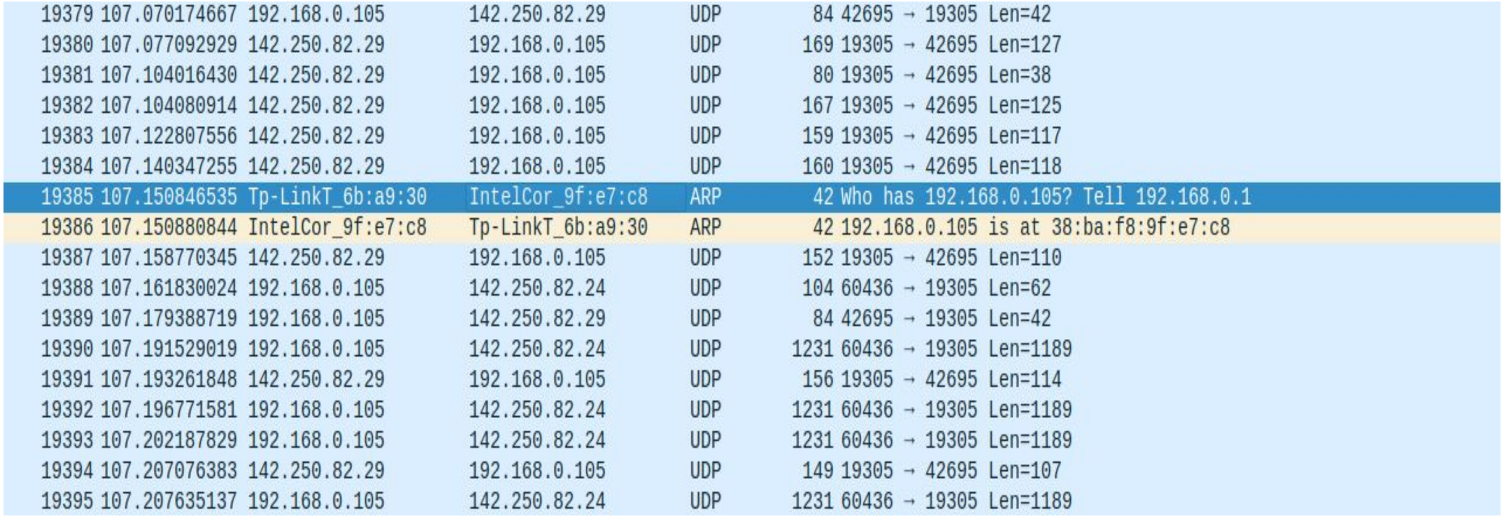
Network Lab: Exp-02

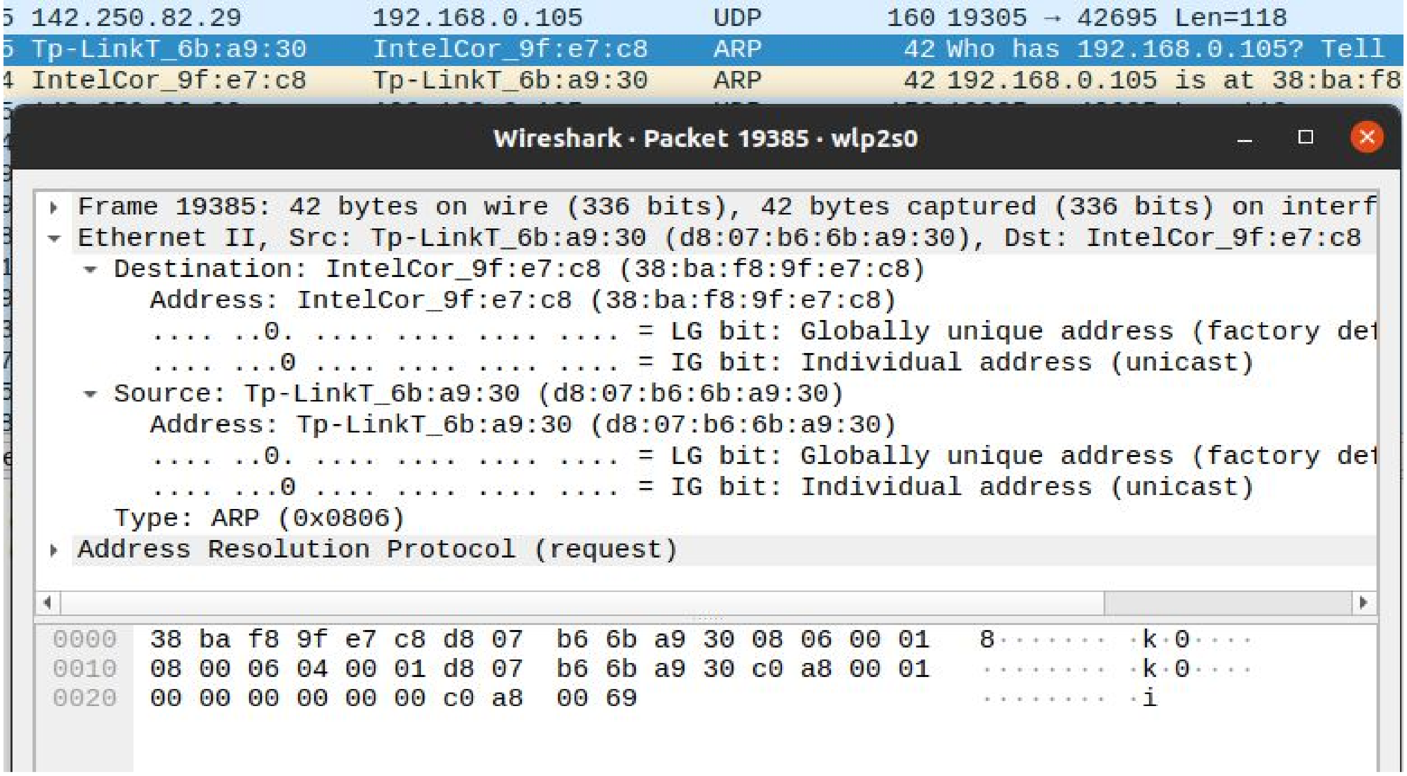
Rahul Kumawat
B180635CS

1. In this experiment, I used *Wireshark* to capture and analyze *ARP* and *IP* packets -



```
19379 107.070174667 192.168.0.105      142.250.82.29     UDP      84 42695 → 19305 Len=42
19380 107.077092929 142.250.82.29      192.168.0.105     UDP     169 19305 → 42695 Len=127
19381 107.104016430 142.250.82.29      192.168.0.105     UDP      80 19305 → 42695 Len=38
19382 107.104080914 142.250.82.29      192.168.0.105     UDP     167 19305 → 42695 Len=125
19383 107.122807556 142.250.82.29      192.168.0.105     UDP     159 19305 → 42695 Len=117
19384 107.140347255 142.250.82.29      192.168.0.105     UDP     160 19305 → 42695 Len=118
19385 107.150846535 Tp-LinkT_6b:a9:30  IntelCor_9f:e7:c8 ARP      42 Who has 192.168.0.105? Tell 192.168.0.1
19386 107.150880844 IntelCor_9f:e7:c8  Tp-LinkT_6b:a9:30 ARP      42 192.168.0.105 is at 38:ba:f8:9f:e7:c8
19387 107.158770345 142.250.82.29      192.168.0.105     UDP     152 19305 → 42695 Len=110
19388 107.161830024 192.168.0.105      142.250.82.24     UDP     104 60436 → 19305 Len=62
19389 107.179388719 192.168.0.105      142.250.82.29     UDP      84 42695 → 19305 Len=42
19390 107.191529019 192.168.0.105      142.250.82.24     UDP    1231 60436 → 19305 Len=1189
19391 107.193261848 142.250.82.29      192.168.0.105     UDP     156 19305 → 42695 Len=114
19392 107.196771581 192.168.0.105      142.250.82.24     UDP    1231 60436 → 19305 Len=1189
19393 107.202187829 192.168.0.105      142.250.82.24     UDP    1231 60436 → 19305 Len=1189
19394 107.207076383 142.250.82.29      192.168.0.105     UDP     149 19305 → 42695 Len=107
19395 107.207635137 192.168.0.105      142.250.82.24     UDP    1231 60436 → 19305 Len=1189
```

a. *ARP* ⇒ *MAC header* of ARP packet contains information about MAC addresses of source and destination of this packet and Protocol ID is *ARP (0x0806)*

