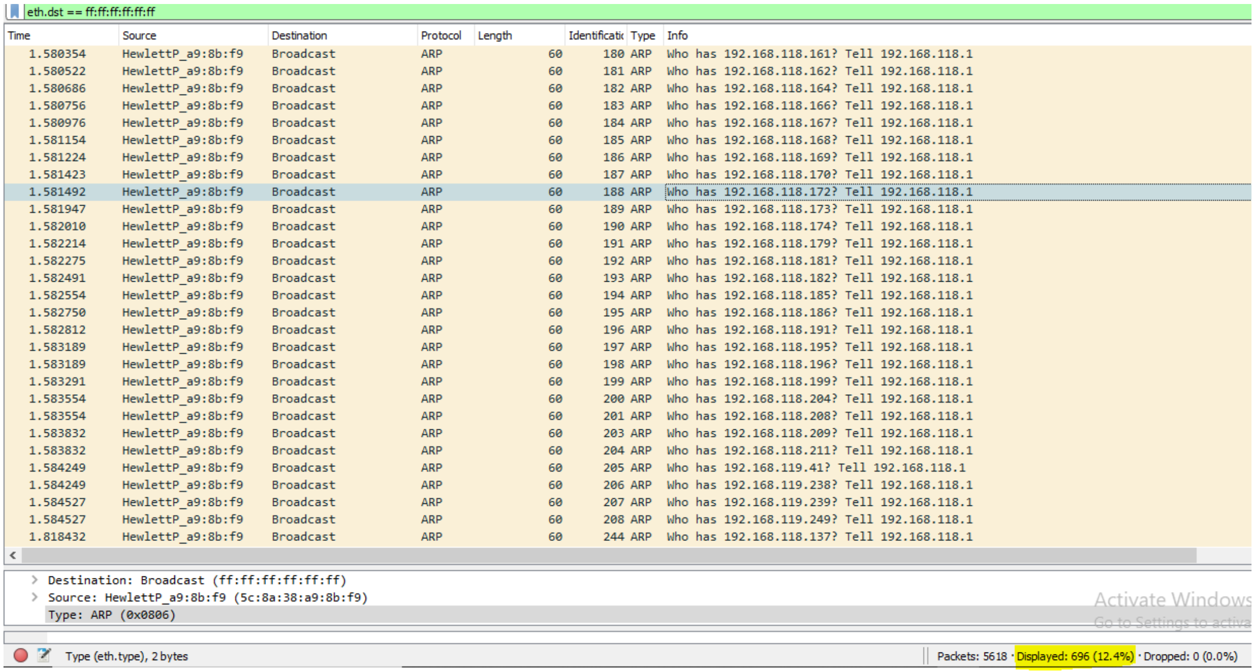
```
eth.type == 0x086dd

Time        Source                 Destination             Protocol  Length  Identificati Type Info
  4.793636  fe80::4040:81ff:fe4…  2a03:2880:f037:113:…  TCP           94    520 IPv6 53134 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 SACK_PERM=1 TSval=1233662353 TSecr=0 WS=256
 16.881919  fe80::7571:4296:e51…  ff02::fb              MDNS         107   1223 IPv6 Standard query 0x0000 PTR _spotify-connect._tcp.local, "QM" question
 22.166925  fe80::58e2:8724:962…  ff02::fb              MDNS         107   1761 IPv6 Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
 22.225668  fe80::78be:bd57:37a…  ff02::1:2             DHCPv6       120   1774 IPv6 Information-request XID: 0x53d557 CID: 0001000121c1bba02cfda1ac49eb
 25.630462  fe80::8f8:9155:207e…  2001:0:2851:fcb0:8e…  TCP           86   2107 IPv6 64587 → 7680 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 WS=256 SACK_PERM=1
 28.643106  fe80::8f8:9155:207e…  2001:0:2851:fcb0:8e…  TCP           86   2150 IPv6 [TCP Retransmission] [TCP Port numbers reused] 64587 → 7680 [SYN] Seq=0 Win=64800 Len=0 MSS=144
 30.474315  fe80::8f8:9155:207e…  fe80::217:7cff:fe37…  ICMPv6        86   2168 IPv6 Neighbor Solicitation for fe80::217:7cff:fe37:5bc8 from 30:e1:71:78:76:c7
 31.115338  fe80::dc9e:69de:8de…  ff02::c               SSDP         591   2245 IPv6 NOTIFY * HTTP/1.1
 31.380339  fe80::dc9e:69de:8de…  ff02::c               SSDP         577   2249 IPv6 NOTIFY * HTTP/1.1
 31.504822  fe80::dc9e:69de:8de…  ff02::c               SSDP         575   2252 IPv6 NOTIFY * HTTP/1.1
 31.770259  fe80::dc9e:69de:8de…  ff02::c               SSDP         511   2263 IPv6 NOTIFY * HTTP/1.1
 32.223624  fe80::dc9e:69de:8de…  ff02::c               SSDP         520   2270 IPv6 NOTIFY * HTTP/1.1
 32.659242  fe80::8f8:9155:207e…  2001:0:2851:fcb0:8e…  TCP           86   2300 IPv6 [TCP Retransmission] [TCP Port numbers reused] 64587 → 7680 [SYN] Seq=0 Win=64800 Len=0 MSS=144
 33.165336  fe80::58e2:8724:962…  ff02::16              ICMPv6       110   2313 IPv6 Multicast Listener Report Message v2
 33.535854  fe80::dc9e:69de:8de…  ff02::c               SSDP         563   2341 IPv6 NOTIFY * HTTP/1.1
 33.621526  fe80::58e2:8724:962…  ff02::16              ICMPv6       110   2352 IPv6 Multicast Listener Report Message v2
 34.114047  fe80::dc9e:69de:8de…  ff02::c               SSDP         591   2369 IPv6 NOTIFY * HTTP/1.1
 34.378605  fe80::dc9e:69de:8de…  ff02::c               SSDP         577   2385 IPv6 NOTIFY * HTTP/1.1
 34.518719  fe80::dc9e:69de:8de…  ff02::c               SSDP         575   2391 IPv6 NOTIFY * HTTP/1.1
 34.784236  fe80::dc9e:69de:8de…  ff02::c               SSDP         511   2398 IPv6 NOTIFY * HTTP/1.1
 35.224276  fe80::dc9e:69de:8de…  ff02::c               SSDP         520   2405 IPv6 NOTIFY * HTTP/1.1
 36.547065  fe80::dc9e:69de:8de…  ff02::c               SSDP         563   2434 IPv6 NOTIFY * HTTP/1.1
 37.126683  fe80::dc9e:69de:8de…  ff02::c               SSDP         591   2463 IPv6 NOTIFY * HTTP/1.1
 37.392472  fe80::dc9e:69de:8de…  ff02::c               SSDP         577   2479 IPv6 NOTIFY * HTTP/1.1
 37.533014  fe80::dc9e:69de:8de…  ff02::c               SSDP         575   2481 IPv6 NOTIFY * HTTP/1.1
 37.798561  fe80::dc9e:69de:8de…  ff02::c               SSDP         511   2485 IPv6 NOTIFY * HTTP/1.1
 38.235988  fe80::dc9e:69de:8de…  ff02::c               SSDP         520   2491 IPv6 NOTIFY * HTTP/1.1

> Frame 520: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: 42:40:81:47:cc:92 (42:40:81:47:cc:92), Dst: Smartlin_37:5b:c8 (00:17:7c:37:5b:c8)
  > Destination: Smartlin_37:5b:c8 (00:17:7c:37:5b:c8)
  > Source: 42:40:81:47:cc:92 (42:40:81:47:cc:92)
    Type: IPv6 (0x86dd)

Type (eth.type), 2 bytes        Packets: 5618 · Displayed: 44 (0.8%) · Dropped: 0 (0.0%)        Profile: D
```

b. In the captured file, how many ARP Requests are present ? How many ARP responses are present ? How do you find if an ARP packet is a request or a response ? Also make a table with a minimum of 5 entries. Each entry will have the details about the ARP Request, ARP Reply and the count of such ARP request-response transactions between two machines. i.e., the respective MAC address from the corresponding ARP Request, the MAC address from its ARP Reply (E.g: If X is sending an ARP request and Y is responding to it, there shall be an entry in the table for X's details, Y's details and another column indicating the count of how many times this request-reply has occurred. If there is no ARP reply for a request, you can place the count as 0).

