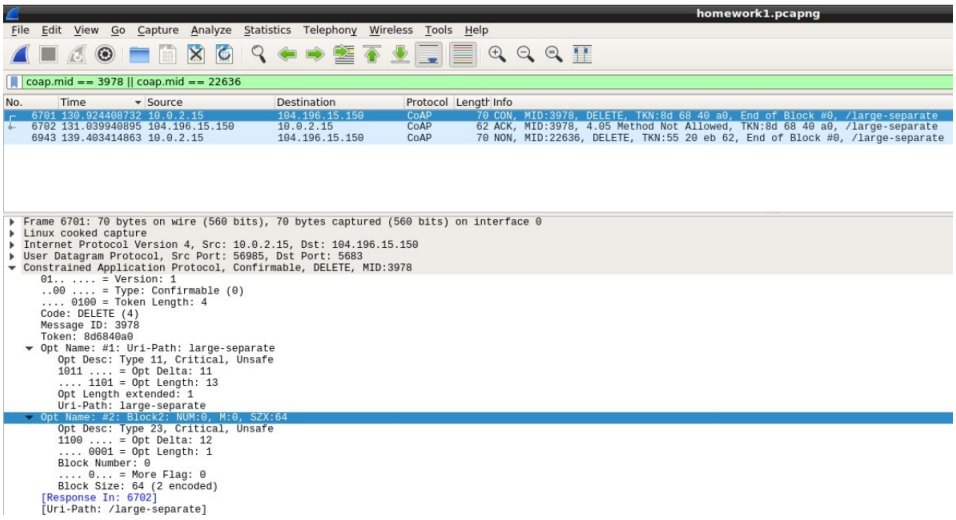
#01: CoAP-MQTT Sniffing

Andrea Calici, C. Persona 10490117, Matricola 944717

1) What's the difference between the message with MID: 3978 and the one with MID: 22636?

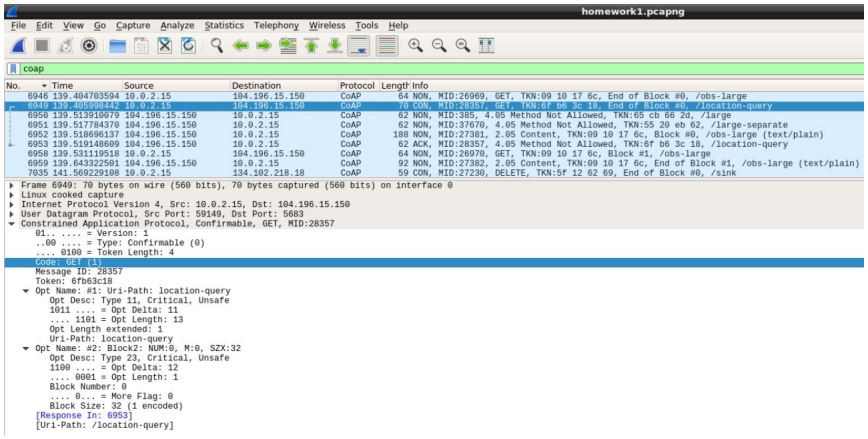
The message with MID = 3978 is a **Confirmable (CON)** message (if it doesn't receive an ACK, it will restransmit after a certain timeout), the message with MID = 22636 is a **Non Confirmable (NON-CON)** message. It's possibile to find them with the following filter:

```
coap.mid == 3978 || coap.mid == 22636
```



2) Does the client receive the response of message No. 6949?

Yes, it receives the response in the message No. 6953 with an ACK that contains a **Method Not Allowed** error. It can be found simply by looking the message No. 6949 without using any particular filter.



3) How many replies of type confirmable and result code "Content" are received by the server "localhost"?
If we look for **ACK** messages that acknowledge **CON** messages and that contain the code **Content**, we can use the filter:

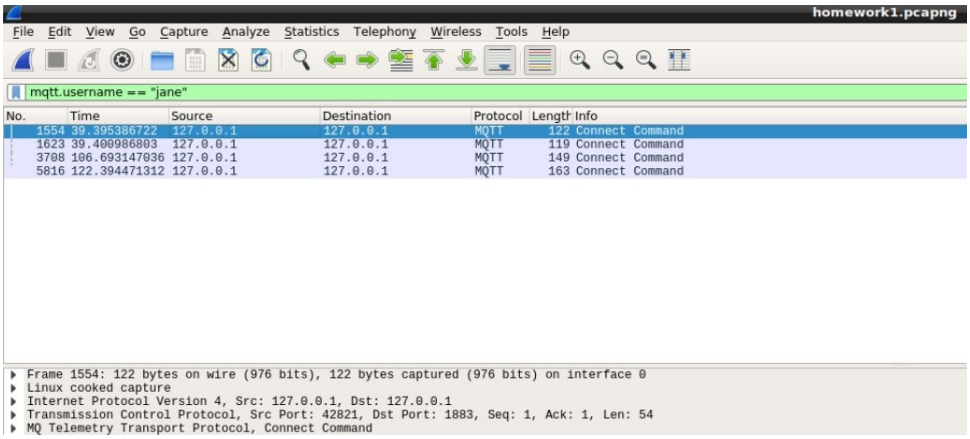
$$\text{coap.code} == 69 \;\&\&\; \text{ip.dst} == 127.0.0.1$$