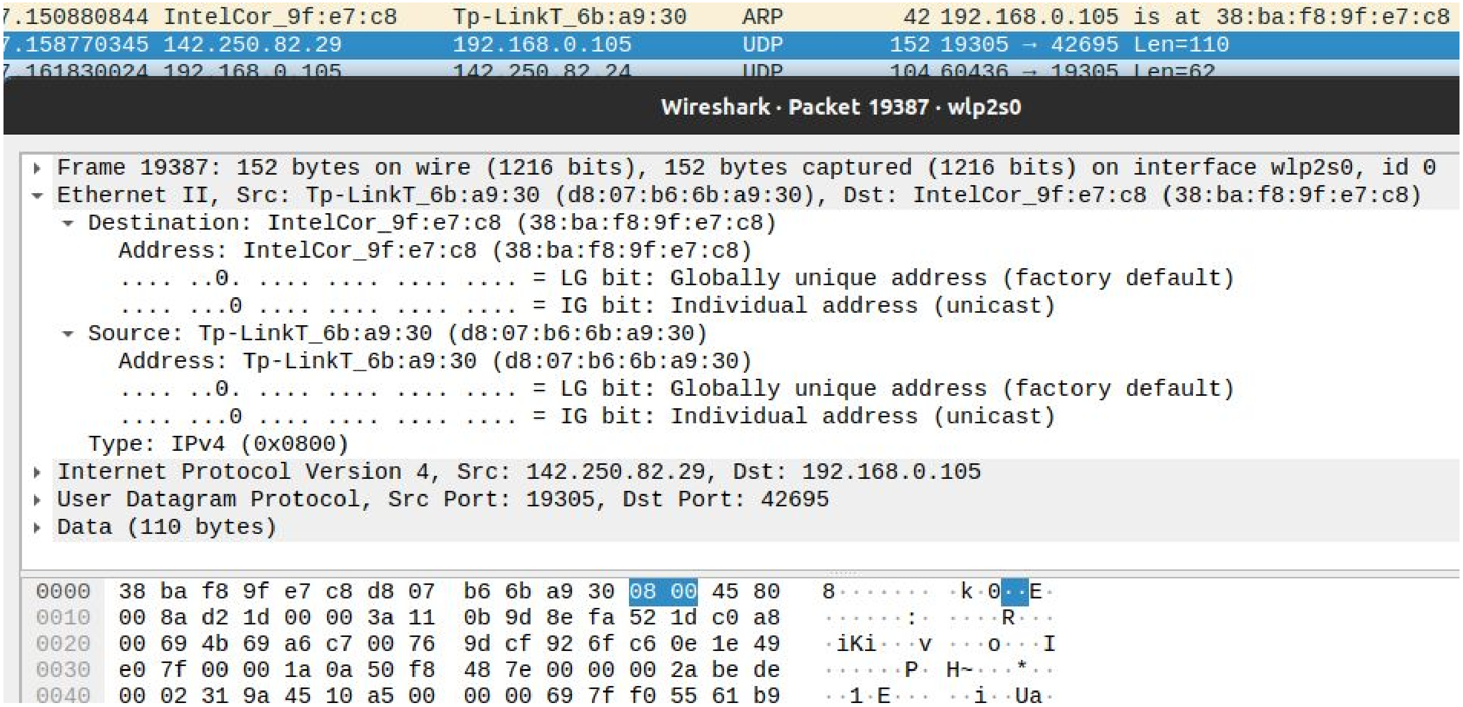


```
5 142.250.82.29       192.168.0.105      UDP      160 19305 → 42695 Len=118
5 Tp-LinkT_6b:a9:30   IntelCor_9f:e7:c8  ARP       42 Who has 192.168.0.105? Tell
4 IntelCor_9f:e7:c8   Tp-LinkT_6b:a9:30  ARP       42 192.168.0.105 is at 38:ba:f8
```

```
▸ Frame 19385: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interf
▾ Ethernet II, Src: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30), Dst: IntelCor_9f:e7:c8
  ▾ Destination: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
      Address: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory def
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
      Address: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory def
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
▸ Address Resolution Protocol (request)

0000  38 ba f8 9f e7 c8 d8 07  b6 6b a9 30 08 06 00 01   8······· ·k·0····
0010  08 00 06 04 00 01 d8 07  b6 6b a9 30 c0 a8 00 01   ········ ·k·0····
0020  00 00 00 00 00 00 c0 a8  00 69                     ········ ·i
```

