IoT MQTT – Every bit counts

Degree programme: BSc in Computer Science | Specialisation: Mobile Computing
Thesis advisor: Dr. Andreas Danuser
Expert: Thomas Jäggi (GIBB – Gewerblich Industrielle Berufsschule Bern)

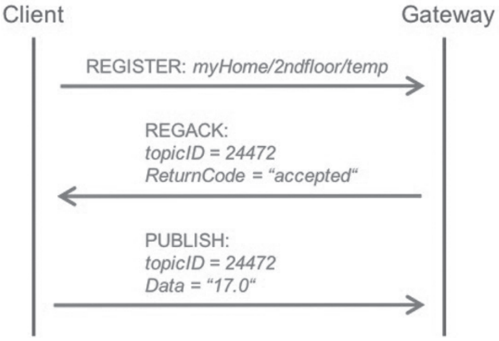
MQTT – a messaging protocol for the Internet of Things - is known to have a very low protocol overhead. Its verbose topic structure is perfect for humans, as it is readable. However, verbose topic names cause the advantage of a light weight protocol to be lost.
In constrained communication environments every bit counts. The elaborated MQTT topic shortener provides a topic management to reduce the ratio between protocol information and data.

Introduction

MQ Telemetry Transport (MQTT) and MQTT for Sensor Networks (MQTT-SN) are a lightweight and open messaging protocol for the Internet of Things. Clients publish or subscribe a topic on a centralised server (Broker). Therefore, each publish message contains a topic name. Based on that a broker decides if a subscribing client is interested in receiving the message or not.
MQTT-SN is optimised for the implementation on hardware constrained devices and a communication environment with low bandwidth and short message length. It has a slightly different message format but is designed to be as close as possible to MQTT.

Problem domain

The topic names in MQTT suffers from human weakness which tend to choose very verbose topic names. A usual MQTT topic looks as follows whereas a temperature value could represented as a single byte (0..255):
iot/smartHome/berlingtonStreet/17/2ndFloor/bedroom/sensor/temperature/
This publish message requires 70-bytes for the topic name and carries an information of 1-byte. For every publisher and every subscriber, this topic-overhead is present in each message – a ratio of 70:1 between protocol information and data. For a battery-powered device, with an unstable, slow and expensive network connectivity is such an overhead not practical.



Registration procedure of topic names in MQTT-SN

MQTT-SN overcomes this topic-overhead through a registration procedure before publishing. Each topic name has to be registered at server side to obtain a corresponding, two-byte long topic ID. Nevertheless, its protocol is tedious and not as comfortable as MQTT for human developer/debugger.

Content

The main goal of this thesis is the combination of MQTT and MQTT-SN to achieve a better data and protocol information ratio – without losing the simplicity of MQTT and with the benefit from MQTT-SN. Thus, the developer remains oblivious to the change, whereas the required bandwidth from and to the client is reduced. The complete solution shall work under the normal MQTT standard but shall also provide its service to 'optimised' clients that are aware of this service.

Outcome

A MQTT-broker is slightly adapted to use the elaborated MQTT topic shortener service (TSS). The broker itself can handle normal MQTT-clients as well as MQTT-clients that uses the service. The MQTT-client is extended to support the shorted topic names and its register procedure.
The topic registering process from MQTT-SN is implemented and used beside the normal client-broker connection for new topic names. A given topic will be shortened on broker side to a maximum length of 10-bytes and sent back to the client. Thus, the worst-case ratio between protocol information and data is 10:1.
The TSS on both side keeps track of each registered topic name and its short topic ID. The short topic ID will always be exchanged to the topic name before its used by the underling application or published on broker side. This guarantees the interoperability between all MQTT-clients and the functionality in case of an error.
The registering process first requires more data effort. But overall, it saves up battery-power, billing cost and data effort for each message.

Raphael Kim Huber