Solution:

Number of ARP Requests = 696

Apply the filter **eth.dst == ff:ff:ff:ff:ff:ff** and in Statistics → Conversations → limit to display filter observe the number of packets captured.



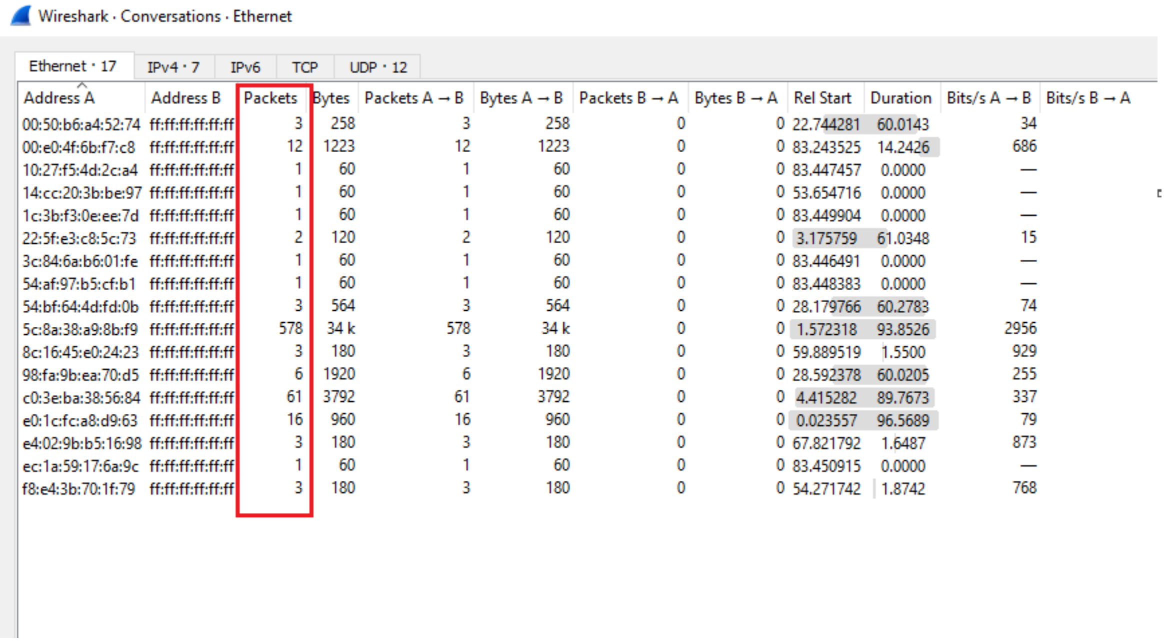
eth.dst == ff:ff:ff:ff:ff:ff

Time	Source	Destination	Protocol	Length	Identificati	Type	Info
1.580354	HewlettP_a9:8b:f9	Broadcast	ARP	60	180	ARP	Who has 192.168.118.161? Tell 192.168.118.1
1.580522	HewlettP_a9:8b:f9	Broadcast	ARP	60	181	ARP	Who has 192.168.118.162? Tell 192.168.118.1
1.580686	HewlettP_a9:8b:f9	Broadcast	ARP	60	182	ARP	Who has 192.168.118.164? Tell 192.168.118.1
1.580756	HewlettP_a9:8b:f9	Broadcast	ARP	60	183	ARP	Who has 192.168.118.166? Tell 192.168.118.1
1.580976	HewlettP_a9:8b:f9	Broadcast	ARP	60	184	ARP	Who has 192.168.118.167? Tell 192.168.118.1
1.581154	HewlettP_a9:8b:f9	Broadcast	ARP	60	185	ARP	Who has 192.168.118.168? Tell 192.168.118.1
1.581224	HewlettP_a9:8b:f9	Broadcast	ARP	60	186	ARP	Who has 192.168.118.169? Tell 192.168.118.1
1.581423	HewlettP_a9:8b:f9	Broadcast	ARP	60	187	ARP	Who has 192.168.118.170? Tell 192.168.118.1
1.581492	HewlettP_a9:8b:f9	Broadcast	ARP	60	188	ARP	Who has 192.168.118.172? Tell 192.168.118.1
1.581947	HewlettP_a9:8b:f9	Broadcast	ARP	60	189	ARP	Who has 192.168.118.173? Tell 192.168.118.1
1.582010	HewlettP_a9:8b:f9	Broadcast	ARP	60	190	ARP	Who has 192.168.118.174? Tell 192.168.118.1
1.582214	HewlettP_a9:8b:f9	Broadcast	ARP	60	191	ARP	Who has 192.168.118.179? Tell 192.168.118.1
1.582275	HewlettP_a9:8b:f9	Broadcast	ARP	60	192	ARP	Who has 192.168.118.181? Tell 192.168.118.1
1.582491	HewlettP_a9:8b:f9	Broadcast	ARP	60	193	ARP	Who has 192.168.118.182? Tell 192.168.118.1
1.582554	HewlettP_a9:8b:f9	Broadcast	ARP	60	194	ARP	Who has 192.168.118.185? Tell 192.168.118.1
1.582750	HewlettP_a9:8b:f9	Broadcast	ARP	60	195	ARP	Who has 192.168.118.186? Tell 192.168.118.1
1.582812	HewlettP_a9:8b:f9	Broadcast	ARP	60	196	ARP	Who has 192.168.118.191? Tell 192.168.118.1
1.583189	HewlettP_a9:8b:f9	Broadcast	ARP	60	197	ARP	Who has 192.168.118.195? Tell 192.168.118.1
1.583189	HewlettP_a9:8b:f9	Broadcast	ARP	60	198	ARP	Who has 192.168.118.196? Tell 192.168.118.1
1.583291	HewlettP_a9:8b:f9	Broadcast	ARP	60	199	ARP	Who has 192.168.118.199? Tell 192.168.118.1
1.583554	HewlettP_a9:8b:f9	Broadcast	ARP	60	200	ARP	Who has 192.168.118.204? Tell 192.168.118.1
1.583554	HewlettP_a9:8b:f9	Broadcast	ARP	60	201	ARP	Who has 192.168.118.208? Tell 192.168.118.1
1.583832	HewlettP_a9:8b:f9	Broadcast	ARP	60	203	ARP	Who has 192.168.118.209? Tell 192.168.118.1
1.583832	HewlettP_a9:8b:f9	Broadcast	ARP	60	204	ARP	Who has 192.168.118.211? Tell 192.168.118.1
1.584249	HewlettP_a9:8b:f9	Broadcast	ARP	60	205	ARP	Who has 192.168.119.41? Tell 192.168.118.1
1.584249	HewlettP_a9:8b:f9	Broadcast	ARP	60	206	ARP	Who has 192.168.119.238? Tell 192.168.118.1
1.584527	HewlettP_a9:8b:f9	Broadcast	ARP	60	207	ARP	Who has 192.168.119.239? Tell 192.168.118.1
1.584527	HewlettP_a9:8b:f9	Broadcast	ARP	60	208	ARP	Who has 192.168.119.249? Tell 192.168.118.1
1.818432	HewlettP_a9:8b:f9	Broadcast	ARP	60	244	ARP	Who has 192.168.118.137? Tell 192.168.118.1

```
> Destination: Broadcast (ff:ff:ff:ff:ff:ff)
> Source: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
  Type: ARP (0x0806)
```

