POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

IoT Challenge #2, Exercise Application Layer

Internet of Things

Authors: **Kevin Ziroldi - 10764177**

Matteo Volpari - 10773593

Professors: Alessandro Redondi, Fabio Palmese
Academic Year: 2024-2025
Version: 1.0
Release date: 6-4-2025

Contents

1 | Exercise temperature sensor and valve

1.1. Data

The data of the exercise is reported here.

- $T_{sensor\_reading} = 5$ minutes
- $T_{average\_computation} = 30$ minutes
- $L_{resource} = L_{topic} = 10$ Bytes
- $L_{payload} = 8$ Bytes
- $E_{TX} = 50$ nJ/bit
- $E_{RX} = 58$ nJ/bit
- Ideal Wi-Fi network
- $E_C = 2.4$ mJ

The message sizes of the two protocols are reported in the following tables.

Message	Size [Byte]
GET Request	60
GET Response	55
PUT Request	77
PUT Response	58
Empty ACK	14

Table 1.1: Message sizes CoAP

Message	Size [Byte]
Subscribe	58
Sub ACK	52
Publish	68
Pub Ack	51
Connect	54
Connect Ack	47
Ping Req	52
Ping Resp	48

Table 1.2: Message sizes MQTT

1.2. EQ1.a Energy consumed using CoAP

We start by computing the energy consumed by the two devices when they communicate using CoAP in the most efficient configuration energy-wise. The temperature sensor acts as a CoAP Server, while the valve as a CoAP Client.
In order to save energy, we adopt the following configurations:

- We use CoAP Non-confirmable requests, which means that requests are not followed by a ACK.
- Since the Client wants to receive the same data from the Server at fixed rate, we use CoAP Observation.

The valve (Client) sends a GET Request with Observe to the temperature sensor (Server). The Server is programmed to send a new value to who observes it, in our scenario the valve, every 5 minutes, with a GET Response. Finally, the valve computes the average every 30 minutes.

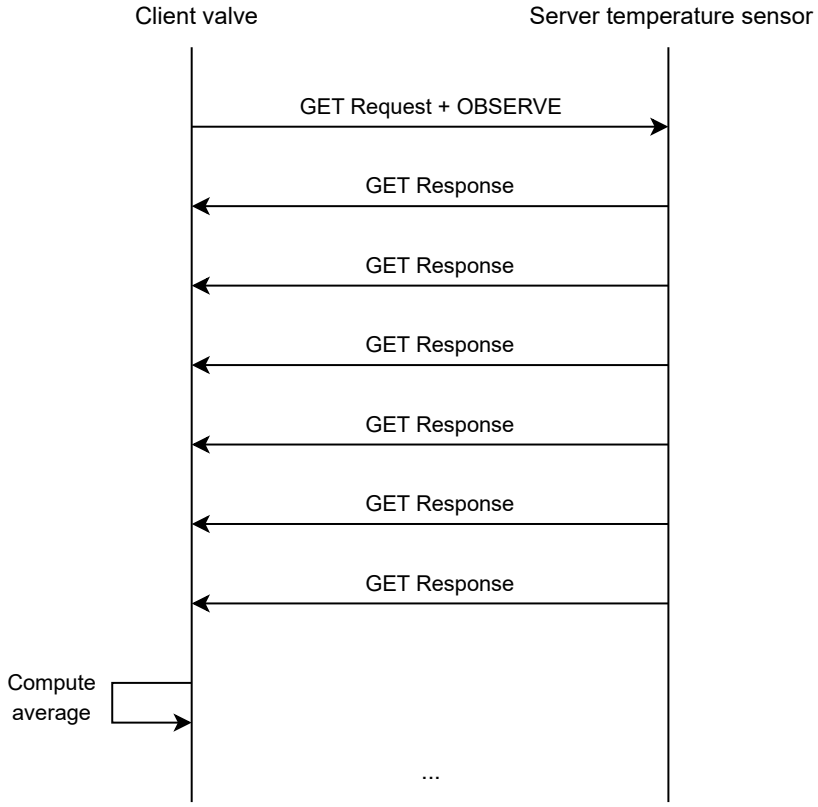

Figure 1.1: Message flow CoAP

We start the energy consumption calculation by computing the number of sensor readings, N_{sensor_read}, and the number of computations of the average, N_{avg}.

