

Networks Lab Experiment-2

Singam Sai Bala Subrahmanyam
B180522CS

The image shows a Wireshark packet capture interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons for packet capture and analysis. The main window displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packets are filtered by 'Apply a display filter: <Ctrl>'. The first 31 packets are shown, including TCP, TLSv1.2, and DNS traffic. The bottom pane shows the details of the first frame, which is a packet of 66 bytes on the wire (528 bits) captured on interface \Device\NPF{119CCAC4-5D12-48B9-BE88-019F6C28127E}. The packet is an IP packet from 192.168.43.71 to 104.85.102.61, containing a TLSv1.2 client hello.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	104.85.102.61	192.168.43.71	TCP	66	443 → 60572 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1370 SACK_PERM=1 WS=128
2	0.000290	192.168.43.71	104.85.102.61	TCP	54	60572 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=0
3	0.000901	192.168.43.71	104.85.102.61	TLSv1.2	273	Client Hello
4	0.009536	104.85.102.61	192.168.43.71	TCP	66	443 → 60571 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1370 SACK_PERM=1 WS=128
5	0.009612	192.168.43.71	104.85.102.61	TCP	54	60571 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=0
6	0.009892	192.168.43.71	104.85.102.61	TLSv1.2	273	Client Hello
7	0.021337	52.184.80.179	192.168.43.71	TCP	54	443 → 60569 [ACK] Seq=1 Ack=1 Win=2047 Len=0
8	0.021337	52.184.80.179	192.168.43.71	TLSv1.2	92	Application Data
9	0.039016	52.184.80.179	192.168.43.71	TCP	54	443 → 60569 [ACK] Seq=39 Ack=1427 Win=2049 Len=0
10	0.050883	52.184.80.179	192.168.43.71	TCP	54	443 → 60569 [ACK] Seq=39 Ack=2797 Win=2049 Len=0
11	0.051904	104.85.102.61	192.168.43.71	TCP	54	443 → 60572 [ACK] Seq=1 Ack=220 Win=64128 Len=0
12	0.051904	104.85.102.61	192.168.43.71	TLSv1.2	1424	Server Hello
13	0.055525	104.85.102.61	192.168.43.71	TCP	1424	443 → 60572 [PSH, ACK] Seq=1371 Ack=220 Win=64128 Len=1370 [TCP segment of a reassembled PDU]
14	0.055525	104.85.102.61	192.168.43.71	TLSv1.2	1410	Certificate [TCP segment of a reassembled PDU]
15	0.055644	192.168.43.71	104.85.102.61	TCP	54	60572 → 443 [ACK] Seq=220 Ack=4097 Win=256 Len=0
16	0.057839	104.85.102.61	192.168.43.71	TLSv1.2	1424	Certificate Status [TCP segment of a reassembled PDU]
17	0.057900	192.168.43.71	104.85.102.61	TCP	54	60572 → 443 [ACK] Seq=220 Ack=5467 Win=256 Len=0
18	0.062027	104.85.102.61	192.168.43.71	TCP	54	443 → 60571 [ACK] Seq=1 Ack=220 Win=64128 Len=0
19	0.065685	104.85.102.61	192.168.43.71	TLSv1.2	1424	Server Hello
20	0.066009	192.168.43.71	52.184.80.179	TCP	54	60569 → 443 [ACK] Seq=3453 Ack=39 Win=254 Len=0
21	0.066285	192.168.43.71	192.168.43.255	DNS	92	Name query NB CEZPLHYVCJFT<00>
22	0.066357	192.168.43.71	192.168.43.255	DNS	92	Name query NB XHADONXD7XBZMPM<00>
23	0.066418	192.168.43.71	192.168.43.255	DNS	92	Name query NB QCLBULIQNQM<00>
24	0.066901	104.85.102.61	192.168.43.71	TCP	1424	443 → 60571 [PSH, ACK] Seq=1371 Ack=220 Win=64128 Len=1370 [TCP segment of a reassembled PDU]
25	0.066901	104.85.102.61	192.168.43.71	TLSv1.2	1410	Certificate [TCP segment of a reassembled PDU]
26	0.066952	192.168.43.71	104.85.102.61	TCP	54	60571 → 443 [ACK] Seq=220 Ack=4097 Win=256 Len=0
27	0.068137	104.85.102.61	192.168.43.71	TLSv1.2	1424	Certificate Status [TCP segment of a reassembled PDU]
28	0.068137	52.184.80.179	192.168.43.71	TCP	54	443 → 60569 [ACK] Seq=39 Ack=3453 Win=2047 Len=0
29	0.068137	52.184.80.179	192.168.43.71	TLSv1.2	96	Application Data
30	0.068174	192.168.43.71	104.85.102.61	TCP	54	60571 → 443 [ACK] Seq=220 Ack=5467 Win=256 Len=0
31	0.070225	192.168.43.71	224.0.0.251	DNS	82	Standard query 0x0000 PTR googlecast.tcp.local "DM" question

