Protocolo MQTT para sistemas de IoT

Um estudo técnico/prático

Larissa L. Wong Marco A. G. Pedroso
Victor E. Almeida

UNIOESTE

29 de abril de 2022

Conteúdo

Colocar mais algo para introduzir

TODO

Criação do Protocolo

Começou a ser projetado durante a década de 1990 por:

Figura 1: Andy Stanford-Clark da IBM

Figura 2: Arlen Nipper da Cirrus Link/Eurotech

Problemas a serem resolvidos

- Resolver o problema de conexão de oleodutos via satélite.
- Limitações:
  - Alta latência;
  - Baixa largura de banda;
  - Dispositivos com pouca bateria.

Requisitos do Protocolo

- Implementação simples;
- Uso de QoS, *Quality of Service* por quem publica a mensagem;
- Uso eficiente de largura de banda, baixo *overhead*;
- Baixo custo energético para envio;
- Possibilidade de enviar qualquer tipo de dado;
- Possibilidade de manter conexões ativas, prontas para enviar e receber dados;

Fase do protocolo proprietário

- Primeira versão implementada no ano de 1999;
- Batizado MQTT, *MQ Telemetry Transport*, em referência ao produto da IBM MQ Series
- Muito utilizado embarcado em produtos da IBM.

Fase do protocolo aberto I

- Demanda/Aplicabilidade IoT;
- Em 2010 o protocolo se tornou livre;
- Primeira versão lançada 3.1;
- Investimentos da IBM através da Eclipse Foundation para criar um ecossistema em torno do protocolo.

Fase do protocolo aberto II

Figura 3: Exemplos de aplicações do ecossistema MQTT

Fase do protocolo aberto III

- No ano de 2013 a IBM buscou padronização com a OASIS;
- 29 de outubro de 2014 o MQTT foi aprovado como padrão pela OASIS na sua versão 3.1.1

Fase atual do protocolo

- A última versão 5.0 março de 2019;
- Funcionalidades modernas como:
  - facilidade de conexão e interação com a nuvem;
  - Tratamento de erros;
- Implementações de clientes para diversos sistemas e linguagens;
- Implementação de diversos brokers;
- Utilizado por grandes empresas tanto software aberto quanto proprietário.

Titulo

Tabela 1: Estrutura básica comum a todos os pacotes MQTT.

Cabeçalho fixo
Cabeçalho variável
Carga útil

Titulo

Tabela 2: Estrutura do cabeçalho fixo utilizado no protocolo MQTT.

Bit	7	6	5	4	3	2	1	0
Byte 1	Tipo de pacote MQTT				Sinais específicos do pacote			
Byte 2	Espaço restante							
. . .								

Titulo

Tabela 3: Tipos de pacotes utilizados pelo protocolo MQTT – Parte I.

Tipo	Código	Remetente	Descrição
Reserved	0	-	Reservado
CONNECT	1	Cliente	Solicitação de conexão
CONNACK	2	Servidor	Confirmação de conexão
PUBLISH	3	Ambos	Publicar mensagem
PUBACK	4	Ambos	Publicar confirmação (QoS 1)
PUBREC	5	Ambos	Publicar recebimento (QoS 2 – parte 1)
PUBREL	6	Ambos	Publicar lançamento (QoS 2 – parte 2)
PUBCOMP	7	Ambos	Publicar conclusão (QoS 2 – parte 3)

Titulo

Tabela 4: Tipos de pacotes utilizados pelo protocolo MQTT – Parte II.

Tipo	Código	Remetente	Descrição
SUBSCRIBE	8	Cliente	Solicitação de inscrição
SUBACK	9	Servidor	Confirmação de inscrição
UNSUBSCRIBE	10	Cliente	Solicitação de cancelamento de inscrição
UNSUBACK	11	Servidor	Confirmação de cancelamento de inscrição
PINGREQ	12	Cliente	Solicitação de PING
PINGRESP	13	Servidor	Resposta de PING
DISCONNECT	14	Ambos	Notificação de desconexão
AUTH	15	Ambos	Troca de autenticação

Titulo I

- **CONNECT** – é o primeiro pacote que deve ser enviado pelo cliente no momento após ter estabelecido um conexão de rede com servidor. Entre as informações enviadas se encontra um identificador único para o cliente, entre outras informações opcionais como usuário e senha.
- **CONNACK** – é o pacote enviado pelo servidor em resposta ao recebimento do pacote CONNECT enviado pelo cliente, e que deve ser recebido antes do pacote AUTH. Em caso do cliente não receber um pacote do tipo CONNACK em um tempo razoável o mesmo deve encerar a conexão com o servidor e tentar novamente.

Titulo II

- **PUBLISH** – são os pacotes encarregados de carregar as mensagens entre o servidor e os clientes podendo ir em ambas as direções.
- **PUBACK** – é o pacote enviado em resposta a um PUBLISH realizado com QoS 1.
- **PUBREC** – é o pacote enviado em resposta a um PUBLISH representa a segunda parte do intercambio de mensagens realizado com QoS 2.
- **PUBREL** – é o pacote enviado em resposta a um PUBREC e representa a terceira parte do intercambio de mensagens realizado com QoS 2.

