Practical Exercises 1

Surname: Hejhal
Name: Jakub

Degree (Ing. del Software/Ing. Informática/Ing. Computadores): Erasmsus student

PC Label in the lab (e.g. 012): personal laptop

Exercise 1. Find an HTTP packet in the packet trace (the first that your computer sent with a GET message) and complete the following schema with the information contained in this packet (make screenshots where these data appear).

MAC address information of your computer.

MAC Address: 40 a3 cc 2a a6 d9

NIC Manufacturer: Intel Corporate

NIC serial number: 2a a6 d9

MAC address information of gateway/router.

MAC Address: 02 10 18 37 91 9c

NIC Manufacturer: not found (https://aruljohn.com/mac/021018) , my router probably uses random mac address after each reboot for security reasons

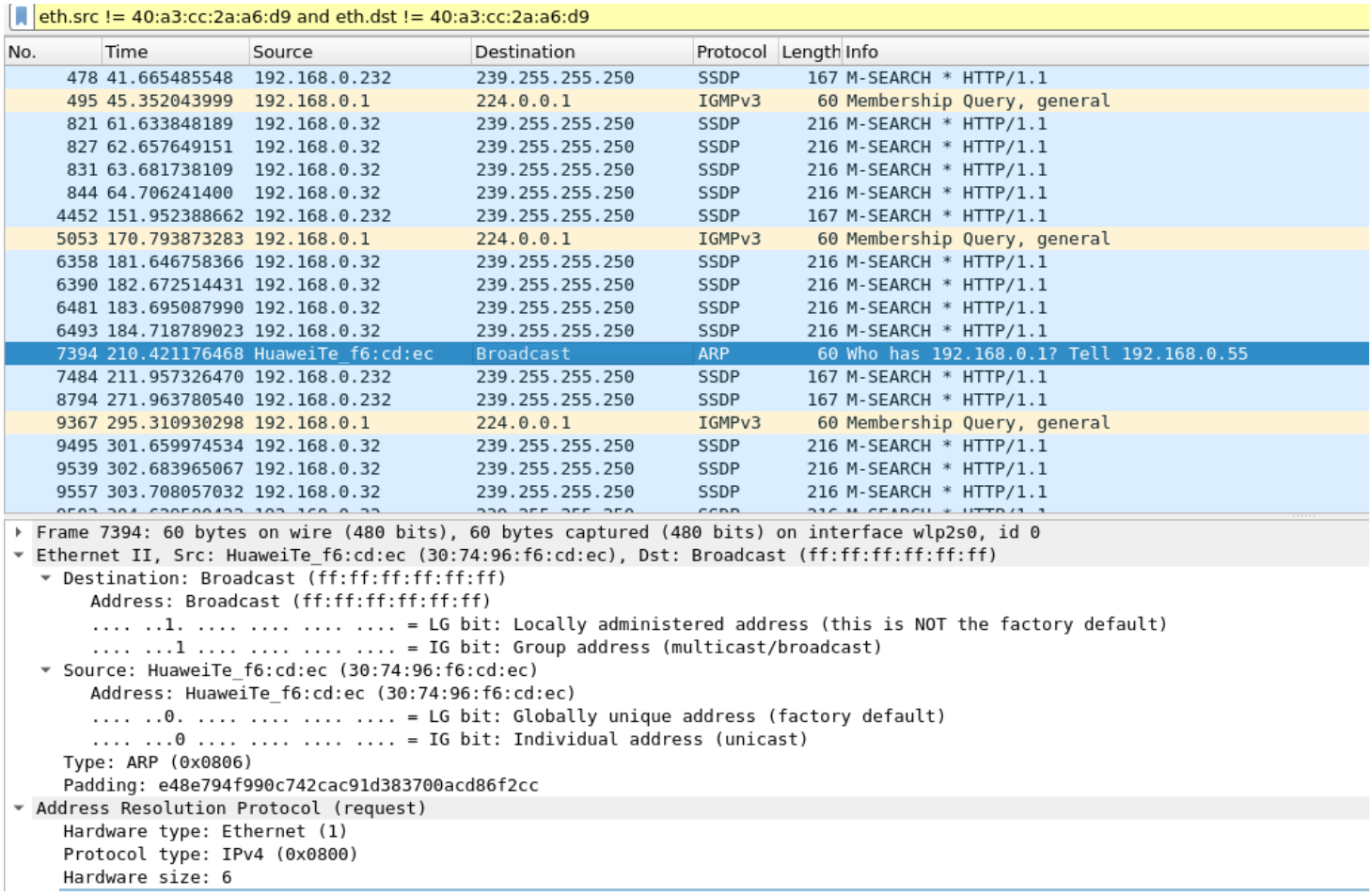
NIC serial number: 37 91 9c

Exercise 2. Which filter do you use to show all frames where your MAC address is not used? Why do you receive these frames? (To answer this question, observe the features of the destination MAC addresses used by these frames)