Frame 1: 66 bytes on wire (528 bits) - 66 bytes captured (528 bits) on interface \Device\NPF{119CCAC4-5D12-48B9-BE88-019F6C28127E}. id 0

0000 d8 f2 ca 0c 48 b0 d8 32 e3 4f 5c b0 08 00 45 28 ...H..2..O....E(
0010 00 34 00 00 40 00 34 06 8c 1a 68 55 66 3d c0 a8 ...4..@.4..hlf=..
0020 2b 47 01 bb ec 9c e1 e5 3c 19 6f 73 71 cc 80 12 +G.....<osq..
0030 fa f0 cc 51 00 00 02 04 05 5a 01 01 04 02 01 03 ...Q.....Z.....
0040 03 07

Flags (3 bits) (p.flags), 1 byte

Packets: 32869 · Displayed: 32869 (100.0%) · Dropped: 0 (0.0%)

Profile: Default

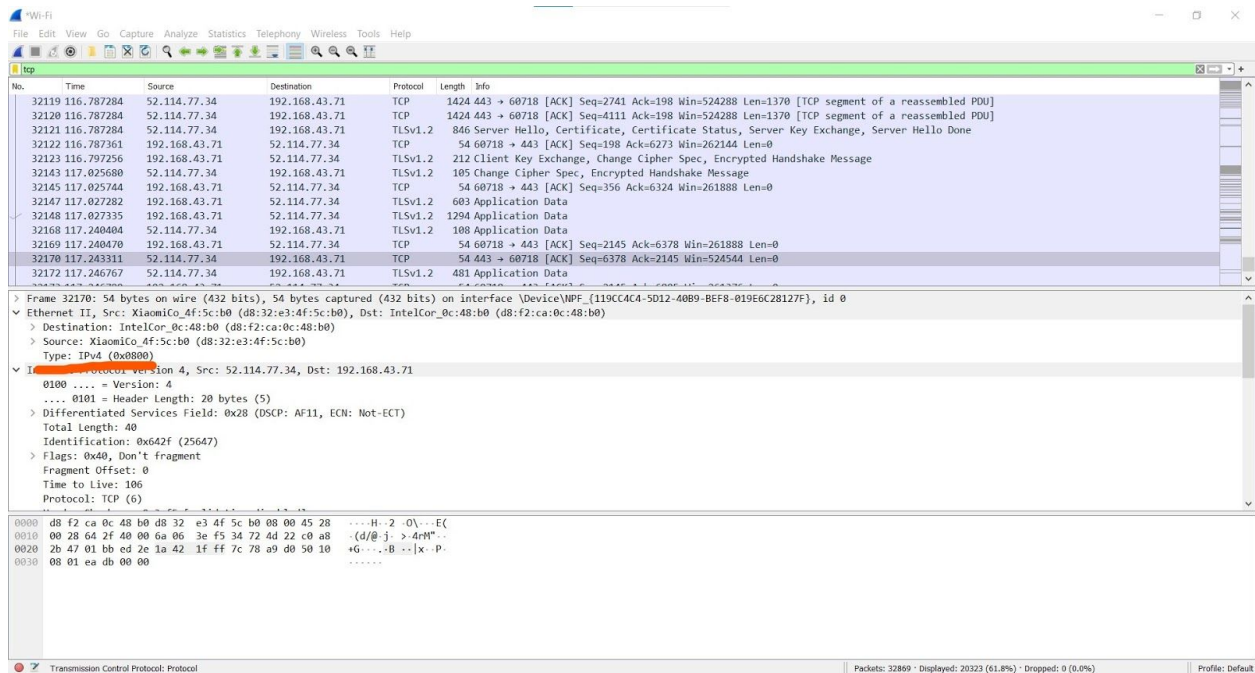
Capturing packets using wireshark for 5 seconds.

