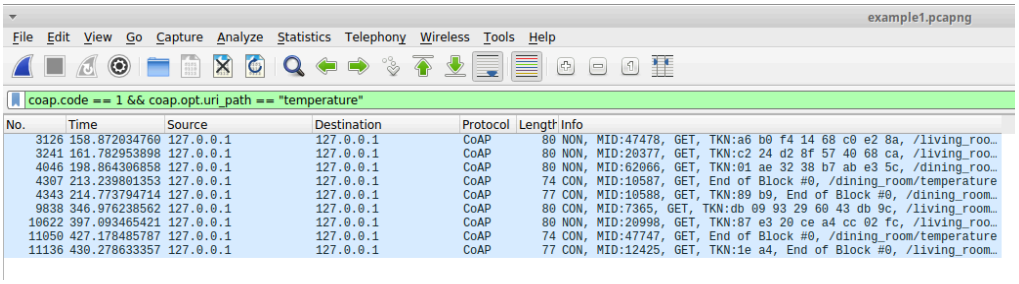
POLITECNICO MILANO 1863

CoAP and MQTT Traffic Analysis

The purpose of this challenge was to analyze the traffic of different communication protocols using Wireshark. In that way, as it is shown below, the implementation of filters was crucial to obtain the answer of the questions present in the current challenge.

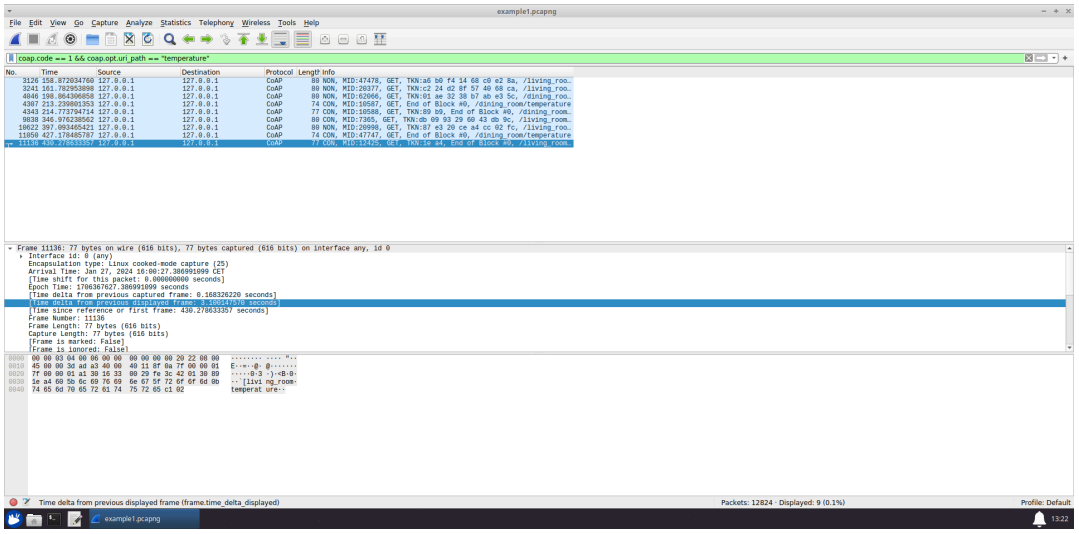
1a. How many different CoAP clients sent a GET request to a temperature resource (.../temperature)?

Answer: After applying the filter *coap.code == 1 && coap.opt.uri_path == "temperature"*, generally found in the 8 packets sent from source to destination, which are the same IP so there is only **1** CoAP Client which sends GET requests to a temperature resource shown in the screen shot .



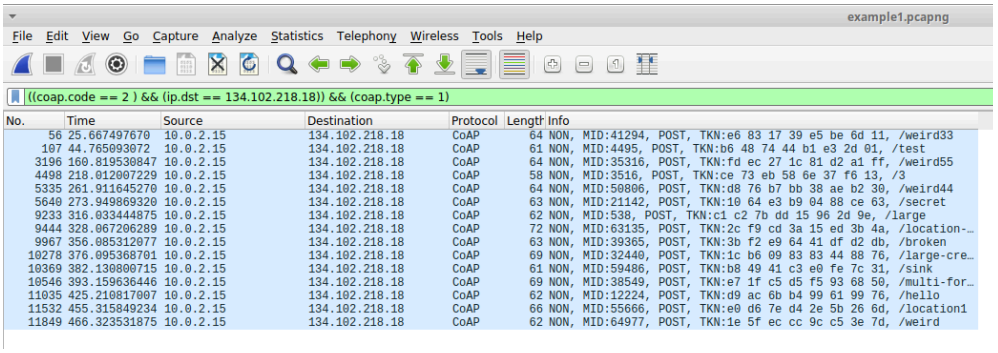
1b. For each of the clients found in 1a), write the MID of the longest CoAP response (any response) received by the client.