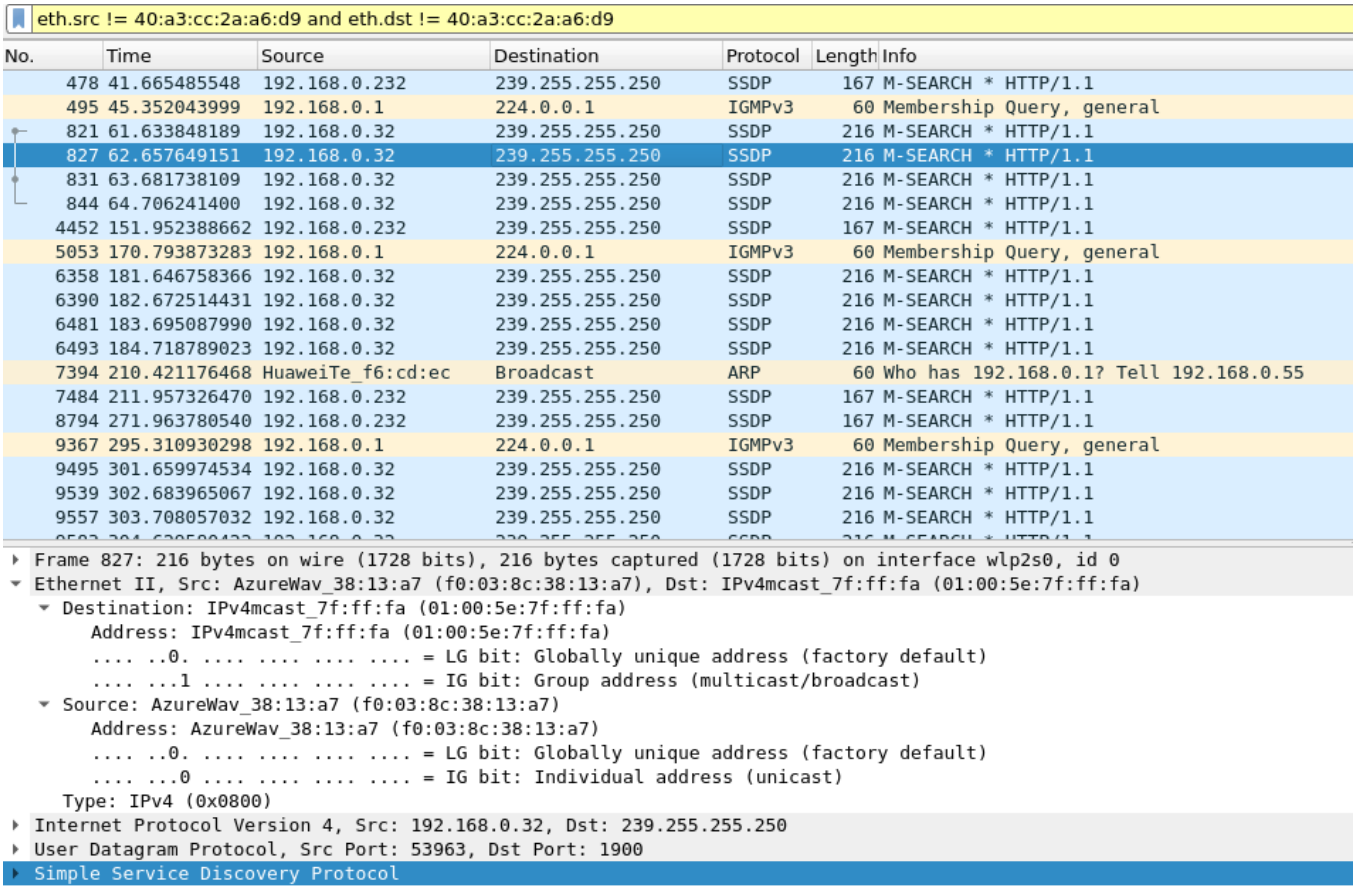
Answer:

My MAC is 40:a3:cc:2a:a6:d9. Therefore the wireshark filter looks like this:

```
eth.src != 40:a3:cc:2a:a6:d9 and eth.dst != 40:a3:cc:2a:a6:d9
```