Type (eth.type), 2 bytes

Packets: 5618 · Displayed: 696 (12.4%) · Dropped: 0 (0.0%)

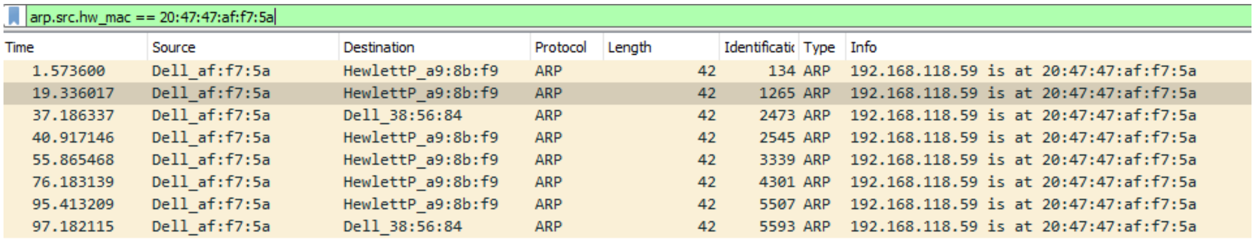


Wireshark · Conversations · Ethernet

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
00:50:b6:a4:52:74	ff:ff:ff:ff:ff:ff	3	258	3	258	0	0	22.744281	60.0143	34	
00:e0:4f:6b:f7:c8	ff:ff:ff:ff:ff:ff	12	1223	12	1223	0	0	83.243525	14.2426	686	
10:27:f5:4d:2c:a4	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.447457	0.0000	—	
14:cc:20:3b:be:97	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	53.654716	0.0000	—	
1c:3b:f3:0e:ee:7d	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.449904	0.0000	—	
22:5f:e3:c8:5c:73	ff:ff:ff:ff:ff:ff	2	120	2	120	0	0	3.175759	61.0348	15	
3c:84:6a:b6:01:fe	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.446491	0.0000	—	
54:af:97:b5:cf:b1	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.448383	0.0000	—	
54:bf:64:4d:fd:0b	ff:ff:ff:ff:ff:ff	3	564	3	564	0	0	28.179766	60.2783	74	
5c:8a:38:a9:8b:f9	ff:ff:ff:ff:ff:ff	578	34 k	578	34 k	0	0	1.572318	93.8526	2956	
8c:16:45:e0:24:23	ff:ff:ff:ff:ff:ff	3	180	3	180	0	0	59.889519	1.5500	929	
98:fa:9b:ea:70:d5	ff:ff:ff:ff:ff:ff	6	1920	6	1920	0	0	28.592378	60.0205	255	
c0:3e:ba:38:56:84	ff:ff:ff:ff:ff:ff	61	3792	61	3792	0	0	4.415282	89.7673	337	
e0:1c:fc:a8:d9:63	ff:ff:ff:ff:ff:ff	16	960	16	960	0	0	0.023557	96.5689	79	
e4:02:9b:b5:16:98	ff:ff:ff:ff:ff:ff	3	180	3	180	0	0	67.821792	1.6487	873	
ec:1a:59:17:6a:9c	ff:ff:ff:ff:ff:ff	1	60	1	60	0	0	83.450915	0.0000	—	
f8:e4:3b:70:1f:79	ff:ff:ff:ff:ff:ff	3	180	3	180	0	0	54.271742	1.8742	768	

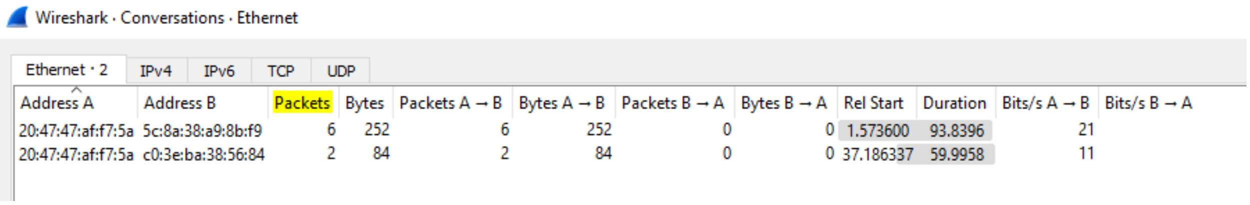
Number of ARP Responses = 8

Apply the filter **arp.src.hw_mac == 20:47:47:af:f7:5a** and in Statistics → Conversations → limit to display filter observe the number of packets captured.



arp.src.hw_mac == 20:47:47:af:f7:5a

Time	Source	Destination	Protocol	Length	Identificati	Type	Info
1.573600	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	134	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
19.336017	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	1265	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
37.186337	Dell_af:f7:5a	Dell_38:56:84	ARP	42	2473	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
40.917146	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	2545	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
55.865468	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	3339	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
76.183139	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	4301	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
95.413209	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	5507	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
97.182115	Dell_af:f7:5a	Dell_38:56:84	ARP	42	5593	ARP	192.168.118.59 is at 20:47:47:af:f7:5a