$$N_{sensor\_read} = (24\ \text{h} \cdot 60\ \text{min/h})/5\ \text{min} = 288$$

$$N_{avg} = (24\ \text{h} \cdot 60\ \text{min/h})/30\ \text{min} = 48$$

We can compute the energy consumed by the two devices as:

$$E_{CoAP\_valve} = E_{TX} \cdot L_{GET\_Req} + N_{sensor\_read} \cdot E_{RX} \cdot L_{GET\_Resp} + N_{avg} \cdot E_C = 122.574\ \text{mJ}$$

$$E_{CoAP\_sensor} = E_{RX} \cdot L_{GET\_Req} + N_{sensor\_read} \cdot E_{TX} \cdot L_{GET\_Resp} = 6.364\ \text{mJ}$$

$$E_{CoAP\_total} = E_{CoAP\_valve} + E_{CoAP\_sensor} = 128.938\ \text{mJ}$$

1.3. EQ1.b Energy consumed using MQTT

In this section, we compute the energy consumption if the devices communicate through MQTT. In this scenario, the Raspberry Pi acts as MQTT Broker, the sensor is a Publisher and the valve is a Subscriber. In order to minimize energy consumption, we adopt the following configurations:

- Messages are exchanged with QoS 0, in "at most once" mode, i.e. no PUBACK messages are exchanged.
- Neither the sensor nor the valve send ping messages. The sensor uses a Keep Alive greater than 5 minutes, so that the periodic PUBLISH message will reset the timer. The valve, instead, uses a Keep Alive 0, which disables the Keep Alive mechanism; disconnections from the valve will be detected at TCP level when the broker will try to communicate the new sensor reading but won't be able to reach the valve.

The sensor (Publisher) connects to the Broker and starts publishing temperature values every 5 minutes. The valve (Subscriber) connects to the Broker, subscribes to the topic and starts receiving temperature values.