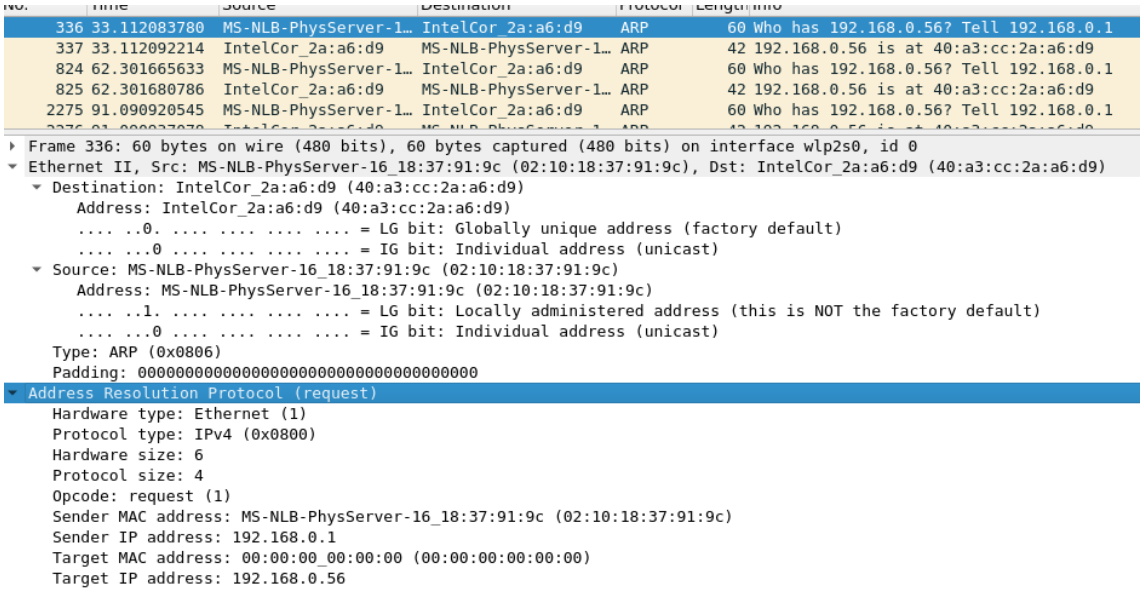
I receive a lot of different frames, whose destination and source MAC is different from mine. The previous screenshot shows ARP request from my phone on the same WIFI. It's destination MAC is Broadcast, so it makes sense I receive this frame.

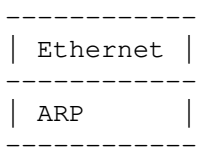


On the other hand, I receive a lot of frames, whose Source and Destination address is not mine MAC and neither Broadcast MAC, as can be seen on the previous screenshot. This is because their Destination MAC is ethernet multicast adddress.

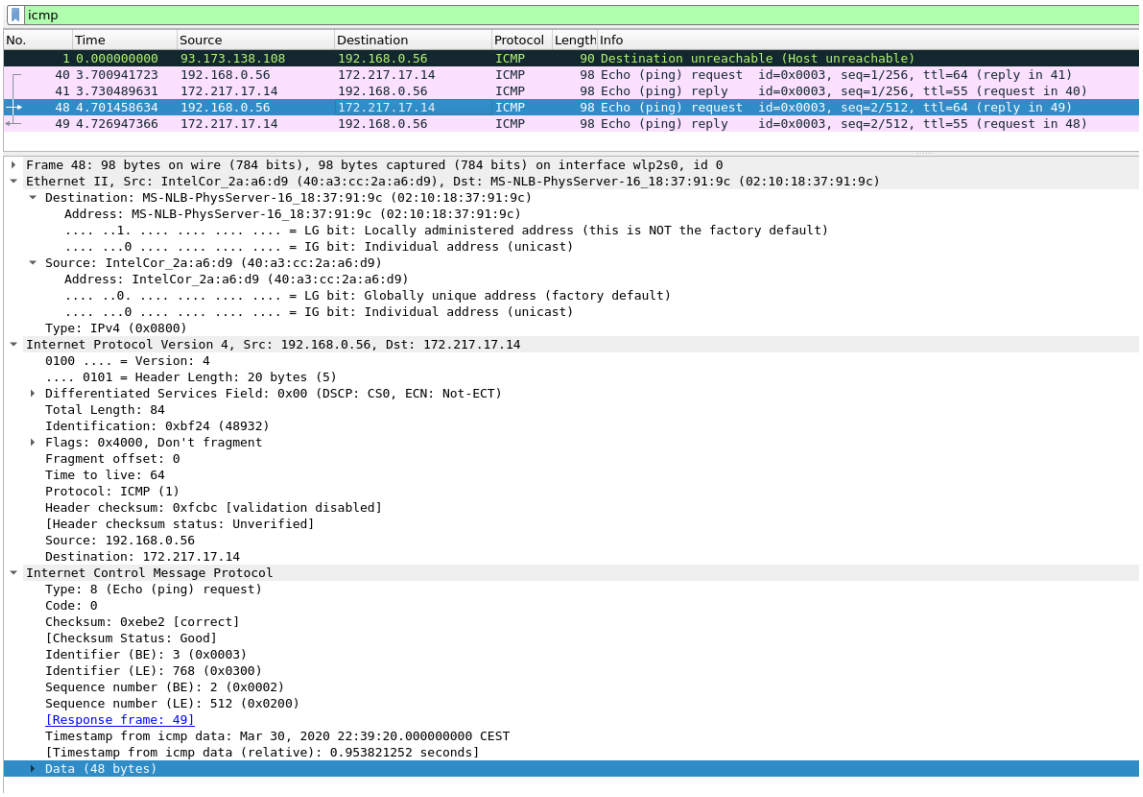
Exercise 3. Draw the protocol stack and the packet encapsulation that corresponds to each of the following packets: ARP, ICMP, DNS and HTTP packets. Include a screenshot with the information you have used to draw each protocol stack (i.e., the center part of the screen that shows the content of the selected packet).