Wireshark · Conversations · Ethernet

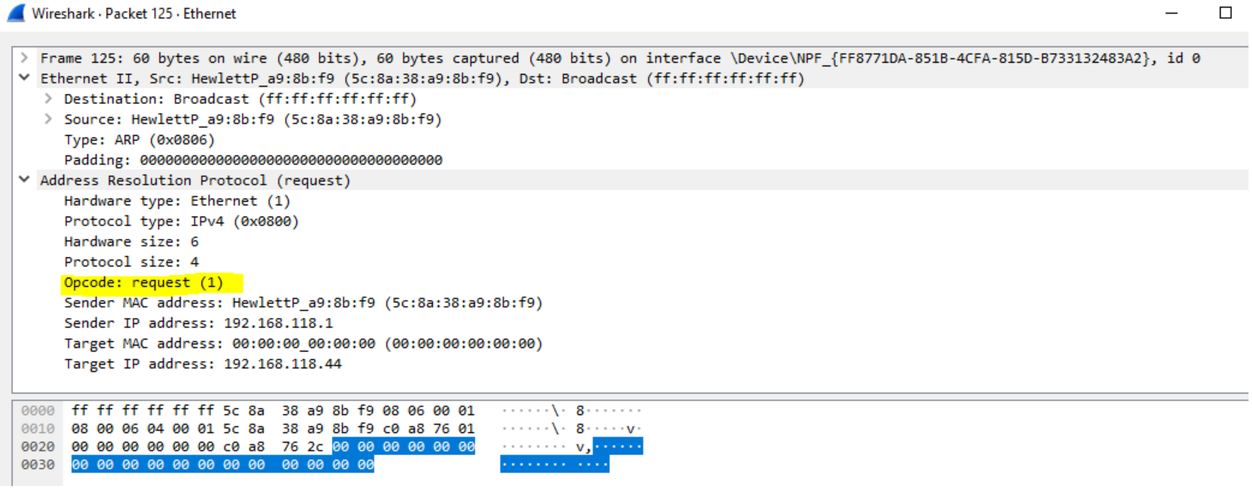
Ethernet · 2 | IPv4 | IPv6 | TCP | UDP

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
20:47:47:af:f7:5a	5c:8a:38:a9:8b:f9	6	252	6	252	0	0	1.573600	93.8396	21	
20:47:47:af:f7:5a	c0:3e:ba:38:56:84	2	84	2	84	0	0	37.186337	59.9958	11	

- An ARP packet is either a request packet or a reply packet.

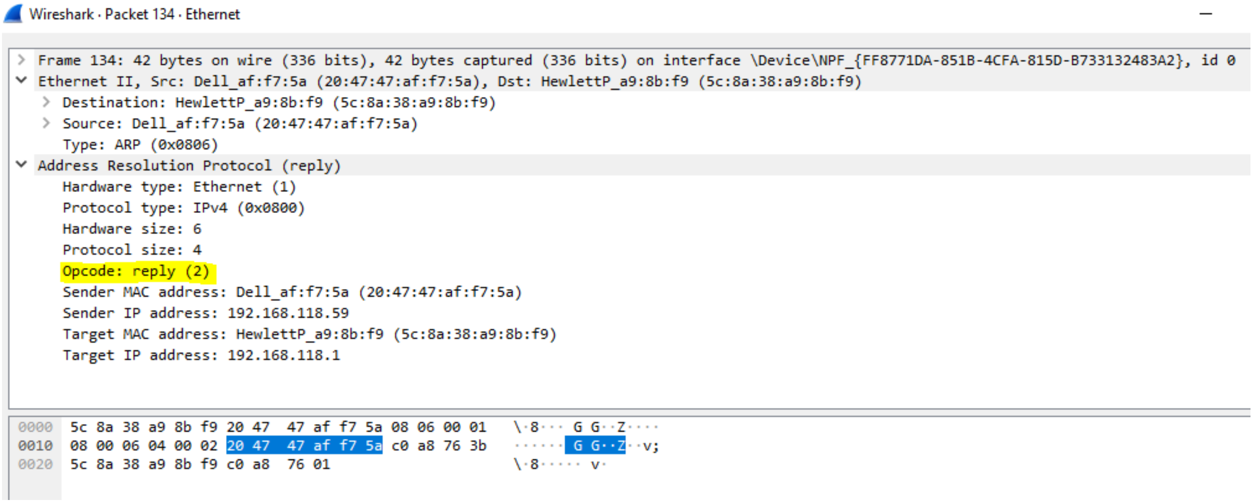
Request packet and Response packet can be differentiated using the Operation field that is **“Opcode”** field in the ARP packet.

If opcode = 1 ⇒ ARP Request
If opcode = 2 ⇒ ARP Reply



Wireshark · Packet 125 · Ethernet

```
> Frame 125: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Sender IP address: 192.168.118.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.118.44

0000  ff ff ff ff ff ff 5c 8a  38 a9 8b f9 08 06 00 01   ······\· 8·······
0010  08 00 06 04 00 01 5c 8a  38 a9 8b f9 c0 a8 76 01   ······\· 8·····v·
0020  00 00 00 00 00 00 c0 a8  76 2c 00 00 00 00 00 00   ········ v,······
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```



Wireshark · Packet 134 · Ethernet

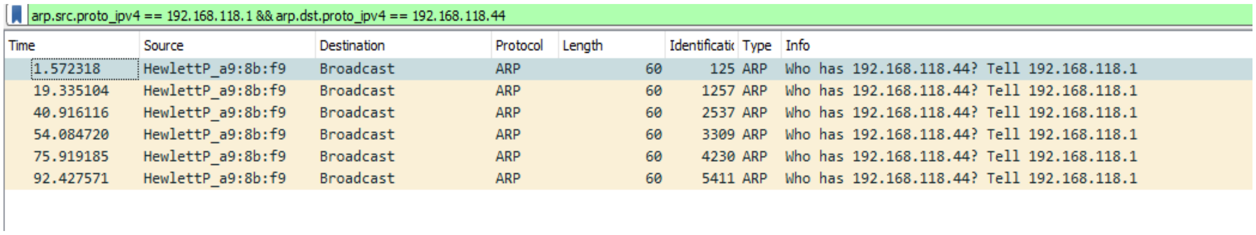
```
> Frame 134: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: Dell_af:f7:5a (20:47:47:af:f7:5a), Dst: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
  > Destination: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
  > Source: Dell_af:f7:5a (20:47:47:af:f7:5a)
    Type: ARP (0x0806)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Dell_af:f7:5a (20:47:47:af:f7:5a)
    Sender IP address: 192.168.118.59
    Target MAC address: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Target IP address: 192.168.118.1

0000  5c 8a 38 a9 8b f9 20 47  47 af f7 5a 08 06 00 01   \·8··· G G··Z····
0010  08 00 06 04 00 02 20 47  47 af f7 5a c0 a8 76 3b   ······ G G··Z··v;
0020  5c 8a 38 a9 8b f9 c0 a8  76 01                     \·8····· v·
```

Table showing 5 entries of ARP:

Source MAC address	Source IP address	Destination MAC address	Destination IP address	Count of ARP Request	Count of ARP Responses
5c:8a:38:a9:8b:f9	192.168.118.1	No Reply	192.168.118.44	6	0
5c:8a:38:a9:8b:f9	192.168.118.1	20:47:47:af:f7:5a	192.168.118.59	6	6
5c:8a:38:a9:8b:f9	192.168.118.1	No Reply	192.168.119.41	6	0
c0:3e:ba:38:56:84	192.168.118.74	No Reply	192.168.118.36	6	0
c0:3e:ba:38:56:84	192.168.118.74	20:47:47:af:f7:5a	192.168.118.59	2	2

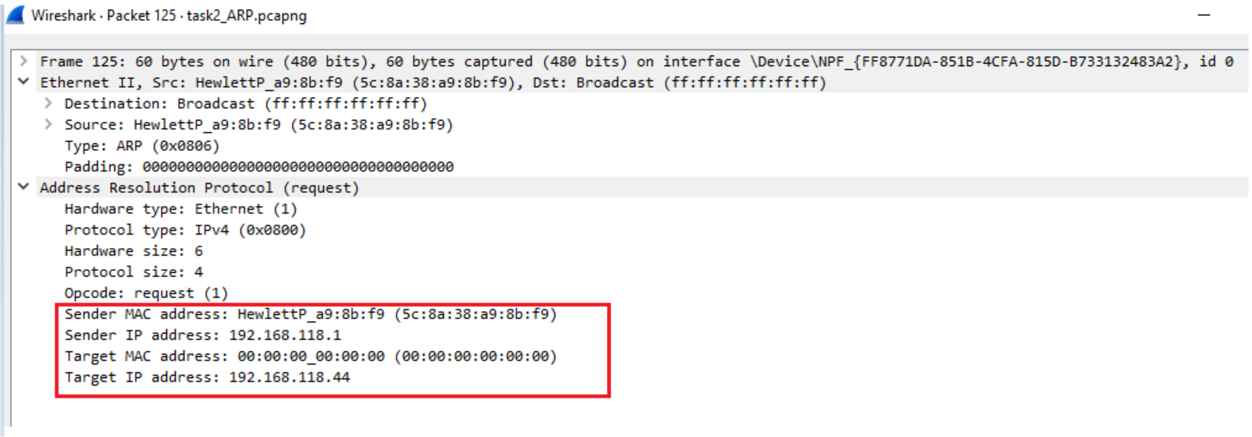
Apply the filter with source and destination IP address to know the count of ARP Requests and Responses.

1st entry:



