A
05

Answer Sheet
Assignment - A05

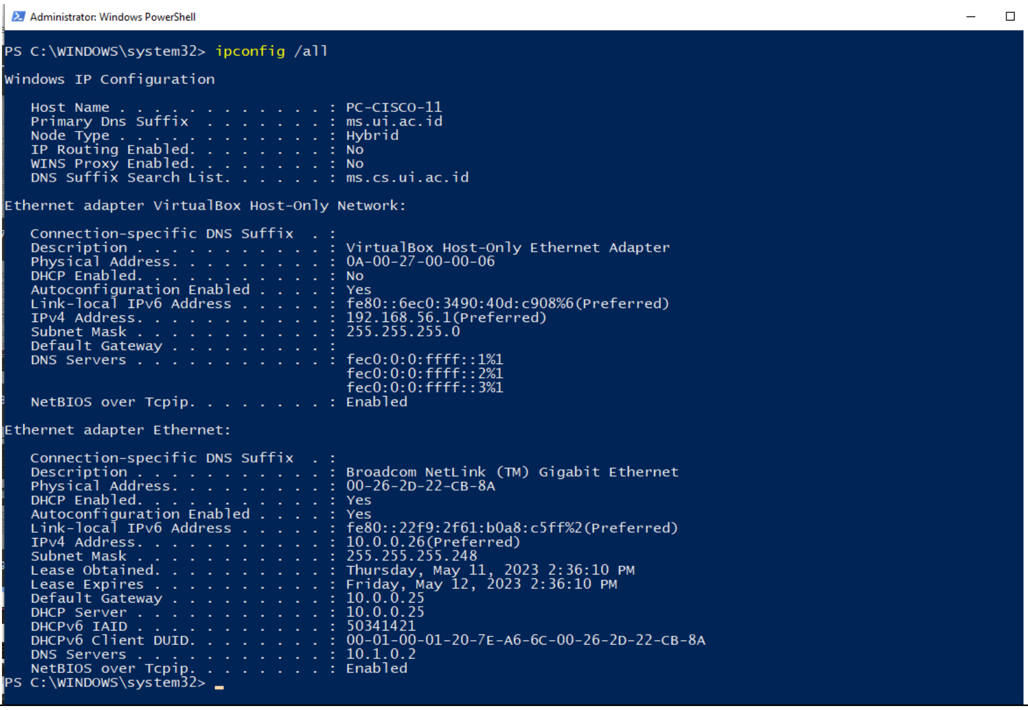
A Day in The Life of a Webpage Request

Name	:	Alvaro Austin
Student ID	:	2106752180

Notes:

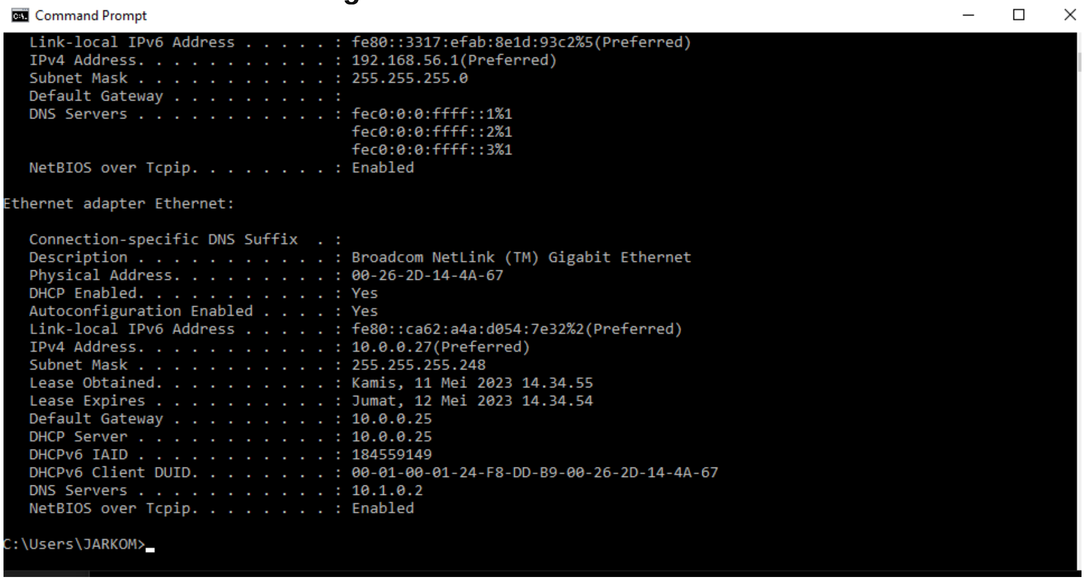
1. All answers you provide must be supported with relevant screenshots in the space provided. You can also highlight the part of the screenshot where it supports your answer.
2. Make sure before submitting a report, all screenshots and answers can be read by the assessment team properly.

Client PC Network Configuration:



```
PS C:\WINDOWS\system32> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : PC-CISCO-11
   Primary Dns Suffix  . . . . . . . : ms.ui.ac.id
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : ms.cs.ui.ac.id

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VirtualBox Host-Only Ethernet Adapter
   Physical Address. . . . . . . . . : 0A-00-27-00-00-06
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::6ec0:3490:40d:c908%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.56.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Broadcom NetLink (TM) Gigabit Ethernet
   Physical Address. . . . . . . . . : 00-26-2D-22-CB-8A
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::22f9:2f61:b0a8:c5ff%2(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.0.0.26(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.248
   Lease Obtained. . . . . . . . . . : Thursday, May 11, 2023 2:36:10 PM
   Lease Expires . . . . . . . . . . : Friday, May 12, 2023 2:36:10 PM
   Default Gateway . . . . . . . . . : 10.0.0.25
   DHCP Server . . . . . . . . . . . : 10.0.0.25
   DHCPv6 IAID . . . . . . . . . . . : 50341421
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-20-7E-A6-6C-00-26-2D-22-CB-8A
   DNS Servers . . . . . . . . . . . : 10.1.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
PS C:\WINDOWS\system32> _
```

Server PC Network Configuration:

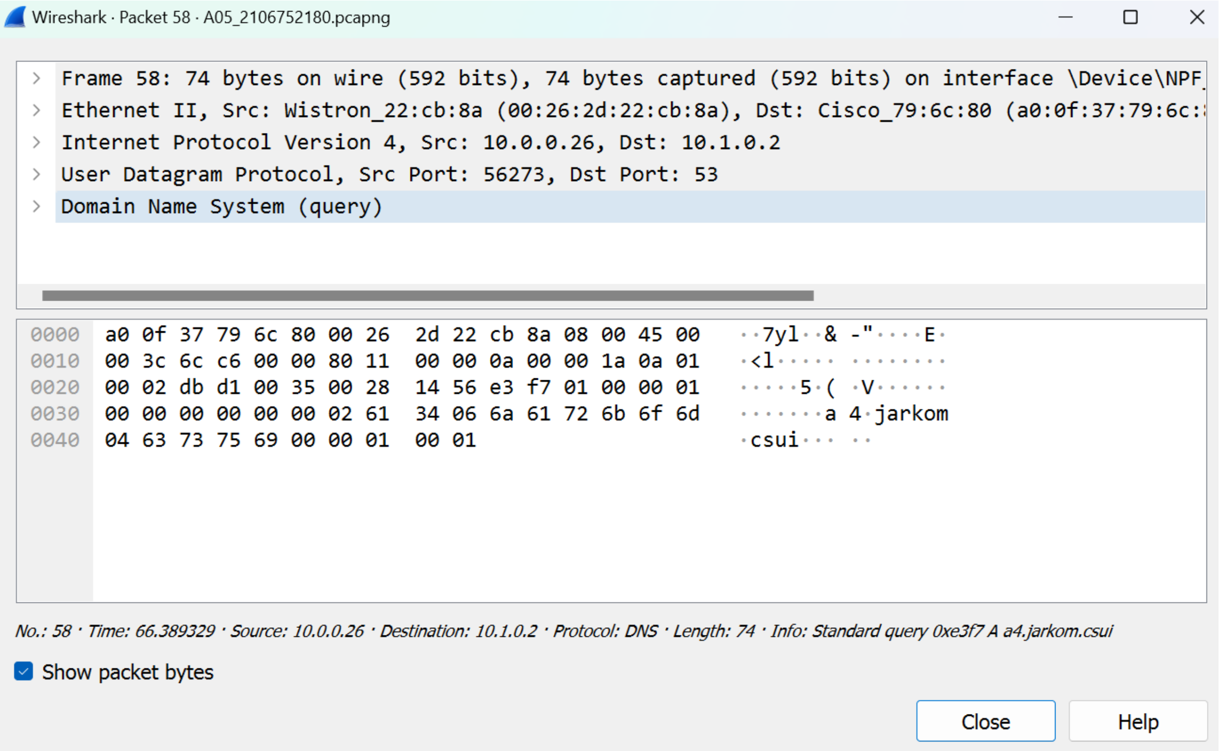


```
   Link-local IPv6 Address . . . . . : fe80::3317:efab:8e1d:93c2%5(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.56.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Broadcom NetLink (TM) Gigabit Ethernet
   Physical Address. . . . . . . . . : 00-26-2D-14-4A-67
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::ca62:a4a:d054:7e32%2(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.0.0.27(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.248
   Lease Obtained. . . . . . . . . . : Kamis, 11 Mei 2023 14.34.55
   Lease Expires . . . . . . . . . . : Jumat, 12 Mei 2023 14.34.54
   Default Gateway . . . . . . . . . : 10.0.0.25
   DHCP Server . . . . . . . . . . . : 10.0.0.25
   DHCPv6 IAID . . . . . . . . . . . : 184559149
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-F8-DD-B9-00-26-2D-14-4A-67
   DNS Servers . . . . . . . . . . . : 10.1.0.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Users\JARKOM>_
```

[10 points] The Protocols

Layer	Protocol	Frame Number
Application Layer	DNS	58

Application Layer Screenshot:



Wireshark · Packet 58 · A05_2106752180.pcapng

```
> Frame 58: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Cisco_79:6c:80 (a0:0f:37:79:6c:
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.1.0.2
> User Datagram Protocol, Src Port: 56273, Dst Port: 53
> Domain Name System (query)
```

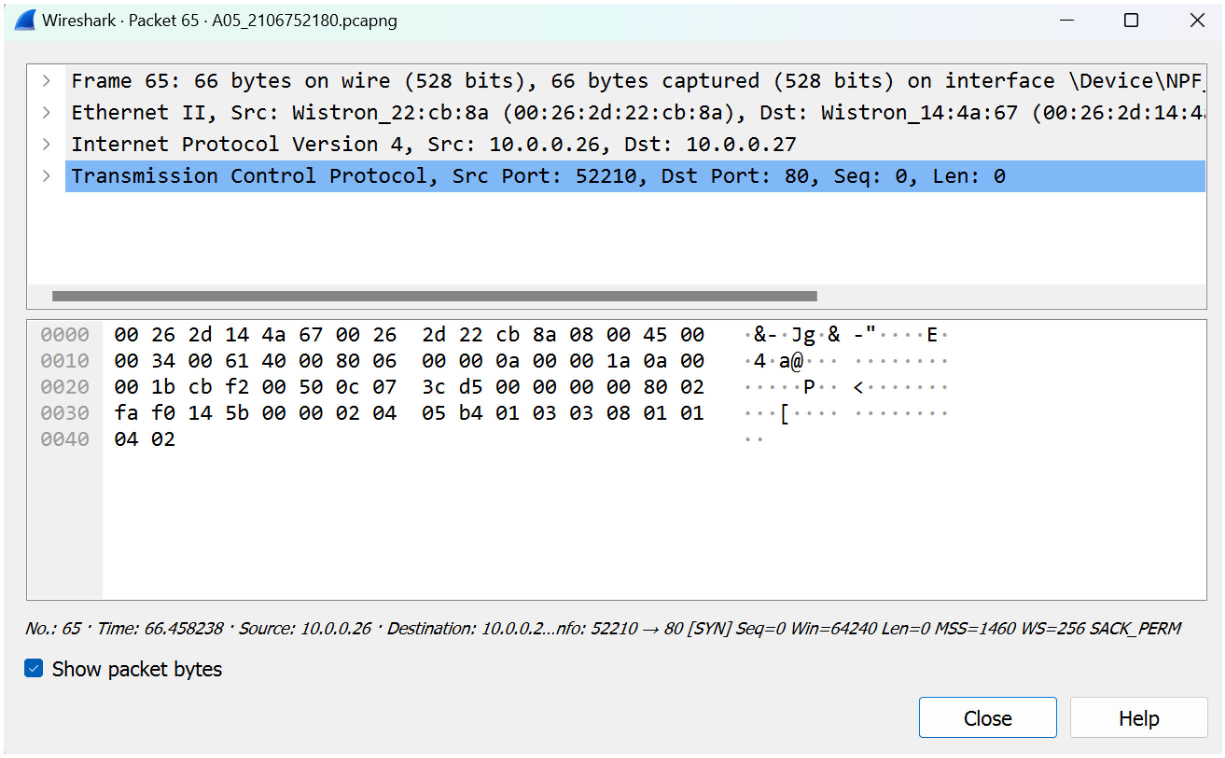
```
0000  a0 0f 37 79 6c 80 00 26  2d 22 cb 8a 08 00 45 00   ··7yl··& -"····E·
0010  00 3c 6c c6 00 00 80 11  00 00 0a 00 00 1a 0a 01   ·<l····· ········
0020  00 02 db d1 00 35 00 28  14 56 e3 f7 01 00 00 01   ·····5·( ·V······
0030  00 00 00 00 00 00 02 61  34 06 6a 61 72 6b 6f 6d   ·······a 4·jarkom
0040  04 63 73 75 69 00 00 01  00 01                     ·csui··· ··
```