ARP



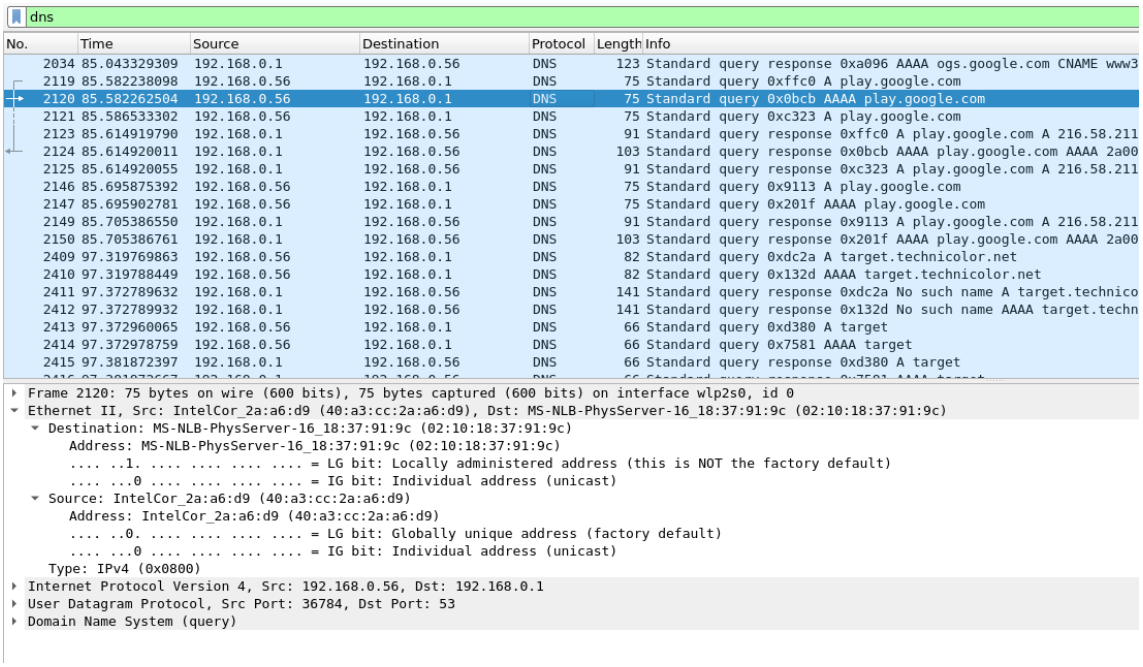


ICMP



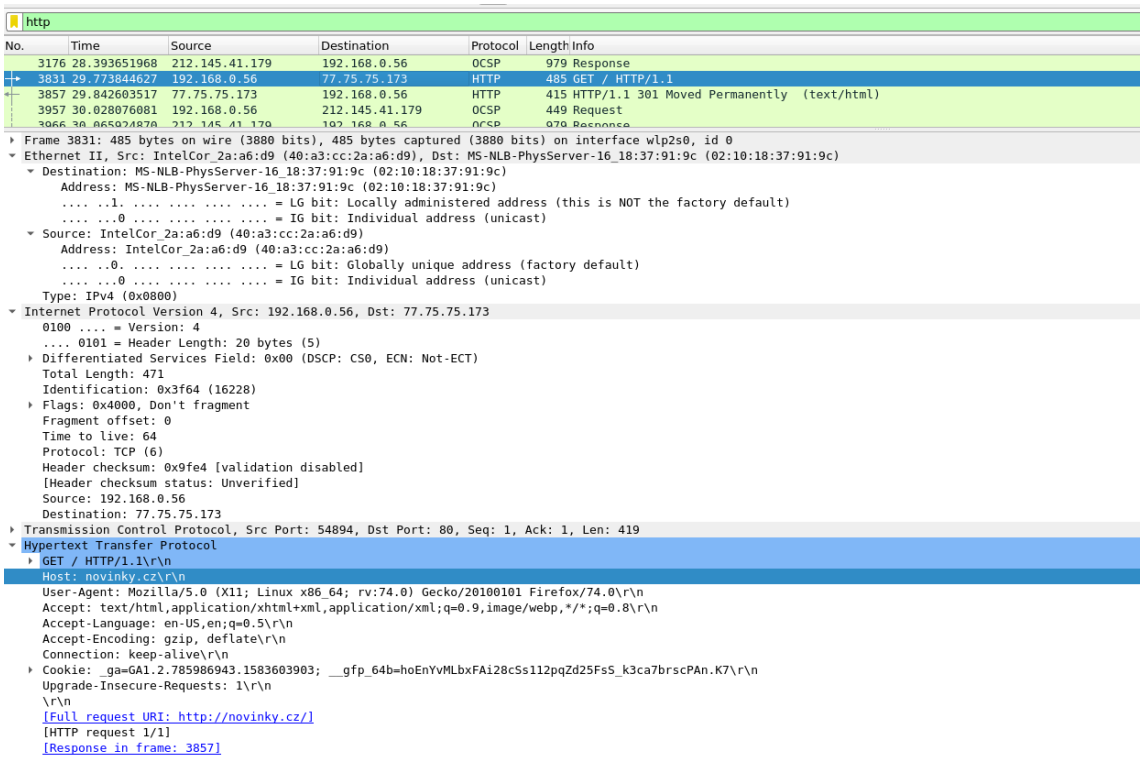

```
------------
| Ethernet |    Link layer
------------
| IPv4     |    Network layer
------------
| ICMP     |    Transport layer
------------
```

DNS




```
------------
| Ethernet |   Link layer
------------
| IPv4     |   Network layer
------------
| UDP      |   Transport layer
------------
| DNS      |   Application layer
------------
```

HTTP

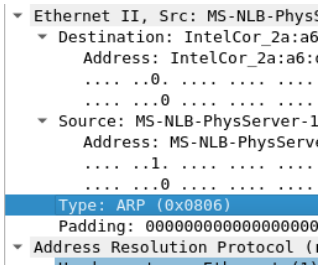



```
-------------
| Ethernet  |   Link layer
-------------
| IPv4      |   Network layer
-------------
| TCP       |   Transport layer
-------------
| HTTP      |   Application layer
-------------
```

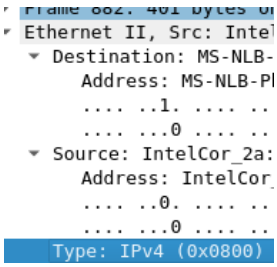
Exercise 4. Observe carefully the Ethernet II **type** field in the obtained trace and make a screenshot for each protocol in the table below. What is the utility of this field? Why different frames contain the same type value? After answering these questions fill the following table with the number that appears for the type field.