Titulo III

- **PUBCOMP** – é o pacote enviado em resposta a um PUBREL e representa a quarta e última parte do intercambio de mensagens realizado com QoS 2.
- **SUBSCRIBE** – é o pacote enviado pelo cliente ao servidor como solicitude para a criação de uma ou mais assinaturas, registrando o interesse em receber informação de um ou mais tópicos, junto com o tipo de qualidade do serviço máxima (QoS) desejada com a qual o servidor pode enviar uma mensagem ao cliente.

Titulo IV

- **SUBACK** – é o pacote enviado pelo servidor ao cliente com o objetivo de confirmar o recebimento e processamento de um pacote do tipo SUBSCRIBE. Entre as informações contidas pelo pacote tem uma lista indicando a qualidade do serviço concedida ao cliente para cada tópico e em caso de ter sido negada a subscrição é enviado um sinal justificando o motivo do mesmo. UNSUBSCRIBE

Titulo

Dispositivos e softwares da parte prática

- Esp32:
  - Sensor Adafruit BMP-280;
  - Led embutido no Esp32;
  - C/C++ (Framework Arduino e ESP-IDF);
- Raspberry:
  - Docker executando o broker mosquitto;
  - Cliente inscrito no tópico “#”

Diagrama da aplicação

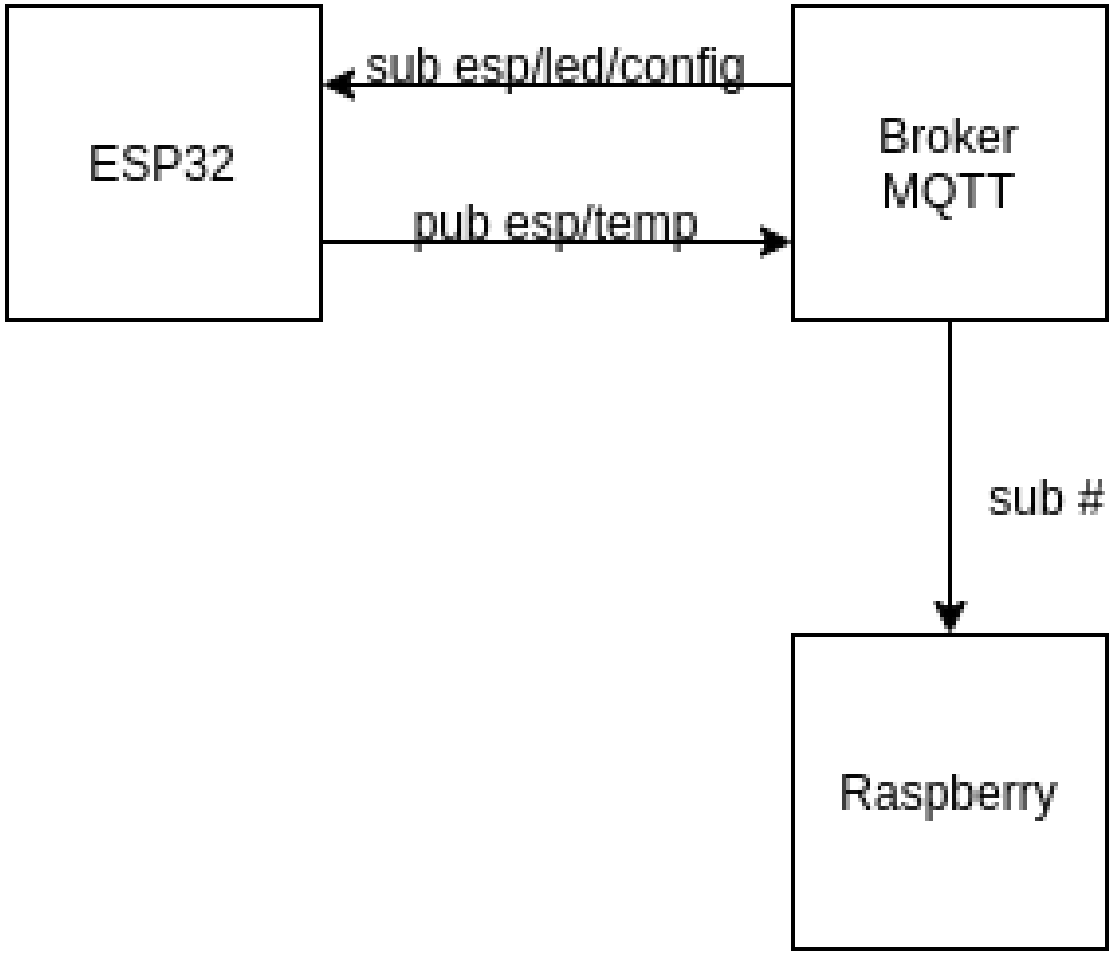

Figura 4: Dispositivos e tópicos utilizados

Códigos Fonte I

```
1  void setup() {
2      if (!sensor.begin(BMP280_ADDRESS)) {
3          if(!sensor.begin(BMP280_ADDRESS_ALT)) {
4              delay(1000);
5              ESP.restart();
6          }
7      }
8      pinMode(LED_PIN, OUTPUT);
9      wifiConnect();
10     MqttConnect();
11     xTaskCreate(taskSendTemperature, "send",
           20000, NULL, 1, &handle);
12 }
```

Códigos Fonte II

```
13
14  void loop() {
15      if (!mqttClient.connected()) {
16          MqttConnect();
17      }
18      mqttClient.loop();
19  }
```

Mão na massa!!

Agradecimentos

Perguntas?

Obrigado pela atenção