IP ⇒ *MAC header* of IP packet contains information about MAC addresses of source and destination of this packet and Protocol ID is *IPv4 (0x0800)*



```
7.150880844 IntelCor_9f:e7:c8     Tp-LinkT_6b:a9:30     ARP        42 192.168.0.105 is at 38:ba:f8:9f:e7:c8
7.158770345 142.250.82.29         192.168.0.105         UDP       152 19305 → 42695 Len=110
7.161830024 192.168.0.105         142.250.82.24         UDP       104 60436 → 19305 Len=62
```

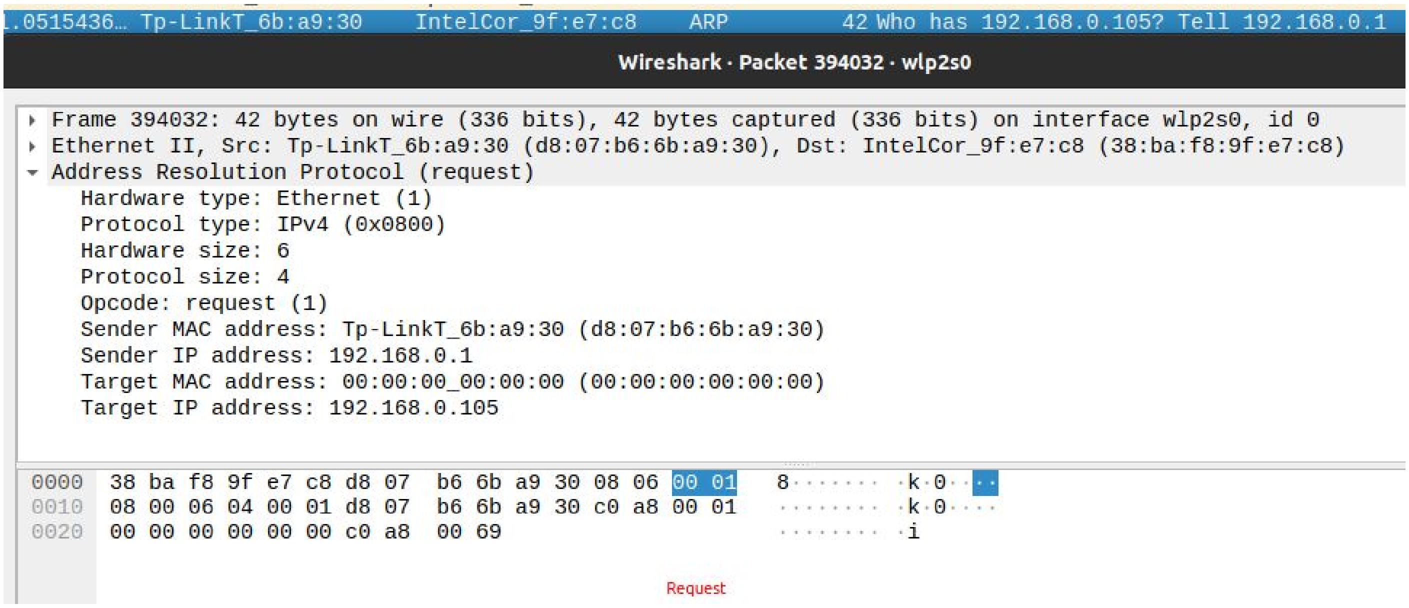
Wireshark · Packet 19387 · wlp2s0

```
▸ Frame 19387: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits) on interface wlp2s0, id 0
▾ Ethernet II, Src: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30), Dst: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
  ▾ Destination: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
      Address: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
      Address: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 142.250.82.29, Dst: 192.168.0.105
▸ User Datagram Protocol, Src Port: 19305, Dst Port: 42695
▸ Data (110 bytes)
```

```
0000  38 ba f8 9f e7 c8 d8 07  b6 6b a9 30 08 00 45 80   8······· ·k·0··E·
0010  00 8a d2 1d 00 00 3a 11  0b 9d 8e fa 52 1d c0 a8   ······:· ····R···
0020  00 69 4b 69 a6 c7 00 76  9d cf 92 6f c6 0e 1e 49   ·iKi···v ···o···I
0030  e0 7f 00 00 1a 0a 50 f8  48 7e 00 00 00 2a be de   ······P· H~···*··
0040  00 02 31 9a 45 10 a5 00  00 00 69 7f f0 55 61 b9   ··1·E··· ··i··Ua·
```