	Tipo en la cabecera Ethernet II
ARP	0x0806
HTTP	0x0800
ICMP	0x0800
DNS	0x0800

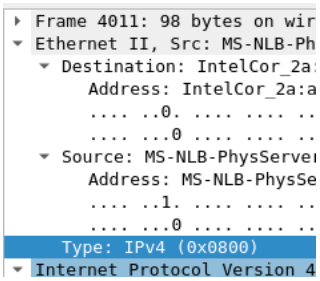
ARP



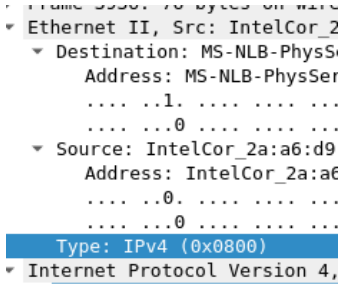
HTTP



ICMP



DNS

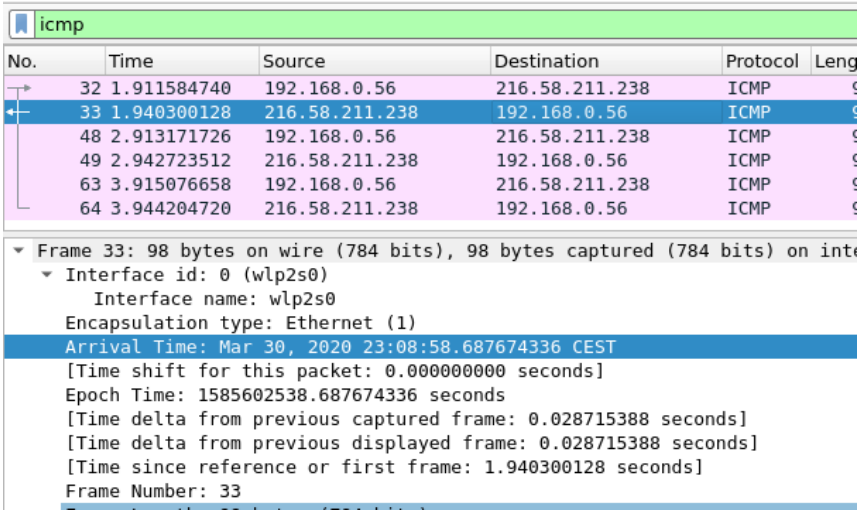


EtherType is a two-octet field in an Ethernet frame. It is used to indicate which protocol is encapsulated in the payload of the frame. ICMP, HTTP and DNS have the same EtherType field, because all of those protocols use IPv4 protocol at network OSI layer.

Exercise 5. Pay attention to the difference between the times of the ICMP request and its answer. How much time is it? What is the concept of the theory that corresponds to this time? The ping command returns some lines with the following format:

```
Respuesta desde 150.214.57.60: bytes=32 tiempo=45ms TTL=50
```

Observe times obtained by the ping command and compare them with the times that appear in the Wireshark trace. Are times obtained by the ping command consistent with times that appear in the Wireshark trace?

