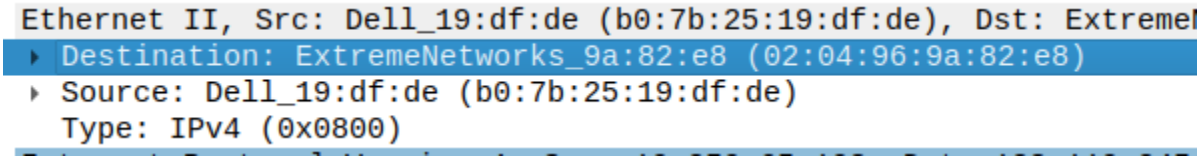
CS315: Assignment-11

Name: Shahil Patel
Roll No.: 200010039

Part-1: Capturing and analyzing Ethernet frames

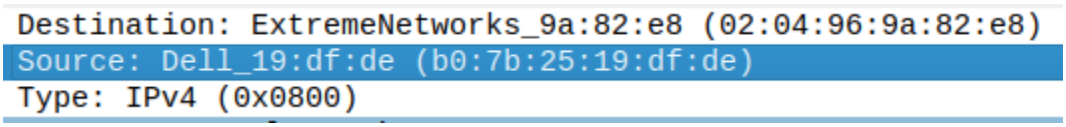
1. What is the 48-bit Ethernet address of your computer?

Ans: The 48-bit Ethernet address of computer is: **b0:7b:25:19:df:de**



2. What is the 48-bit destination address in the Ethernet frame? Is this the Ethernet address of gaia.cs.umass.edu? (Hint: the answer is *no*). What device has this as its Ethernet address?

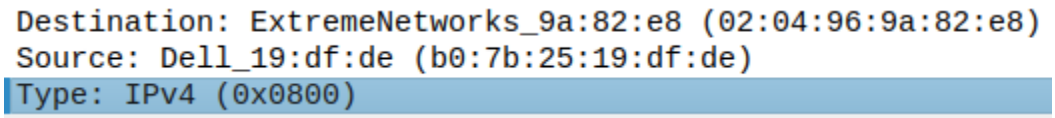
Ans: The 48-bit destination address is: **02:04:96:9a:82:e8**



No, This ethernet address belongs to the **router** (*Gateway to the Internet*).

3. What is the hexadecimal value for the two-byte Frame type field in the Ethernet frame carrying the HTTP GET request? What upper layer protocol does this correspond to?

Ans: The hexadecimal value for the two-byte Frame type field in the Ethernet frame is: **0x0800.** It corresponds to the **IPv4** layer protocol.



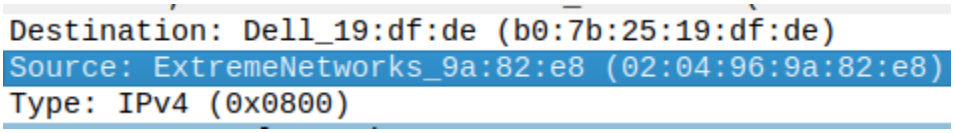
4. How many bytes from the very start of the Ethernet frame does the ASCII “G” in “GET” appear in the Ethernet frame? Do not count any preamble bits in your count, i.e., assume that the Ethernet frame begins with the Ethernet frame's destination address.

Ans: The ASCII “G” in GET appears **67** times in the Ethernet frame.

5. What is the value of the Ethernet source address? Is this the address of your computer, or of gaia.cs.umass.edu (Hint: the answer is *no*). What device has this as its Ethernet address?

Ans: The Ethernet source address is **02:04:96:9a:82:e8.**

No, this is the address of the **router** (*Gateway to the Internet*).



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

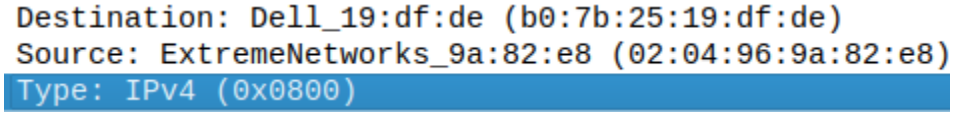
Ans: The destination address in the Ethernet Frame is: **b0:7b:25:19:df:de.**

Yes, this is the Ethernet address of our computer.

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

Ans: The hexadecimal value for the two-byte Frame type field is: **0x0800.**

The upper layer protocol is: **IPv4**



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? Do not count any preamble bits in your count, i.e., assume that the Ethernet frame begins with the Ethernet frame's destination address.

Ans: The ASCII “O” in “OK” appears in the **79** bytes from the very start of the Ethernet frame.

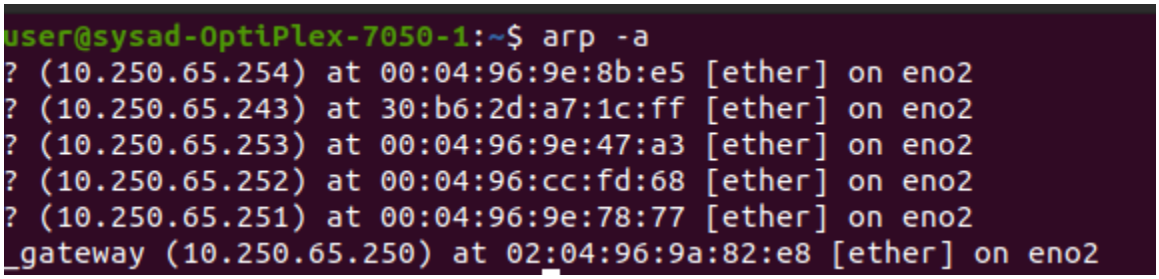
9. How many Ethernet frames (each containing an IP datagram, each containing a TCP segment) carry data that is part of the complete HTTP “OK 200 ...” reply message?