Answer: The biggest packet related to the “Time delta from previous capture frame “ which shown in the picture, the MID if the “ time delta from previous capture frame “ is : **12425**



2a. How many CoAP POST requests directed to the "coap.me" server did NOT produce a successful result?

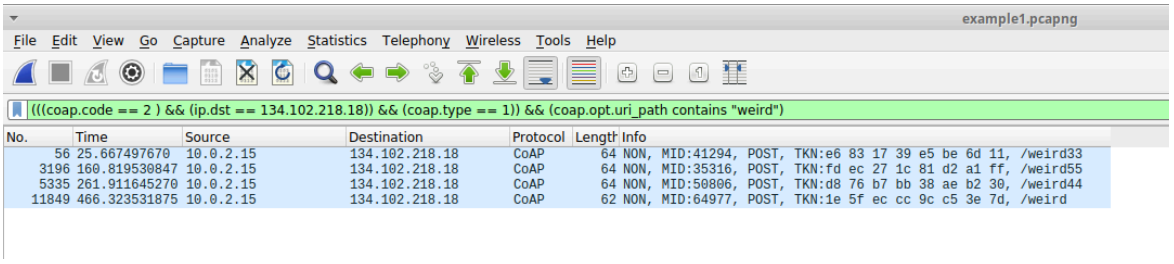
Answer: There are **15** non confirmable requests which satisfies this question, which can be reached with the filter *coap.type == 1,*used to find non-confirmable or not successful requests). In addition, it was implemented the filter *(coap.code == 2) && (ip.dst == 134.102.218.18)) && (coap.type == 1)* to find CoAP POST requests directed to the "coap.me" server that did NOT produce a successful.



2b. How many requests from 2a) are directed to a "weird" resource? (resources like /weirdXX)?

Answer: There are **4** packets with these attributes.

Filter: (((coap.code == 2) && (ip.dst == 134.102.218.18)) && (coap.type == 1)) && (coap.opt.uri_path contains "weird")



3a. How many MQTT Publish messages with qos=2 are RECEIVED by the clients running in the machine capturing the traffic?

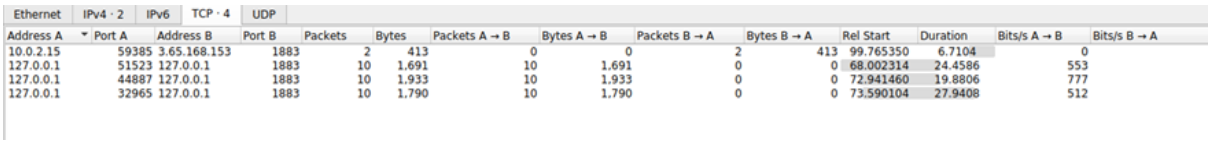
Answer: 32 packets – 9 bad formed (no content packets) = 23 packets

Filter(s) :
mqtt.msgtype == 3 && mqtt.qos == 2 && (ip.dst == 127.0.0.0/8 or ip.dst == 10.0.0.0/8)
mqtt.msgtype == 3 && mqtt.qos == 2 && sll.pkttype == 0

Motivation: Both filters get the same results. The first gets all the messages received by clients which are in the auto-loop domain and in the router domain addresses. There could be other addresses but in this case they are not, so it is useful.
It is checked with another filter, which gets the packets labeled as “send to us”; In the ‘Linux cooked capture v1 / Packet type’ section with the value ‘Unicast to us (0)’. As it was said the same number of packets are obtained.

3b. How many clients are involved in the messages found in 3a)?

Answer: With any of the filters used before look into the *Statistics* > *Conversations* > *UDP* section and check how many different lines are there (because the packets are sent just in one direction of the communication). It is not looked at the IPv4 section because there are clients with the same IP address but using different ports.



Ethernet | IPv4 · 2 | IPv6 | TCP · 4 | UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.15	59385	3.65.168.153	1883	2	413	0	0	2	413	99.765350	6.7104	0	
127.0.0.1	51523	127.0.0.1	1883	10	1.691	10	1.691	0	0	68.002314	24.4586	553	
127.0.0.1	44887	127.0.0.1	1883	10	1.933	10	1.933	0	0	72.941460	19.8806	777	
127.0.0.1	32965	127.0.0.1	1883	10	1.790	10	1.790	0	0	73.590104	27.9408	512	

3c. What are the MQTT Message identifiers (ID) of the subscribe requests that let the client receive the messages found in 3a)?

mqtt.msgtype == 8 -> check manually for the topics we got in the 3a answer.

