Computer Networking Lab

Assignment No 04.

Date: 25/01/2023

Name: Harshavardhan Anil Bamane

Frame 918: 955 bytes on wire (7640 bits), 955 bytes captured (7640 bits) on interface wlp0s20f3, id 0 Ethernet II, Src: ExtremeN ai:fb:0b (00:04:90:a1:fb:0b), Dst: IntelCor_e9:59:b7 (b0:3c:dc:e9:59:b7) Internet Protocol Version 4, Src: 104.115.39, 42, Dst: 10.40.5.223 Transmission Control Protocol, Src Port: 80, Dst Port: 44346, Seq: 890, Ack: 847, Len: 889

PRN: 22510112

Wireshark Lab: Ethernet

```
pertext Transfer Protocol
HTTP/1.1 260 0KYr\n
Server: nginx\r\n
Content-Type: application/ocsp-response\r\n
Content-Length: 503\r\n
ETag: "4DB425892520EB11ACF7A14927B83B274D788AB3D46B07E45226AAB8E6CFF80E"\r\n
Content-Length: 503\\n\
Frag: "AD82589252DEB11ACF7A14927B83B274D788AB3D46B07E45226AAB8E6CFF80E'
Last-Modified: Wed, 31 Jan 2024 08:54:00 UTC\r\n
Cache-Control: public, no-transform, must-revalidate, max-age=16461\r\n
Expires: Thu, 01 Feb 2024 15:45:05 GMT\r\n
Date: Thu, 01 Feb 2024 11:10:44 GMT\r\n
Connection: keep-alive\r\n
                                                                                                                                                                                                                               Wireshark · Packet 916 · wlp0s20f3
        Frame 916: 489 bytes on wire (3912 bits), 489 bytes captured (3912 bits) on interface wlp0s20f3, id 0
   Ethernet II, Src: IntelCor_e9:59:b7 (b0:3c:dc:e9:59:b7), Dst: ExtremeN_a1:fb:0b (00:04:96:a1:fb:0b)

> Destination: ExtremeN_a1:fb:0b (00:04:96:a1:fb:0b)

> Source: IntelCor_e9:59:b7 (b0:3c:dc:e9:59:b7)

Type: IPv4 (0x0800)
     Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 10.40.5.223, Dst: 104.115.39.42
Transmission Control Protocol, Src Port: 44346, Dst Port: 80, Seq: 424, Ack: 890, Len: 423
Hypertext Transfer Protocol

POST / HTTP/1.1\r\n

[Expert Info (Chat/Sequence): POST / HTTP/1.1\r\n]
Request MRthod: POST
Request URI: /
Request Version: HTTP/1.1
Host: r3.o.lencr.org\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:122.0) Gecko/20100101 Firefox/122.0\r\n
Accept: */*\r\n
            ent: Moz illa/5.0

(X11; U buntu; L

inux x86 _64; rv:

122.0) G ecko/201

00101 Fi refox/12

2.0 - Acc ept: */*

- Accept - Languag

e: en-US ,en;q=0.

5. - Accep t- Encodi
                                                                                                   74 20 45
2c 20 64
6e 74 2d
74 69 6f
74 0d 0a
68 3a 20
6e 3a 20
72 61 67
                                                                                                                                                   6c 61 74
70 65 3a
6f 63 73
6e 74 65
0d 0a 43
65 70 2d
                                                                                                                                                                                       ng: gzip , deflat
e-Conte nt-Type:
applica tion/ocs
p-reques t-Conte
nt-Lengt h: 85-C
onnectio n: keep-
alive-P ragma: n
                                                                                                                                 54 79
6e 2f
43 6f
38 35
6b 65
6d 61
                                                  43 6f
                                                43 6f 6e 74 65
70 6c 69 63 61
65 71 75 65 73
4c 65 6e 67 74
65 63 74 69 6f
76 65 0d 0a 50
                                                                                                                                                   70 65
6f 63
6e 74
0d 0a
65 70
3a 20
```

- 1. What is the 48-bit Ethernet address of your computer?
 - The 48-bit Ethernet address (MAC address) of my computer is b0:3c:dc:e9:59:b7.
- 2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of www.google.com?

• The 48-bit destination address in the Ethernet frame is 00:04:96:a1:fb:0b. This is not the Ethernet address of www.google.com.

3. What device has this as its Ethernet address?

• The device with the MAC address 00:04:96:a1:fb:0b is identified as ExtremeN_a1:fb:0b.

4. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

- The hexadecimal value for the two-byte Frame type field is 0x0800, which corresponds to the upper-layer protocol IPv4.
- 5. How many bytes from the very start of the Ethernet frame does the ASCII "G" in "GET" appear in the Ethernet frame? Next, answer the following questions, based on the contents of the Ethernet frame containing the first byte of the HTTP response message.
 - The ASCII "P" in "POST" appears 34 bytes from the very start of the Ethernet frame containing the HTTP response message.
- 5. What is the value of the Ethernet source address? Is this the address of your computer, or of www.google.com(Hint: the answer is no). What device has this as its Ethernet address?
 - Ethernet source address is 00:04:96:a1:fb:0b. This is not the address of your computer, nor is it the address of www.google.com.

6. What is the destination address in the Ethernet frame? Is this the Ethernet address of your computer?

- The destination address in the Ethernet frame is b0:3c:dc:e9:59:b7.
- wlp0s20f3 (Wi-Fi interface): MAC Address: b0:3c:dc:e9:59:b7

AS BOTH ARE THE SAME, IT IS THE ADDRESS OF MY COMPUTER!

7. Give the hexadecimal value for the two-byte Frame type field. What upper layer protocol does this correspond to?

 The hexadecimal value for the two-byte Frame type field is 0x0800. This corresponds to the upper-layer protocol IPv4. In Ethernet frames, the Frame type field indicates the type of the payload encapsulated within the frame. In this case, 0x0800 indicates that the payload is an IPv4 packet.

8. How many bytes from the very start of the Ethernet frame does the ASCII "O" in "OK" (i.e., the HTTP response code) appear in the Ethernet frame?

• To determine the position of the ASCII "O" in "OK" within the Ethernet frame, we need to examine the payload of the Ethernet frame containing the HTTP response message. Based on the provided output, the payload of the Ethernet frame starts at byte offset 0x46.

Scanning through the payload, we find the ASCII "O" in "OK" at byte offset 0x47 within the payload. However, to determine the byte offset from the very start of the Ethernet frame, we need to add the byte offset of the payload within the frame to the byte offset of the "O" within the payload. The byte offset of the payload within the Ethernet frame is 0x46, and the byte offset of the "O" within the payload is 0x47. Therefore, the total byte offset from the very start of the Ethernet frame to the ASCII "O" in "OK" is 0x46 + 0x47 = 0x8D. So, the ASCII "O" in "OK" appears 141 bytes from the very start of the Ethernet frame.