No.: 58 · Time: 66.389329 · Source: 10.0.0.26 · Destination: 10.1.0.2 · Protocol: DNS · Length: 74 · Info: Standard query 0xe3f7 A a4.jarkom.csui

☑ Show packet bytes

Layer	Protocol	Frame Number
Transport Layer	Transmission Control Protocol (TCP)	65

Transport Layer Screenshot



Wireshark · Packet 65 · A05_2106752180.pcapng

```
> Frame 65: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14:4
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
> Transmission Control Protocol, Src Port: 52210, Dst Port: 80, Seq: 0, Len: 0
```

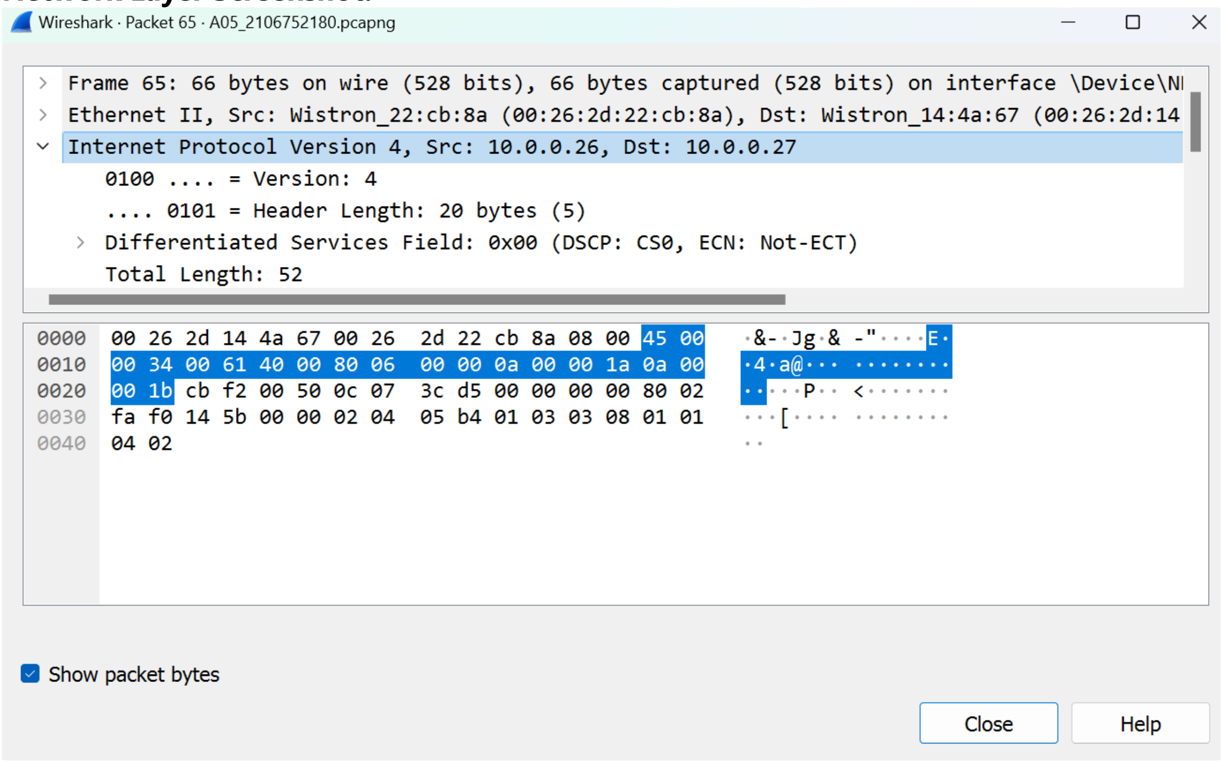
```
0000  00 26 2d 14 4a 67 00 26  2d 22 cb 8a 08 00 45 00   ·&-·Jg·& -"····E·
0010  00 34 00 61 40 00 80 06  00 00 0a 00 00 1a 0a 00   ·4·a@··· ········
0020  00 1b cb f2 00 50 0c 07  3c d5 00 00 00 00 80 02   ·····P·· <·······
0030  fa f0 14 5b 00 00 02 04  05 b4 01 03 03 08 01 01   ···[···· ········
0040  04 02                                              ··
```

No.: 65 · Time: 66.458238 · Source: 10.0.0.26 · Destination: 10.0.0.2...nfo: 52210 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

☑ Show packet bytes

Network Layer	Internet Protocol (IP)	65

Network Layer Screenshot:



Wireshark · Packet 65 · A05_2106752180.pcapng

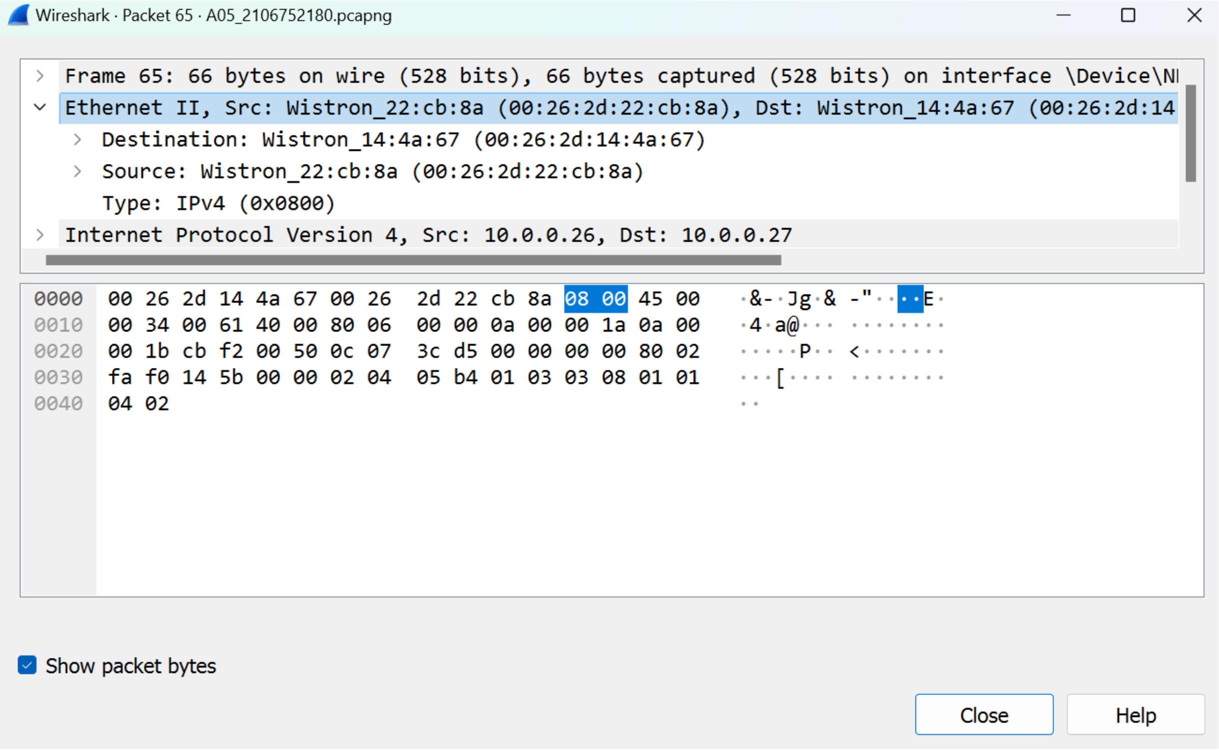
```
> Frame 65: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NI
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14
v Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 52
```

```
0000  00 26 2d 14 4a 67 00 26  2d 22 cb 8a 08 00 45 00   ·&-·Jg·& -"····E·
0010  00 34 00 61 40 00 80 06  00 00 0a 00 00 1a 0a 00   ·4·a@··· ········
0020  00 1b cb f2 00 50 0c 07  3c d5 00 00 00 00 80 02   ·····P·· <·······
0030  fa f0 14 5b 00 00 02 04  05 b4 01 03 03 08 01 01   ···[···· ········
0040  04 02                                              ··
```

☑ Show packet bytes

Link Layer	Ethernet	65

Link Layer Screenshot:



```
> Frame 65: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NI
v Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14
    > Destination: Wistron_14:4a:67 (00:26:2d:14:4a:67)
    > Source: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
```

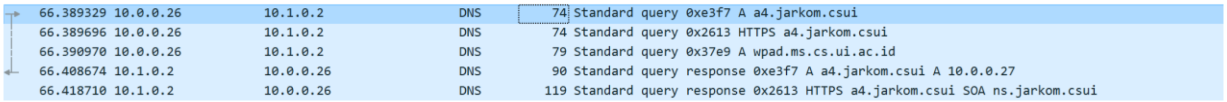
```
0000  00 26 2d 14 4a 67 00 26  2d 22 cb 8a 08 00 45 00   ·&-·Jg·& -"····E·
0010  00 34 00 61 40 00 80 06  00 00 0a 00 00 1a 0a 00   ·4·a@··· ········
0020  00 1b cb f2 00 50 0c 07  3c d5 00 00 00 00 80 02   ·····P·· <·······
0030  fa f0 14 5b 00 00 02 04  05 b4 01 03 03 08 01 01   ···[···· ········
0040  04 02                                              ··
```

☑ Show packet bytes

[20 points] The First Step

1. **[2 points]** Based on your observation, which frame is the first frame related to the communication between the client machine and the server machine?

Screenshot:



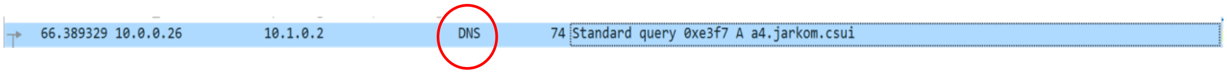
```
66.389329 10.0.0.26      10.1.0.2       DNS     74 Standard query 0xe3f7 A a4.jarkom.csui
66.389696 10.0.0.26      10.1.0.2       DNS     74 Standard query 0x2613 HTTPS a4.jarkom.csui
66.390970 10.0.0.26      10.1.0.2       DNS     79 Standard query 0x37e9 A wpad.ms.cs.ui.ac.id
66.408674 10.1.0.2       10.0.0.26      DNS     90 Standard query response 0xe3f7 A a4.jarkom.csui A 10.0.0.27
66.418710 10.1.0.2       10.0.0.26      DNS    119 Standard query response 0x2613 HTTPS a4.jarkom.csui SOA ns.jarkom.csui
```

Explanation:

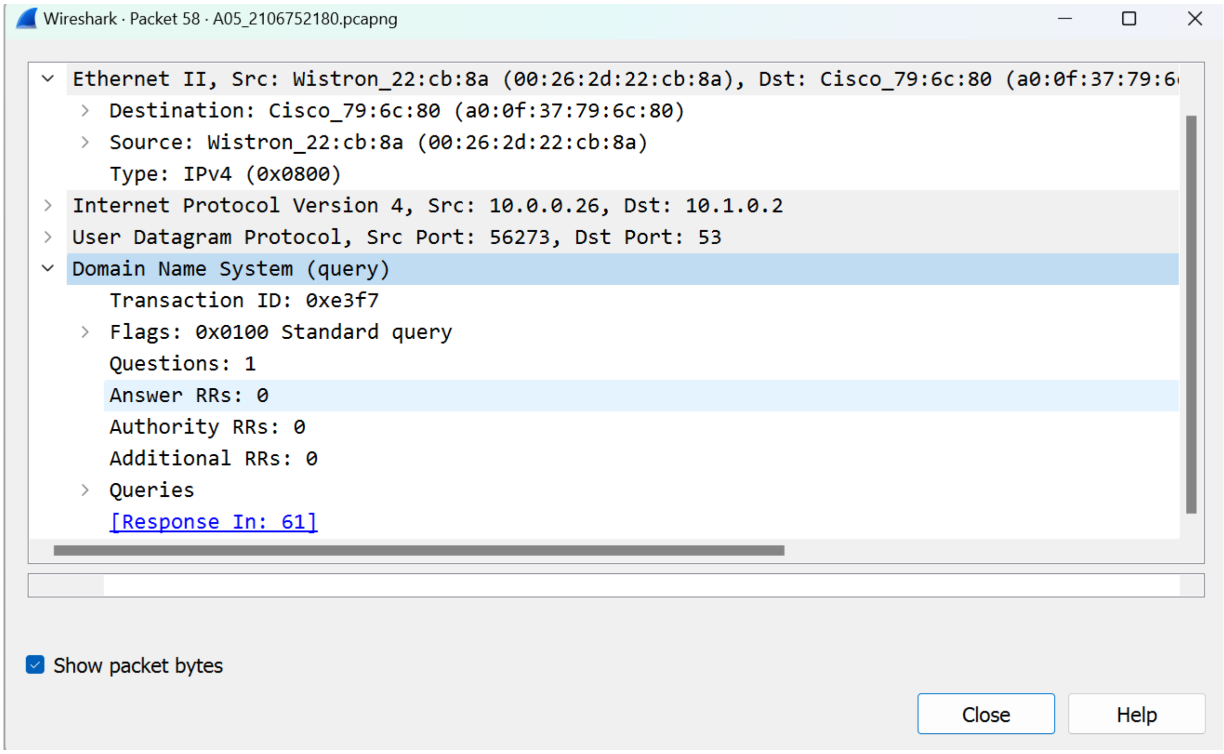
The first frame that is related to communication between the client machine and the server machine is frame 58. This indicates the start of the communication, when the client try to access the web of a4.jarkom.csui. DNS is an application layer protocol that is used to mapped an IP Address to specific domain name. In this case, the DNS server (10.1.0.2), finds the server which was 10.0.0.27 and told the client about this information.

