

Monitoramento de ambientes com sensores utilizando MQTT

William Barbosa de Macedo

¹Núcleo de Computação Eletrônica – NCE – Universidade Federal do Rio de Janeiro (UFRJ)
Prédio do CCMN - Bloco C, Caixa Postal: 2324 - CEP: 20.010-974

²Departamento de Ciência da Computação – Prédio do CCMN - UFRJ
Rio de Janeiro, Brasil

³Instituto de Matemática
Centro de Tecnologia – Rio de Janeiro, RJ – Brasil

`william.macedo@poli.ufrj.br`

Abstract. *This paper describes how MQTT will be used for subscribing and publishing temperature data using.*

Resumo. *Este artigo visa descrever alguns conceitos e as principais funcionalidades do protocolo MQTT.*

1. Introdução

Para este trabalho foi utilizado um microcontrolador com interface de rádio embutido e suporte ao protocolo IEEE 802.11 que se integrará com o Broker Mosquitto para troca de mensagens.

2. MQTT

O protocolo de rede da Internet é o TCP/IP. Desenvolvido com base na pilha TCP/IP, o MQTT (Message Queue Telemetry Transport) tornou-se o padrão para comunicações de IoT. O MQTT é um protocolo de rede leve e flexível que oferece o equilíbrio ideal para os desenvolvedores de IoT. O protocolo leve permite a implementação em hardware de dispositivo altamente restringido e em redes de largura de banda limitada e de alta latência. Sua flexibilidade possibilita o suporte a diversos cenários de aplicativo para dispositivos e serviços de IoT.

2.1. Cliente MQTT

Um cliente MQTT pode ser qualquer dispositivo como um microcontrolador ou um PC rodando um server que utiliza a bibliotecas MQTT. Clientes podem ser tanto publicadores, quanto assinantes. Clientes assinam tópicos para publicar e receber mensagens.

2.2. Broker

O Broker tem como principal função receber, filtrar e encaminhar as mensagens aos clientes. O Broker comanda os tópicos e recebe assinaturas dos clientes.

2.3. Sessões, assinaturas e tópicos

Uma sessão é identificada como o estabelecimento de conexão entre um cliente e um broker. Diferente de uma sessão, uma assinatura estabelece uma conexão ou melhor um vínculo entre um cliente e um tópico. Além disso, tópicos permitem clientes trocarem informações dentro de um campo semântico e são baseados no padrão publish/subscribe.

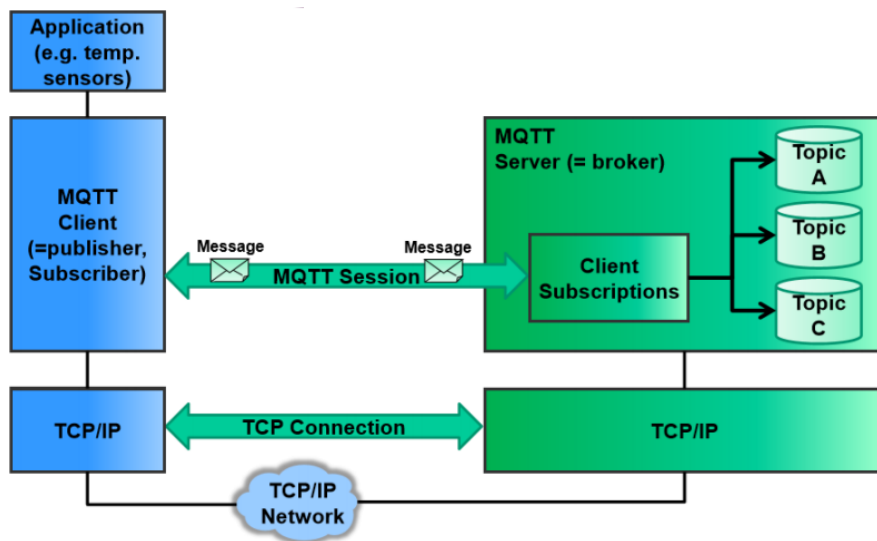


Figura 1. Diagrama de relações dentro do universo MQTT

2.4. Modelo Publish-Subscribe

Alternativa ao tradicional modelo cliente-servidor, o Publicador e Assinante não precisam se conhecer(pelo ip ou port). Além disso, o Publicador e Assinante não precisam estar sincronizados e o Intermediário(Broker) conhece tanto o publicador, quanto o assinante.

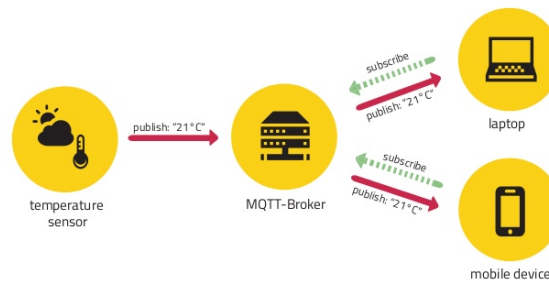


Figura 2. Esquema de comunicação entre o sensor de temperatura e o Broker através do protocolo MQTT

```

1 def send_sample(obj):
2     print("publishing: ", obj)
3     client.publish("temp/random", str(obj))
4
5     print(temperature)
6     send_sample(temperature)
7     if temperature%10==0:
8         publish_to_self()
9 except Exception as e:
10     print(e)
11

```

A função `sendsample()` é responsável por publicar os valores de temperatura para o broker.

O código completo pode ser acessado neste link: [github](#)

Referências

- [1] Zerinth Middleware. *Installation Guide*, <https://docs.zerynth.com/latest/index.html>
- [2] O protocolo MQTT, *Leve e simples. Perfeito para IoT e sistemas embarcados*, <https://www.gta.ufrj.br/ensino/eel878/redes1-2018-1/trabalhos-vf/mqtt/>, Jose E.B. Nascimento, Matheus Molin, Luis Octávio, Rio de Janeiro, Universidade Federal do Rio de Janeiro, 2018
- [3] Things flow Python, *Streaming dataflow library for IoT applications. Program at a higher level, with reusable "things"*, <https://github.com/mpi-sws-rse/thingflow-python>, Author: jfischer
- [4] Mosquitto, *hosts a publicly available Mosquitto MQTT server/broker.*, <https://test.mosquitto.org/>,