Ans: **4** Ethernet frames carry the data that is part of the complete HTTP “OK 200…” reply message.

Part-2: The Address Resolution Protocol

1. How many entries are stored in your ARP cache?

Ans: No. of entries in ARP cache = **5**

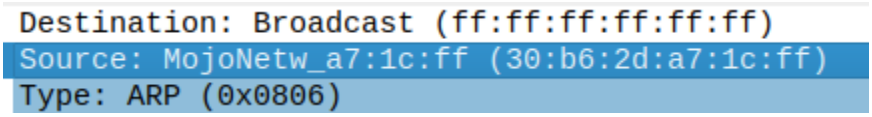


2. What is contained in each displayed entry of the ARP cache?

Ans: The ARP cache contains entries that map **IP addresses** to **MAC addresses**. A static ARP table contains entries that are user-configured.

3. What is the hexadecimal value of the source address in the Ethernet frame containing the ARP request message sent out by your computer?

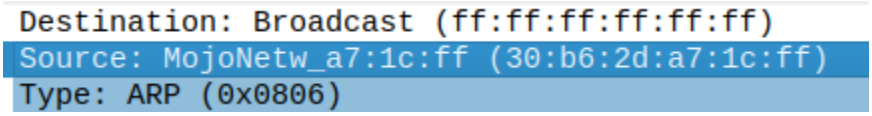
Ans: The hexadecimal value of the source address in the Ethernet frame is: **30:b6:2d:a7:1c:ff.**



4. What is the hexadecimal value of the destination addresses in the Ethernet frame containing the ARP request message sent out by your computer? And what device(if any) corresponds to that address (e.g,, client, server, router, switch or otherwise...)?

Ans: The hexadecimal value of the destination addresses in the Ethernet The frame containing the ARP request message sent out by our computer is: **ff:ff:ff:ff:ff:ff.**

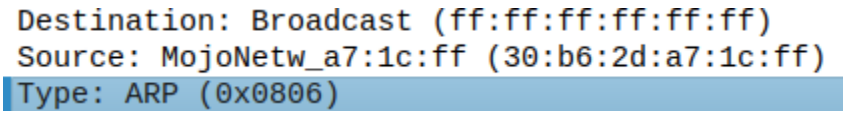
This is the **Ethernet address** of the **router** (*Gateway to the Internet*).



5. What is the hexadecimal value for the two-byte Ethernet Frame *type* field? What upper layer protocol does this correspond to?

Ans: The hexadecimal value for the two-byte Ethernet frame type field is: **ARP (0x0806).**

The upper layer protocol is: **ARP.**



6. How many bytes from the very beginning of the Ethernet frame does the ARP *opcode* field begin?

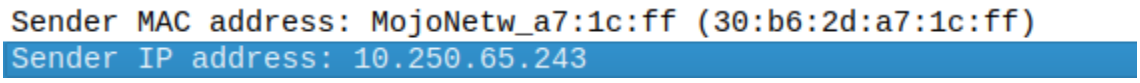
Ans: The ARP opcode begins from **20** byte from the beginning of the Ethernet frame.

7. What is the value of the *opcode* field within the ARP request message sent by your computer?

Ans: The value of the opcode filed within the ARP request message sent by our computer is: **request (1)**

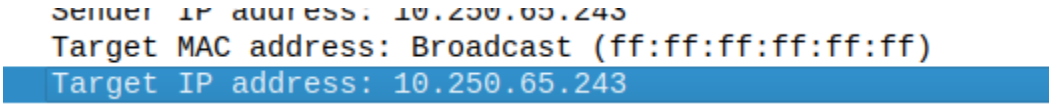
8. Does the ARP request message contain the IP address of the sender? If the answer is yes, what is that value?

Ans: Yes, the ARP request message contains the IP address of the sender which is: **10.250.65.243**



9. What is the IP address of the device whose corresponding Ethernet address is being requested in the ARP request message sent by your computer?

Ans: The IP address of the device whose corresponding Ethernet address is being requested in the ARP request message is: **10.250.65.243**

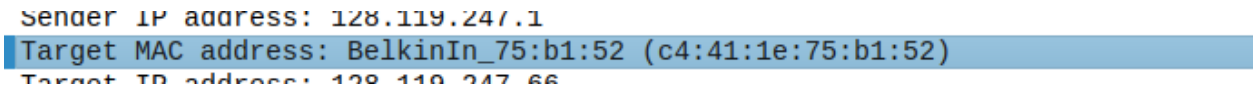


10. What is the value of the *opcode* field within the ARP reply message received by your computer?

Ans: The value of the opcode filed with the ARP reply message is: **reply (2)**

11. *Finally (!),* let's look at the **answer** to the ARP request message! What is the Ethernet address corresponding to the IP address that was specified in the ARP request message sent by your computer?

Ans: The Ethernet address corresponding to the IP address that was specified in the ARP request message sent by our computer is **c4:41:1e:75:b1:52.**



12. We've looked at the ARP request message sent by your computer running Wireshark, and the ARP reply message sent in response. But there are other devices in this network that are also sending ARP request messages that you can find in the trace. Why are there no ARP replies in your trace that are sent in response to these other ARP request messages?

Ans: We are not able to see the responses that are being sent to other ARP requests because even though the ARP requests are broadcasted but the replies to their corresponding requests aren't broadcasted instead it is directly *sent to the device's Ethernet Address requesting it*. Hence, we will be only seeing the response for our request and not the responses for other requests.