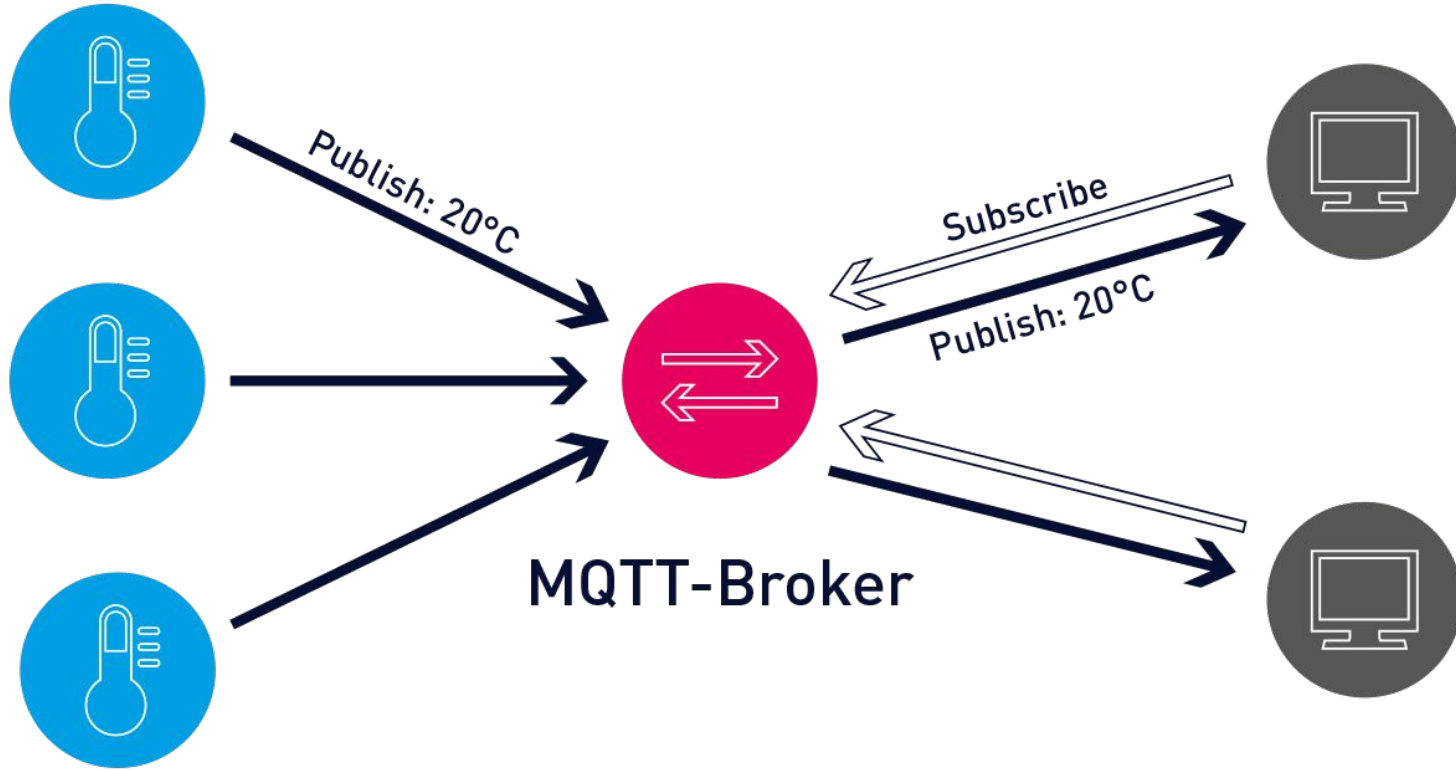


Diseño y Análisis de Algoritmos

MSc. Enrique Paiva - 2023

¿Qué es MQTT?

MQTT es un protocolo de mensajería basado en estándares, o un conjunto de reglas, que se utiliza para la comunicación de un equipo a otro. Los sensores inteligentes, los dispositivos portátiles y otros dispositivos de Internet de las cosas (IoT) generalmente tienen que transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado. Estos dispositivos IoT utilizan MQTT para la transmisión de datos, ya que resulta fácil de implementar y puede comunicar datos IoT de manera eficiente. MQTT admite la mensajería entre dispositivos a la nube y la nube al dispositivo.



¿Por qué es importante el protocolo MQTT?

Ligero y eficiente

La implementación de MQTT en el dispositivo IoT requiere recursos mínimos, por lo que se puede usar incluso en pequeños microcontroladores. Por ejemplo, un mensaje de control MQTT mínimo puede tener tan solo dos bytes de datos. Los encabezados de los mensajes MQTT también son pequeños para poder optimizar el ancho de banda de la red.

Escalable

La implementación de MQTT requiere una cantidad mínima de código que consume muy poca energía en las operaciones. El protocolo también tiene funciones integradas para admitir la comunicación con una gran cantidad de dispositivos IoT. Por tanto, puede implementar el protocolo MQTT para conectarse con millones de estos dispositivos.

Fiable

Muchos dispositivos IoT se conectan a través de redes celulares poco fiables con bajo ancho de banda y alta latencia. MQTT tiene funciones integradas que reducen el tiempo que tarda el dispositivo IoT en volver a conectarse con la nube. También define tres niveles diferentes de calidad de servicio a fin de garantizar la fiabilidad para los casos de uso de IoT: como máximo una vez (0), al menos una vez (1) y exactamente una vez (2).