2. **[2 points]** What protocol is primarily being used in the frame (and is identified by Wireshark as that protocol)?

Screenshot:



```
66.389329 10.0.0.26          10.1.0.2          DNS          74 Standard query 0xe3f7 A a4.jarkom.csui
```



Wireshark · Packet 58 · A05_2106752180.pcapng

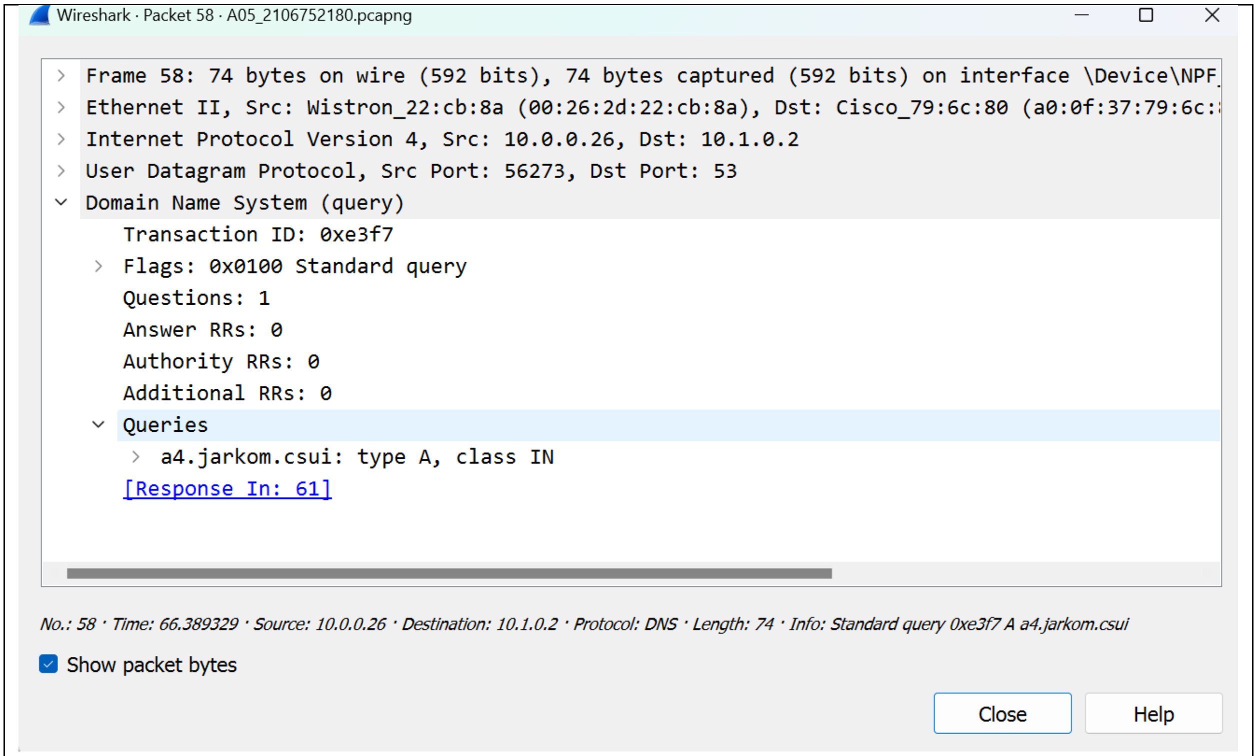
```
∨ Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Cisco_79:6c:80 (a0:0f:37:79:6
    > Destination: Cisco_79:6c:80 (a0:0f:37:79:6c:80)
    > Source: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
      Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.1.0.2
> User Datagram Protocol, Src Port: 56273, Dst Port: 53
∨ Domain Name System (query)
      Transaction ID: 0xe3f7
    > Flags: 0x0100 Standard query
      Questions: 1
      Answer RRs: 0
      Authority RRs: 0
      Additional RRs: 0
    > Queries
      [Response In: 61]
```

☑ Show packet bytes

Explanation:
Protocol that is primarily being used in the frame is the application layer protocol, the Domain Name System Protocol.

3. **[5 points]** What function does the communication done with this frame (and its counterpart) serve as part of the communication between the client and server machines?

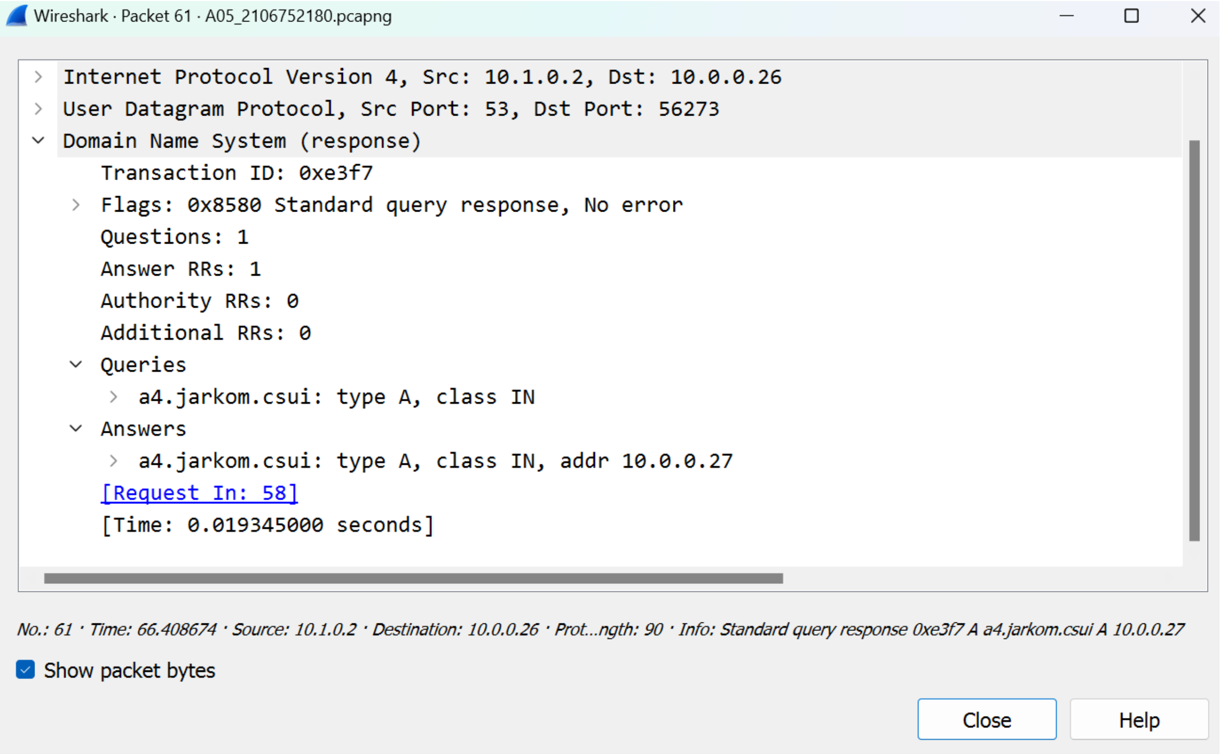
Screenshot:
Frame 58:



```
> Frame 58: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Cisco_79:6c:80 (a0:0f:37:79:6c:
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.1.0.2
> User Datagram Protocol, Src Port: 56273, Dst Port: 53
v Domain Name System (query)
      Transaction ID: 0xe3f7
    > Flags: 0x0100 Standard query
      Questions: 1
      Answer RRs: 0
      Authority RRs: 0
      Additional RRs: 0
    v Queries
        > a4.jarkom.csui: type A, class IN
      [Response In: 61]
```

No.: 58 · Time: 66.389329 · Source: 10.0.0.26 · Destination: 10.1.0.2 · Protocol: DNS · Length: 74 · Info: Standard query 0xe3f7 A a4.jarkom.csui

Frame 61:



```
> Internet Protocol Version 4, Src: 10.1.0.2, Dst: 10.0.0.26
> User Datagram Protocol, Src Port: 53, Dst Port: 56273
v Domain Name System (response)
      Transaction ID: 0xe3f7
    > Flags: 0x8580 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 0
      Additional RRs: 0
    v Queries
        > a4.jarkom.csui: type A, class IN
    v Answers
        > a4.jarkom.csui: type A, class IN, addr 10.0.0.27
      [Request In: 58]
      [Time: 0.019345000 seconds]
```

No.: 61 · Time: 66.408674 · Source: 10.1.0.2 · Destination: 10.0.0.26 · Prot...ngth: 90 · Info: Standard query response 0xe3f7 A a4.jarkom.csui A 10.0.0.27

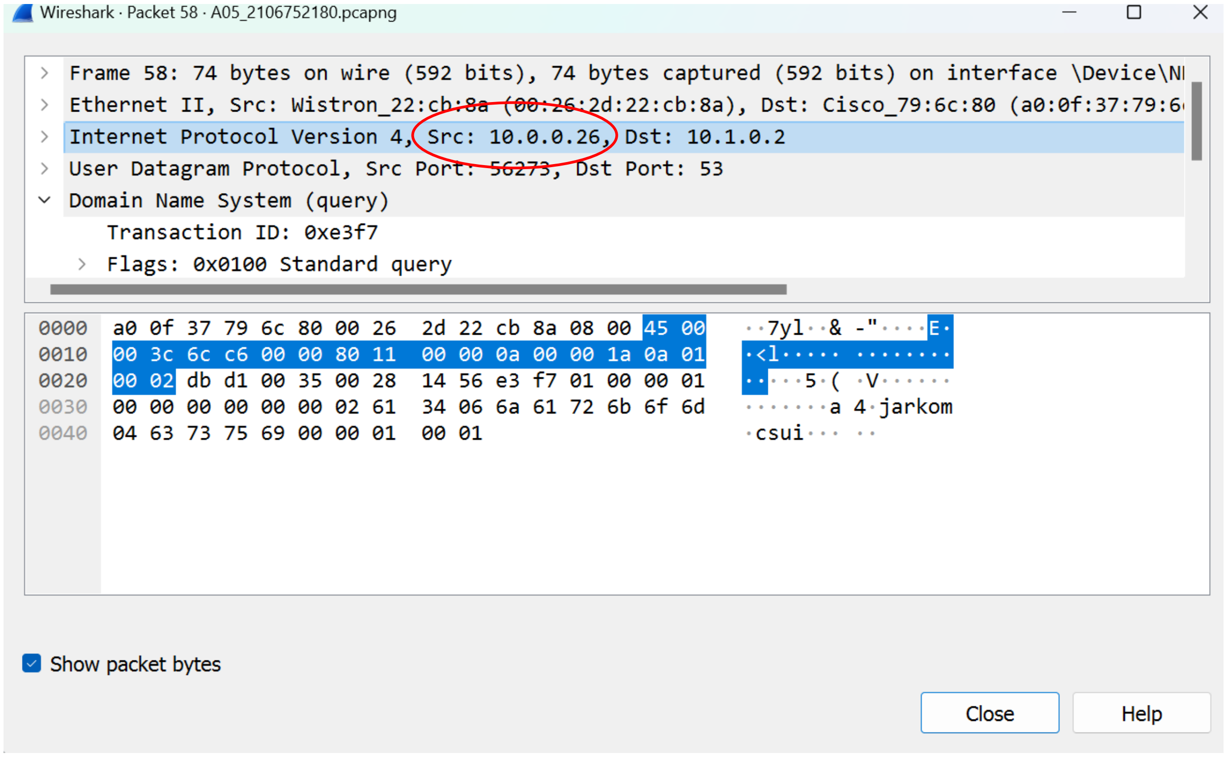
Explanation:

The role of this frame is to translate the domain name using DNS protocol to get the desired IP that mapped to the domain name. As you can see in the screenshot above, the first frame that was related to the communication between the client and the server is frame 58, in which, DNS protocol, has 1 question. The question is related to the IP Address of a4.jarkom.csui.Then, in frame 61, DNS gave their answer, which highlighted in blue color in the screenshot above.

From the highlights, you can see that when the client access a4.jarkom.csui, DNS protocol translate that domain name using DNS protocol, then returns the IP address once it's found.

4. **[2 points]** Who is the sender of the frame? The answer options are "Client", "Server", and "DNS Server". However, you need to elaborate on how you deduce who the sender is based on existing information.

