Team 18 - Homework 3

Andrea Parolin
Politecnico di Torino
Torino, Italia
s291462@studenti.polito.it

MohammadSadegh Radmehr
Politecnico di Torino
Torino, Italia
s301623@studenti.polito.it

Sara Ferrua
Politecnico di Torino
Torino, Italia
s292946@studenti.polito.it

I. Exercise 1

In this point are listed the reasons why MQTT is a better choice than REST as the communication protocol for this application.

- **Low bandwidth**: MQTT is a lightweight protocol that is designed to minimize the amount of data transmitted over the network due to the small size of the overhead (2 bytes). It uses binary encoding for messages, which is more efficient than the text-based encoding used by REST. This makes it well-suited for use in low-bandwidth and high-latency environments(e.g. IOT sensors).
- **Efficient use of resources**: MQTT uses a publish/subscribe model, which allows clients to send messages to multiple recipients without the need for each recipient to establish a separate connection using a broker. This can be more efficient in terms of network and server resources than the request/response model used by REST. One potential benefit is reduced power consumption from edge side.
- **QoS**: optional functionality that allows the assurance of message delivery, which can be important in mission-critical or time-sensitive applications. For instance with QoS level equal to 1 the devices keeps the message in memory until it receives from the recipient.
- **Scaleability**: MQTT is generally considered to be a scalable protocol, especially for applications involving a large number of clients(One-to-Many), which allows clients to send messages to multiple recipients without the need for each recipient to establish a separate connection. This can help reduce the load on the server and network resources.

II. Exercise 2

In Table I are inserted some of the methods of GET, POST, PUT, and DELATE that best fit the specific endpoint and its associated description. In the below list is turned off in detail each row of the Table I.

1) Data can be retrieved from a server using GET requests. In this specific case, it is used this method to obtain the list of mac addresses that are part of the dataset's key.
2) The GET method is used for retrieving a resource that is the battery status associated with a specific *mac_address* for a specific time range.
3) The DELETE method is used to delete a specific resource from the server.

TABLE I: HTTP method - Endpoint

Method	Endpoint	Motivation
GET	/devices	Retrieve the list of MAC addresses of the monitored devices.
GET	/device/{$mac\_address$}	Retrieve battery status information of the device with the specified MAC address in the specified time range.
DELATE	/device/{$mac\_address$}	Delete the timeseries associated to the specified MAC address.

The two timeseries connected to *mac_address: battery* and *mac_address: power* in this software can be deleted with the use of the delete method.