- a. For an IP and ARP packet, compare the MAC header of these two packets and find the protocol ID for ARP and IP, if exists.

Each Ethernet frame starts with an Ethernet header or MAC header. The header features destination and source MAC addresses (each six octets in length), the EtherType field. A MAC address is a hardware identification number that uniquely identifies each device on a network. MAC addresses function at the data link layer (layer 2 in the OSI model).

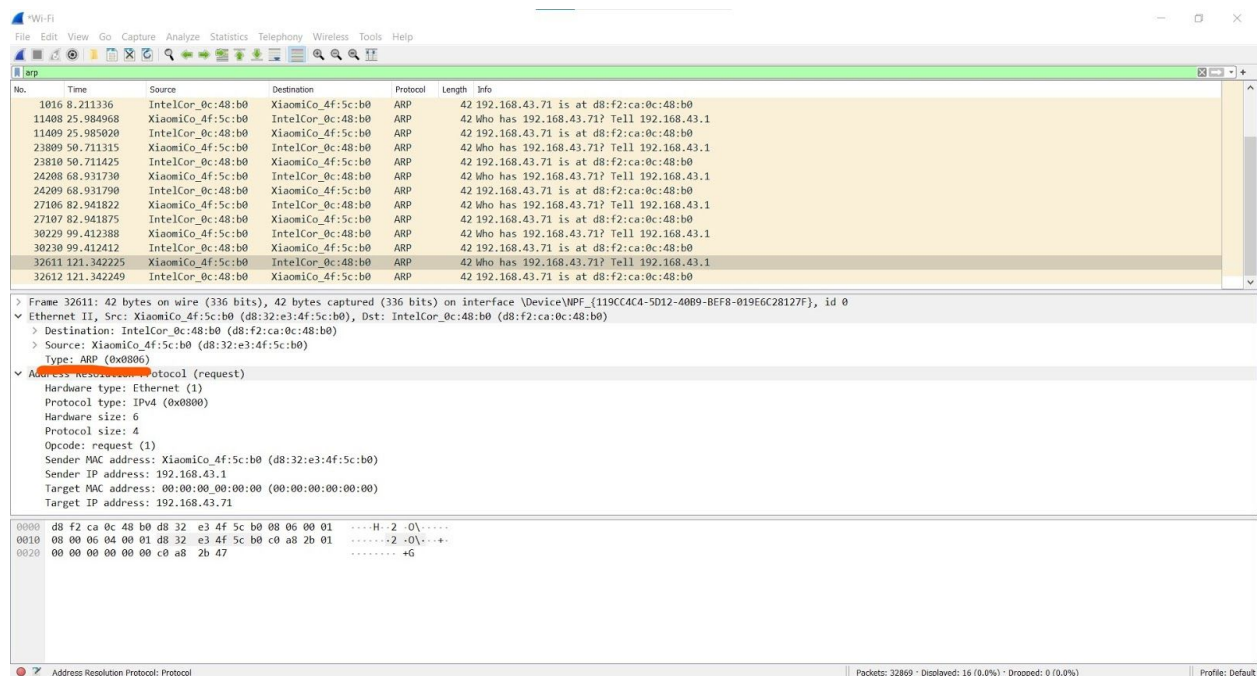
Both IP and ARP packet's MAC header contains

- a. Source MAC address
- b. Destination MAC address
- c. Ether type



IP packet mac header contains:

1. Source MAC address: IntelCor_0c:48:b0 (d8:f2:ca:0c:48:b0)
2. Destination MAC address: XiaomiCo_4f:5c:b0 (d8:32:e3:4f:5c:b0)
3. Type: IPv4 (**Protocol ID: 0x0800**)



ARP packet mac header contains:

4. Source MAC address: IntelCor_0c:48:b0 (d8:f2:ca:0c:48:b0)
1. Destination MAC address: XiaomiCo_4f:5c:b0 (d8:32:e3:4f:5c:b0)
2. Type: ARP (**Protocol ID: 0x0806**)

b. Is the destination address of the ARP packet a broadcast address or a unicast address?