and find 8 messages that satisfy the request.

4) How many messages containing the topic "factory/ department*/+" are published by a client with user name: "jane"? Where * replaces the dep. number, e.g. factory/department1/+, factory/department2/+ and so on.
First of all we can use the filter:

```
mqtt.username == "jane"
```

to highlight all the **Connect** messages sent by a client with this username.



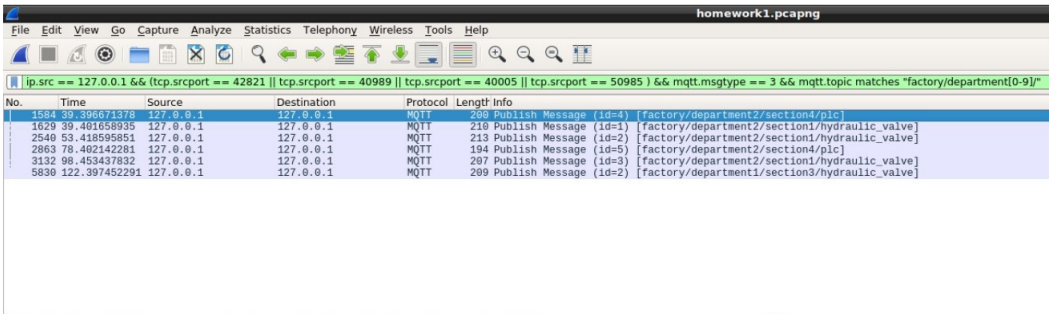
We can find the source ports of this client looking at the TCP part of the 4 messages: 42821, 40989, 40005 and 50985. Then using the following filter:

```
ip.src == 127.0.0.1 && (tcp.srcport == 42821 || tcp.srcport ==
40989 || tcp.srcport == 40005 || tcp.srcport == 50985 ) &&
mqtt.msgtype == 3 && mqtt.topic matches
"factory/department[0-9]/"
```

it's possible to find that all the 6 messages doesn't match the request because are in the form:

```
factory/department[0-9]/+/+
```

so the answer is 0.

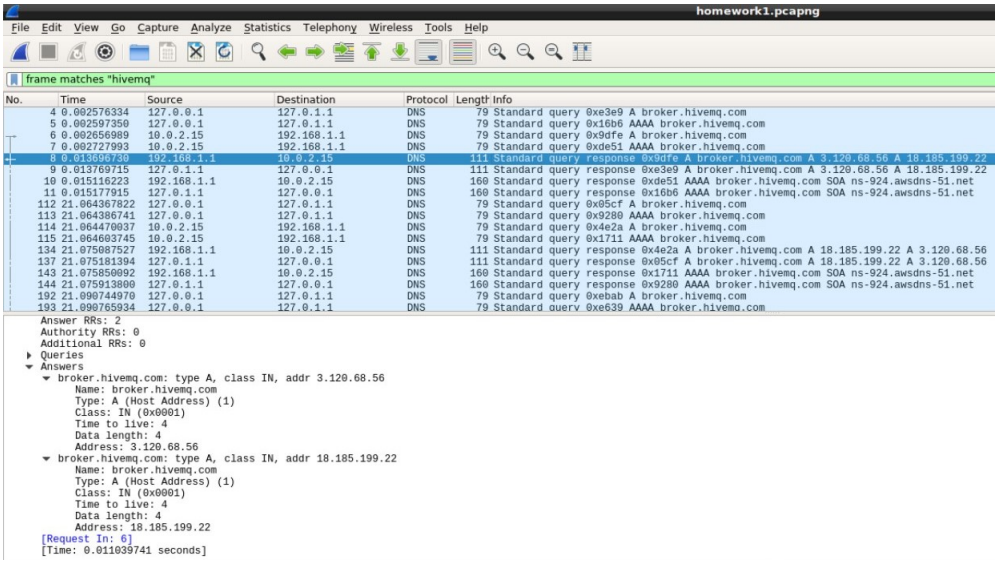


5) How many clients connected to the broker "hivemq" have specified a will message?