Screenshot:



Wireshark · Packet 58 · A05_2106752180.pcapng

```
> Frame 58: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NI
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Cisco_79:6c:80 (a0:0f:37:79:6
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.1.0.2
> User Datagram Protocol, Src Port: 56273, Dst Port: 53
v Domain Name System (query)
     Transaction ID: 0xe3f7
   > Flags: 0x0100 Standard query
```

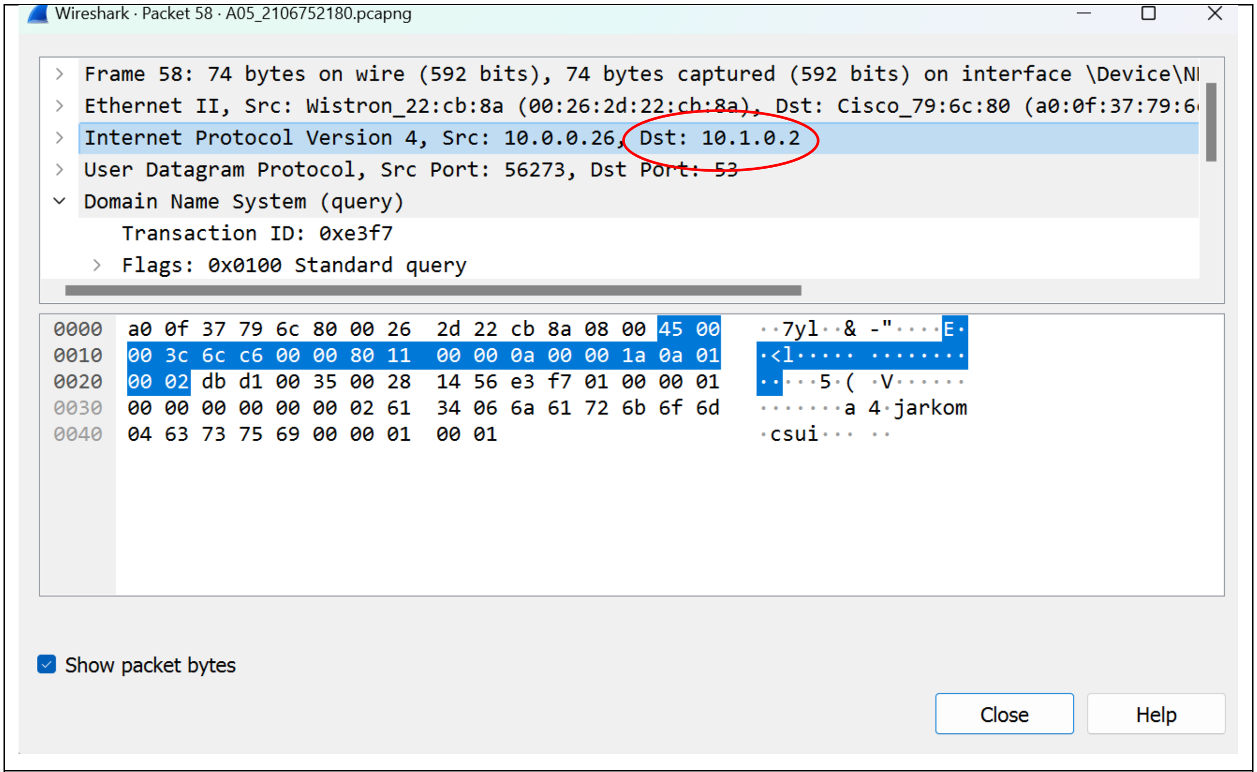
```
0000  a0 0f 37 79 6c 80 00 26  2d 22 cb 8a 08 00 45 00   ··7yl··& -"····E·
0010  00 3c 6c c6 00 00 80 11  00 00 0a 00 00 1a 0a 01   ·<l····· ········
0020  00 02 db d1 00 35 00 28  14 56 e3 f7 01 00 00 01   ·····5·( ·V······
0030  00 00 00 00 00 00 02 61  34 06 6a 61 72 6b 6f 6d   ·······a 4·jarkom
0040  04 63 73 75 69 00 00 01  00 01                     ·csui··· ··
```

☑ Show packet bytes

Explanation:
The sender of the frame is the client. The client in this case is the Client A4 with IP of 10.0.0.26. I can deduce this because the first frame has the source IP address of the client.

5. **[2 points]** Who is the recipient of the frame? The answer options are "Client", "Server", and "DNS Server". However, you need to elaborate on how you deduce who the recipient is based on existing information.

Screenshot:



```
> Frame 58: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NI
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Cisco_79:6c:80 (a0:0f:37:79:6
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.1.0.2
> User Datagram Protocol, Src Port: 56273, Dst Port: 53
v Domain Name System (query)
     Transaction ID: 0xe3f7
   > Flags: 0x0100 Standard query
```

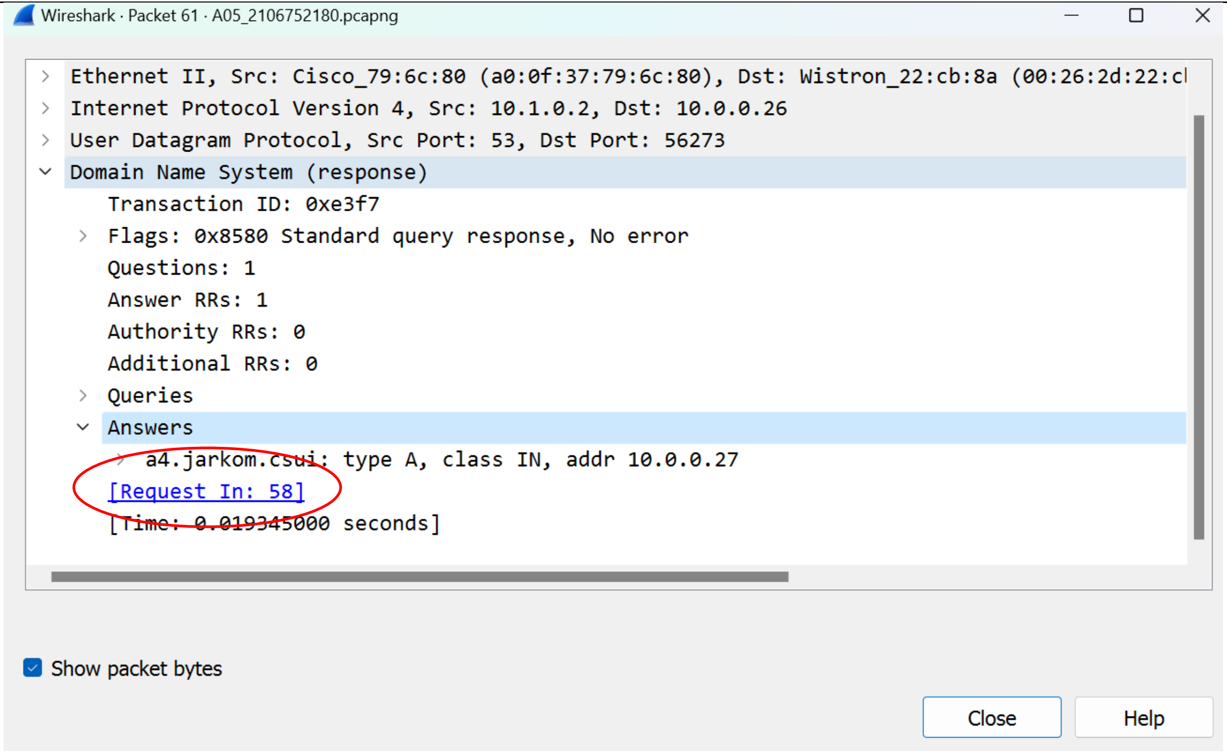
```
0000  a0 0f 37 79 6c 80 00 26  2d 22 cb 8a 08 00 45 00   ··7yl··& -"····E·
0010  00 3c 6c c6 00 00 80 11  00 00 0a 00 00 1a 0a 01   ·<l····· ········
0020  00 02 db d1 00 35 00 28  14 56 e3 f7 01 00 00 01   ·····5·( ·V······
0030  00 00 00 00 00 00 02 61  34 06 6a 61 72 6b 6f 6d   ·······a 4·jarkom
0040  04 63 73 75 69 00 00 01  00 01                     ·csui··· ··
```

☑ Show packet bytes

Explanation:
The recipient of the frame 58 is the DNS Server. DNS Server acts as the local DNS server to map the domain name to find the desired IP address. The reason I chose this, is because the destination of the frame is towards the IP address of DNS Server.

6. **[2 points]** Based on your observation, which frame is the response for the frame that you identified in the first number?

Screenshot:

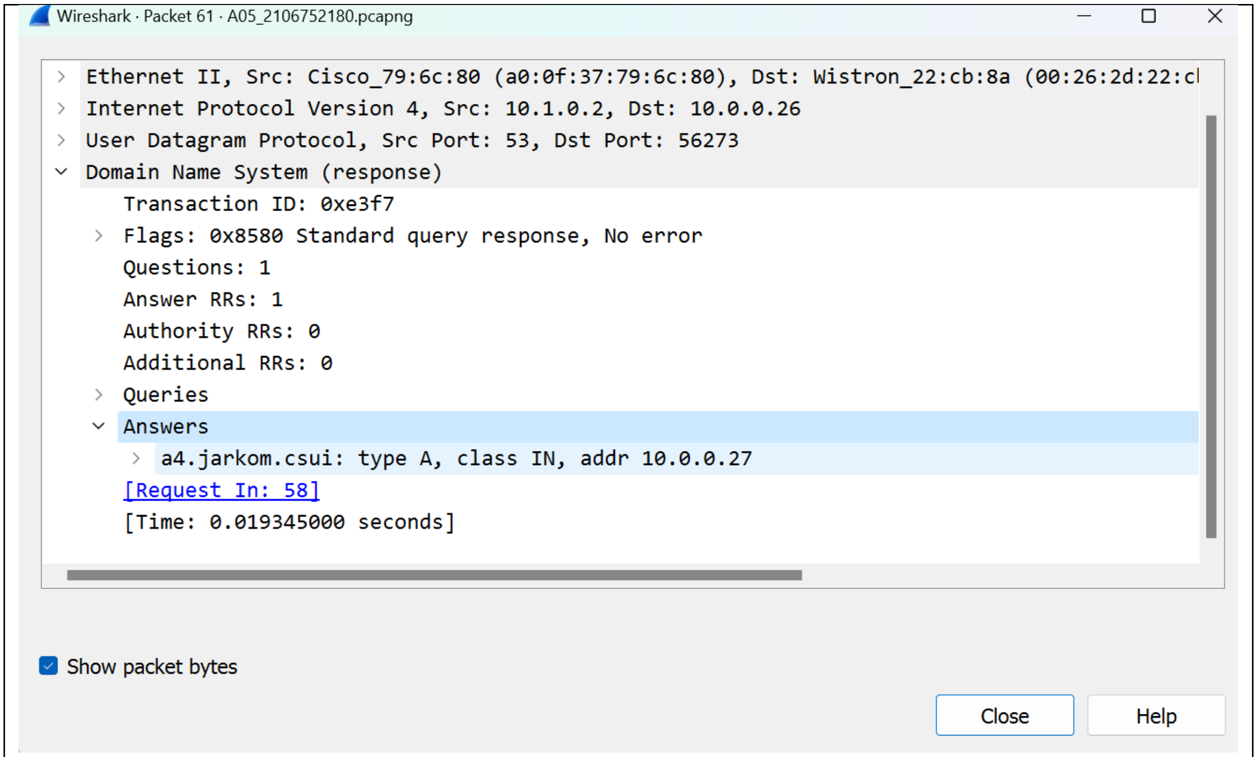


```
>  Ethernet II, Src: Cisco_79:6c:80 (a0:0f:37:79:6c:80), Dst: Wistron_22:cb:8a (00:26:2d:22:c
>  Internet Protocol Version 4, Src: 10.1.0.2, Dst: 10.0.0.26
>  User Datagram Protocol, Src Port: 53, Dst Port: 56273
v  Domain Name System (response)
      Transaction ID: 0xe3f7
   >  Flags: 0x8580 Standard query response, No error
      Questions: 1
      Answer RRs: 1
      Authority RRs: 0
      Additional RRs: 0
   >  Queries
   v  Answers
      >  a4.jarkom.csui: type A, class IN, addr 10.0.0.27
      [Request In: 58]
      [Time: 0.019345000 seconds]
```

Explanation:
The frame that was the response of my request is frame 61. This is because the response above were sent by DNS server back to the client for the answers. As you can see in the above screenshots, there were also answers. You can see in the first number screenshot, there is “Response in: 61”.

7. **[5 points]** What is the information that is contained in the response that fulfills the function of this communication? Please be specific regarding the information type.