Address Resolution Protocol (ARP) is a procedure for mapping a dynamic Internet Protocol address (IP address) to a permanent physical machine address in a local area network (LAN). The physical machine address is also known as a Media Access Control or MAC address.

The job of the ARP is essentially to translate 32-bit addresses to 48-bit addresses and vice-versa. This is necessary because an IP address is 32-bits long, but MAC addresses are 48-bits long.

ARP works between network layers 2 and 3 of OSI model. The MAC address exists on layer 2 of the OSI model, the data link layer, while the IP address exists on layer 3, the network layer.

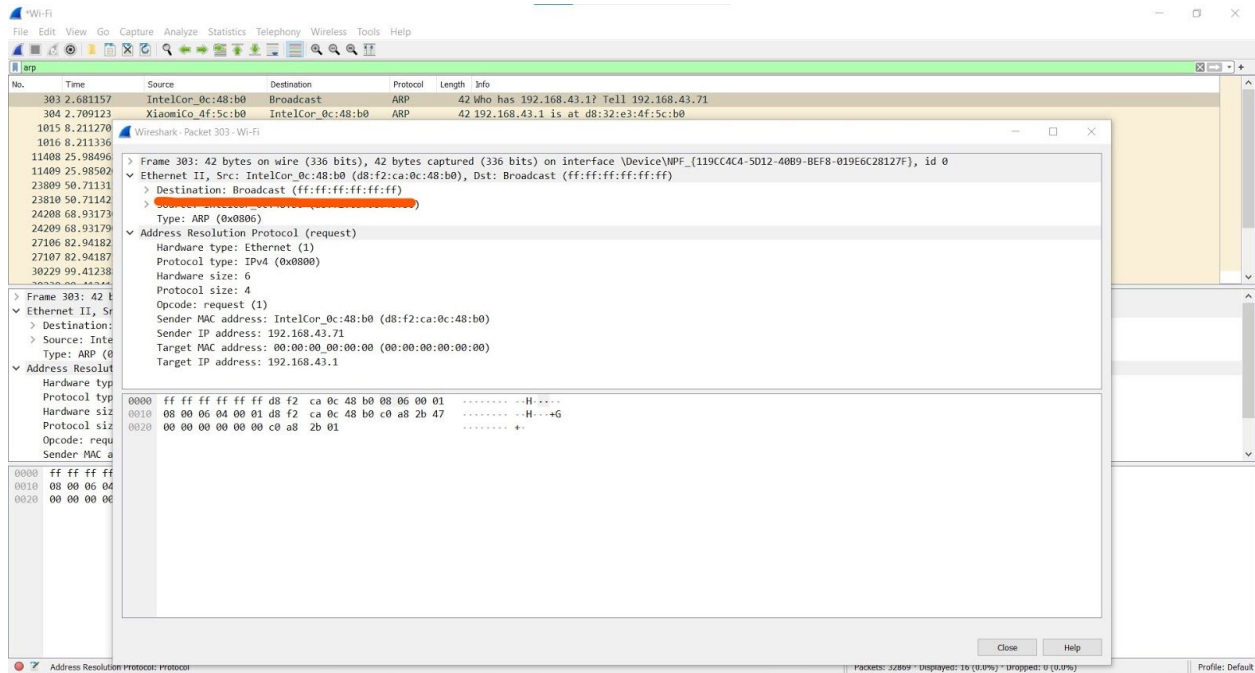
All operating systems in an IPv4 Ethernet network keep an ARP cache. Every time a host requests a MAC address in order to send a packet to another host in the LAN, it checks its ARP cache to see if the IP to MAC address translation already exists. If it does, then a new ARP request is unnecessary. If the translation does not already exist, then the request for network addresses is sent and ARP is performed.