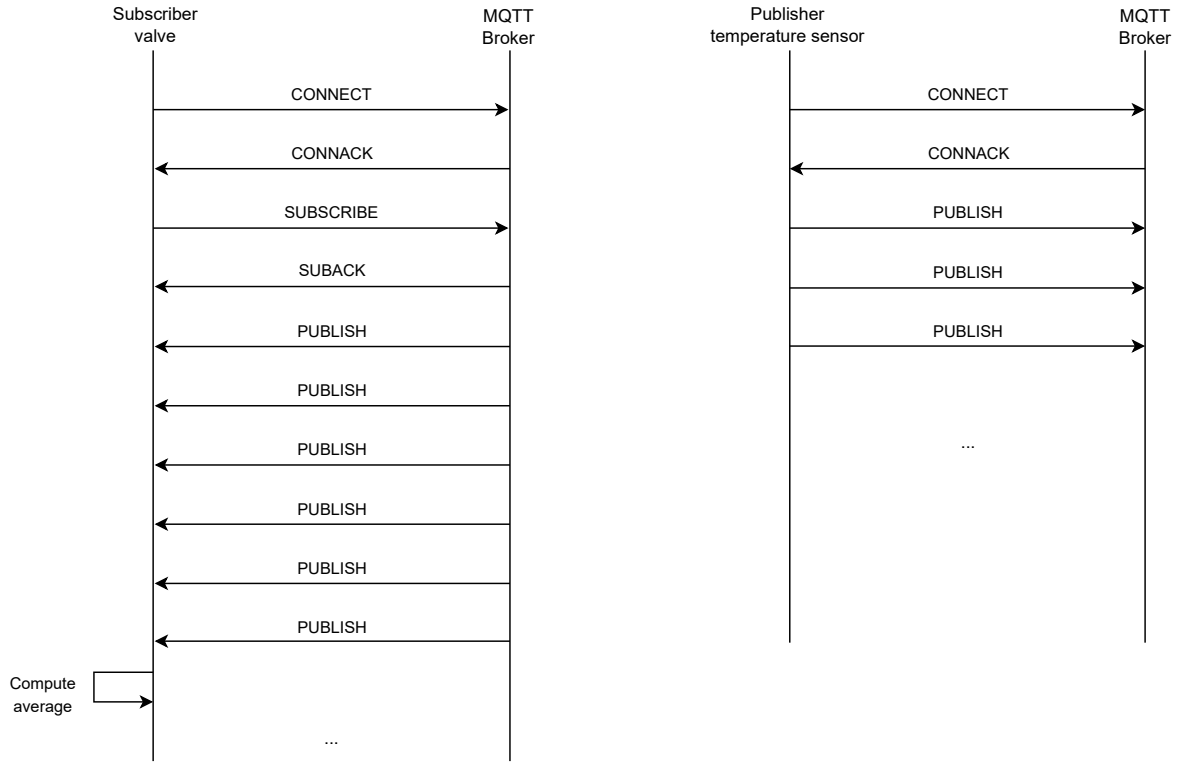

Figure 1.2: Message flow MQTT

We can compute the energy consumed by the two devices as:

$$E_{MQTT\_valve} = E_{TX} \cdot L_{CONNECT} + E_{RX} \cdot L_{CONNACK} + E_{TX} \cdot L_{SUBSCRIBE} +$$

$$+ E_{RX} \cdot L_{SUBACK} + N_{sensor\_read} \cdot E_{RX} \cdot L_{PUBLISH} + N_{avg} \cdot E_C = 124.378 \text{ mJ}$$

$$E_{MQTT\_sensor} = E_{TX} \cdot L_{CONNECT} + E_{RX} \cdot L_{CONNACK} + N_{sensor\_read} \cdot E_{TX} \cdot L_{PUBLISH} =$$

$$= 7.877 \text{ mJ}$$

$$E_{MQTT\_total} = E_{MQTT\_valve} + E_{MQTT\_sensor} = 132.255 \text{ mJ}$$

1.4. EQ2 Improvements

In this section, we propose two ways to decrease the energy consumed by the two device while using the Raspberry Pi as a broker.

1.4.1. Using MQTT-SN

The first improvement consists in using MQTT-SN instead of MQTT. Since we have not been provided with exact values for MQTT-SN messages we are going to make some assumptions. We assume that the sizes of the SUBSCRIBE and PUBLISH messages are:

$$L_{SUBSCRIBE\_MQTT\_SN} = L_{SUBSCRIBE\_MQTT} - L_{TOPIC\_MQTT} + L_{TOPIC\_MQTT\_SN} =$$

$$= 58 \text{ B} - 10 \text{ B} + 2 \text{ B} = 50 \text{ B}$$

$$L_{PUBLISH\_MQTT\_SN} = L_{PUBLISH\_MQTT} - L_{TOPIC\_MQTT} + L_{TOPIC\_MQTT\_SN} =$$

$$= 68 \text{ B} - 10 \text{ B} + 2 \text{ B} = 60 \text{ B}$$

Moreover we assume that the size of REGISTER is 59 B and the size of REGISTER ACK is 51 B, as seen during the exercise session.
Using MQTT-SN introduces a trade-off:

- Disadvantage: the Publisher needs to send a REGISTER message to the broker, represented by the Raspberry Pi, before being able to send PUBLISH messages.
- Advantage: PUBLISH messages have a reduced size, since they have a 2 Bytes long topic, instead of the 10 Bytes long one of MQTT messages.