Screenshot:



Wireshark · Packet 61 · A05_2106752180.pcapng

```
> Ethernet II, Src: Cisco_79:6c:80 (a0:0f:37:79:6c:80), Dst: Wistron_22:cb:8a (00:26:2d:22:c
> Internet Protocol Version 4, Src: 10.1.0.2, Dst: 10.0.0.26
> User Datagram Protocol, Src Port: 53, Dst Port: 56273
v Domain Name System (response)
     Transaction ID: 0xe3f7
   > Flags: 0x8580 Standard query response, No error
     Questions: 1
     Answer RRs: 1
     Authority RRs: 0
     Additional RRs: 0
   > Queries
   v Answers
      > a4.jarkom.csui: type A, class IN, addr 10.0.0.27
     [Request In: 58]
     [Time: 0.019345000 seconds]
```

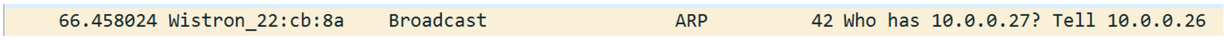
☑ Show packet bytes

Explanation:
The information that is contained in the response is the type, which was, A record, towards address 10.0.0.27 (Server). This response basically meant that, a4.jarkom.csui is mapped to IP address of 10.0.0.27, from then on, the client will know that when they sent request to a4.jarkom.csui, it will automatically refer to IP 10.0.0.27 until TTL is expired.

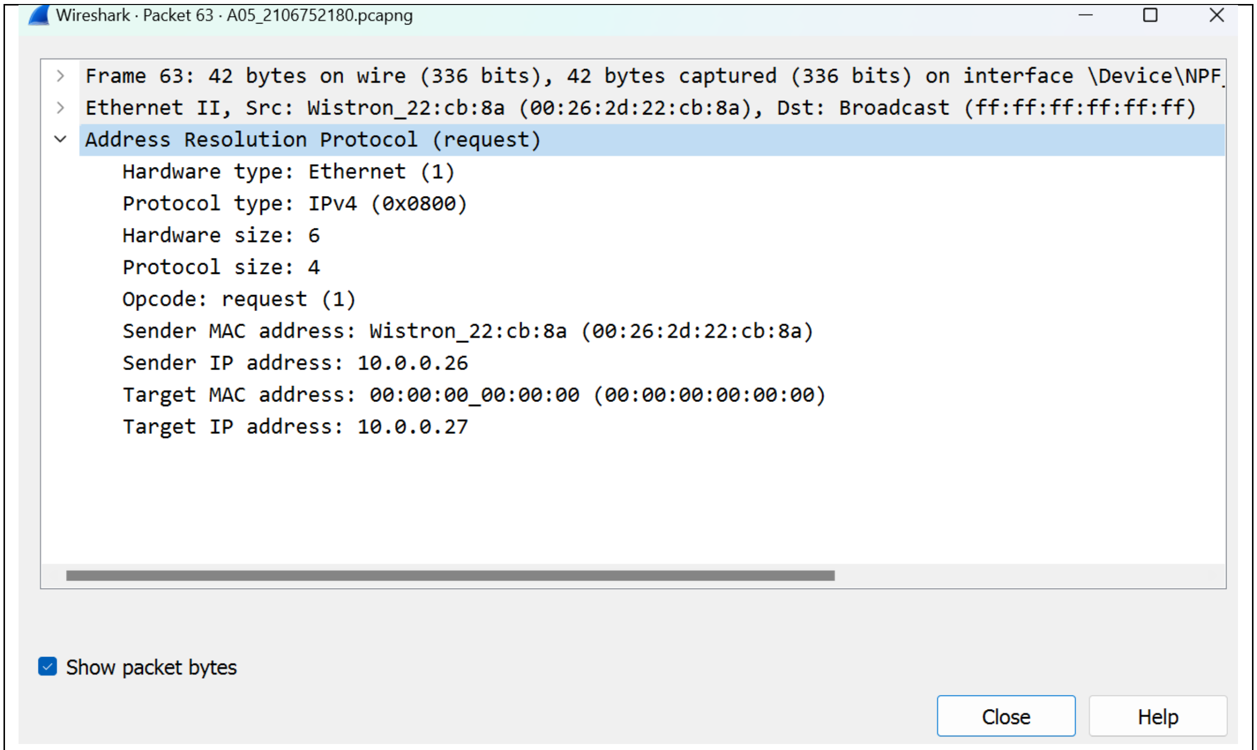
[20 points] The Second Step

1. **[2 points]** Based on your observation, which frame is the next frame, after the first step and its response, related to the communication between the client machine and the server machine?

Screenshot:



```
    66.458024 Wistron_22:cb:8a     Broadcast                   ARP          42 Who has 10.0.0.27? Tell 10.0.0.26
```



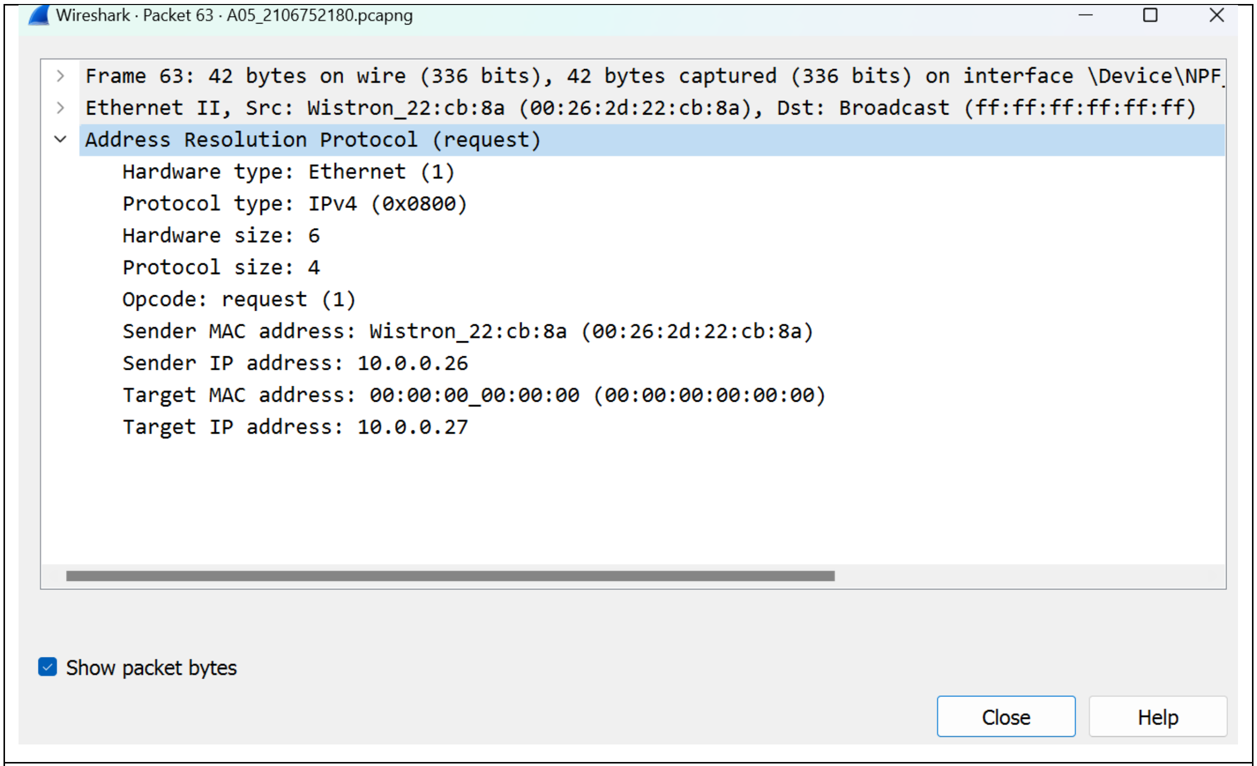
```
> Frame 63: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request)
     Hardware type: Ethernet (1)
     Protocol type: IPv4 (0x0800)
     Hardware size: 6
     Protocol size: 4
     Opcode: request (1)
     Sender MAC address: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
     Sender IP address: 10.0.0.26
     Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
     Target IP address: 10.0.0.27
```

Explanation:

The frame that is the next frame is frame 63. This frame is right after the last frame of DNS protocol frames related to communication between the client and the server.

2. **[5 points]** What protocol is primarily being used in the frame (and is identified by Wireshark as that protocol)?

Screenshot:

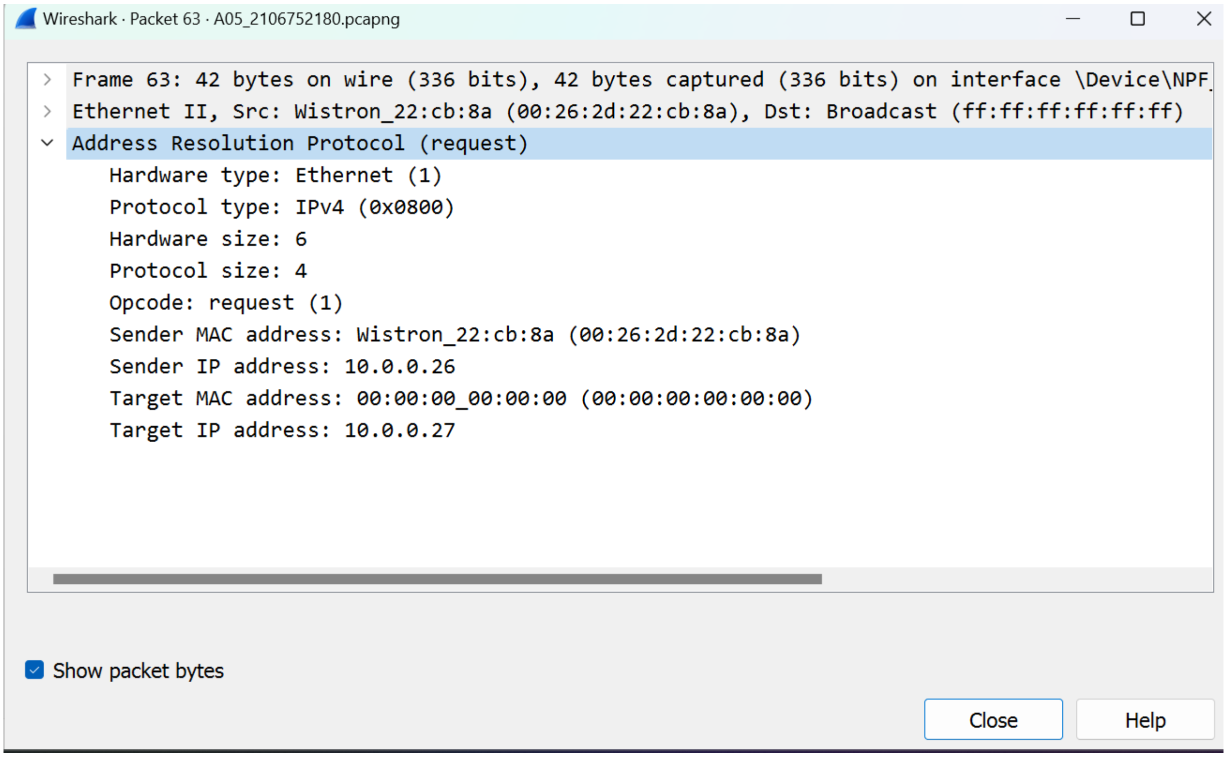


```
> Frame 63: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request)
     Hardware type: Ethernet (1)
     Protocol type: IPv4 (0x0800)
     Hardware size: 6
     Protocol size: 4
     Opcode: request (1)
     Sender MAC address: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
     Sender IP address: 10.0.0.26
     Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
     Target IP address: 10.0.0.27
```

Explanation:
Protocol that is primarily used in the frame is Address Resolution Protocol (ARP). ARP is a protocol or procedure that connects an ever-changing Internet Protocol (IP) address to a fixed physical machine address, also known as a media access control (MAC) address, in a local-area network (LAN).

3. **[2 points]** What function does the communication done with this frame (and its counterpart) serve as part of the communication between the client and server machines?