Fig - An IP packet

b. The destination address of the *ARP* packets broadcast for request and unicast for a response.



```
1.0515436… Tp-LinkT_6b:a9:30     IntelCor_9f:e7:c8     ARP        42 Who has 192.168.0.105? Tell 192.168.0.1
```

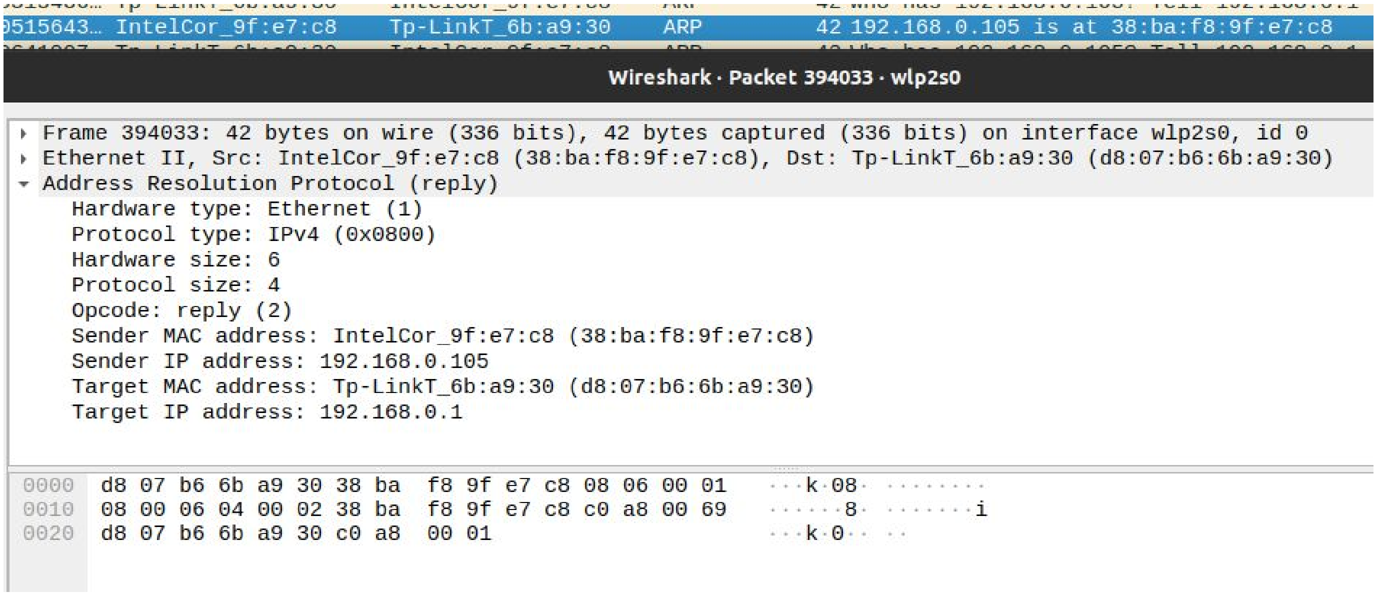
Wireshark · Packet 394032 · wlp2s0

```
▸ Frame 394032: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp2s0, id 0
▸ Ethernet II, Src: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30), Dst: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
    Sender IP address: 192.168.0.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.0.105
```

```
0000  38 ba f8 9f e7 c8 d8 07  b6 6b a9 30 08 06 00 01   8······· ·k·0····
0010  08 00 06 04 00 01 d8 07  b6 6b a9 30 c0 a8 00 01   ········ ·k·0····
0020  00 00 00 00 00 00 c0 a8  00 69                     ········ ·i
```