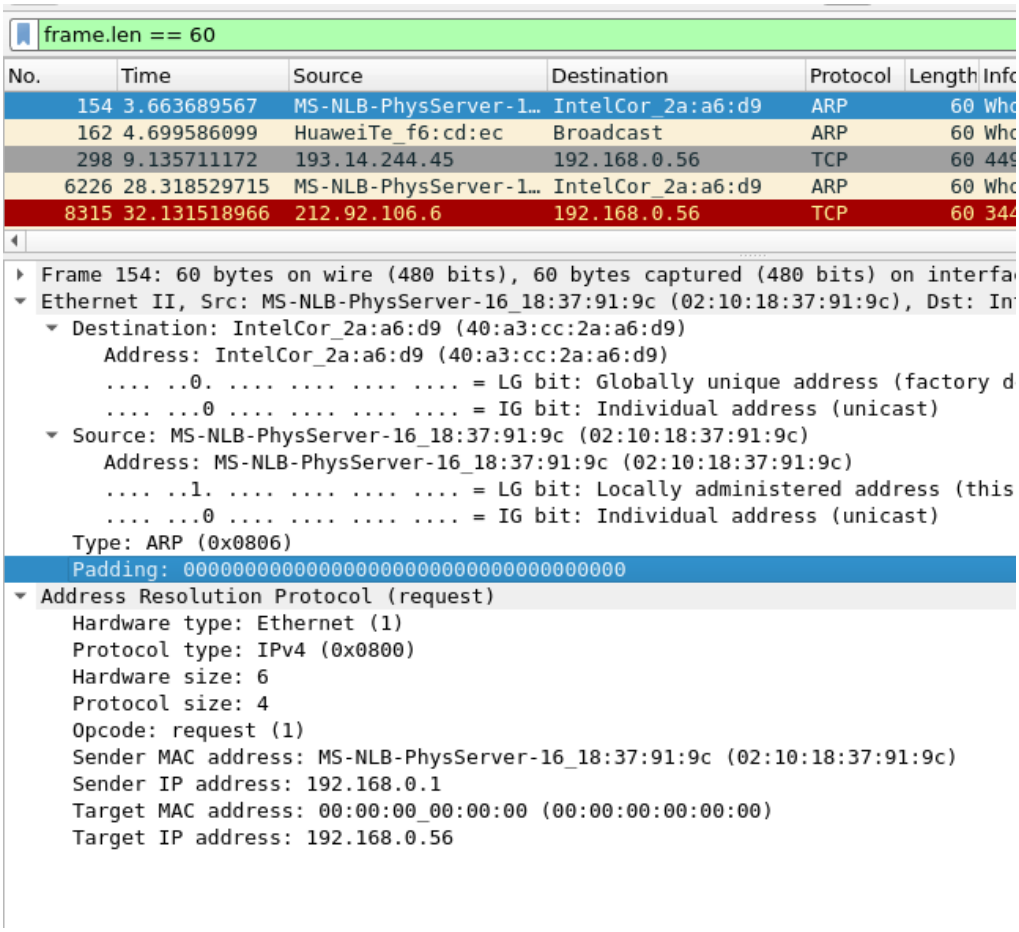


Time difference between ICMP request and ICMP response frames (Time delta from previous displayed frame field) is 0.028715388 seconds, which is 28.71 ms. This time is called round trip time.

```
kubik@terminator ~/l/networking (master)> ping google.com