After implementing the filter mentioned before, it is shown below a table with the topics of the published messages in 3a, where each of them is associated with the topic for which each client is subscribed and its corresponding ID.

Topic Publication Message:	**Topic Subscription Message**	**ID**
hospital/facility2/room4	*hospital/+/+*	*4*
hospital/facility2/section0	*hospital/+/+*	*4*
hospital/facility2/room4/temperature	*hospital/+/#*	*7*
hospital/building5/area2/temperature	*hospital/+/#*	*7*
hospital/department5	*hospital/+*	*6*
hospital/facility2/section0	*hospital/+/+*	*4*
hospital/facility2/room4/temperature	*hospital/+/#*	*7*
hospital/building5/area2/temperature	*hospital/+/#*	*7*
hospital/department5	*hospital/+*	*6*

Topic Publication Message:	**Topic Subscription Message**	**ID**
factory/facility2	*factory/+ || factory/#*	*3 & 1 & 17*
factory/room2/area2	*factory/#*	*1&17 (2 subscriptions for the same topic)*
factory/room2/room4	*factory/#*	*1&17 (2 subscriptions for the same topic)*

Topic Publication Message:	**Topic Subscription Message**	**ID**
factory/building5	*factory/+ || factory/#*	*3 & 1 & 17*

Topic Publication Message:	**Topic Subscription Message**	**ID**
metaverse/facility2/room4	*metaverse/+/+*	*3*
metaverse/facility2/floor1	*metaverse/+/+*	*3*
metaverse/room2	*metaverse/+*	*19*
metaverse/building5	*metaverse/+*	*19*
metaverse/facility2/room4	*metaverse/+/+*	*3*
metaverse/facility2/section0/hydraulic_valve	-	-
metaverse/facility2/area2/pollution	-	-
metaverse/facility2/section0/deposit	-	-
metaverse/building5/room4/deposit	-	-
metaverse/facility2/room4/deposit	-	-

Topic Publication Message:	**Topic Subscription Message**	**ID**
university/room2/area2	-	-
university/facility2/floor1/pollution	-	-
university/department5	-	-
university/building5/area2	-	-

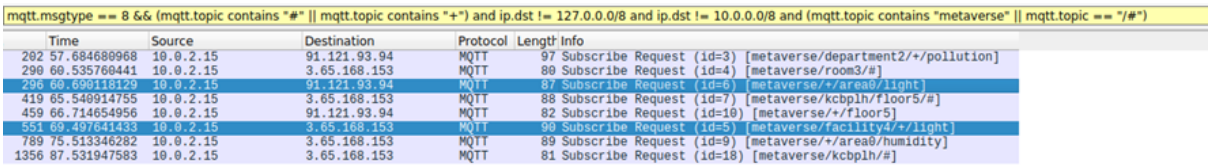
4a. How many MQTT clients sent a subscribe message to a public broker using at least one wildcard?

Filter: mqtt.msgtype == 8 && (mqtt.topic contains "#" || mqtt.topic contains "+") && (ip.dst != 127.0.0.0/8) && (ip.dst != 10.0.0.0/8)

Motivation: Considering messages to a public broker, it is considered public brokers all the addresses that are not the auto-loop and the router interface ones. For that reason, those directions were excluded and checked if those were actually delivered to public brokers.