We can use the filter

```
frame matches "hivemq"
```

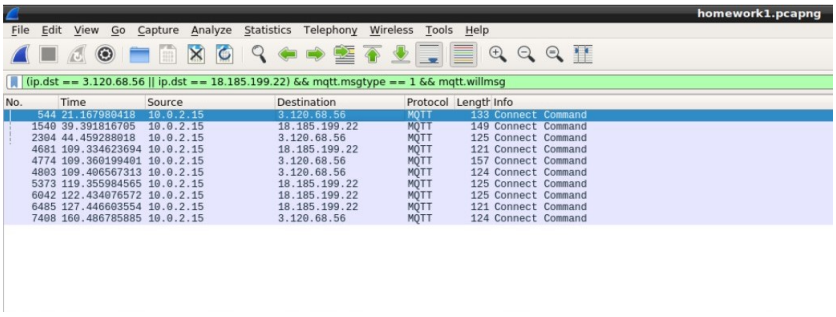
in order to highlight the IP addresses of the broker: 3.120.68.56 and 18.185.199.22.



We can then use the filter

```
(ip.dst == 3.120.68.56 || ip.dst == 18.185.199.22) &&
mqtt.msgtype == 1 && mqtt.willmsg
```

to retrieve all the **connect** messages directed to the broker, with the **will flag** true and that provide a **will message**.



There are 10 messages:

- 2 of them have an unique **Client ID** (No. 1540 and No. 4774);

- 2 of them have the same **Client ID**, but different **TCP port** (No. 4083 and No. 7408);
- the last 6 don't provide a **Client ID**, but have the **Clean Session Flag**.

So there are at least 3 different clients and at maximum 9 (10 if we consider different the clients with the same ClientID, but different TCP port).

6) How many publishes with QoS 1 don't receive the ACK?
It's possible to interpeter the question in 2 different ways:

1) If we consider all the published messages with QoS 1 sent by the clients and the broker that satisfy the filter:

```
mqtt.msgtype == 3 && mqtt.qos == 1
```

are 124. The **PUBACK** are 74, due to:

```
mqtt.msgtype == 4
```

If we count also the duplicates with the filter:

```
mqtt.msgtype == 3 && mqtt.dupflag == 1
```

there are 2 messages. It's possible to find 124 - 74 - 2 = 48 messages.

2) Otherwise if we consider only the messages sent by the clients we find that 77 of them are sent to the broker. If we use the filter:

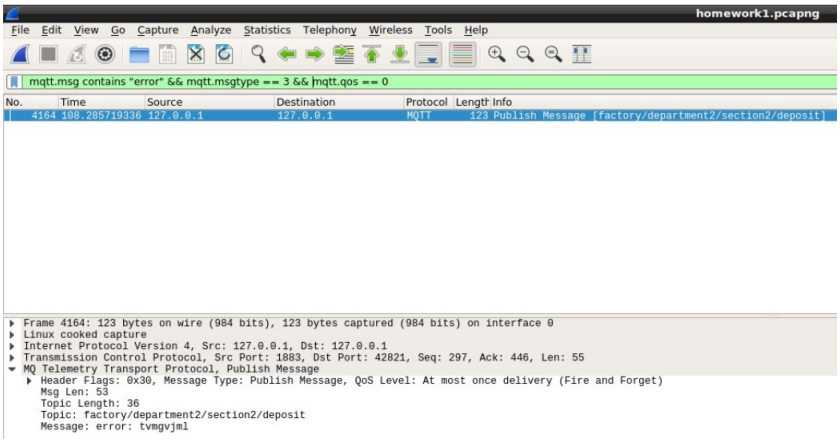
```
mqtt.msgtype == 4
```

there are 74 **PUBACK** messages. So only 3 messages with QoS 1 don't receive the ack.

