IRTA Implementation

Version control

Date	Version	Description	Author
20/01/2018	0.1.0	Document creation with packets REGISTER, PUBLISH y SUBSCRIBE	M. Finochietto
23/01/2018	0.1.1	Fix all paquets headers to fit MQTT format: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718020 (period, latency and timestamp are moved from fixed section to variable section).	M. Finochietto
28/05/2019	0.2.0	Content reorganization. Packets, Description and Code section have been created.	M. Finochietto

Description

This document describes the modifications needed to be done to an MQTT v3.1.1 implementation in order to translate the SRTI model into a messaging system for application development. There are two points to take into account for this work. The first one deals with how an MQTT broker should interact with the IRTA real-time agent. The second one indicates how to adapt the specification of this version of MQTT, so that publishers, subscribers and brokers can send the additional information needed by the IRTA real-time agent, to provide the functionalities already described in previous sections.

As shown in Figure 1, the communication between the MQTT broker and the IRTA real-time agent is bidirectional; this means that both components can send and receive data between them. On the one hand, the broker will send information about both the subscriptions and the publications that it receives. On the other hand, the IRTA agent will store the information of the subscriptions (topic, timestamp, deadline and period), and every time it receives a publication, it will perform the following steps:

1. If there is a function defined by the user, it will be applied taking the publication message as an argument. The output of this function will be a new publication message that will replace the previous one. For example, the original publication could be a temperature value with the topic "Full_Temperature" with an accuracy of 2 bytes; however, the function applied rounds it up to 1 byte because that much precision is not needed; therefore, to save bandwidth, it is published under a new topic named "Shrunken_Temperature".
2. The publication message product of step 1 is submitted to an analysis of temporal restrictions to validate if it is within the expectations of the subscriber or not. This analysis corresponds to the transition VI of the state machine described in section 4.
3. If from the previous step it is decided that the publication is valid, it is queued in the output buffer of the broker following the EDF scheduling algorithm.

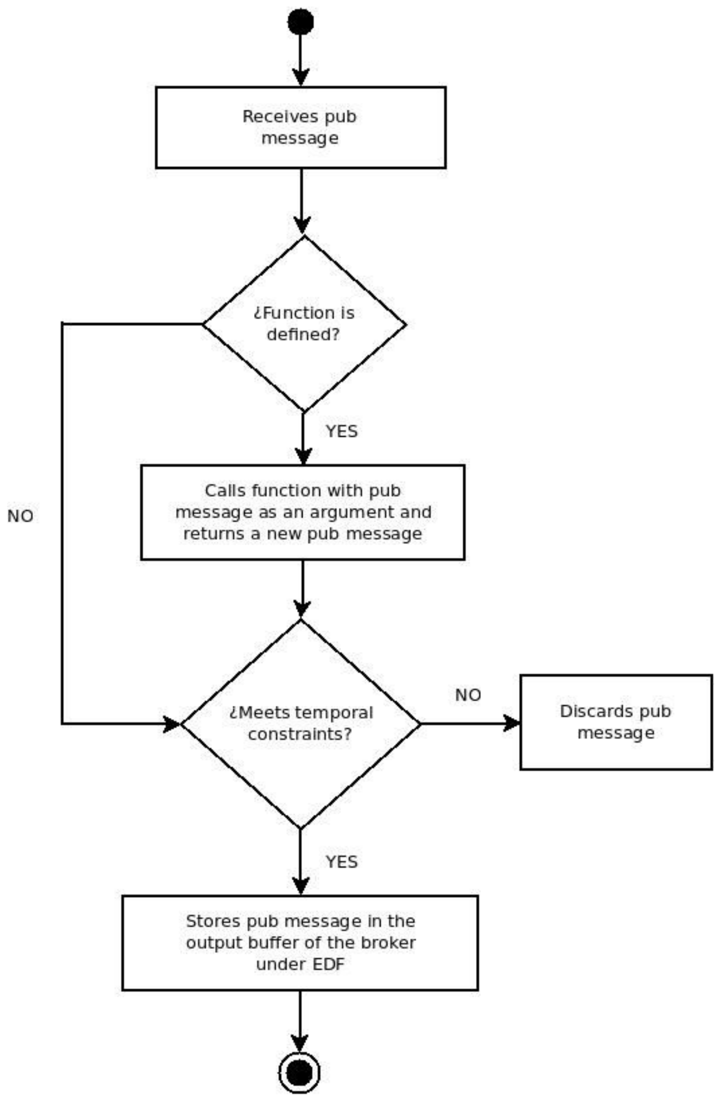
Receives pub message
¿Function is defined?
YES
NO
Calls function with pub message as an argument and returns a new pub message
¿Meets temporal constraints?
NO
Discards pub message
YES
Stores pub message in the output buffer of the broker under EDF

Figure 1 shows an activity diagram of the IRTA agent where this process is described.

Regarding the MQTT specification, there are 3 major changes suggested. The first one is to add a topic registration MQTT Control Packet. While an MQTT variant, known as MQTT-SN, introduces a way to register a topic in order to reduce packet sizes. The aim of the topic registration in SRTI is not to shrink the packets, but to inform the temporal data associated to the publisher and the periodicity of the topic. Therefore, its variable header should have the timestamp of the publisher, the period in which it will send the corresponding data of that topic, and the length of the topic along with its name. The packet identifier value could be 0 or 15, both currently unused.

The second suggestion is to modify the headers of the PUBLISH and SUBSCRIBE MQTT Control Packets, adding in both a timestamp of the issuer. Moreover, in the case of SUBSCRIBE also add the period and deadline in which it is expected to receive the data of that topic.

The third suggestion is to add a new return code in the SUBACK package, so that the broker can reject a subscription in case it cannot offer data with the temporary parameters required by the subscriber. The value of the return code could be any between the Failure code 0x80 and the maximum available 0xFF.

Code repository

An implementation as the one described in this document, has been done using Node.js. Both code and instructions on how to use it, can be found at https://github.com/marianofino/realtime-mqtt

Packets

REGISTER - Registers topic

[New Packet] It registers a topic in the broker, by giving the name of the topic and its temporal parameters.

Fixed header

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	MQTT Control Packet type (15)				Reserved			
	1	1	1	1	0	0	0	0
byte 2...	Remaining Length							

Variable header

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	**Period**							
byte 2	**Timestamp MSB**							
byte 3	**Timestamp**							
byte 4	**Timestamp**							
byte 5	**Timestamp LSB**							
byte 6	Length Topic MSB							

byte 7	Length Topic LSB
byte 8...	Topic Name

Payload
(empty)

PUBLISH - Publishes a message

[Modified Packet] Adds timestamp of the message to the variable header.

Fixed header

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	MQTT Control Packet type (3)				DUP flag	QoS level		RETAIN
	0	0	1	1	X	X	X	X
byte 2...	Remaining Length							

Variable header

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	**Timestamp MSB**							
byte 2	**Timestamp**							
byte 3	**Timestamp**							
byte 4	**Timestamp LSB**							

SUBSCRIBE - Subscribes to topics

[Modified Packet] Adds the period and latency required by the subscriber and the timestamp of the message to the variable header.

Fixed header

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	MQTT Control Packet type (8)				Reserved			
	1	0	0	0	0	0	0	0

byte 2...	Remaining Length

Variable header

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
byte 1	Packet Identifier MSB							
byte 2	Packet Identifier LSB							
byte 3	**Period**							
byte 4	**Latency**							
byte 5	**Timestamp MSB**							
byte 6	**Timestamp**							
byte 7	**Timestamp**							
byte 8	**Timestamp LSB**							