```
arp.src.proto_ipv4 == 192.168.118.1 && arp.dst.proto_ipv4 == 192.168.118.44
```

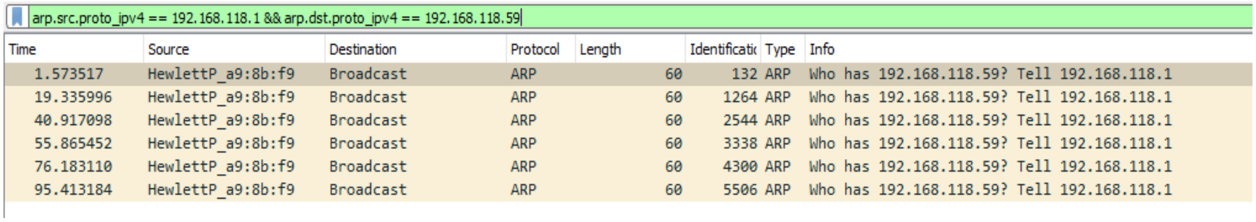
```
Time        Source              Destination     Protocol  Length  Identificati Type  Info
 1.572318   HewlettP_a9:8b:f9   Broadcast       ARP           60      125 ARP   Who has 192.168.118.44? Tell 192.168.118.1
 19.335104  HewlettP_a9:8b:f9   Broadcast       ARP           60     1257 ARP   Who has 192.168.118.44? Tell 192.168.118.1
 40.916116  HewlettP_a9:8b:f9   Broadcast       ARP           60     2537 ARP   Who has 192.168.118.44? Tell 192.168.118.1
 54.084720  HewlettP_a9:8b:f9   Broadcast       ARP           60     3309 ARP   Who has 192.168.118.44? Tell 192.168.118.1
 75.919185  HewlettP_a9:8b:f9   Broadcast       ARP           60     4230 ARP   Who has 192.168.118.44? Tell 192.168.118.1
 92.427571  HewlettP_a9:8b:f9   Broadcast       ARP           60     5411 ARP   Who has 192.168.118.44? Tell 192.168.118.1
```



Wireshark · Packet 125 · task2_ARP.pcapng

```
> Frame 125: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{FF8771DA-851B-4CFA-815D-B733132483A2}, id 0
v Ethernet II, Src: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: HewlettP_a9:8b:f9 (5c:8a:38:a9:8b:f9)
    Sender IP address: 192.168.118.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.118.44
```

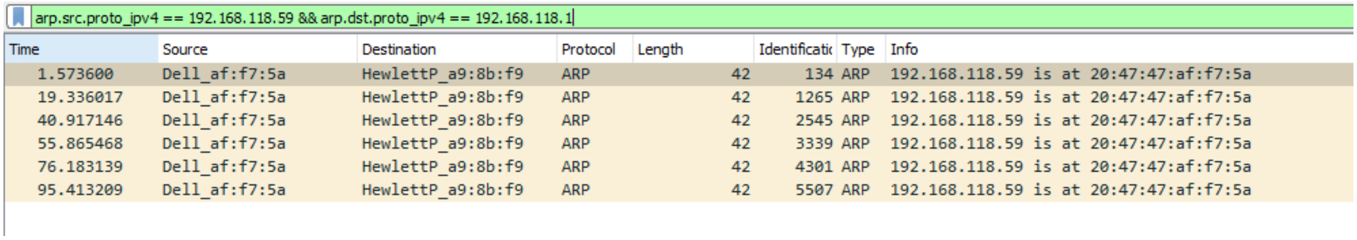
2nd entry: Request



```
arp.src.proto_ipv4 == 192.168.118.1 && arp.dst.proto_ipv4 == 192.168.118.59
```

Time	Source	Destination	Protocol	Length	Identificatic	Type	Info
1.573517	HewlettP_a9:8b:f9	Broadcast	ARP	60	132	ARP	Who has 192.168.118.59? Tell 192.168.118.1
19.335996	HewlettP_a9:8b:f9	Broadcast	ARP	60	1264	ARP	Who has 192.168.118.59? Tell 192.168.118.1
40.917098	HewlettP_a9:8b:f9	Broadcast	ARP	60	2544	ARP	Who has 192.168.118.59? Tell 192.168.118.1
55.865452	HewlettP_a9:8b:f9	Broadcast	ARP	60	3338	ARP	Who has 192.168.118.59? Tell 192.168.118.1
76.183110	HewlettP_a9:8b:f9	Broadcast	ARP	60	4300	ARP	Who has 192.168.118.59? Tell 192.168.118.1
95.413184	HewlettP_a9:8b:f9	Broadcast	ARP	60	5506	ARP	Who has 192.168.118.59? Tell 192.168.118.1

2nd entry: Response