Screenshot:



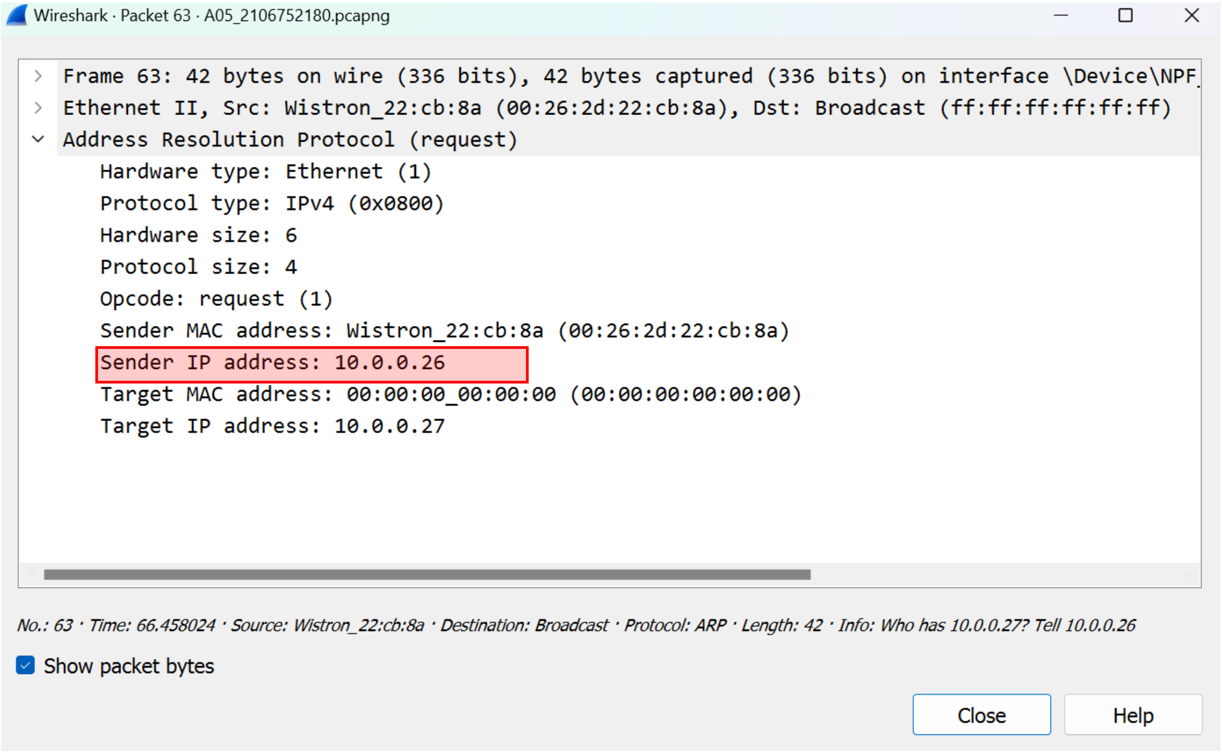
```
> Frame 63: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
v Address Resolution Protocol (request)
     Hardware type: Ethernet (1)
     Protocol type: IPv4 (0x0800)
     Hardware size: 6
     Protocol size: 4
     Opcode: request (1)
     Sender MAC address: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
     Sender IP address: 10.0.0.26
     Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
     Target IP address: 10.0.0.27
```

☑ Show packet bytes

Explanation:
This ARP protocol is used to find the MAC address of a PC. This MAC address is used to uniquely identify PC's related. The MAC address is then used for each PC to communicate with each other. This frame is basically broadcast to all PC, to find the mac address of the PC that has the IP address of 10.0.0.27.

4. **[2 points]** Who is the sender of the frame? The answer options are "Client", "Server", and "DNS Server". However, you need to elaborate on how you deduce who the sender is based on existing information.

Screenshot:



Wireshark · Packet 63 · A05_2106752180.pcapng

```
> Frame 63: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
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: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
     Sender IP address: 10.0.0.26
     Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
     Target IP address: 10.0.0.27
```

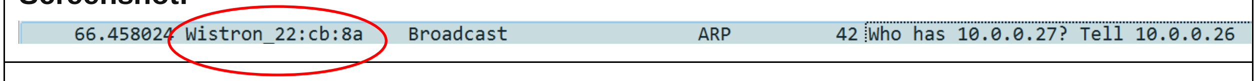
No.: 63 · Time: 66.458024 · Source: Wistron_22:cb:8a · Destination: Broadcast · Protocol: ARP · Length: 42 · Info: Who has 10.0.0.27? Tell 10.0.0.26

☑ Show packet bytes

Explanation:
The sender of the frame is the client. The reason for this, is because Wistron_22::cb:8a (sender) is the MAC Address of the client. Hence the client, is the sender of the frame. This information is backed because from the detail, you can see that Sender IP Address is 10.0.0.26, which is the client.

5. **[2 points]** Why is the sender address format different between the first and second step?

Screenshot:

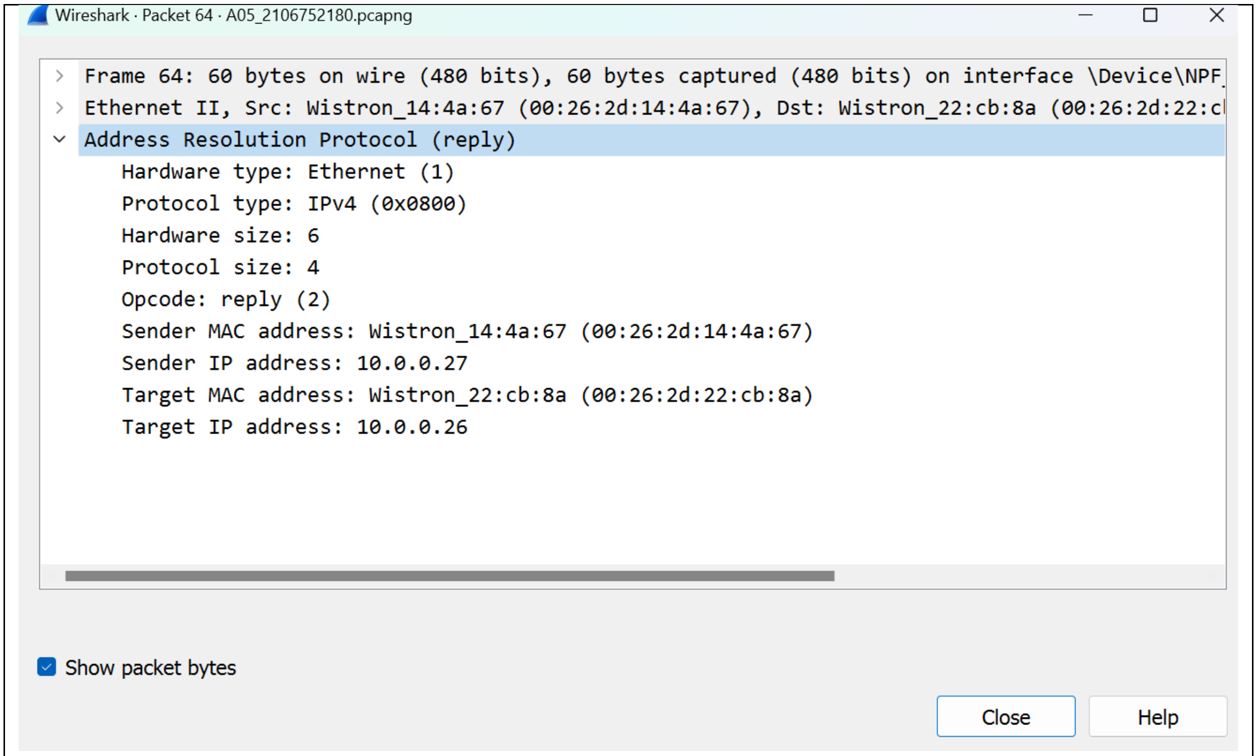


```
      66.458024 Wistron_22:cb:8a     Broadcast                ARP          42 Who has 10.0.0.27? Tell 10.0.0.26
```

Explanation:
The sender address format in second step is the MAC address of the sender of first step (Client's MAC address). So these two is different because MAC address identify physical address of a device (Client), and IP address identify the device (client) globally.

6. **[2 points]** Based on your observation, which frame is the response for the frame that you identified in the first number?

Screenshot:

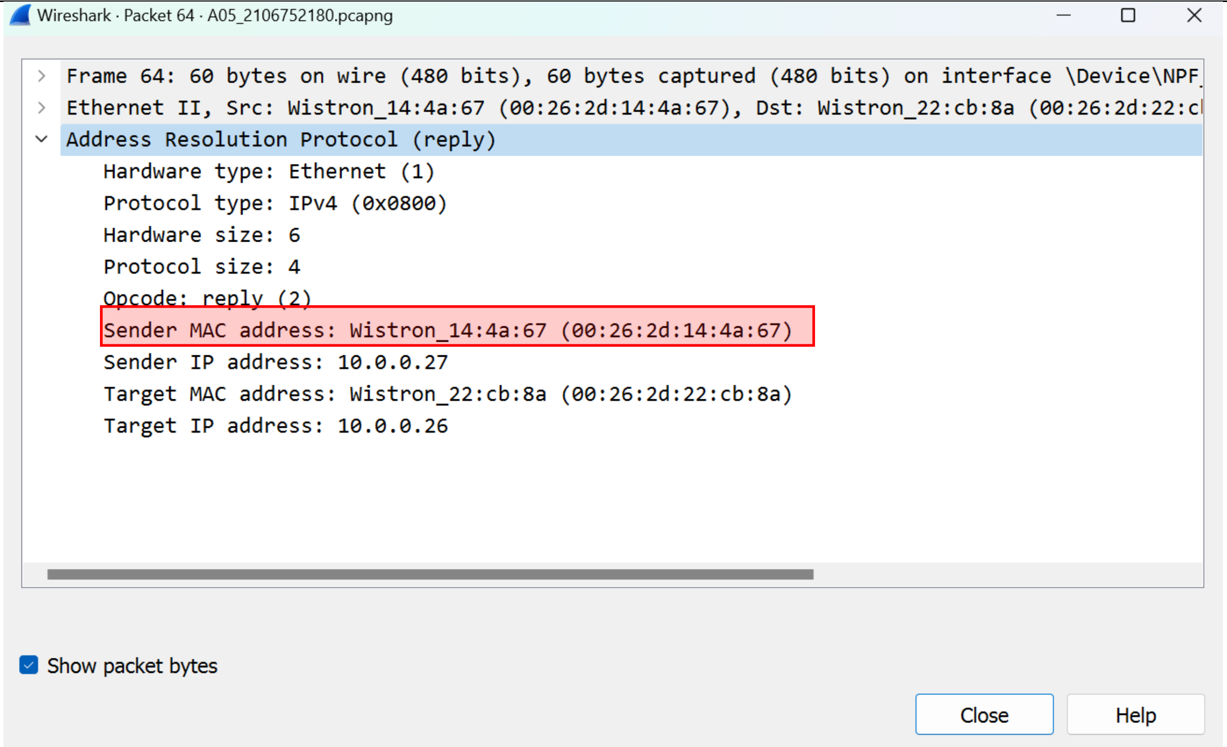


```
> Frame 64: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_14:4a:67 (00:26:2d:14:4a:67), Dst: Wistron_22:cb:8a (00:26:2d:22:c
v Address Resolution Protocol (reply)
     Hardware type: Ethernet (1)
     Protocol type: IPv4 (0x0800)
     Hardware size: 6
     Protocol size: 4
     Opcode: reply (2)
     Sender MAC address: Wistron_14:4a:67 (00:26:2d:14:4a:67)
     Sender IP address: 10.0.0.27
     Target MAC address: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
     Target IP address: 10.0.0.26
```

Explanation:
The frame that gives the response for previous frame is **frame 64.**

7. **[5 points]** What is the information that is contained in the response that fulfills the function of this communication? Please be specific regarding the information type.

Screenshot:



```
> Frame 64: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_
> Ethernet II, Src: Wistron_14:4a:67 (00:26:2d:14:4a:67), Dst: Wistron_22:cb:8a (00:26:2d:22:c
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: Wistron_14:4a:67 (00:26:2d:14:4a:67)
     Sender IP address: 10.0.0.27
     Target MAC address: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
     Target IP address: 10.0.0.26
```

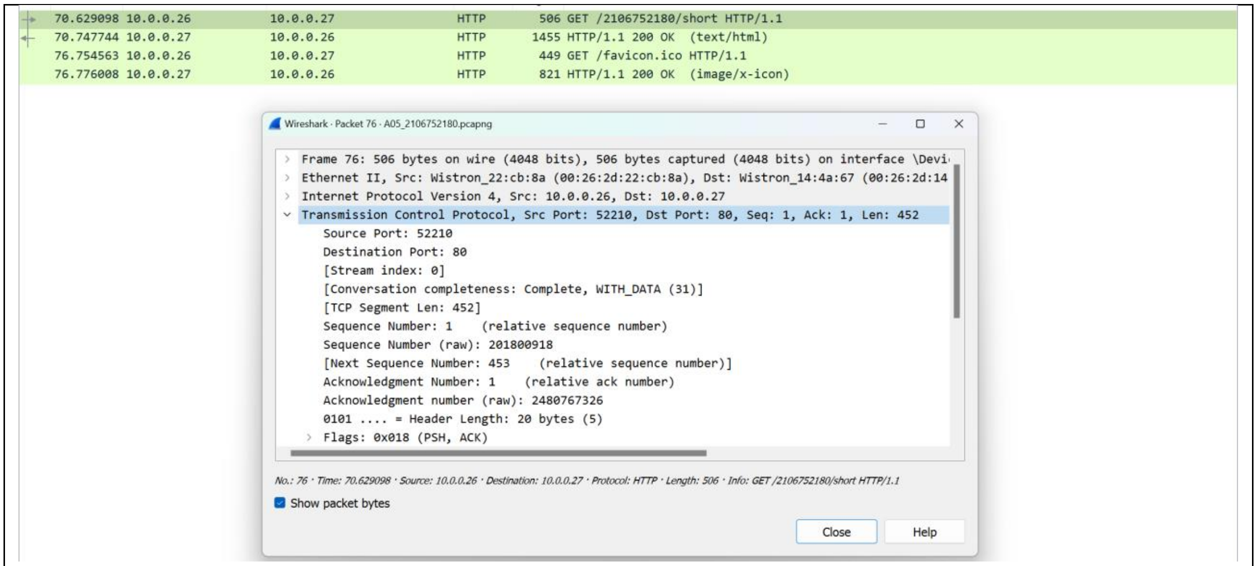
Explanation:

As you can see in the screenshot above, the reason that the response fulfills the function of this communication is because the response replied back to the initial sender (client). This happened because when previously, the client ask for server's MAC Address (initially, MAC address for server is 00:00:00:00:00:00). Then, as you can see from the screenshot, the server sent back a reply with the sender MAC address of 00:26:2d:14:4a:67. Hence, ARP is fulfilled because the ARP request has been replied with the information of server's MAC Address.