ARP broadcasts a request packet to all the machines on the LAN and asks if any of the machines know they are using that particular IP address. When a machine recognizes the IP address as its own, it sends a reply so ARP can update the cache for future reference and proceed with the communication.

Types of ARP packets:

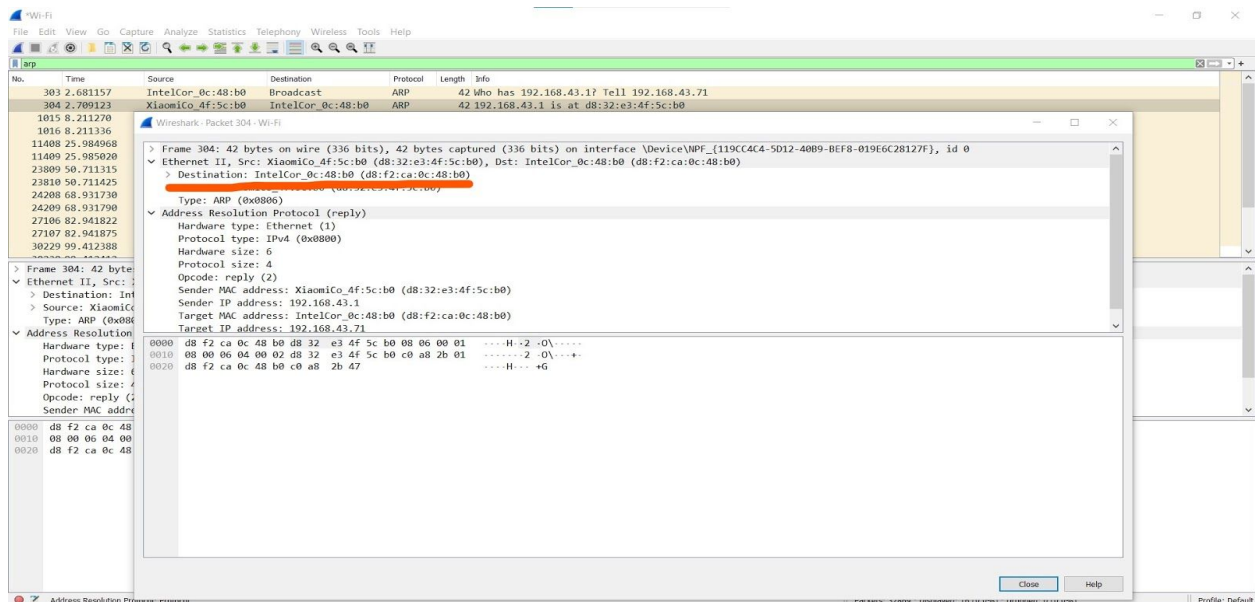
- a. ARP Request packets
- b. ARP Reply packets

a. ARP Request packet



Destination address of ARP Request packet is a broadcast address since the packet is broadcasted to all machines in local area network asking if any of them has the particular IP address.

