Logging Service Application for a LoRa Gateway

Author: Anne Nasimiyu Makhanu

Date: 05/05/2023

Project Overview

This project aims to build a logging service application that reports the connection status of the network for a LoRa gateway that has a LoRa gateway module attached to a Raspberry Pi. Its functionality includes connecting to the MQTT broker, recording the time it got and lost the connection, then sending the network statistics. By monitoring the statistics, it is possible to assess the network health status and identify any trends or errors that may require further investigation.

Implementation

Tools

The application is built using Python 3 and the Paho MQTT library. It connects to a public MQTT broker- Host: broker.emqx.io Port: 1883

The Python Code

The points below explain the flow of the python script:

1. The necessary libraries, including the paho.mqtt.client library and the time library are imported.
2. The following functions are defined:
    - ***on_connect*** is called when the client successfully connects to the MQTT broker. It takes four arguments:
      client - the MQTT client instance.
      userdata - user-defined data that can be passed to the mqtt.Client() constructor.
      flags - contains response flags sent by the broker.
      rc - the connection result code returned by the broker.
      In this code, on_connect() prints a message to the console indicating whether or not the connection was successful.
    - ***on_disconnect*** is called when the client is unexpectedly disconnected from the MQTT broker. It takes three arguments: client, userdata, and rc. In this code, on_disconnect() prints a message to the console indicating the time of the

disconnection, waits for 5 seconds, and then attempts to reconnect to the MQTT broker using client.reconnect().

- ***statistics*** sends network statistics to the MQTT topic. It creates a message containing information about the time the connection was lost, the current time, and the number of connection retries that have occurred, and then publishes the message to the "/stats/health/device_id/network" topic using the client.publish() method.
- ***main*** runs the application. It creates a new MQTT client instance, sets the event handlers for connecting and disconnecting (event handlers are functions that are called by the MQTT client instance to handle specific events or situations), and connects to the MQTT broker. It then initializes a retry counter to 0 and enters an infinite loop.

  Inside the ***infinite loop***, the function client.loop() processes incoming MQTT messages and network traffic. It must be called frequently (in this code it's in a loop) to ensure that messages are received and that the client stays connected to the broker. The function then checks whether the client is disconnected from the MQTT broker.

  If the client is disconnected, the retry counter is incremented. If the client is reconnected after one or more retries, the ***statistics function*** is called to send network statistics with the lost connection time and number of retries, and the retry counter is reset to 0. The function also records the time of the last successful connection. Finally, the function waits for 1 second before checking the MQTT connection again.

3. time.strftime('%Y-%m-%d %H:%M:%S') is a function that returns a formatted string representing the current date and time. The %Y-%m-%d %H:%M:%S specifies the format of the string, where %Y is the year, %m is the month, %d is the day, %H is the hour in 24-hour format, %M is the minute, and %S is the second. So, the resulting string will look something like this: 2023-05-05 12:07:45.
4. time.sleep() is a Python function that pauses the execution of the program for a specified number of seconds. In this context, it is used to wait for 5 seconds before attempting to reconnect to the MQTT broker if a disconnection is detected. This allows time for the

network to potentially recover and for any issues causing the disconnection to be resolved before attempting to reconnect.

Conclusion

A logging service application that uses MQTT to report the network connection status of a LoRa gateway has been built successfully. The program meets the project's specifications and offers a simple tool for checking the LoRa gateway's network health status.

References

Eclipse Paho MQTT Python client library documentation https://pypi.org/project/paho-mqtt/#on-connect