Request

Fig - Request



```
9515643… IntelCor_9f:e7:c8    Tp-LinkT_6b:a9:30    ARP      42 192.168.0.105 is at 38:ba:f8:9f:e7:c8
```

```
▸ Frame 394033: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface wlp2s0, id 0
▸ Ethernet II, Src: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8), Dst: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: IntelCor_9f:e7:c8 (38:ba:f8:9f:e7:c8)
    Sender IP address: 192.168.0.105
    Target MAC address: Tp-LinkT_6b:a9:30 (d8:07:b6:6b:a9:30)
    Target IP address: 192.168.0.1

0000  d8 07 b6 6b a9 30 38 ba  f8 9f e7 c8 08 06 00 01   ···k·08· ········
0010  08 00 06 04 00 02 38 ba  f8 9f e7 c8 c0 a8 00 69   ······8· ·······i
0020  d8 07 b6 6b a9 30 c0 a8  00 01                     ···k·0·· ··
```

Fig - Response

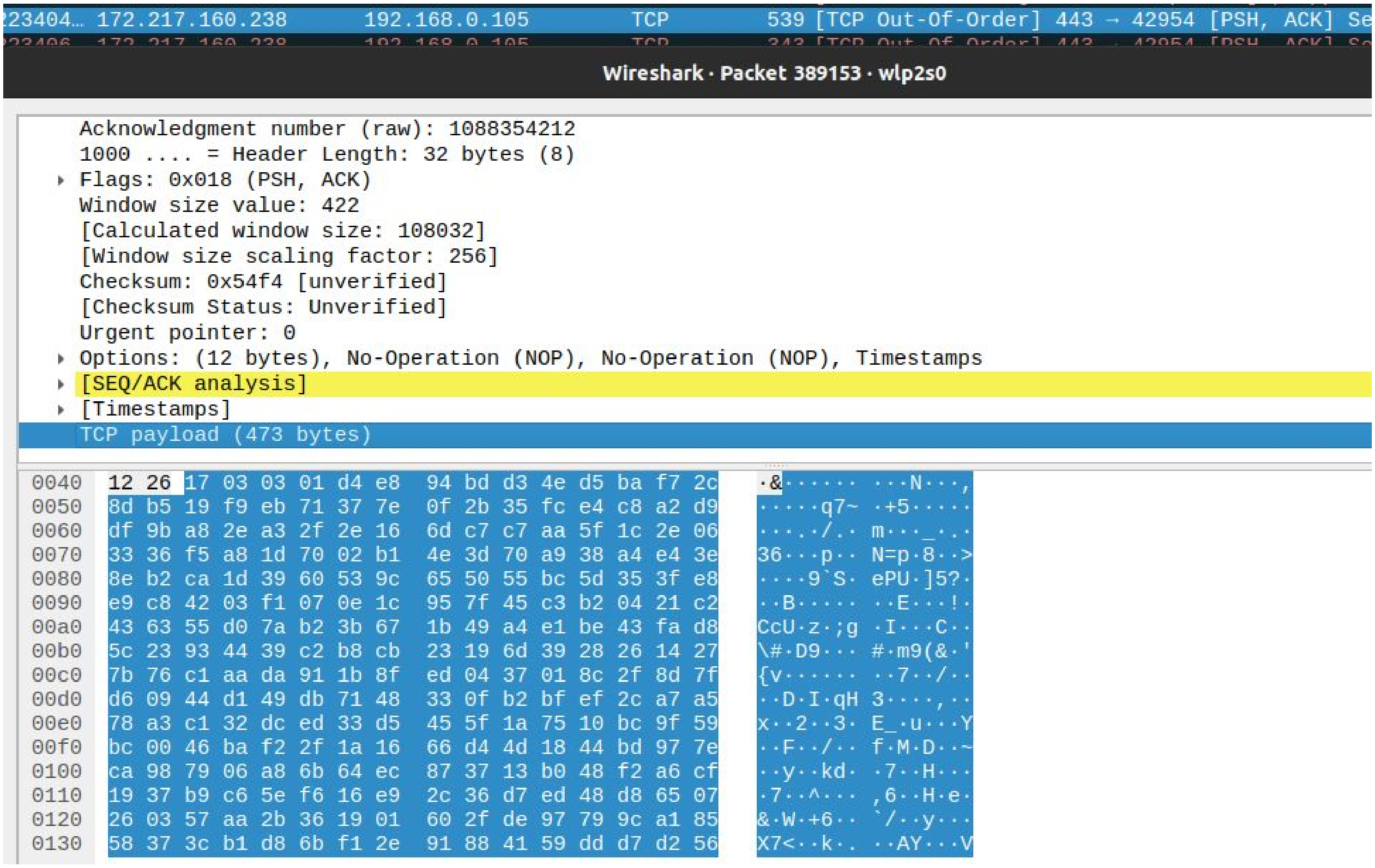
c. An ARP packet is both request and reply ⇒

