Virtual smart home automation system simulating MQTT network

Rafat Khan

May 8, 2022

1 Introduction

The goal of this project is to create a virtual smart home automation system simulating a complete MQTT network, which is an OASIS standard messaging protocol for the Internet of Things (IoT). The MQTT (Message Queue Telemetry Transport) data will be cached by an HTTP server and made available via HTTP as well as via WebSockets to allow for real time updates in the dashboard. I will build separate program for different components which will communicate using MQTT network and can be deployed to IOT devices using azure-iot-device functions.

2 Overview

The Internet of things describes physical objects with sensors, processing ability, software, and other technologies that connect and exchange data with other devices and systems over the Internet or other communications networks. Communication protocols are important in such systems because they create consistency and universality for the sending and receiving of messages. But unlike other networking protocols, for IoT devices, we need a lightweight protocol with a fast response time and minimum overhead, which requires a very low bandwidth. It makes the interaction between devices efficient, whatever the number of devices there is. The MQTT IoT protocol is such a protocol which has become a standard for IoT messaging. Simulating a MQTT network is the main goal of this project.

2.1 MQTT Protocol

MQTT Message Queuing Telemetry Transport) is a lightweight IoT messaging protocol based on the publish/subscribe model. It is a simple messaging protocol which provides an easy solution for communication between constrained devices. It can provide real-time and reliable messaging services for networked devices with very little code and bandwidth. It is widely used in the industries such as the IoT, mobile Internet, smart hardware, Internet of Vehicles and power energy.

2.2 Components

I have worked with some basic component in this project. The main features of this project is the smart management of heating systems and blind based on the weather update. Our system will fetch weather information and publish the data using MQTT broker while the heating and blind control programs will subscribe these data and change the room temperature and will open/close the blinds.

2.2.1 Weather station

The weather station fetch weather update and publish the data using the MQTT broker every 10 minutes and looking for alerts in the selected region (data is fetched from openweathermap API).

2.2.2 Heating System

Compare ideal temperature with outside temperature and adjust the heater accordingly for different rooms. This MQTT client publish Data (Current temperature) and subscribe different topics (Weather, Heating and Simulation Speed).

2.2.3 Blinds Control

Decide if the blinds will be open or closed based on sunrise and sunset data from the weather station. There is a configuration option to set the mode of the blinds to manual and change the value of how open we want the blinds to be. If set to automatic, blinds will remain shut between sunset and sunrise and open between sunrise and sunset.This MQTT client publish data (Roller Blind) and subscribe different topics (Weather, Blinds).

3 Implementation

The project is implemented using using JavaScript and node JS framework. The major components are described bellow.

3.1 HTTP Server

I have used a HTTP server which contain the weather updates for being accessed from the dashboard.

3.2 MQTT Broker

An MQTT broker is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients. An MQTT client is any device (from a micro controller up to a fully-fledged server) that runs an MQTT library and connects to an MQTT broker over a network.

In this project, I have used 'Aedes', which is a stream-based MQTT broker.

3.3 MQTT Clients

Publish and subscribe messages to the connected MQTT broker.

```
    mqttClient.publish();
    mqttClient.on("connect", () => {
    subscribe();
});
```

3.4 Web Socket

MQTTWebSocketBridge is used to enable web socket. In the case of MQTT over Websockets the websockets connection forms an outer pipe for the

MQTT protocol. The MQTT broker places the MQTT packet into a websockets packet, and sends it to the client. The client unpacks the MQTT packet from the websockets packet and then processes it as a normal MQTT packet.

4 Conclusion

The purpose of MQTT is to provide a protocol, which is bandwidth-efficient and uses little battery consumption. MQTT uses our existing Internet home network to send messages to the IoT devices and respond to those messages. This connectivity is one of the major components of IoT. Though this project only simulate a solution for smart home, it can be deployed to IOT devices using azure-iot-device or similar services.