IOT Meta Projects — Specifications

Dolt.ch

Table of Contents

Glue

Incomming Message Queue

Specs ID	glue:incomming-message-queue
Start Date	2017-05-22
Implemented	No
Rev	1
Tracking issues	No

Summary

Incomming message from multiple Lorawan network are stored in multiple queue in Redis, using the FPort lorawan field as key.

Motivation

We need to support multiple Lorawan Network and multiple sensors:

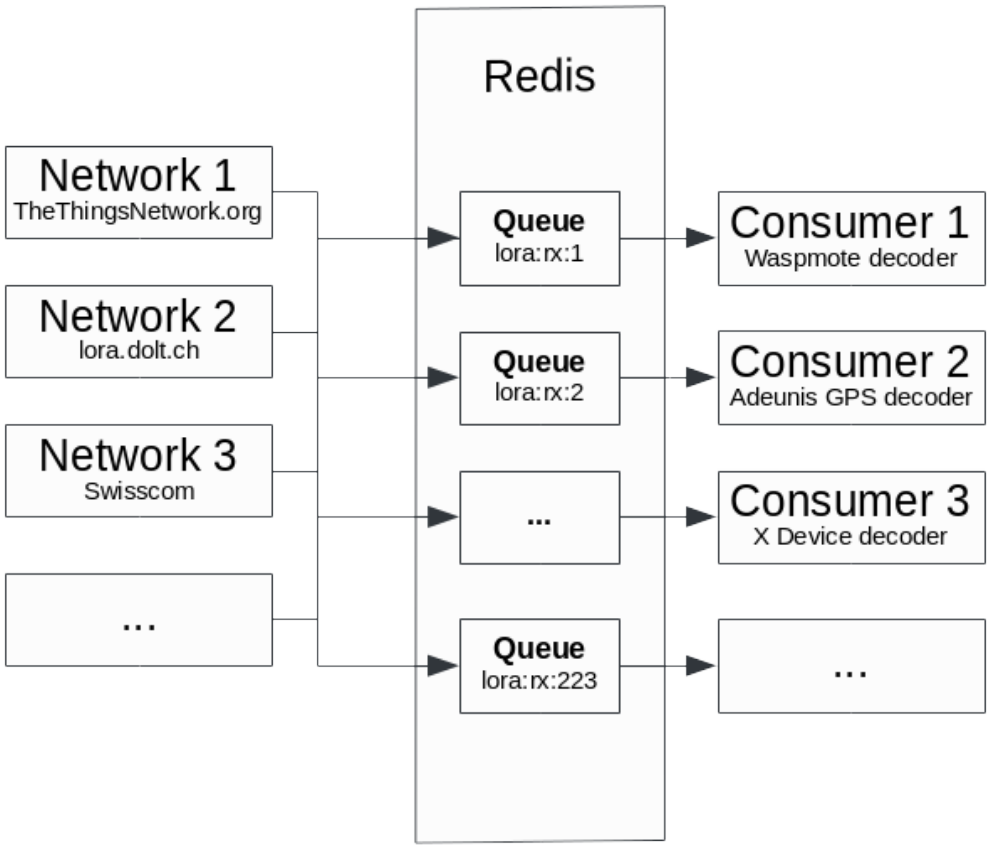
- Each Lorawan network come with a different protocol
- Each Lorawan sensors type use a different encoding to send sensors status

Our stack use the microservice patterns, so it would be nice to have:

- One process per network
- One process per sensor, or per group of sensors who use the same encoding scheme.

Incoming message with raw, undecoded content are not usefull, we don't need to save them. It would be great to keep them in volatile memory.