```
arp.src.proto_ipv4 == 192.168.118.59 && arp.dst.proto_ipv4 == 192.168.118.1
```

Time	Source	Destination	Protocol	Length	Identificatic	Type	Info
1.573600	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	134	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
19.336017	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	1265	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
40.917146	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	2545	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
55.865468	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	3339	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
76.183139	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	4301	ARP	192.168.118.59 is at 20:47:47:af:f7:5a
95.413209	Dell_af:f7:5a	HewlettP_a9:8b:f9	ARP	42	5507	ARP	192.168.118.59 is at 20:47:47:af:f7:5a

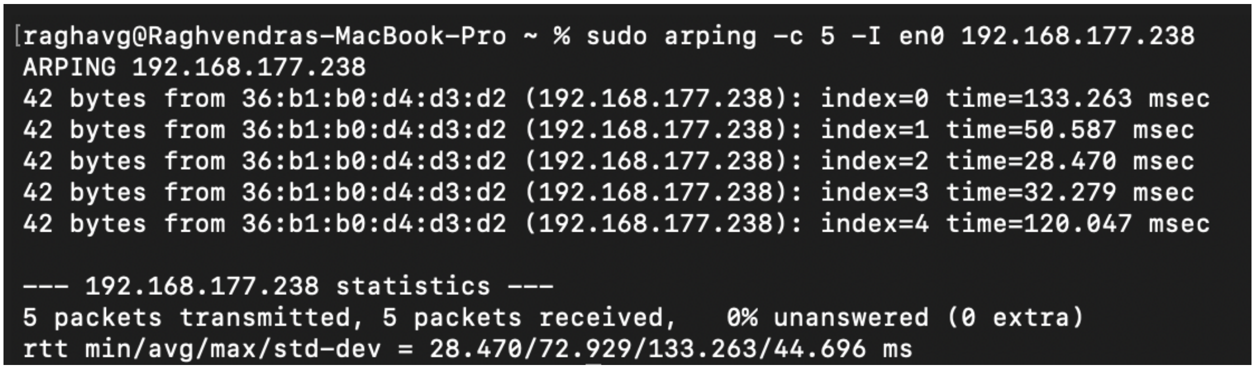
Similarly, do the following for other entries.

3. Start the Wireshark and start capturing. Execute sudo arping -I <network interface name> <ip address>. Please note that the network interface name shall be the one which has active network connection like wlp2s0, eno1, eno2, etc. For the <ip address> field in the command, you may use any IP address within your Default gateway subnet. E.g. If your m/c ip address is 192.168.137.13, you can try arping to 192.168.137.X. If you don't get any response for the tried ip address as above,simply try the default Gateway IP like 192.168.0.1 based on your m/c IP address.
Stop capturing now. Using the captured file, answer the following questions.
Solution:

NOTE: *I was unable to capture packets in Wireshark due to a non-compatible system, hence by seeking Sir's permission, I have captured the packets from my friend's system and analysis is as follows.*

arping is a tool for probing hosts in a network. Unlike the ping command, which operates at the network layer, arping operates at the data link layer and uses the Address Resolution Protocol (ARP). Using it involves sending ARP requests to a destination host and waiting for ARP replies.

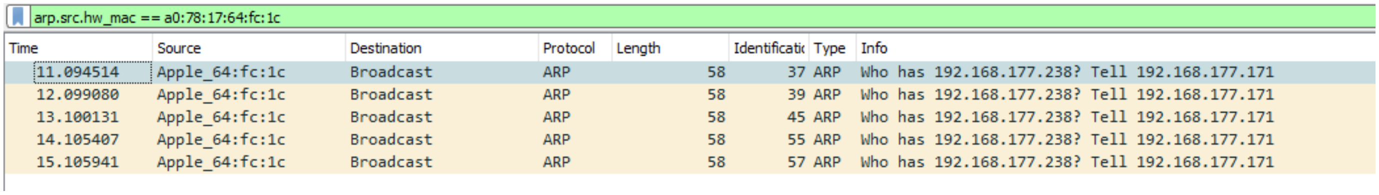
If we only supply the destination to arping, it'll send ARP requests to the destination forever. However, we can pass the desired number of ARP requests with the **-c** option: So, here we have restricted to 5 ARP requests as shown below.



```
[raghavg@Raghvendras-MacBook-Pro ~ % sudo arping -c 5 -I en0 192.168.177.238
 ARPING 192.168.177.238
 42 bytes from 36:b1:b0:d4:d3:d2 (192.168.177.238): index=0 time=133.263 msec
 42 bytes from 36:b1:b0:d4:d3:d2 (192.168.177.238): index=1 time=50.587 msec
 42 bytes from 36:b1:b0:d4:d3:d2 (192.168.177.238): index=2 time=28.470 msec
 42 bytes from 36:b1:b0:d4:d3:d2 (192.168.177.238): index=3 time=32.279 msec
 42 bytes from 36:b1:b0:d4:d3:d2 (192.168.177.238): index=4 time=120.047 msec

 --- 192.168.177.238 statistics ---
 5 packets transmitted, 5 packets received,   0% unanswered (0 extra)
 rtt min/avg/max/std-dev = 28.470/72.929/133.263/44.696 ms
```

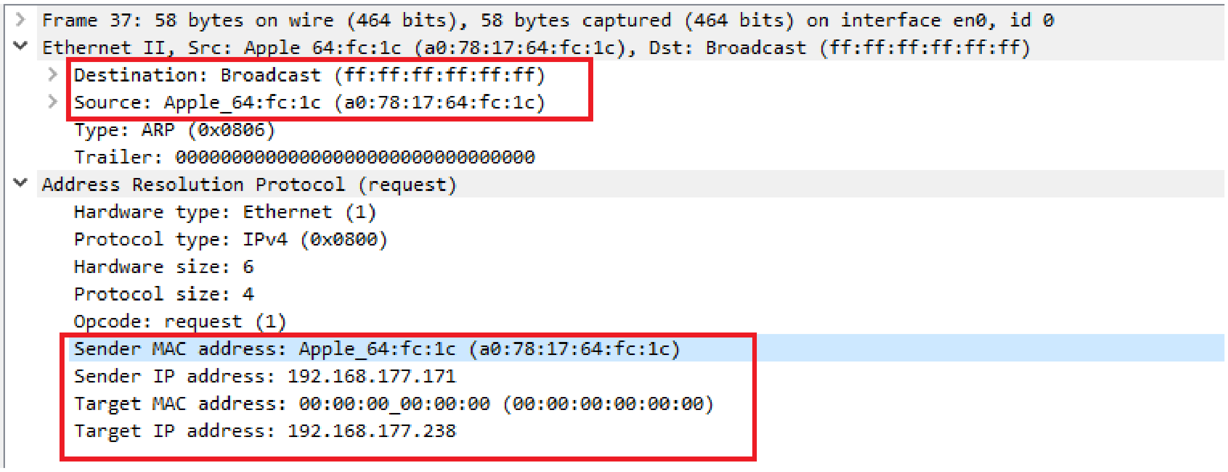
a. In the captured file, use the appropriate filter on Wireshark to show the ARP conversation for the above arping command. Clearly show the screenshot of Wireshark with appropriate filter applied and output of the command from the terminal.
Solution :

ARP Request:



arp.src.hw_mac == a0:78:17:64:fc:1c

Time	Source	Destination	Protocol	Length	Identificati	Type	Info
11.094514	Apple_64:fc:1c	Broadcast	ARP	58	37	ARP	Who has 192.168.177.238? Tell 192.168.177.171
12.099080	Apple_64:fc:1c	Broadcast	ARP	58	39	ARP	Who has 192.168.177.238? Tell 192.168.177.171
13.100131	Apple_64:fc:1c	Broadcast	ARP	58	45	ARP	Who has 192.168.177.238? Tell 192.168.177.171
14.105407	Apple_64:fc:1c	Broadcast	ARP	58	55	ARP	Who has 192.168.177.238? Tell 192.168.177.171
15.105941	Apple_64:fc:1c	Broadcast	ARP	58	57	ARP	Who has 192.168.177.238? Tell 192.168.177.171

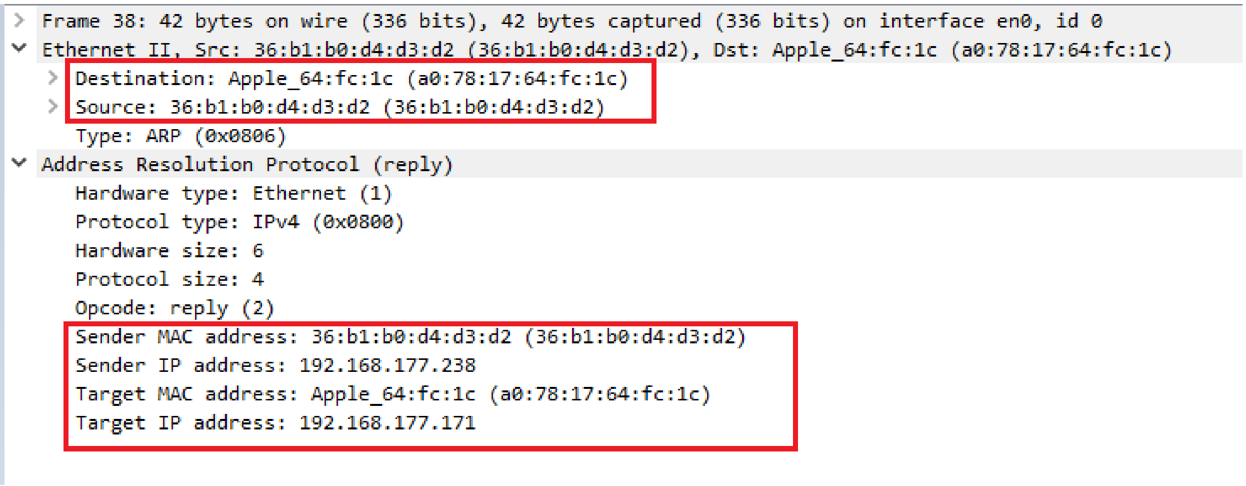