7) How many last will messages with QoS set to 0 are actually delivered?
We can see that all the **Last Will messages** contain the string 'error'. Then it's possibile to use the filter:

```
mqtt.msg contains "error" && mqtt.msgtype == 3 && mqtt.qos == 0
```

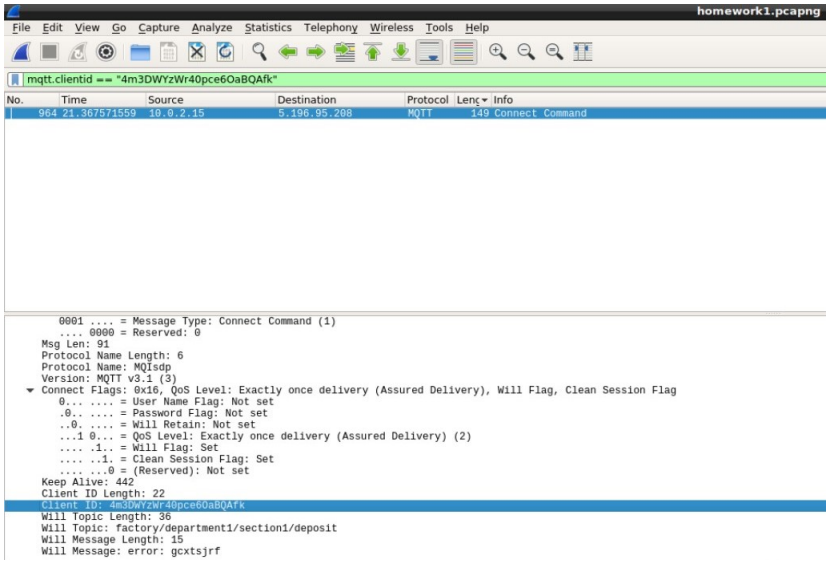
to find a single message that satisfy the query: No. 4164.



8) Are all the messages with QoS > 0 published by the client "4m3DWYzWr40pce6OaBQAfk" correctly delivered to the subscribers?
Using the filter:

```
mqtt.clientid == "4m3DWYzWr40pce6OaBQAfk"
```

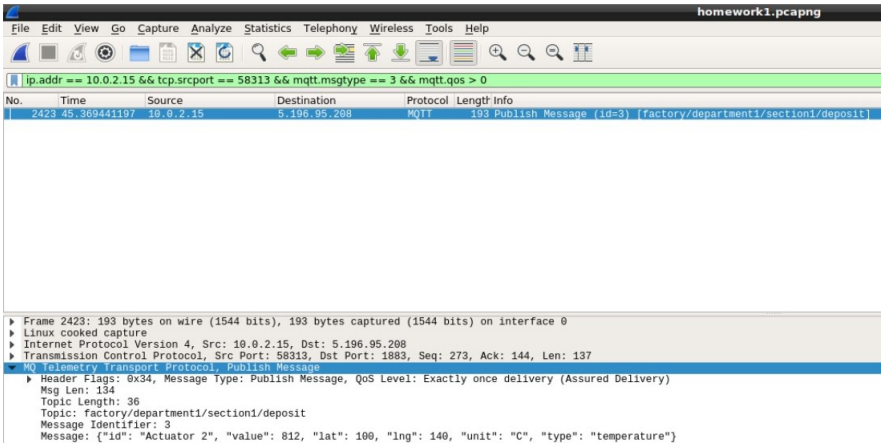
it's possible to discover the **IP address** (10.0.2.15) and the **TCP port** (58313) of the requested client.



Then, with the filter:

```
ip.addr == 10.0.2.15 && tcp.srcport == 58313 && mqtt.msgtype ==
3 && mqtt.qos > 0
```

it's possible to find a single message delivered to the broker with IP address 5.196.95.208.



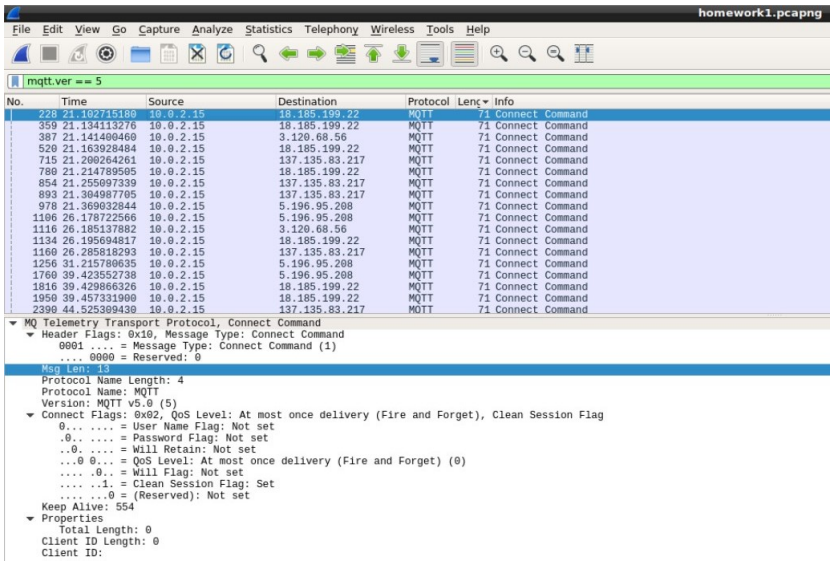
Then, we can check if this message was correctly delivered by the broker to the subscribers, with the filter:

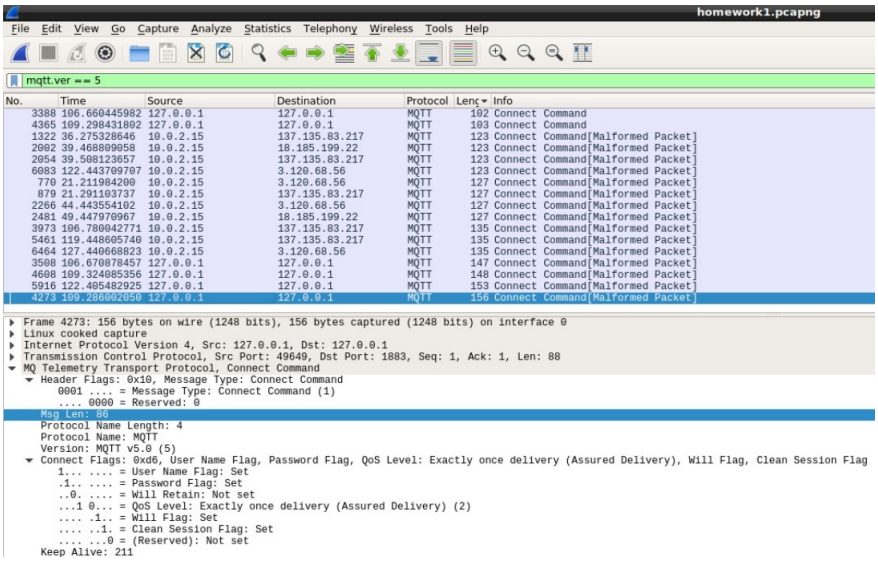
```
mqtt.msg == '{"id": "Actuator 2", "value": 812, "lat": 100,
"lng": 140, "unit": "C", "type": "temperature"}' && ip.src ==
5.196.95.208 && mqtt.msgtype == 3
```

and find that there are no messages that satify the request.

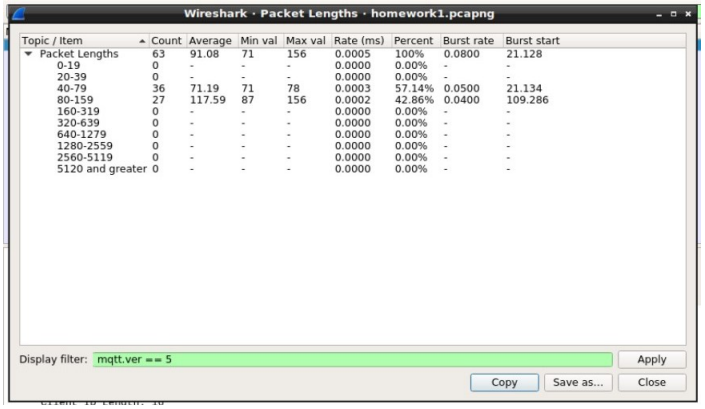
9) What is the average message length of a connect msg using mqttv5 protocol? Why messages have different size?
There are 63 connect messages using **mqttv5** protocol. We can distinguish between 2 cases (message length and packets length):

Message Length: they range from 13 to 86 because some of them add more properties (User Name, Password, ClientID...) and others are malformed. The length of the majority is 13. If we calculate the average message length including also the malformed packets we obtain 30.22. If we exclude the malformed packets we obtain 16.93.





Packet Lenght: if the question requests the average **packet length** it's possible to find it in the *Statistics/Packet Lengths* panel and it's equals to 91.08.



Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	63	91.08	71	156	0.0005	100%	0.0800	21.128
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	36	71.19	71	78	0.0003	57.14%	0.0500	21.134
80-159	27	117.59	87	156	0.0002	42.86%	0.0400	109.286
160-319	0	-	-	-	0.0000	0.00%	-	-
320-639	0	-	-	-	0.0000	0.00%	-	-
640-1279	0	-	-	-	0.0000	0.00%	-	-
1280-2559	0	-	-	-	0.0000	0.00%	-	-
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120 and greater	0	-	-	-	0.0000	0.00%	-	-

10) Why there aren't any REQ/RESP pings in the pcap?

If there are no REQ/RESP pings it means that all the interactions between the clients and the broker occurred before the expiration of the keep alive time, or the broker disconnect the client due to the connection drop.