i. When the sender wants to send a packet to the receiver it is needed to know the receiver’s MAC address. For this sender broadcast a ARP packet requesting that “if anyone has this dest address response back with your MAC address”. So it is a request sent by sender.

ii. When a machine got such a request and find the dest ip same as its ip address it response back to sender with an ARP packet.

Note:- *Refer to part **b** images*

d. Payload of IP packet ⇒



```
23404… 172.217.160.238    192.168.0.105    TCP    539 [TCP Out-Of-Order] 443 → 42954 [PSH, ACK] Se
```

```
    Acknowledgment number (raw): 1088354212
    1000 .... = Header Length: 32 bytes (8)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 422
    [Calculated window size: 108032]
    [Window size scaling factor: 256]
    Checksum: 0x54f4 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (473 bytes)
```

```
0040  12 26 17 03 03 01 d4 e8  94 bd d3 4e d5 ba f7 2c   ·&······ ···N···,
0050  8d b5 19 f9 eb 71 37 7e  0f 2b 35 fc e4 c8 a2 d9   ·····q7~ ·+5·····
0060  df 9b a8 2e a3 2f 2e 16  6d c7 c7 aa 5f 1c 2e 06   ·····/.· m···_·.·
0070  33 36 f5 a8 1d 70 02 b1  4e 3d 70 a9 38 a4 e4 3e   36···p·· N=p·8··>
0080  8e b2 ca 1d 39 60 53 9c  65 50 55 bc 5d 35 3f e8   ····9`S· ePU·]5?·
0090  e9 c8 42 03 f1 07 0e 1c  95 7f 45 c3 b2 04 21 c2   ··B····· ··E···!·
00a0  43 63 55 d0 7a b2 3b 67  1b 49 a4 e1 be 43 fa d8   CcU·z·;g ·I···C··
00b0  5c 23 93 44 39 c2 b8 cb  23 19 6d 39 28 26 14 27   \#·D9··· #·m9(&·'
00c0  7b 76 c1 aa da 91 1b 8f  ed 04 37 01 8c 2f 8d 7f   {v······ ··7··/··
00d0  d6 09 44 d1 49 db 71 48  33 0f b2 bf ef 2c a7 a5   ··D·I·qH 3····,··
00e0  78 a3 c1 32 dc ed 33 d5  45 5f 1a 75 10 bc 9f 59   x··2··3· E_·u···Y
00f0  bc 00 46 ba f2 2f 1a 16  66 d4 4d 18 44 bd 97 7e   ··F··/·· f·M·D··~
0100  ca 98 79 06 a8 6b 64 ec  87 37 13 b0 48 f2 a6 cf   ··y··kd· ·7··H···
0110  19 37 b9 c6 5e f6 16 e9  2c 36 d7 ed 48 d8 65 07   ·7··^··· ,6··H·e·
0120  26 03 57 aa 2b 36 19 01  60 2f de 97 79 9c a1 85   &·W·+6·· `/··y···
0130  58 37 3c b1 d8 6b f1 2e  91 88 41 59 dd d7 d2 56   X7<··k·. ··AY···V
```

An IP packet has *IP header + payload* in it. So in the above image payload data is highlighted which is sent through this IP packet.

Payload of ARP packet ⇒
The payload of the packet consists of four addresses, the hardware and protocol address of the sender and receiver hosts.
Note - Refer to part B images

e. transport layer protocols used in Skype and Zoom ⇒

i. UDP ⇒ UDP protocol is used for video streaming because here speed of data transfer matters more than end-to-end communication.
ii. TCP ⇒ the chat section where the entire message is important they use TCP protocol which promises end-to-end complete data transfer.