On the long run, using MQTT-SN is beneficial, since we send more messages and compensate the initial cost of the REGISTER message with the lowered cost of PUBLISH

messages. We start by computing the number of packets that the Publisher should send with MQTT-SN in order to save energy with respect to MQTT. To do so, we compute the energy needed to send the initial packages, publish a variable number of packages m and compute the average every 30 minutes both with MQTT and MQTT-SN.

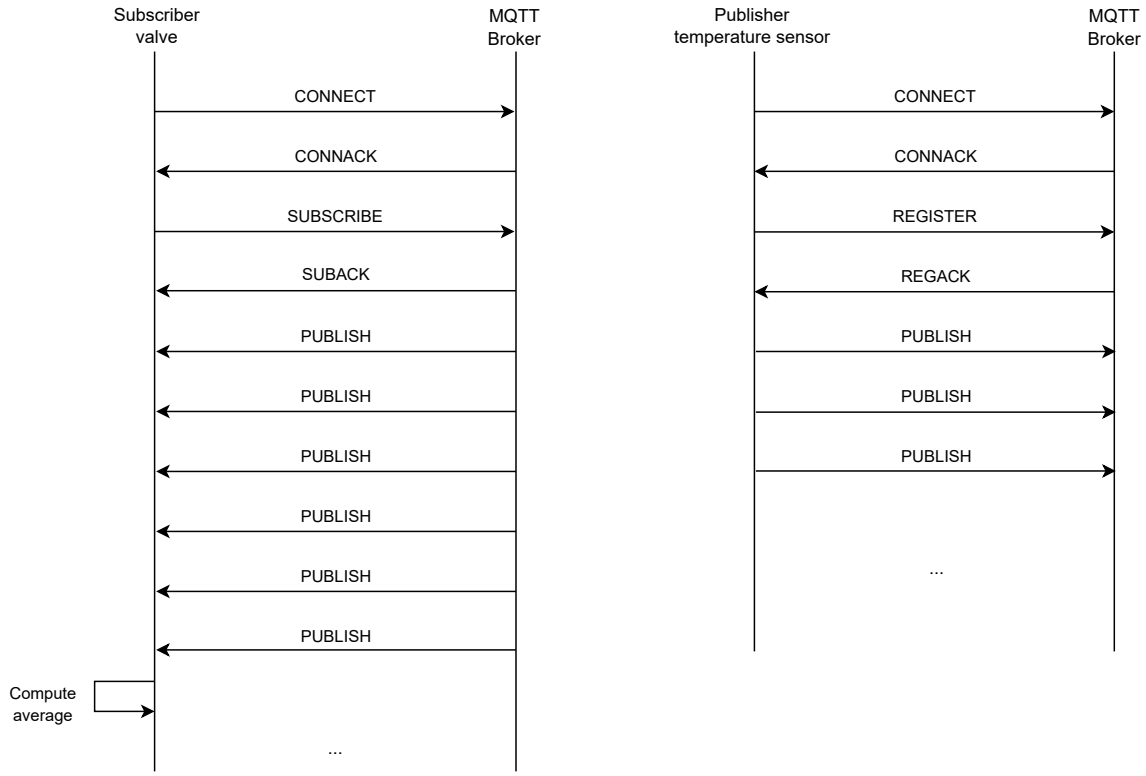

Figure 1.3: Message flow MQTT-SN

The energy consumption to send m packets with MQTT is:

$$E_{MQTT\_valve} = E_{TX} \cdot L_{CONNECT} + E_{RX} \cdot L_{CONNACK} + E_{TX} \cdot L_{SUBSCRIBE} +$$

$$+ E_{RX} \cdot L_{SUBACK} + m \cdot E_{RX} \cdot L_{PUBLISH} + N_{avg} \cdot E_C = 115.291 \text{ mJ} + m \cdot 3.1552 \cdot 10^{-5} \text{ J}$$