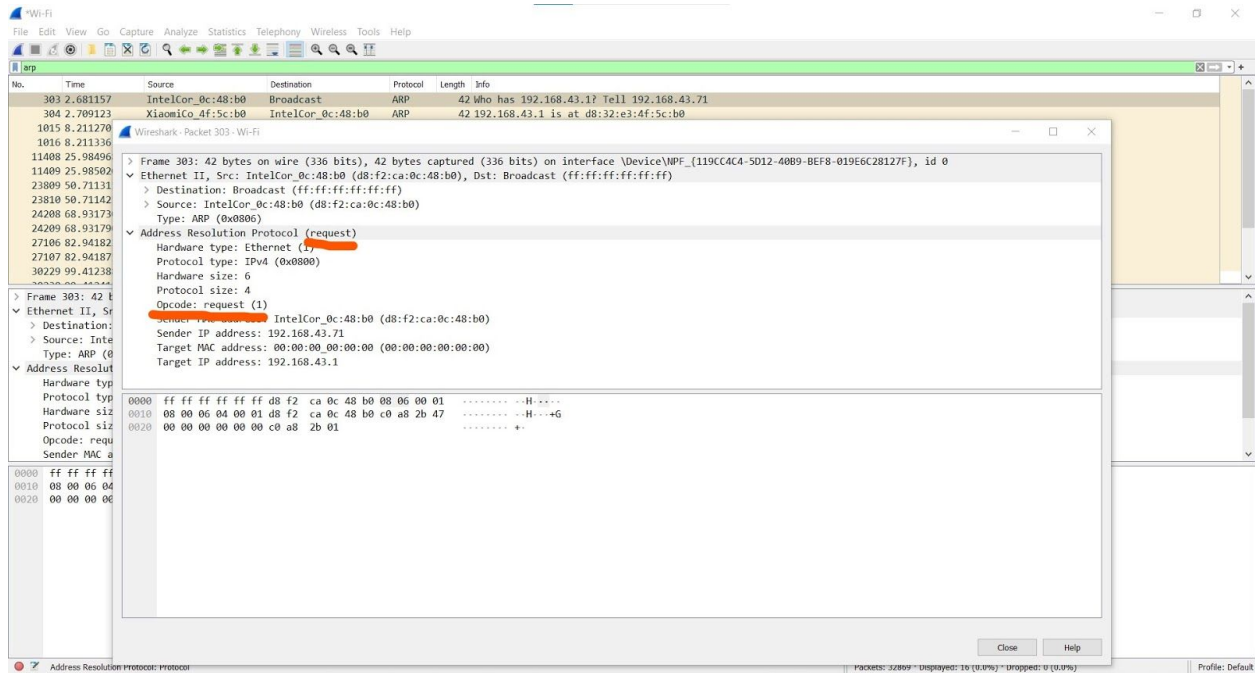
b. ARP Reply packet



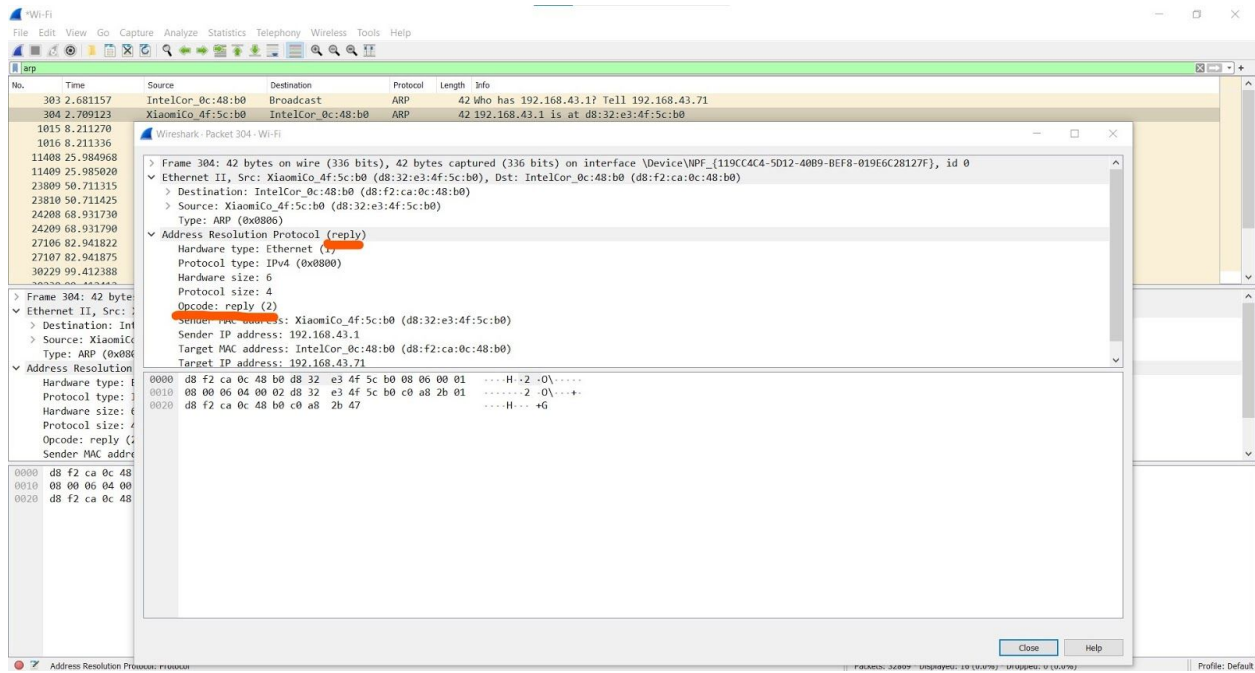
Destination address of ARP Reply packet is a unicast address as the destination will be the broadcasted source of ARP request.