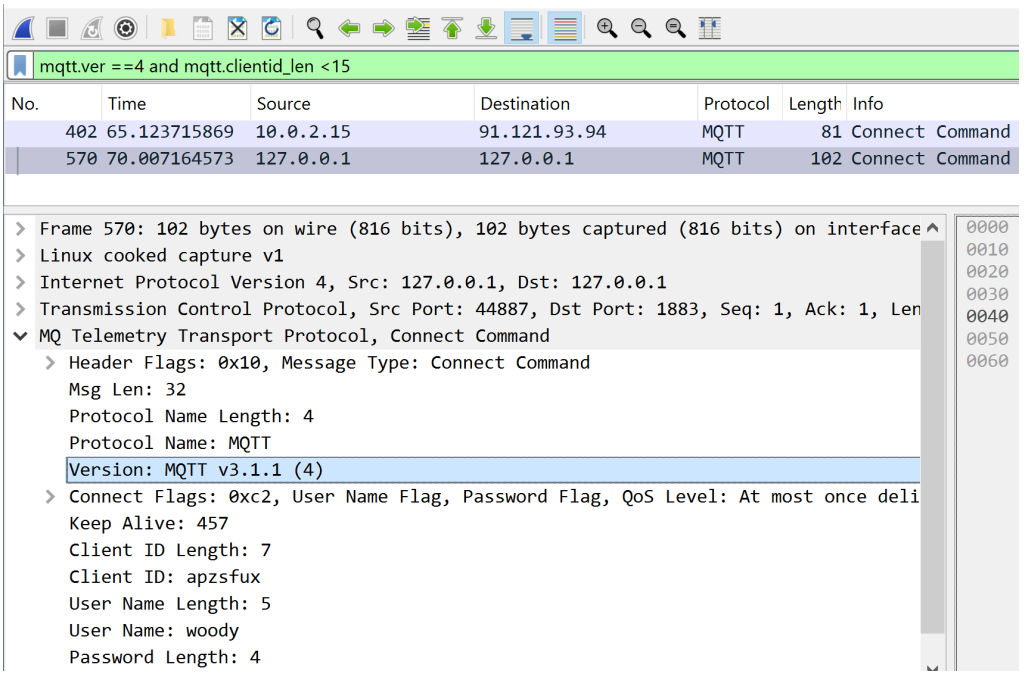
Answer: 16

4b. Considering clients found in 4a), how many of them WOULD receive a publish message directed to the topic: "metaverse/facility4/area0/light“



5. How many MQTT ACK messages in total are received by clients who connected to brokers specifying a client identifier shorter than 15 bytes and using MQTT version 3.1.1?

Answer: In total there are 2 MQTT ACK messages using version 3.1.1 and with a client identifier length shorter than 15 bytes. To find them, the filter *mqtt.ver == 4 and mqtt.clientid_len <15* was used, where the mqtt.ver equal to 4 is corresponding to the 3.1.1 version and the client identifier was discriminated to be less than 15 bytes.

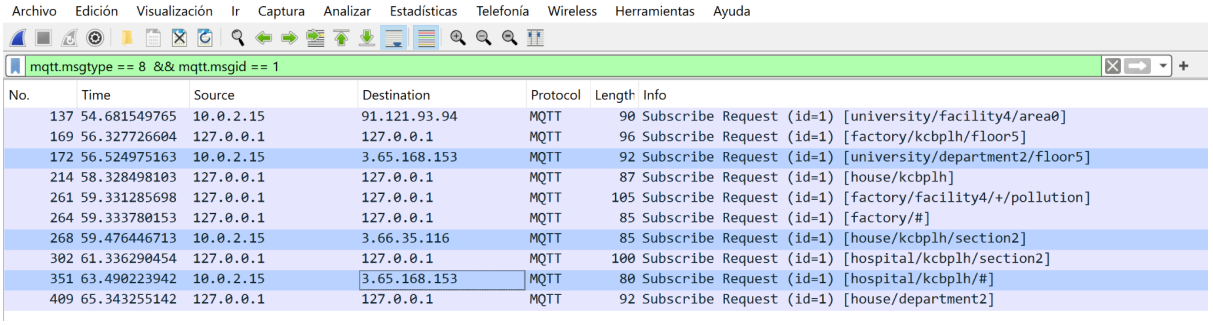


6a. How many MQTT subscribe requests with message ID=1 are directed to the HiveMQ broker?*

Answer: First, the filter *dns.qry.name matches "HiveMQ"* was implemented, where it was found that the answer for the broker is given in the IPs 3.65.168.153 and 3.66.35.116 as it is shown in the following image.

```
Answers
> broker.hivemq.com: type A, class IN, addr 3.65.168.153
> broker.hivemq.com: type A, class IN, addr 3.66.35.116
```

Moreover, after applying a filter for the type of message equal to subscribe request (8) and the message identifier equal to 1, for the IP addresses of the HiveMQ broker, **3** subscribe requests were detected, for the topics “university/department2/floor5”, “house/kcbplh/section2” and “hospital/kcbplh/#”



6b. How many publish messages are received by the clients thanks to the subscribe requests found in 6a)

Answer: If only the topics of the subscribe request message in point 6a are considered, there are not published messages that were received. To confirm, the filter *(mqtt.msgtype == 3 or mqtt.msgtype == 5)* was used to discriminate just the massage type under Publish messages or Publish Received, but after searching specifically the subscribed topics of point 6a, there was not any result.

7a. How many MQTT-SN (on port 1885) publish messages sent after the hour 3.59PM (Milan Time) are directed to topic 6?

Answer: To find all the published messages directed to topic 6, the filter *mqttsn.topic.id == 6* was used, however only **3** published messages were sent after 3:59 PM for topic 6 in the MQTT-SN protocol, on January 27th 2024.

7b. Explain possible reasons why messages in 7a) are not handled by the server

Answer: One of the possible reasons the messages are not handled by the server is because the specific port required for the communication (in this case, port 1885 for MQTT-SN) is unreachable or closed. In addition, as MQTT-SN is implemented for a sensor network where the connection is not reliable, there could be connectivity issues between the MQTT-SN client and the server, such as network outages or firewall restrictions, thus the publish message may not reach the server.