Politenico di Milano

Dipartimento Elettronica, Informazione e Bioingegneria

Homework IoT Project

Packets Analysis

Author:
Francesco Monti
Matr: 919755

Supervisor:
Dr. Edoardo Longo
Dr. Matteo Cesana

May 14, 2020

Abstract

This document contains the answers for the third activity for the course ”Internet of Things”, Academic Year 2019/2020.
This document has been also uploaded on the following GitHub repository: https://github.com/Framonti/IoT_Projects

0.1 Answers

1. - **Answer**: The difference is the presence of an ACK from the server (104.196.15.150) for first message (MID = 3978); this ACK informs the client that the DELETE request wasn't successful (response code: 4.05 - Method Not Allowed).
   - **Filter**: coap.mid == 3978 or coap.mid == 22636

2. - **Answer**: Yes, the server (104.196.15.150) answers the GET request with an ACK, code 4.05 - Method Not Allowed.
   - **Filter**: First scroll to message with No. 6949, then check its MID (28357), finally identify the eventual response with coap.mid == 28357

3. - **Answer**: There are 8 replies.
   - **Comments**: The question asks for message of type ”Connectable”, which we interpreted as a typo of ”Confirmable”.
   - **Filter**: ip.addr == 127.0.0.1 and coap.code == 69 and coap.type == 2, then check the Displayed count.

4. - **Answer**: None, wildcard in Topics are allowed only when subscribing, not when publishing
   - **Filter**: Not necessary.

5. - **Answer**: Ten clients connected to hivemq have a will message.
   - **Filter**: Firstly, we find the IP address of hivemq with dns.resp.name == ”broker.hivemq.com”, that returns 3.120.68.56 or 18.185.199.22; then, we check the Displayed count of the messages selected with mqtt.willmsg and (ip.dst == 3.120.68.56 or ip.dst == 18.185.199.22)

6. - **Answer**: Fifty publishes with QoS of 1 didn't receive an ACK back.

- **Filter**: First we check the number of publishes with QoS of 1 with mqtt.qos == 1 and mqtt.msgtype == 3 (they are 124), then check how many ACKs to QoS 1 are sent back with mqtt.msgtype == 4 (they are 74), therefore 124 − 74 = 50 messages didn’t receive their ACK.

7. - **Answer**: Only one LWT with QoS equal to 0 is actually sent.
   - **Filter**: We rapidly check with mqtt.willmsg that all the will messages start with ”error: ”, then we serched mqtt.msg contains ”error: ” and mqtt.msgtype == 3 and mqtt.qos == 0

8. - **Answer**: No client is subscribed to the topic the client is sending data, therefore no publish is delivered.
   - **Filter**: We find source IP address an TPC port of the user with mqtt.clientid == ”4m3DWYzWr40pce6OaBQAfk” (10.0.2.15 and 58313 respectively). We check how many messages the user has made with ip.src == 10.0.2.15 and tcp.srcport == 58313 and mqtt.qos >0, and found that it only publish one message with the required QoS (and the server, 5.196.95.208, confirm it has received it, message No. 2425). However, the broker sent this publish to nobody (i.e. nobody is subscribed to this topic yet), as the only message with ip.src == 5.196.95.208 and mqtt.topic == ”factory/department1/section1/deposit” (the topic of the publish) returns only one value, with message = {”id”: ”Sensor 10002”, ”value”: 2381, ”lat”: 102, ”lng”: 103, ”unit”: ”lumen”, ”type”: ”light”}, different from the one sent by the client ”4m3DWYzWr40pce6OaBQAfk” (message = {”id”: ”Actuator 2”, ”value”: 812, ”lat”: 100, ”lng”: 140, ”unit”: ”C”, ”type”: ”temperature”})

9. - **Answer**: The average length is 91.08. The different size depends both on some errors (Malformed Packet) and on the different payloads (e.g. the password has variable length).
   - **Filter**: We go to Statistics ->Packet Length and apply the filter mqtt.ver == 5 and mqtt.msgtype == 1

10. - **Answer**: Under the standard, REQ Pings are sent by clients to keep alive the connection to the broker if no messages are exchanged during a timeout. This timeout is specified in the connection message (Keep Alive field). If no REQ pings are sent,

it means that all the clients have at least one of the following properties:

  - Have a Keep Alive greater than the duration of the traffic sniffed.
  - Properly disconnect
  - Unexpectedly disconnect (and fire a LWT message)
  - The broker and the client send to each other at least one message before the timeout expires

- **Filter**: No need