$$E_{MQTT\_sensor} = E_{TX} \cdot L_{CONNECT} + E_{RX} \cdot L_{CONNACK} + m \cdot E_{TX} \cdot L_{PUBLISH} =$$

$$= 4.3408 \cdot 10^{-5} \text{ J} + m \cdot 2.72 \cdot 10^{-5} \text{ J}$$

The energy consumption to send m packets in MQTT-SN is:

$$E_{MQTT\_SN\_valve} = E_{TX} \cdot L_{CONNECT} + E_{RX} \cdot L_{CONNACK} + E_{TX} \cdot L_{SUBSCRIBE} +$$

$$+ E_{RX} \cdot L_{SUBACK} + m \cdot E_{RX} \cdot L_{PUBLISH} + N_{avg} \cdot E_C = 115.288 \text{ mJ} + m \cdot 2.784 \cdot 10^{-5} \text{ J}$$

$$E_{MQTT\_SN\_sensor} = E_{TX} \cdot L_{CONNECT} + E_{RX} \cdot L_{CONNACK} + E_{TX} \cdot L_{REGISTER} +$$

$$+ E_{RX} \cdot L_{REGACK} + m \cdot E_{TX} \cdot L_{PUBLISH} = 9.0672 \cdot 10^{-5} \text{ J} + m \cdot 2.4 \cdot 10^{-5} \text{ J}$$

We now find the number of packets that the publisher should publish in order for the communication to be cheaper with MQTT-SN.
For the valve:

$$E_{MQTT\_SN\_valve} \leq E_{MQTT\_valve}$$

$$115.288 \text{ mJ} + m \cdot 2.784 \cdot 10^{-5} \text{ J} \leq 115.291 \text{ mJ} + m \cdot 3.1552 \cdot 10^{-5} \text{ J}$$

Which is true for every value of m (since it is positive), i.e. the valve always consumes less energy using MQTT-SN over MQTT.
For the sensor, instead:

$$E_{MQTT\_SN\_sensor} \leq E_{MQTT\_sensor}$$

$$9.0672 \cdot 10^{-5} \text{ J} + m \cdot 2.4 \cdot 10^{-5} \text{ J} \leq 4.3408 \cdot 10^{-5} \text{ J} + m \cdot 2.72 \cdot 10^{-5} \text{ J}$$

Which is true for $m \geq 14.77$, i.e. MQTT-SN is convenient if the publisher sends 15 packets or more.

Since the sensor sends 288 PUBLISH messages, using MQTT-SN is convenient. The energy consumption using MQTT-SN is computed by substituting $m = 288$:

$$E_{MQTT\_SN\_valve} = 123.306 \text{ mJ}$$

$$E_{MQTT\_SN\_sensor} = 7.003 \text{ mJ}$$

$$E_{MQTT\_SN\_total} = 130.309 \text{ mJ}$$

We can see that the new energy consumption is smaller than the one computed using MQTT, as expected by our calculations on the minimum number of packets m.

1.4.2. Sending less frequent updates

Another strategy we can adopt to reduce the energy consumption is to reduce the rate with which the sensor sends updates to the valve. The valve still computes the average every 30 minutes, but with 3 new readings, instead of 5, so the result will be less precise. The trade-off in this case is between saving energy and having an accurate measurement of the average.
For example, the sensor can send a temperature reading to the valve every 10 minutes. In this case, the number of readings published by the sensor will be:

$$N_{sensor\_read} = (24\ \text{h} \cdot 60\ \text{min/h})/10\ \text{min} = 144$$

Since 144 is stil greater than 15, using MQTT-SN is better than MQTT, so we can compute the new energy consumption by using $m = 144$:

$$E_{MQTT\_SN\_valve} = 119.297\ \text{mJ}$$

$$E_{MQTT\_SN\_sensor} = 3.547\ \text{mJ}$$

$$E_{MQTT\_SN\_total} = 122.844\ \text{mJ}$$