c. Is the ARP packet a request or reply packet? Justify

ARP packets are of two types i.e ARP Request packets and ARP Reply packets. The opcode in the ARP field specifies the type of packet.



ARP Request packet



ARP Reply packet

ARP request packets are broadcasted from the source to all the machines in a local area network in search of a machine with a particular IP address.

Whereas ARP reply packets are sent by the machine with particular IP sending its MAC address to the requested source address.

d. Examine the payload of the packet.

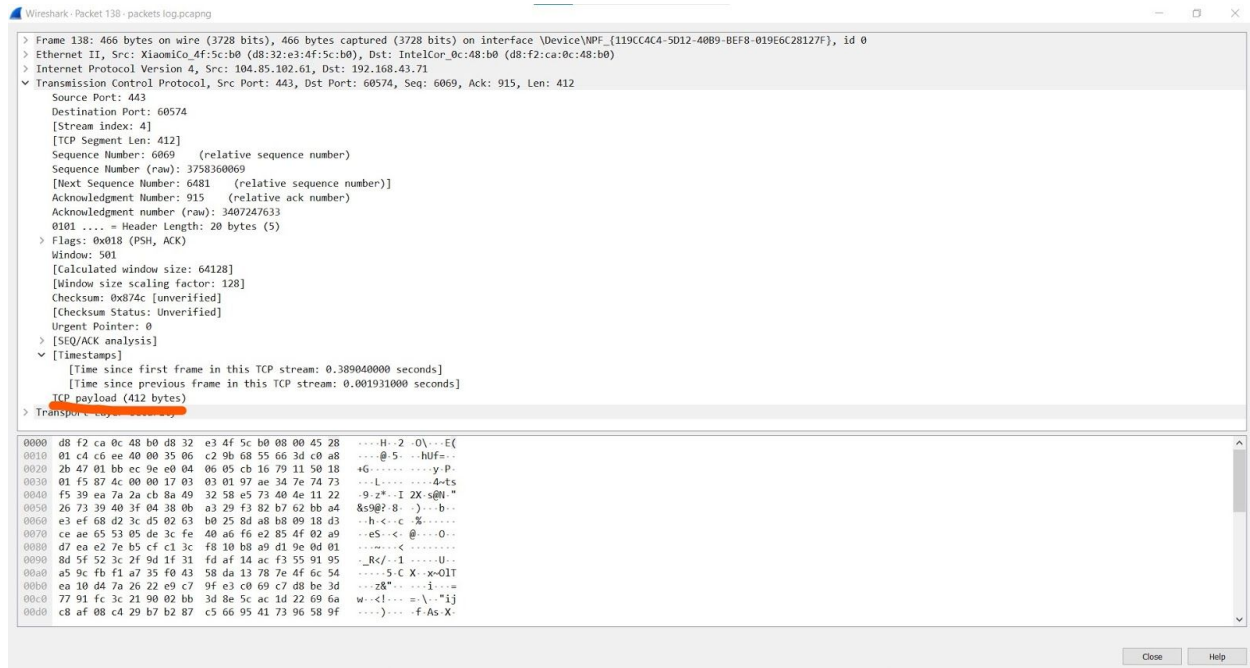
ARP header consists of

- a. Hardware type
- b. Hardware size
- c. Protocol type
- d. Protocol size
- e. Opcode
- f. Sender MAC address
- g. Sender IP address
- h. Target MAC address
- i. Target IP address

