Internet of Things

Home Challenge I

Politecnico di Milano A.A. 2020-2021

Student : Laura Colzani **ID**: 10539060

Student : Dario Crippa **ID**: 10529288

Wireshark : Latest version available for MacOS

CoAP

QUESTION I

What is the difference between the message with MID: 3978 and the one with MID: 22636?

While the message with `MID=3978` is of type `CONFIRMABLE` and it needs to be acknowledged, the second one is `NOT-CONFIRMABLE`, so the client does not need any kind of feedback after the transmission.
In order to find this information we have used the Wireshark filter `coap.mid==MESSAGE_ID`.

QUESTION II

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

Using the Wireshark filter `frame.number==6949` we have found the `MESSAGE_ID` related to the message `No.6949`.
Through the filter `coap.mid==MESSAGE_ID` (28357) we tracked down the ACK related to the message.
We encounter an error : Method Not Allowed.

QUESTION III

How many replies of type confirmable and result code "Content" are received by the server "localhost"?

In order to answer this question we combined these filters :
- `ip.src==127.0.0.1` where `127.0.0.1` is the localhost IP address
- `coap.code=="2.05 Confirmable"`.

We have identified 8 ACK messages that satisfy these constraints.

MQTT

QUESTION IV

How many messages containing the topic "factory/department/+" are published by a client with username: "jane"? Where * replaces the dep. number, e.g. factory/department1/+, factory/department2/+ and so on.*

First of all we used the Wireshark filter `mqtt.username=="jane"` in order to identify the client.
At this point, for each frame, we have collected the stream index.

To reply to this question we had to use this composition of filters :
`(tcp.stream==112||tcp.stream==121||tcp.stream==230||tcp.stream==354)&&mqtt.topic=="factory/department*/+")`.

The answer is 0, due the fact the + is a "single level" wildcard.

QUESTION V

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

First of all we had to identify the IP address related to the HIVEMQ.
In order to do that we searched for the DNS packets that contain this information.

This broker has two IP addresses : `3.120.68.56` and `18.185.199.22`.

To get the answer for this question we used a composition of Wireshark filters :
`(ip.addr == 3.120.68.56 || ip.addr == 18.185.199.22) && mqtt.conflag.willflag == 1`.
We did this to identify the connected clients with a will message.

We discovered that there are :
• 9 clients connected to `3.120.68.56`;
• 7 clients connected to `18.185.199.22`.

QUESTION VI

How many publishes with QoS 1 don't receive the ACK?

To detect the publish messages with `QoS=1` we have used the filter combination :
`mqtt.qos==1 && mqtt.msgtype=="Publish Message" && (ip.src== 10.0.2.15||ip.src== 127.0.0.1).`

Thanks to the Wireshark functionality `statistics > Capture File Properties` we could calculate the number of these messages, which is 124.

On the other hand using the filter `mqtt.msgtype == "Publish Ack"` we discovered the number of the PUBACKs, referred only to publish messages with `QoS=1`.

The differences between these two values is the answer to the question, which is 50.

QUESTION VII

How many last will messages with QoS set to 0 are actually delivered?

Due the fact that `QoS=0` there is no guarantee of delivery.
This means that we can't count the number of the last will messages that have been delivered in a proper way.

QUESTION VIII

Are all the messages with QoS > 0 published by the client "4m3DWYzWr40pce6OaBQAfk" correctly delivered to the subscribers?

Using the Wireshark filter `mqtt.clientid=="4m3DWYzWr40pce6OaBQAfk"` we are able to identify the stream index related to this client.

At this point with the filter :
`tcp.stream==67&&mqtt.qos>0&&mqtt.msgtype=="Publish Message"`
we have identified all the messages published by the client with the given constraints, which is just one.

Due the fact that this message has a `QoS=2`, we need to identify three messages to prove the correct delivery to all the subscribers :
- `PUBREC` and `PUBCOMP` from the broker;
- `PUBREL` from the client.

Unluckily we have just the `PUBREC`, so we can't provide an answer to this question.

QUESTION IX

What is the average message length of a connect msg using mqttv5 protocol? Why messages have different size?

The average length for this kind of messages is 33 bytes while the average size of these packets is 91 bytes.

We reached this result passing through the filter `mqtt.msgtype=="Connect Command" && mqtt.ver==5`.
Thanks to the Wireshark functionality `Statistics > Capture File Properties` we have discovered the average packets size.
The average message length is 33 bytes, calculated as 91 - 58 bytes.

The number 58 refers to the size of the previous levels headers and payloads, which are TCP + IPv4 + datalink + physical.

Messages could have different sizes due the fact that a Connect message could optional fields.

QUESTION X

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

A client pings the broker when the `Keep Alive` interval is exceeded without any exchange of messages between the parts during this amount of time.

Using the filters `mqtt.msgtype == "Ping Response"` and `mqtt.msgtype == "Ping Request"` we don't see these messages.

The main reason of this is due to the fact that the `Keep Alive` interval is never exceeded.