Detailed design

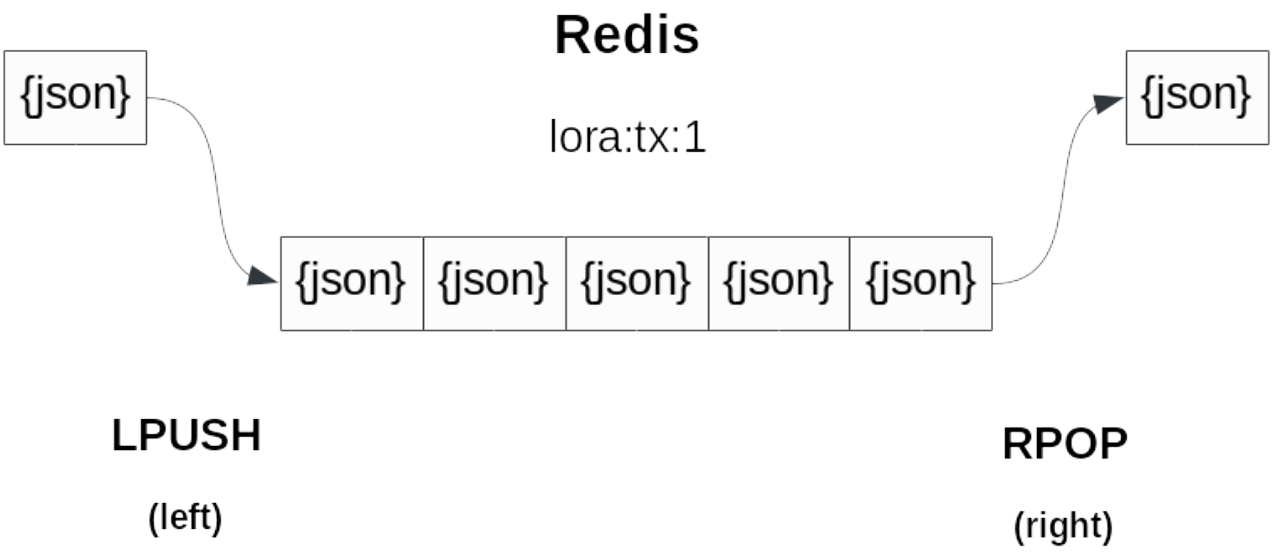


FPort as device type selector

We use the FPort field to choose in which queue a message should go.

This implies that if two device share the same port, they have a compatible data encoding format.

Redis queue



Producers

Each incomming message are converted in a simple JSON format:

```
{
    "device"    : "0123456789ABCDEF" ①
    "payload"   : "AQIDB==" ②
    "metadata"  : {
    "network" : {
        "time": "1970-01-01T00:00:00Z" ③
    }
    }
}
```

① 64bit end-device identifier (EUI-64), coded as hexadecimal string

② Base64 encoded payload

③ Time when the Lora server received the message, coded as a ISO-8601 string

IMPORTANT	Extra field are not allowed.

NOTE	Minified JSON (one-liner, without extra blank space and return line) or JSON with extra blank space, extra line or other indentation are valid

Then we add them in Redis, using FPort as a key to a FIFO queue

```
LPUSH lora:rx:${FPort} ${JSON_MESSAGE} ① ②
```

① Replace ${FPort} by the FPort number, an integer between 1 and 223 (see LoraWan 1R0 specs, section 4.3.2)

② Replace ${JSON_MESSAGE} by a correct json message, as specified above

IMPORTANT	Producers are not allowed to add extra field

Consumers

Consumers have an assigned FPort number, or a list of compatible FPort number.

They can get the oldest message for a given FPort number using the following Redis command:

```
RPOP lora:rx:${FPort} ①
```

① ${FPort} should be replaced by a number between 1 and 223

IMPORTANT	Consumers should be able to parse a message with extra field NOTE: Blocking version of RPOP (BRPOP) should be used when possible

Drawbacks

Using FPort as device type selector

With this solution, only 222 different device type are possible. For now this is not a problem, and we didn't want to add overcomplexity (less is more).

When this becomes a probleme, a solution might be to use multiple application_id per network and to use lora:APP_ID:FPort

Using JSON

JSON is not the fastest serialisation format. A binary alternative like msgpack or CBOR could be more efficient (speed, memory space).

But JSON has the advantage to be human friendly, a works well with redis-cli.

If one day we want more perf, it should be easy to swap JSON for by eg. Msgpack.

The JSON encoded message didn't have a version field

As JSON encoded raw message have a short and volatile lifetime, it's not necessary to know wheter they use this specs in version 1 or in any futur revision.

Alternatives

None

Unresolved questions

None

ttn2redis

redis2influx