
EXPERIMENT - 2

NETWORKS LAB

Submitted by
Mohammed Ismail C
B180437CS
B Batch

Wireshark captured packets

No.	Time	Source	Destination	Protocol	Length	Info
1401	135.964286	e6:c4:83:bb:6d:65	IntelCor_e2:6a:df	ARP	42	192.168.43.1 is at e6:c4:83:bb:6d:65
1402	136.026910	192.168.43.44	142.250.76.206	UDP	75	55874 → 443 Len=33
1403	136.151907	142.250.76.206	192.168.43.44	UDP	68	443 → 55874 Len=26
1404	136.352471	192.168.43.44	142.250.76.206	UDP	75	55874 → 443 Len=33
1405	136.477147	142.250.76.206	192.168.43.44	UDP	68	443 → 55874 Len=26
1406	136.679272	IntelCor_e2:6a:df	Broadcast	ARP	42	Who has 192.168.43.1? Tell 192.168.43.44
1407	136.682209	e6:c4:83:bb:6d:65	IntelCor_e2:6a:df	ARP	42	192.168.43.1 is at e6:c4:83:bb:6d:65
1408	136.682228	192.168.43.44	142.250.76.206	UDP	75	55874 → 443 Len=33
1409	136.746597	192.168.43.44	142.250.77.46	QUIC	1392	Initial, DCID=a9f14bd32c9988d3
1410	136.807031	142.250.76.206	192.168.43.44	UDP	68	443 → 55874 Len=26
1411	136.880903	142.250.77.46	192.168.43.44	QUIC	1392	Initial, SCID=a9f14bd32c9988d3
1412	136.916955	142.250.77.46	192.168.43.44	QUIC	1392	Initial, SCID=a9f14bd32c9988d3
1413	136.916955	142.250.77.46	192.168.43.44	QUIC	100	Protected Payload (KP0)
1414	136.916955	142.250.77.46	192.168.43.44	QUIC	274	Handshake, SCID=a9f14bd32c9988d3
1415	136.918010	192.168.43.44	142.250.77.46	QUIC	190	Protected Payload (KP0), DCID=a9f14bd32c9988d3
1416	136.918358	192.168.43.44	142.250.77.46	QUIC	1388	Protected Payload (KP0), DCID=a9f14bd32c9988d3
1417	136.918401	192.168.43.44	142.250.77.46	QUIC	1388	Protected Payload (KP0), DCID=a9f14bd32c9988d3
1418	136.918434	192.168.43.44	142.250.77.46	QUIC	335	Protected Payload (KP0), DCID=a9f14bd32c9988d3
1419	137.005459	142.250.77.46	192.168.43.44	QUIC	118	Protected Payload (KP0)
1420	137.005657	142.250.77.46	192.168.43.44	QUIC	654	Protected Payload (KP0)
1421	137.005771	192.168.43.44	142.250.77.46	QUIC	75	Protected Payload (KP0), DCID=a9f14bd32c9988d3
1422	137.005908	192.168.43.44	142.250.77.46	QUIC	75	Protected Payload (KP0), DCID=a9f14bd32c9988d3
1423	137.007602	192.168.43.44	142.250.76.206	UDP	75	55874 → 443 Len=33
1424	137.022622	142.250.77.46	192.168.43.44	QUIC	67	Protected Payload (KP0)

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.

```
> Frame 1393: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7F898685-9ABC-4CF5-91B8-C29863E5B4B7}, id 0
▼ Ethernet II, Src: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Dst: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  > Destination: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  > Source: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65)
    Type: ARP (0x0806)
  > Address Resolution Protocol (reply)
```

Fig 1 : Showing MAC header of an ARP packet

```
> Frame 1363: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{7F898685-9ABC-4CF5-91B8-C29863E5B4B7}, id 0
▼ Ethernet II, Src: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Dst: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  > Destination: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  > Source: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65)
    Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 204.79.197.222, Dst: 192.168.43.44
  > Transmission Control Protocol, Src Port: 443, Dst Port: 54449, Seq: 7033, Ack: 1450, Len: 0
```

Fig 2 : Showing MAC header of an TCP/IP packet

For both ARP Packets and TCP/IP packets, There will be a Destination, Source and Type fields. Destination and source fields are the corresponding MAC address(48 bits) of the devices. The only difference noticed is that for ARP packet type field(or protocol ID) value is 0806 and for TCP/IP, it is 0800(both values are in hexadecimal).