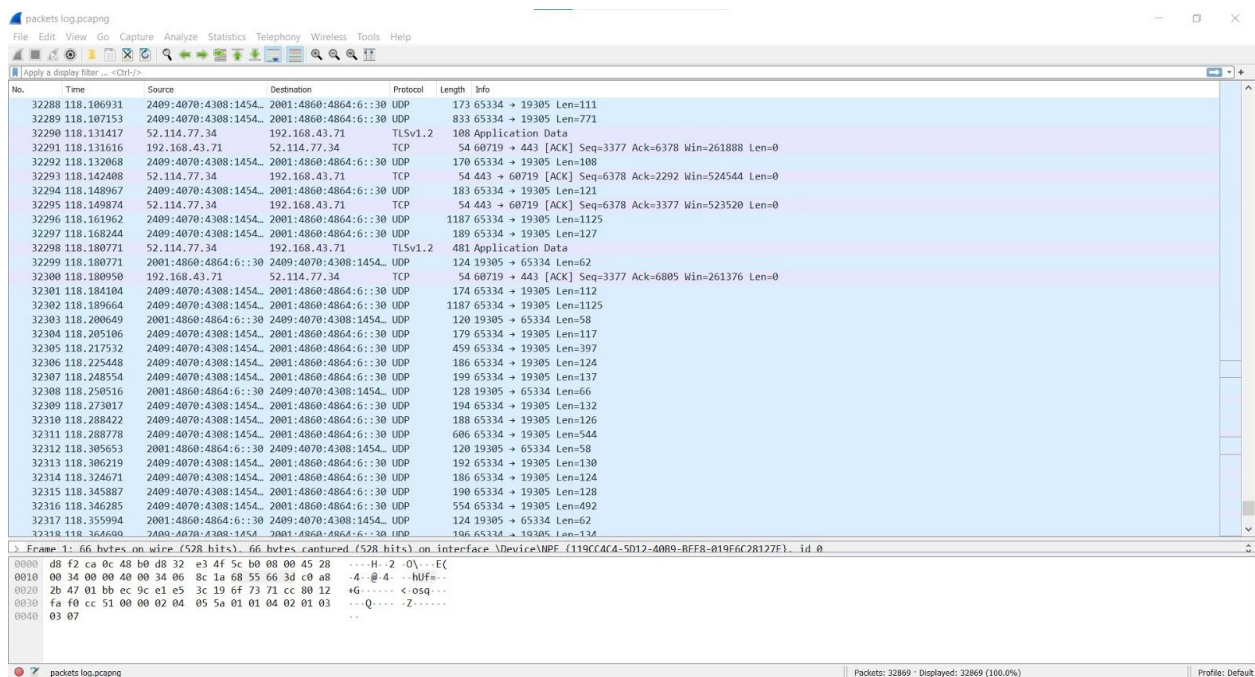
```
type: 1 (Ethernet)  
▼ Address Resolution Protocol (request)  
  Hardware type: Ethernet (1)  
  Protocol type: IPv4 (0x0800)  
  Hardware size: 6  
  Protocol size: 4  
  Opcode: request (1)  
  Sender MAC address: IntelCor_0c:48:b0 (d8:f2:ca:0c:48:b0)  
  Sender IP address: 192.168.43.71  
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)  
  Target IP address: 192.168.43.1
```

The payload of the packet consists of four addresses, the hardware and protocol address of the sender and receiver hosts which are Sender MAC, IP addresses and Target MAC, IP addresses as shown above.

Internet protocol (IP) is network layer protocol. Data segments from the transport layer are divided into packets and this encapsulated segment is IP payload.



e. What transport layer protocols are used in Skype and Zoom.



As seen from the wireshark capture, TCP and UDP packets are present during a video call.

Skype and zoom use both transport layer protocols UDP and TCP.

They send Audio and video over UDP then use TCP to initiate connection or to bypass some firewalls that block UDP packets. TCP is also used for any text communications during video calls.

UDP connections are fast and efficient but do not provide any error checking. So they are preferred for video conferencing and calls.

Whereas TCP connections are more reliable so they are used for text communications etc