```
> Frame 37: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface en0, id 0
v Ethernet II, Src: Apple_64:fc:1c (a0:78:17:64:fc:1c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000000000
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Sender IP address: 192.168.177.171
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.177.238
```

Apply the filter **arp.src.hw_mac == a0:78:17:64:fc:1c** to filter out ARP packets on the basis of source MAC address.
Destination MAC address is set to **ff:ff:ff:ff:ff:ff** as it is a broadcast packet.

Observing " Address Resolution Protocol " of a packet, we see destination MAC address is set to all zeroes as it is unknown whereas destination IP address is the default gateway address.

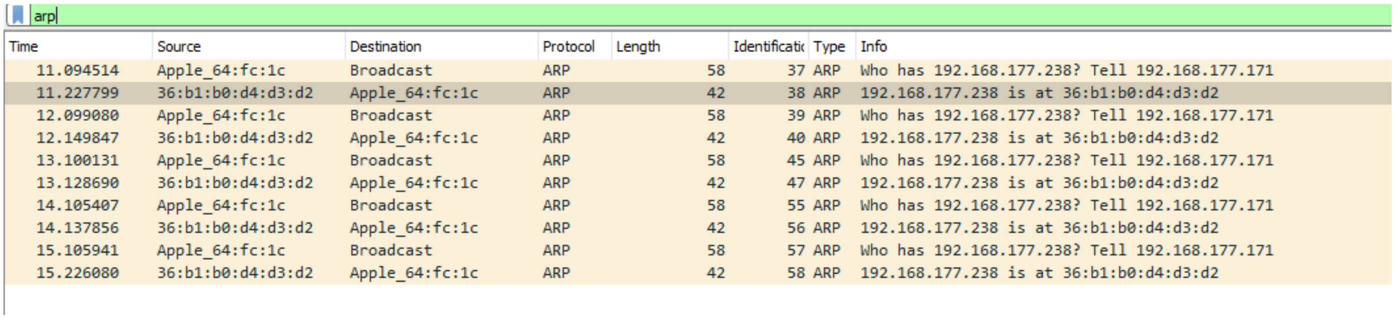
<u>ARP Reply:</u>