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

The destination address of an ARP packet can be either unicast or broadcast address. If it is an ARP reply packet then the destination address will be unicast.

```
> Frame 1407: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7F898685-9A8C-4A8C-8A8C-8A8C}
▼ Ethernet II, Src: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Dst: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  ▼ Destination: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
    Address: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
  > Source: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65)
    Type: ARP (0x0806)
  > Address Resolution Protocol (reply)
```

ARP reply packet

For ARP request packet, the destination address can be either unicast or broadcast based on the situation as shown in fig 4 and fig 5.

```
> Frame 1406: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7F898685-9A8C-4A8C-8A8C-8A8C}
▼ Ethernet II, Src: IntelCor_e2:6a:df (04:ea:56:e2:6a:df), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▼ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ..1. .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ..1. .... = IG bit: Group address (multicast/broadcast)
  > Source: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
    Type: ARP (0x0806)
  > Address Resolution Protocol (request)
```

Fig 4: ARP request packet broadcasted

```
> Frame 1488: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7F898685-9A8C-4A8C-8A8C-8A8C}
▼ Ethernet II, Src: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Dst: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  ▼ Destination: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
    Address: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
      .... ..0. .... = LG bit: Globally unique address (factory default)
      .... ..0. .... = IG bit: Individual address (unicast)
  > Source: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65)
    Type: ARP (0x0806)
  > Address Resolution Protocol (request)
```

Fig 5: ARP request packet (unicasted)

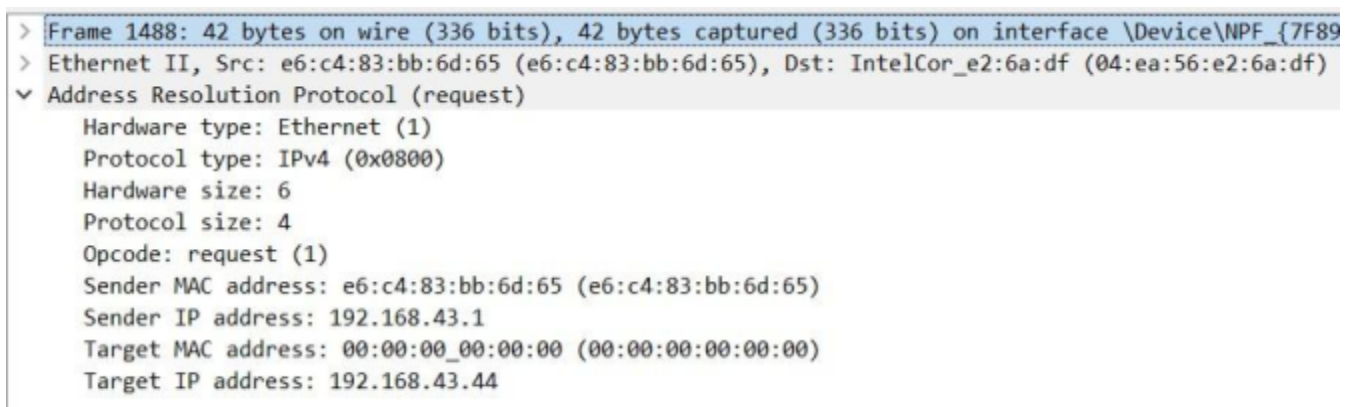
In Fig 4, destination address of ARP request packet is broadcasted and if a device with matching ip address got the packet, then that device will respond with an ARP reply packet. From the source MAC address of the reply packet, we can get the MAC address that we needed earlier.

In Fig 5, It is showing that ARP request packet can also be unicasted. There is an arp cache present in the system. In an attempt to refresh an expired, or expiring, ARP entry, many Client OS's will issue a targeted ARP query to the MAC address they already expect. Most of the time, this prompts a response from the intended target and allows the entry to be refreshed without sending a broadcast to the entire network.

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

ARP packet can be either a request or a reply packet as shown in fig 3,4 and 5. ARP protocol is used to get the MAC address of the target device in a network. If we need to send a packet to a device with particular IP address in a network we also need to know the MAC address of that device. If we have't that, then an ARP request packet is broadcasted to all devices in the network. Then the ip address matching device will respond to that request packet with an ARP reply packet from which we can know the MAC address of that device and initiate further communications.