¿Por qué es importante el protocolo MQTT?

Seguro

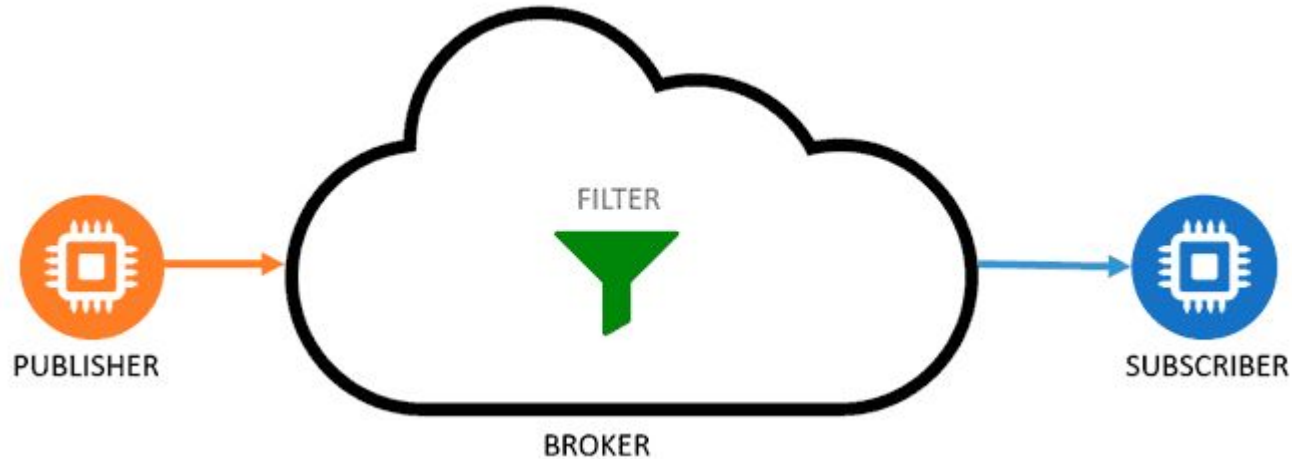
MQTT facilita a los desarrolladores el cifrado de mensajes y la autenticación de dispositivos y usuarios mediante protocolos de autenticación modernos, como OAuth, TLS1.3, certificados administrados por el cliente, etc.

Admitido

Varios lenguajes, como Python, tienen un amplio soporte para la implementación del protocolo MQTT. Por lo tanto, los desarrolladores pueden implementar rápidamente con una codificación mínima en cualquier tipo de aplicación.

MQTT Broker

La clave principal de la comunicación MQTT es el **Broker**, que es el programa que se encarga de receptionar los mensajes enviados por los clientes y distribuirlos entre sí en un sistema pub-sub.



MQTT Broker Open Source

- **Mosquitto.** Como decíamos es el broker MQTT más conocido en el sector doméstico/#maker. Es un broker Open Source desarrollado por la fundación Eclipse y distribuido bajo licencia EPL/EDL. Está programado en C, y es multiplataforma. Es un broker liviano y adecuado para uso en servidores de baja potencia.
- **Mosca.** Es un broker MQTT Open Source para Node.js, desarrollado en Javascript por Matteo Collina. Puede ser empleado como aplicación independiente o embebido en cualquier proyecto de Node.js
- **Aedes.** Del mismo autor que Mosca, Aedes es un servidor broker MQTT Open Source para Node.js diseñado para ser un reemplazo de Mosca.
- **HBMQTT.** Es un broker MQTT Open Source escrito en Python que funciona sobre asyncio, introducido en Python 3.4.
- **EMQTT.** Erlan MQTT Broker es Open Source, desarrollado en Erlang/OTP, está diseñado para aplicaciones con grandes exigencias en escalabilidad.
- **RabbitMQ.** Es un popular broker de mensajería AMQP Open Source, que también permite emplear el protocolo MQTT a través de un Adaptador.
- **HiveMQ CE.** La versión Community del popular HiveMQ es un broker Open Source basado en Java.
- **ActiveMQ.** Es un broker de mensajería JMS (Java Message Script) Open Source desarrollado por Apache, que también admite protocolo MQTT.

Referencias

[**https://aws.amazon.com/es/what-is/mqtt/**](https://aws.amazon.com/es/what-is/mqtt/)

Examen

SmartGrid

Grupo 1 - Luces Inteligentes

Función Objetivo: $100 + 0.05 * (\log_{10}(x) - 3 \times \cos(x) + x^2/20)$

- Al menos 5 atributos.
- Usar una estructura de datos (set).
- Hallar el mínimo consumo (en 40 días)
- Enviar On y Off, para prender el foco.

Grupo 2 - Sensores de Temperatura

Función Objetivo: $30 * (\cos(x)/10 + \sin(x)^2/x + 1) * \sqrt{x}/(x+1)^2 + (x-3)^2/2$

- Al menos 5 atributos.
- Usar una estructura de datos (stack).
- Hallar máximo consumo en 30 días (Hill Climbing).
- Graficar la temperatura, enviar cada 0.5 segundos.

SmartGrid

Grupo 3 - Cargador Solar

Función Objetivo: $P * (\cos(x)/10 + \sin(x)^2/x + 1) * \sqrt{x} / (x+1)^2 + (x-3)^2/2 \rightarrow (800 < a < 1000)$

- Al menos 5 atributos.
- Usar una estructura de datos (queue).
- Hallar la mínima potencia generada (30 días).
- Dar un valor a "P" y enviar.