PING google.com (216.58.211.238) 56(84) bytes of data.
64 bytes from mad01s24-in-f14.1e100.net (216.58.211.238): icmp_seq=1 ttl=55 time=28.7 ms
```

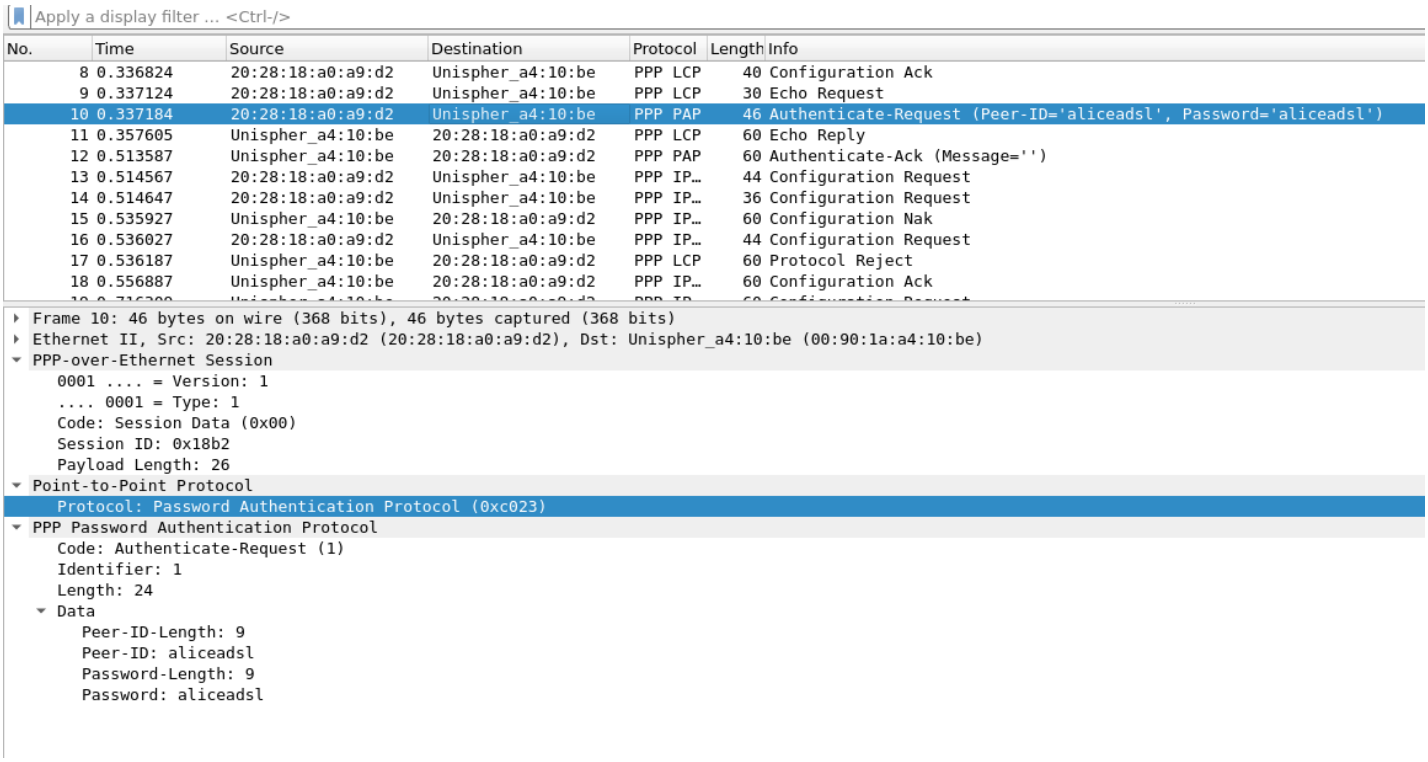
This round trip time value is consistent with the output of ping command.