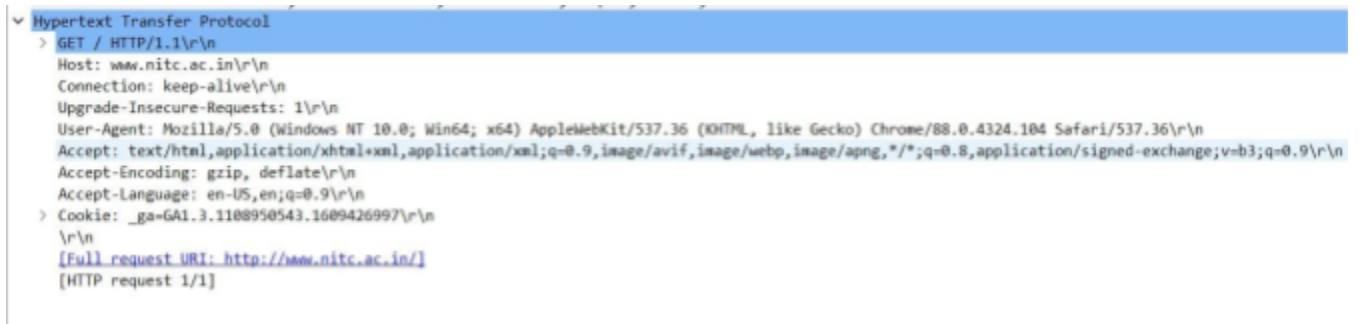
d) Examine the payload of the packet

A screenshot of a Wireshark packet capture window. The top pane shows 'Frame 1488: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7F89...}'. The middle pane shows 'Ethernet II, Src: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Dst: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)'. The bottom pane shows the expanded 'Address Resolution Protocol (request)' with the following details: Hardware type: Ethernet (1), Protocol type: IPv4 (0x0800), Hardware size: 6, Protocol size: 4, Opcode: request (1), Sender MAC address: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Sender IP address: 192.168.43.1, Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00), and Target IP address: 192.168.43.44.

```
> Frame 1488: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_{7F89...}
> Ethernet II, Src: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65), Dst: IntelCor_e2:6a:df (04:ea:56:e2:6a:df)
  Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: e6:c4:83:bb:6d:65 (e6:c4:83:bb:6d:65)
    Sender IP address: 192.168.43.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.43.44
```

Fig 6: Showing payload of an ARP packet

The Payload of an ARP packet shows many informations like Hardware type, protocol type etc. For the above packet the Hardware type value is 1 which implies ethernet. Protocol type gives the protocol ID. In the above figure it is an IPv4 so it's value is 0800(in hexadecimal). Hardware size is the MAC address size which is 6 bytes or 48 bits. Protocol size is IP address size which is 4 bytes or 32 bits for IPv4. opcode bit value tells us whether the packet is a request (1) or reply (2). Then Sender MAC address, sender IP address, Target MAC address and Target IP address are shown. MAC address are represented in Hexadecimal and IP address in decimal as octet. For request packet Target MAC address is not known, so it's value will be filled with zero. For ARP reply packet, Target MAC address will be known.



```

Hypertext Transfer Protocol
> GET / HTTP/1.1\r\n
Host: www.nitc.ac.in\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.104 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
> Cookie: _ga=GA1.3.1188950543.1689426997\r\n
\r\n
[Full request URI: http://www.nitc.ac.in/]
[HTTP request 1/1]

```

Payload of an HTTP packet

In an HTTP packet, There will be header for IPv4, TCP and HTTP. The payload for the TCP is the application layer HTTP which contain lot of information about application layer. In the above figure, It is GET request and the host is www.nitc.ac.in. The Connection general header controls whether or not the network connection stays open after the current transaction finishes. In our packet it is shown keep-alive impile that it is a persistent connection. The User-Agent request header is a characteristic string that lets servers and network peers identify the application, operating system, vendor, and/or version of the requesting user agent. The Accept request HTTP header advertises which content types the client is able to understand. The Accept-Encoding request HTTP header advertises which content encoding, usually a compression algorithm, the client is able to understand. Like that accept language advertises which language client

able to understand. The Cookie HTTP request header contains stored HTTP Cookie associated with the server

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

Skype and Zoom use both transport layer protocols UDP and TCP. For example, it sends Audio and video over UDP then uses TCP to initiate connection or to bypass some firewalls that block UDP packets.

71770	722.494806	192.168.43.44	134.224.211.170	TLSv1.2	92 Application Data
71771	722.509695	134.224.211.170	192.168.43.44	UDP	1071 8801 → 50904 Len=1029
71772	722.509945	134.224.211.170	192.168.43.44	UDP	1071 8801 → 50904 Len=1029
71773	722.510072	35.197.145.196	192.168.43.44	TCP	66 [TCP Dup ACK 71538#4] 443 → 55859 [ACK] Seq=3156 Ack=22369 Win=1963 Len=0 SLE=21703 SRE=22369
71774	722.510291	142.250.76.35	192.168.43.44	QUIC	73 Protected Payload (KP0)
71775	722.513272	134.224.211.170	192.168.43.44	TCP	54 443 → 56578 [ACK] Seq=16100 Ack=40257 Win=65536 Len=0
71776	722.515507	134.224.211.170	192.168.43.44	UDP	1071 8801 → 50904 Len=1029
71777	722.518387	134.224.211.170	192.168.43.44	UDP	1071 8801 → 50904 Len=1029
71778	722.518478	134.224.211.170	192.168.43.44	TLSv1.2	92 Application Data
71779	722.518499	192.168.43.44	134.224.211.170	TCP	66 [TCP Dup ACK 71757#1] 56578 → 443 [ACK] Seq=42450 Ack=14878 Win=66304 Len=0 SLE=16062 SRE=16138