```
> Frame 38: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface en0, id 0
v Ethernet II, Src: 36:b1:b0:d4:d3:d2 (36:b1:b0:d4:d3:d2), Dst: Apple_64:fc:1c (a0:78:17:64:fc:1c)
  > Destination: Apple_64:fc:1c (a0:78:17:64:fc:1c)
  > Source: 36:b1:b0:d4:d3:d2 (36:b1:b0:d4:d3:d2)
    Type: ARP (0x0806)
v Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 36:b1:b0:d4:d3:d2 (36:b1:b0:d4:d3:d2)
    Sender IP address: 192.168.177.238
    Target MAC address: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Target IP address: 192.168.177.171
```

Apply the filter **arp.dst.hw_mac == a0:78:17:64:fc:1c && arp.src.proto_ipv4 == 192.168.177.238** to filter ARP Reply packets received from default gateway.

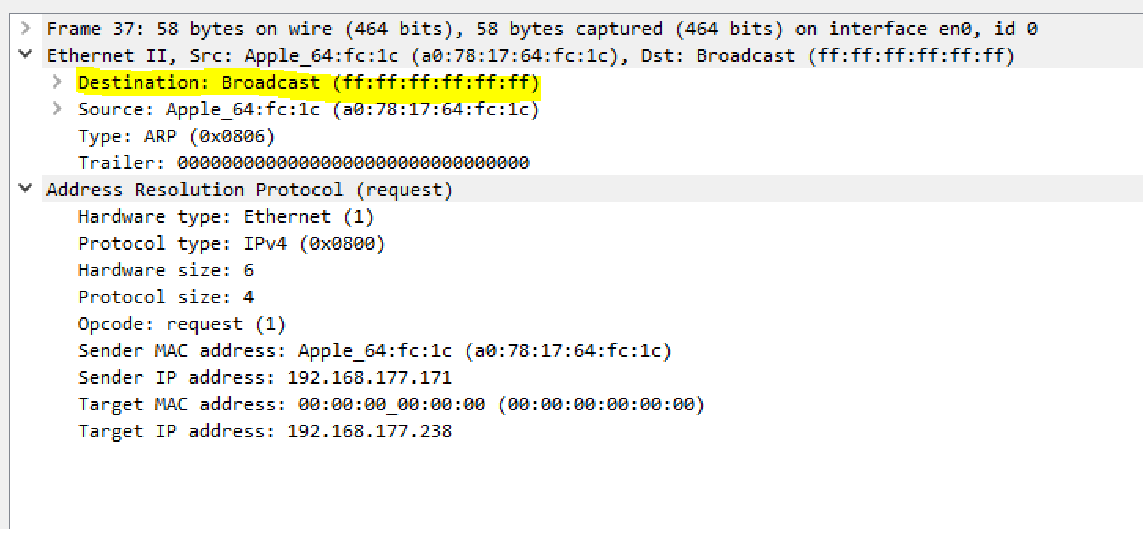
b. Explain what is happening here in this ARP conversation. Is the conversation successful ?. If yes why, if not why ?
Solution:



arp

Time	Source	Destination	Protocol	Length	Identificati	Type	Info
11.094514	Apple_64:fc:1c	Broadcast	ARP	58	37	ARP	Who has 192.168.177.238? Tell 192.168.177.171
11.227799	36:b1:b0:d4:d3:d2	Apple_64:fc:1c	ARP	42	38	ARP	192.168.177.238 is at 36:b1:b0:d4:d3:d2
12.099080	Apple_64:fc:1c	Broadcast	ARP	58	39	ARP	Who has 192.168.177.238? Tell 192.168.177.171
12.149847	36:b1:b0:d4:d3:d2	Apple_64:fc:1c	ARP	42	40	ARP	192.168.177.238 is at 36:b1:b0:d4:d3:d2
13.100131	Apple_64:fc:1c	Broadcast	ARP	58	45	ARP	Who has 192.168.177.238? Tell 192.168.177.171
13.128690	36:b1:b0:d4:d3:d2	Apple_64:fc:1c	ARP	42	47	ARP	192.168.177.238 is at 36:b1:b0:d4:d3:d2
14.105407	Apple_64:fc:1c	Broadcast	ARP	58	55	ARP	Who has 192.168.177.238? Tell 192.168.177.171
14.137856	36:b1:b0:d4:d3:d2	Apple_64:fc:1c	ARP	42	56	ARP	192.168.177.238 is at 36:b1:b0:d4:d3:d2
15.105941	Apple_64:fc:1c	Broadcast	ARP	58	57	ARP	Who has 192.168.177.238? Tell 192.168.177.171
15.226080	36:b1:b0:d4:d3:d2	Apple_64:fc:1c	ARP	42	58	ARP	192.168.177.238 is at 36:b1:b0:d4:d3:d2

Yes, the conversation was successful as we had sent 5 ARP requests by specifying -c 5 and we have received 5 ARP Reply as shown above.
We need to send an ARP request to default gateway, but we dont know it's MAC address hence we broadcast the message as **ff:ff:ff:ff:ff:ff:ff**



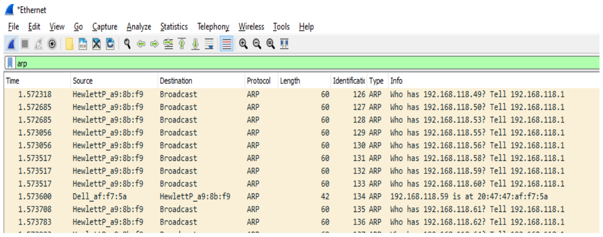
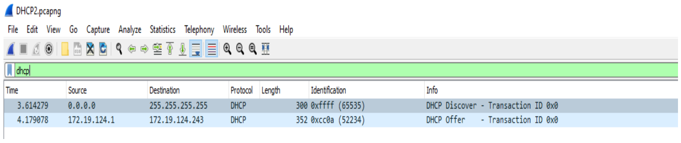
```
> Frame 37: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface en0, id 0
v Ethernet II, Src: Apple_64:fc:1c (a0:78:17:64:fc:1c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Type: ARP (0x0806)
    Trailer: 00000000000000000000000000000000
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Apple_64:fc:1c (a0:78:17:64:fc:1c)
    Sender IP address: 192.168.177.171
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.177.238
```

If there exists an entry for the host, ARP Server will send a unicast reply containing the MAC address of the host.

c. Compare and contrast this ARP conversation with the DHCP conversation in question #1 above.

Solution:

	ARP	DHCP
1)	The Address Resolution Protocol (ARP) is a protocol used by the Internet Protocol (IP), specifically IPv4 to map IP addresses to the hardware addresses(MAC address).	Dynamic Host Configuration Protocol (DHCP)enables a server to automatically assign an IP address to a computer from a defined range of numbers configured for a given network.
2)	Works on Link layer	Works on Application layer
3)	The term address resolution refers to the process of finding an address of a computer in a network. The address is resolved using a protocol in which a piece of information is sent by a client process executing on the local computer to a server process executing on a remote computer. The information received by the server allows the server to uniquely identify the network system for which the address was required and therefore to provide the required address. The address resolution procedure is completed when the client receives a response from the server containing the required address.	The DHCP server may have three methods of allocating IP-addresses: 1.dynamic allocation: A network administrator assigns a range of IP addresses to DHCP, and each client requests an IP address from the DHCP server . The request-and-grant process uses a lease concept with a controllable time period, allowing the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed. 2.automatic allocation: The DHCP server permanently assigns a free IP address to a requesting client from the range defined by the administrator. 3.static allocation: The DHCP server allocates an IP address based on a table with MAC address/IP address pairs, which are manually filled in (perhaps by a network administrator). Only clients with a MAC address listed in this table will be allocated an IP address.
4)	When the source sends **ARP Request**, the network ID of the packet is validated with the destination IP's network ID of the packet and if it's equal then it responds to the source with the MAC address of the destination (**ARP Reply**). Else the packet reaches the gateway of the network and broadcasts packet to the devices it is connected with and validates their network ID.	When a host requests for an IP address, it is assigned from a pool of addresses. In DHCP, the client and the server exchange mainly 4 DHCP messages in order to make a connection, namely **1.DHCP Discover** **2. DHCP Offer** **3. DHCP Request** **4. DHCP ACK**

<table>
<tr><td>5)</td><td>EtherType for ARP is <b>0x0806</b></td><td>DHCP port number for server is <b>67</b> and for the client is <b>68</b></td></tr>
<tr><td>6)</td><td>
*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp
<table>
<tr><th>Time</th><th>Source</th><th>Destination</th><th>Protocol</th><th>Length</th><th>Identificatic</th><th>Type</th><th>Info</th></tr>
<tr><td>1.572318</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>126</td><td>ARP</td><td>Who has 192.168.118.49? Tell 192.168.118.1</td></tr>
<tr><td>1.572685</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>127</td><td>ARP</td><td>Who has 192.168.118.50? Tell 192.168.118.1</td></tr>
<tr><td>1.572685</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>128</td><td>ARP</td><td>Who has 192.168.118.53? Tell 192.168.118.1</td></tr>
<tr><td>1.573056</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>129</td><td>ARP</td><td>Who has 192.168.118.55? Tell 192.168.118.1</td></tr>
<tr><td>1.573056</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>130</td><td>ARP</td><td>Who has 192.168.118.56? Tell 192.168.118.1</td></tr>
<tr><td>1.573517</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>131</td><td>ARP</td><td>Who has 192.168.118.58? Tell 192.168.118.1</td></tr>
<tr><td>1.573517</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>132</td><td>ARP</td><td>Who has 192.168.118.59? Tell 192.168.118.1</td></tr>
<tr><td>1.573517</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>133</td><td>ARP</td><td>Who has 192.168.118.60? Tell 192.168.118.1</td></tr>
<tr><td>1.573600</td><td>Dell_af:f7:5a</td><td>HewlettP_a9:8b:f9</td><td>ARP</td><td>42</td><td>134</td><td>ARP</td><td>192.168.118.59 is at 20:47:47:af:f7:5a</td></tr>
<tr><td>1.573708</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>135</td><td>ARP</td><td>Who has 192.168.118.61? Tell 192.168.118.1</td></tr>
<tr><td>1.573783</td><td>HewlettP_a9:8b:f9</td><td>Broadcast</td><td>ARP</td><td>60</td><td>136</td><td>ARP</td><td>Who has 192.168.118.62? Tell 192.168.118.1</td></tr>
</table></td><td>
DHCP2.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dhcp
<table>
<tr><th>Time</th><th>Source</th><th>Destination</th><th>Protocol</th><th>Length</th><th>Identification</th><th>Info</th></tr>
<tr><td>3.614279</td><td>0.0.0.0</td><td>255.255.255.255</td><td>DHCP</td><td>300</td><td>0xffff (65535)</td><td>DHCP Discover - Transaction ID 0x0</td></tr>
<tr><td>4.179078</td><td>172.19.124.1</td><td>172.19.124.243</td><td>DHCP</td><td>352</td><td>0xcc0a (52234)</td><td>DHCP Offer - Transaction ID 0x0</td></tr>
</table></td></tr>
</table>

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honor violations by other students if I become aware of it.
Name of the student : **SHRUSTI**
Roll No : **CS22MTECH11017**