Exercise 6. According to what we have seen in theory classes, Ethernet frames must have a minimum length of 64 bytes (46 bytes of data). On the other hand, Wireshark does not show the CRC field (it is processed by the NIC), so the length of the showed frame will be equal or greater than 60 bytes. Look for a 60 bytes size frame (define the filter: frame.len == 60). What is the mechanism used to fulfil the length limit constraint if the size of the data to be transmitted is lower than 46 bytes?



We can see, that the rest of the Ethernet frame is padded by zeros.

Exercise 7. What is the mechanism used for the authentication in the traffic captured? What are the frames that negotiate the use of this field?



PPP uses Password Authentication Protocol (PAP) for client authentication. We can see 2 PAP frames.
1) PAP Authenticate-Request with Peer-ID: aliceadsl and Password: aliceadsl
2) Server response PAP frame - Authenticate Acknowledgement

Exercise 8. In the trace, it is possible to see the process that corresponds to the phases of stablishing, authenticating and networking. Take screenshots of frames that corresponds to these phases indicating the phase that correspond in each case. What is the protocol at the network level that will be used to transmit the data?

Establishing PPP connection:

5	0.133822	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP LCP	36 Configuration Request
6	0.336644	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP LCP	60 Configuration Request
7	0.336664	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP LCP	60 Configuration Ack
8	0.336824	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP LCP	40 Configuration Ack

PPP Authentication:

10	0.337184	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP PAP	46 Authenticate-Request (Peer-ID='aliceadsl', Password='aliceadsl')
12	0.513587	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP PAP	60 Authenticate-Ack (Message='')

Networking:
In the traffic captured, we can see only configuration of IPv4 protocol over PPP, there is no useful IPv4 traffic encapsulated.

Here are the frames that correspond to configuration of IPv4 addresses and client primary and secondary DNS servers.

13	0.514567	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP IPCP	44 Configuration Request
14	0.514647	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP IPV6CP	36 Configuration Request
15	0.535927	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP IPCP	60 Configuration Nak
16	0.536027	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP IPCP	44 Configuration Request
17	0.536187	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP LCP	60 Protocol Reject
18	0.556887	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP IPCP	60 Configuration Ack
19	0.716309	Unispher_a4:10:be	20:28:18:a0:a9:d2	PPP IPCP	60 Configuration Request
20	0.716449	20:28:18:a0:a9:d2	Unispher_a4:10:be	PPP IPCP	32 Configuration Ack

```
    Code: Session Data (0x00)
    Session ID: 0x18b2
    Payload Length: 24
▾ Point-to-Point Protocol
    Protocol: Internet Protocol Control Protocol (0x8021)
▾ PPP IP Control Protocol
    Code: Configuration Ack (2)
    Identifier: 2 (0x02)
    Length: 22
  ▾ Options: (18 bytes), IP Address, Primary DNS Server IP Address, Secondary DNS Server IP Address
    ▾ IP Address
        Type: IP Address (3)
        Length: 6
        IP Address: 79.51.70.114
    ▾ Primary DNS Server IP Address
        Type: Primary DNS Server IP Address (129)
        Length: 6
        Primary DNS Address: 85.37.17.41
    ▾ Secondary DNS Server IP Address
        Type: Secondary DNS Server IP Address (131)
        Length: 6
        Secondary DNS Address: 85.38.28.83
```

Exercise 9. Develop a program in Java that list all the network interfaces of our computer that have a MAC address assigned using the **NetworkInterface** class. In addition, the program should provide the following information about the interface: short name of the interface, MAC, if the MAC is globally or locally managed and if the interface is up and running. The output should be as follows:

```
wlan1: 68:07:15:1F:99:A2 – global – (down).
wlan2: 6A:07:15:1F:99:A1 – local – (up).
eth2: D8:CB:8A:F5:8B:4E – global – (up).
wlan3: 68:07:15:1F:99:A1 – global – (down).
```

The code is self-explanatory, with the help of some comments.

Guidelines of the report

- It is recommended the use of the provided Word template.
- Include in the same report all the exercises corresponding to Part 1 (exercises 1, 2 and 3).
- The front page must inform about: (i) list of the exercises included; (ii) data that identifies the student (i.e. name, group, etc). Please **use the template** provided in the Campus Virtual
- Start each exercise on a separate page.
- For each exercise include both the statement of the exercise as well as the solution. Define custom styles for the different parts of the document (e.g. title, statement, solution, etc.)
- In the screenshots (using <alt>+<impr pant>) the student must mark those parts corresponding to what is requested by the exercise (using a drawing utility). Also add a brief text explaining what is shown.
- The student must upload the report in PDF.
- Upload a .zip file with the report and the file .pcap with the trace of the captured packets.
- The report should include an explicative schema detailing the most significant parts of the code. You must deliver the source code (in independent files) jointly with the report.