[8 points] HTTP and Its Transport

1. **[3 points]** Look for a frame that has information about the HTTP request to the domain that has been loaded. What transport layer protocol supports the selected frame?

Screenshot:



```
70.629098 10.0.0.26          10.0.0.27          HTTP        506 GET /2106752180/short HTTP/1.1
70.747744 10.0.0.27          10.0.0.26          HTTP       1455 HTTP/1.1 200 OK  (text/html)
76.754563 10.0.0.26          10.0.0.27          HTTP        449 GET /favicon.ico HTTP/1.1
76.776008 10.0.0.27          10.0.0.26          HTTP        821 HTTP/1.1 200 OK  (image/x-icon)
```

```
> Frame 76: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface \Devi
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
v Transmission Control Protocol, Src Port: 52210, Dst Port: 80, Seq: 1, Ack: 1, Len: 452
    Source Port: 52210
    Destination Port: 80
    [Stream index: 0]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 452]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 201800918
    [Next Sequence Number: 453    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 2480767326
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x018 (PSH, ACK)
```

No.: 76 · Time: 70.629098 · Source: 10.0.0.26 · Destination: 10.0.0.27 · Protocol: HTTP · Length: 506 · Info: GET /2106752180/short HTTP/1.1

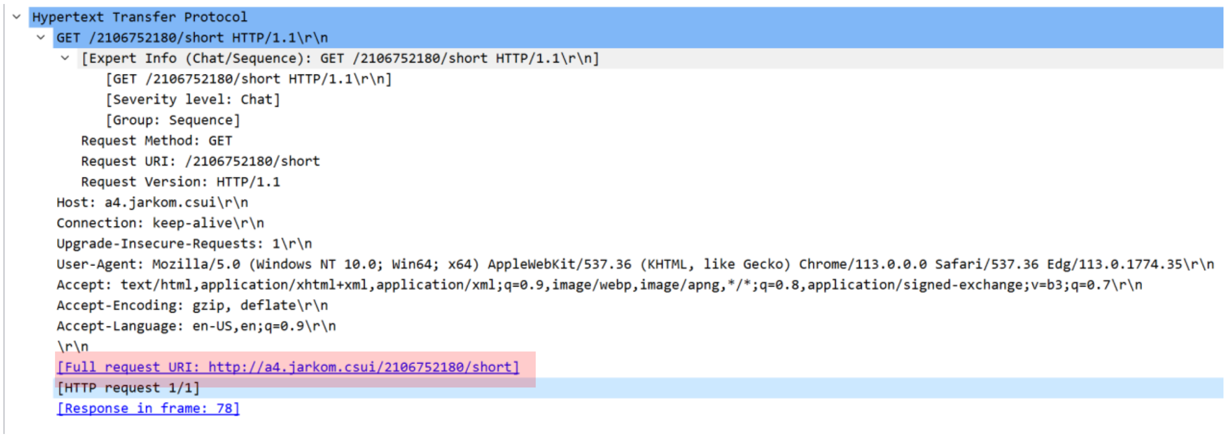
☑ Show packet bytes

Explanation:
Transport layer protocol that supported it’s selected frame is Transmission Control Protocol (TCP). This can be found in the details of the request, above Hypertext Transfer Protocol.

2. **[5 points]** Does the HTTP protocol include information regarding the origin and destination of the request? Explain specifically the type of information it contains, for example MAC Address, FQDN, and others!

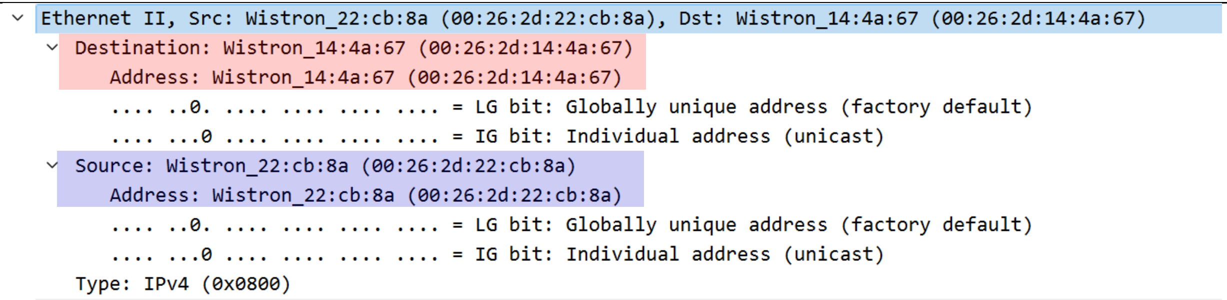
Screenshot:

Origin: -
Information Regarding Destination of the Request in HTTP Protocol:



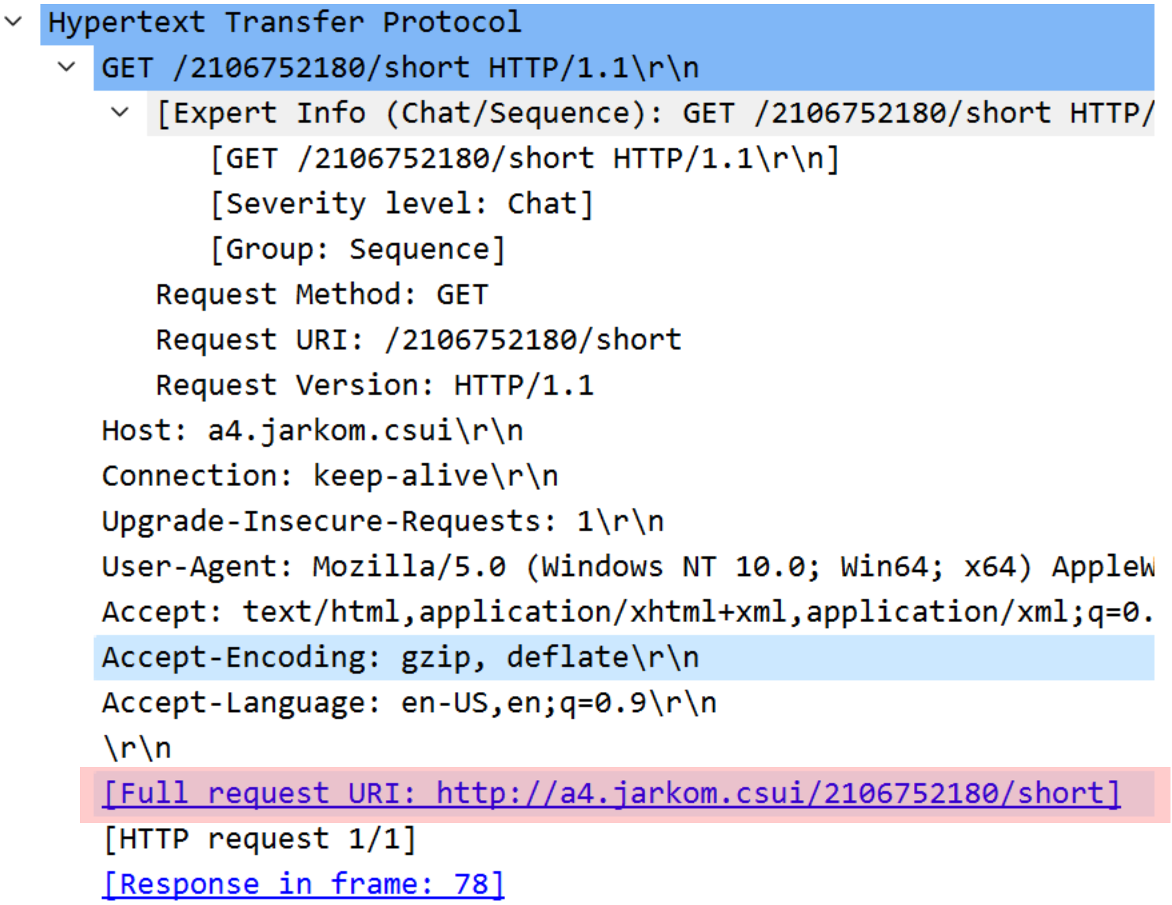
```
v Hypertext Transfer Protocol
  v GET /2106752180/short HTTP/1.1\r\n
    v [Expert Info (Chat/Sequence): GET /2106752180/short HTTP/1.1\r\n]
        [GET /2106752180/short HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /2106752180/short
      Request Version: HTTP/1.1
    Host: a4.jarkom.csui\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/113.0.0.0 Safari/537.36 Edg/113.0.1774.35\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://a4.jarkom.csui/2106752180/short]
    [HTTP request 1/1]
    [Response in frame: 78]
```

MAC Address:



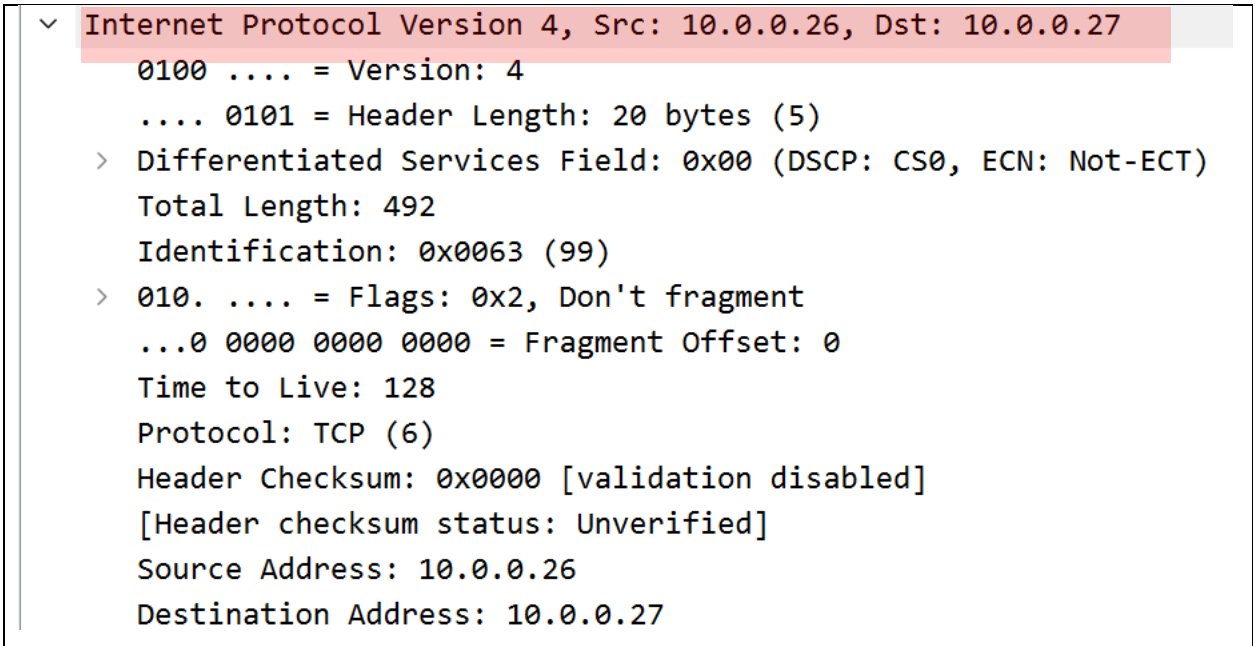
```
∨ Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14:4a:67)
  ∨ Destination: Wistron_14:4a:67 (00:26:2d:14:4a:67)
      Address: Wistron_14:4a:67 (00:26:2d:14:4a:67)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ∨ Source: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
      Address: Wistron_22:cb:8a (00:26:2d:22:cb:8a)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

FQDN:



```
∨ Hypertext Transfer Protocol
  ∨ GET /2106752180/short HTTP/1.1\r\n
    ∨ [Expert Info (Chat/Sequence): GET /2106752180/short HTTP/
        [GET /2106752180/short HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /2106752180/short
      Request Version: HTTP/1.1
    Host: a4.jarkom.csui\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleW
    Accept: text/html,application/xhtml+xml,application/xml;q=0.
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://a4.jarkom.csui/2106752180/short]
    [HTTP request 1/1]
    [Response in frame: 78]
```

IP Address of Source and Destination:



```
∨ Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 492
     Identification: 0x0063 (99)
   > 010. .... = Flags: 0x2, Don't fragment
     ...0 0000 0000 0000 = Fragment Offset: 0
     Time to Live: 128
     Protocol: TCP (6)
     Header Checksum: 0x0000 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 10.0.0.26
     Destination Address: 10.0.0.27
```

Explanation:

Partially, yes and no. HTTP protocol doesn’t include the origin of the request, but they do include the destination of request in form of FQDN. Meanwhile, Network layer does show the source and destination IP address but network layer is not considered HTTP protocol.

The HTTP protocol itself does not include information about MAC addresses. MAC addresses are part of the lower-level network protocols, such as Ethernet, which operate at the data link layer of the networking stack. MAC addresses are used for communication between devices within a local network.

Basically, MAC Address and source address doesn’t directly visible on HTTP protocol, but in other protocol in HTTP request, it has the supporting data there.

[15 points] TCP Connection Life Cycle

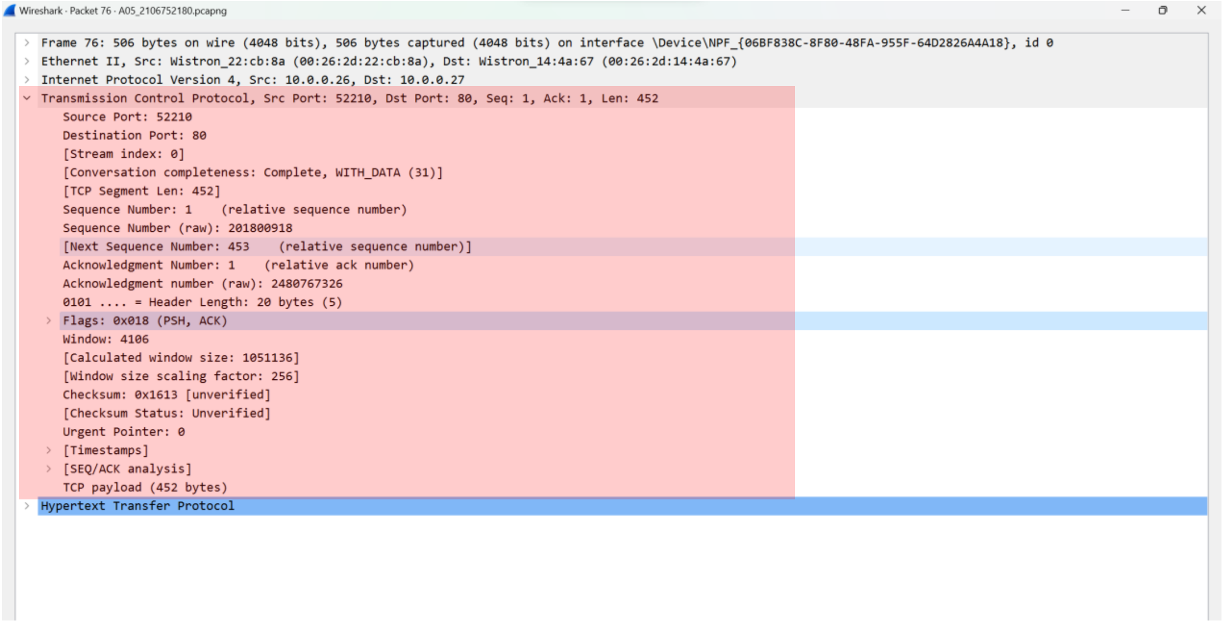
1. **[10 points]** Identify all frames that form the mechanism for establishing and disbanding the TCP connection in the web page request! Fill in the following table based on the information that you obtain from the identification (add additional rows as needed)!

Frame Number	TCP Flag	Role
65	SYN	The client initiates the TCP connection by sending a SYN packet (synchronize) to the server.
66	SYN, ACK	The server responds with a SYN-ACK packet (synchronize-acknowledge) to acknowledge the client's request and

		establish the connection.
67	ACK	The client sends an ACK packet (acknowledge) to confirm the server's response and complete the three-way handshake.
97	FIN, ACK	The client sends a FIN packet, indicating its intention to terminate the connection.
98	ACK	The server acknowledges the receipt of the client's FIN packet.
100	FIN, ACK	The server also sends a FIN packet, indicating its intention to terminate the connection.
101	ACK	The client acknowledges the receipt of the server's FIN packet, and the connection is fully terminated.

2. **[5 points]** Does the TCP include information regarding the origin and destination of the request? Explain specifically the type of information loaded, for example MAC Address, FQDN, and others!

Screenshot:



Wireshark · Packet 76 · A05_2106752180.pcapng

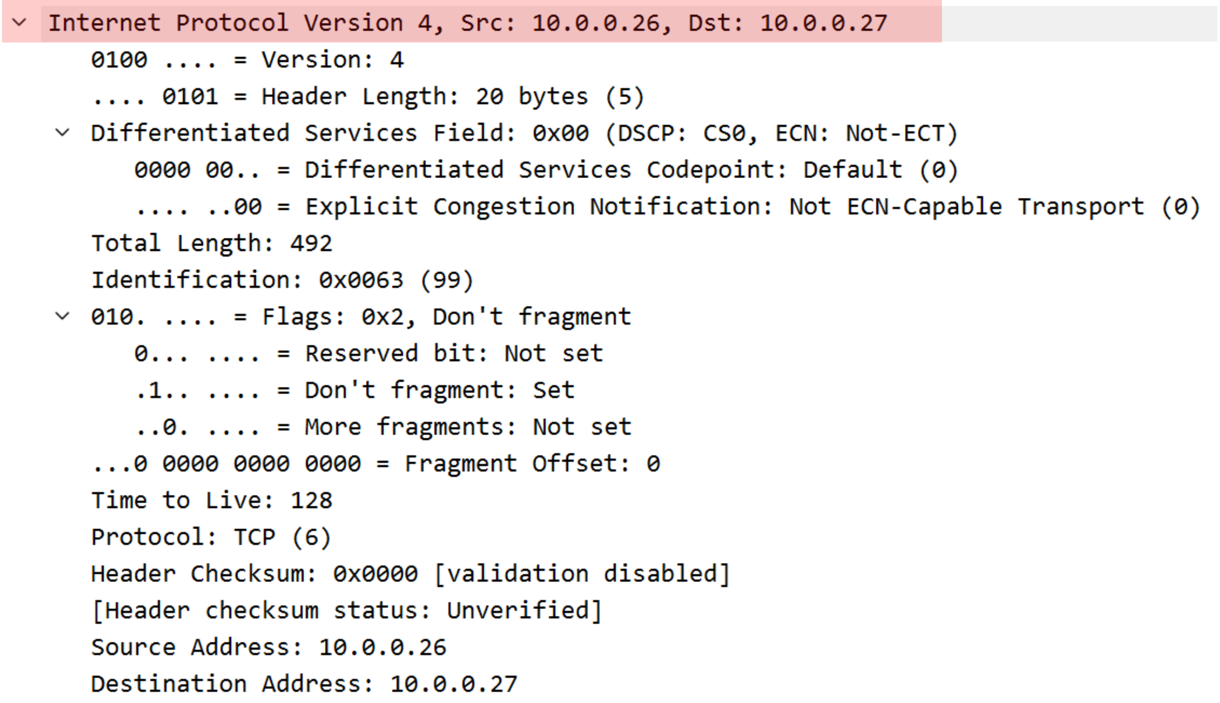
```
> Frame 76: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface \Device\NPF_{06BF838C-8F80-48FA-955F-64D2826A4A18}, id 0
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14:4a:67)
> Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
v Transmission Control Protocol, Src Port: 52210, Dst Port: 80, Seq: 1, Ack: 1, Len: 452
     Source Port: 52210
     Destination Port: 80
     [Stream index: 0]
     [Conversation completeness: Complete, WITH_DATA (31)]
     [TCP Segment Len: 452]
     Sequence Number: 1    (relative sequence number)
     Sequence Number (raw): 201800918
     [Next Sequence Number: 453    (relative sequence number)]
     Acknowledgment Number: 1    (relative ack number)
     Acknowledgment number (raw): 2480767326
     0101 .... = Header Length: 20 bytes (5)
   > Flags: 0x018 (PSH, ACK)
     Window: 4106
     [Calculated window size: 1051136]
     [Window size scaling factor: 256]
     Checksum: 0x1613 [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   > [Timestamps]
   > [SEQ/ACK analysis]
     TCP payload (452 bytes)
> Hypertext Transfer Protocol
```

Explanation:
TCP doesn't gives information about the origin and destination of the request. But, they did give information about source and destination port of the request.

[7 points] The Network Layer

1. **[2 points]** What is the network layer protocol used by the frame that you have selected? Please be specific in the protocol name!

Screenshot:

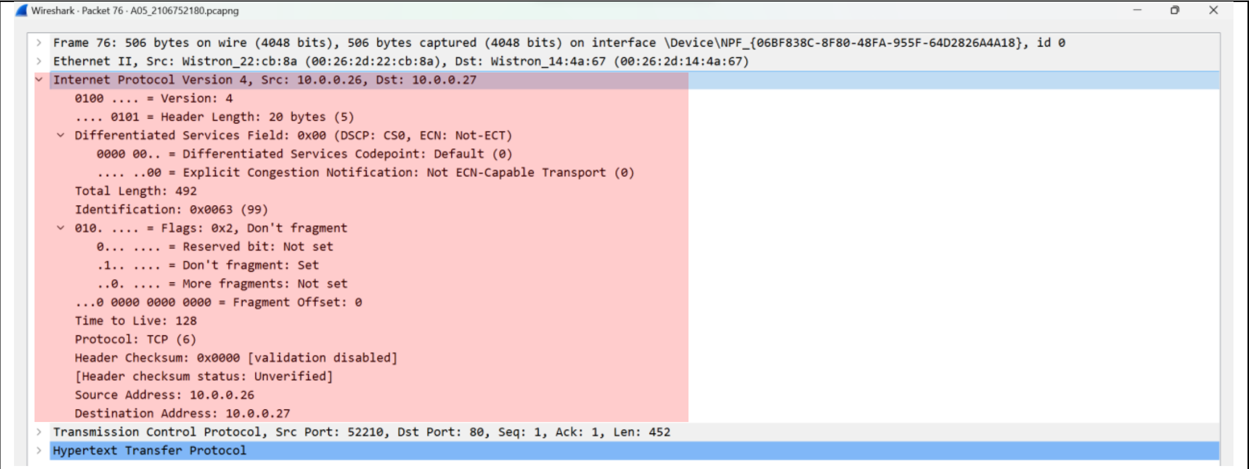


```
⌄ Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
   ⌄  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
         0000 00.. = Differentiated Services Codepoint: Default (0)
         .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
      Total Length: 492
      Identification: 0x0063 (99)
   ⌄  010. .... = Flags: 0x2, Don't fragment
         0... .... = Reserved bit: Not set
         .1.. .... = Don't fragment: Set
         ..0. .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment Offset: 0
      Time to Live: 128
      Protocol: TCP (6)
      Header Checksum: 0x0000 [validation disabled]
      [Header checksum status: Unverified]
      Source Address: 10.0.0.26
      Destination Address: 10.0.0.27
```

Explanation:
Network layer protocol that is used by the frame is Internet Protocol (IP).

2. **[5 points]** Does the Network Layer include information regarding the origin and destination of the request? Explain specifically the type of information loaded, for example MAC Address, FQDN, and others!

Screenshot:



Wireshark · Packet 76 · A05_2106752180.pcapng

```
> Frame 76: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface \Device\NPF_{06BF838C-8F80-48FA-955F-64D2826A4A18}, id 0
> Ethernet II, Src: Wistron_22:cb:8a (00:26:2d:22:cb:8a), Dst: Wistron_14:4a:67 (00:26:2d:14:4a:67)
v Internet Protocol Version 4, Src: 10.0.0.26, Dst: 10.0.0.27
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   v Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
        0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
     Total Length: 492
     Identification: 0x0063 (99)
   v 010. .... = Flags: 0x2, Don't fragment
        0... .... = Reserved bit: Not set
        .1.. .... = Don't fragment: Set
        ..0. .... = More fragments: Not set
     ...0 0000 0000 0000 = Fragment Offset: 0
     Time to Live: 128
     Protocol: TCP (6)
     Header Checksum: 0x0000 [validation disabled]
     [Header checksum status: Unverified]
     Source Address: 10.0.0.26
     Destination Address: 10.0.0.27
> Transmission Control Protocol, Src Port: 52210, Dst Port: 80, Seq: 1, Ack: 1, Len: 452
> Hypertext Transfer Protocol
```

Explanation:

Yes. They give the information regarding the origin and destination of the request. Although, they didn't specify the information about MAC, FQDN, Port, etc. In the screenshot above, you can see that, there is origin and destination IP address of the request.

[20 Points] Synthesis

Based on your observations on this task, create a 150-300 word long narrative, explaining how a web page request can occur and cover all the steps taken, starting from the initial connection that you identified in the second part until the TCP connection is closed after the response is received. Resize the table or add more pages as needed.

Based on the observation above, when a **client** (**IP: 10.0.0.26**) tried to access a webpage, in this case (a4.jarkom.csui), they needed to see the content in the server (if there is any). In order to see the content on the server, we need to use the domain name to find the correct IP address that was mapped to that particular domain name. So when the client access the domain name (a4.jarkom.csui), they perform a DNS protocol that happened on application layer to **DNS Server (10.1.0.2)** available. This DNS server will perform query that mapped the domain name to find the correct IP address. Once the DNS finds it, they will return the IP address of the **server (10.0.0.27)** to the client.

After that, ARP is performed by the client to find the physical (MAC) address of the server because they were located in the local network. After the server replies back, then the communication can be started.

The client then starts a new TCP connection to the server with the sign of **SYN bit to 1**. Afterwards, the server responds back with a packet of SYN-ACK that acknowledges the client request and establishes the connection. Client sends an ACK packet and confirms the server response and completes the three way handshake. Here, the client will send requests to the server and the server will respond back. When the client finishes the request, they send a FIN-ACK to close the connection. Then the server replied back with ACK to acknowledge the FIN packet. Although in my case, there were still packets to be acknowledged, hence the need to have a FIN-ACK packet by the server to terminate the session and an ACK packet replied by the client.