

Uma avaliação do Broker Mosquitto no contexto de Internet das Coisas

An evaluation of Broker Mosquitto in the context of Internet of Things

RESUMO

O paradigma de Internet das Coisas permite a qualquer objeto enviar e receber mensagens através da Internet. São diversos os protocolos, padrões e ferramentas envolvidas. Algumas ferramentas requerem assinatura de um plano mensal para permitir o acesso as funções mais avançadas. No presente trabalho será apresentado uma avaliação do broker Mosquitto, de código aberto e gratuito, visando a sua adoção nos projetos de IoT do campus Toledo. Será apresentado o protocolo MQTT e seu funcionamento dentro do broker e a integração do Mosquitto com a rede LoRaWAN, presente no campus.

PALAVRAS-CHAVE: Internet das Coisas. MQTT. Mosquitto.

ABSTRACT

The Internet of Things paradigm allows any object to send and receive messages over the Internet. There are several protocols, standards and tools involved. Some require the subscription of a monthly plan to allow access to more advanced functions. In the present work, an evaluation of the free and open source broker Mosquitto is presented, aiming at its adoption in IoT projects of Campus Toledo. The MQTT protocol and its operation using the broker is presented. Also the Mosquitto integration with the LoRaWAN network, installed on the campus, is described.

KEYWORDS: Internet of Things. MQTT. Mosquitto.

José Eduardo de Souza
josesouza@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Edson Tavares de Camargo
edson@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Fabio Alexandre Spanhol
fspanhol@gmail.com
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Alvaro Ricieri Castro e Souza
alvarosouza@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

O paradigma de Internet das Coisas (Internet of Things - IoT), segundo Gubbi et al. (2013) permite a um dispositivo eletrônico, sem a intervenção humana coletar informações ao seu redor através de sensores especializados e enviá-las através de uma rede de comunicação para a Internet. Isso faz com que as coisas do dia a dia se tornem mais automatizadas e abre um enorme leque de possibilidades. IoT vem sendo aplicado em segmentos como casas, cidades, agricultura e saúde. Envolve conhecimentos que vão desde eletrônica e computação à apresentação dos dados. Para concretizar o conceito de IoT, há diversos padrões, tecnologias e ferramentas para auxiliar nos desenvolvimentos de aplicações. Conhecer todas essas tecnologias é um desafio em si.

A interação entre um dispositivo eletrônico e um cliente, que pode ser uma pessoa ou outro dispositivo eletrônico geralmente ocorre usando um mediador, chamado de broker, através de uma rede de comunicação sem fio usando o protocolo MQTT (Message Queuing Telemetry Transport). O mediador é geralmente hospedado em servidores na rede local ou na nuvem. Entre as redes sem fio destaca-se as redes WiFi (padrão 802.11) e as redes de longo alcance e baixa potência, como o padrão aberto LoRaWAN que opera sobre a tecnologia sem fio LoRa Andrade na Yoo (2019). LoRaWAN é o protocolo que define a rede, bem como implementa suas configurações. MQTT é um protocolo leve empregado para troca de mensagens. Várias ferramentas, como o brokers Ponte¹ e Mosquitto² implementam o protocolo MQTT e atuam como mediadores (brokers) de acordo com modelo de comunicação publicação-assinatura (publish-subscribe).

Apesar de haver várias plataformas abertas que agregam a função de mediador com a gerência de dispositivos e a apresentação dos dados, como a ThingsBoard³, ThingSpeak⁴ e Tago.io⁵ geralmente o uso de funções mais avançadas dessas plataformas exige a assinatura de um plano mensal ou anual. Ou seja, as funções avançadas somente são acessíveis mediante pagamento. A UTFPR/Toledo possui a rede LoRaWAN instalada em seu câmpus, faz uso das plataformas mencionadas, mas procura uma solução aberta, gratuita e que permita o desenvolvimento de soluções mais avançadas em seus projetos IoT, incluindo os desenvolvidos com o padrão LoRaWAN.

O presente trabalho tem o objetivo de avaliar e mostrar a integração entre o Broker MQTT Mosquitto com microcontroladores e a comunicação entre o Mosquitto e a The Things Network (TTN). TTN é uma infraestrutura de rede comunitária para o padrão LoRaWAN. Um estudo de caso onde dois dispositivos eletrônicos se comunicam através da Internet usando o mediador Mosquitto é apresentado. Um exemplo de como o Mosquitto pode ser integrado a TTN também é apresentado.

O presente trabalho foi desenvolvido no contexto do projeto de extensão Disseminação da tecnologia de Internet das Coisas na Região Oeste do Paraná. O objetivo do projeto é propagar a tecnologia de Internet das Coisas (IoT) entre

¹ <https://www.eclipse.org/ponte/>

² <https://mosquitto.org/>

³ <https://thingsboard.io/>

⁴ <https://thingspeak.com/>

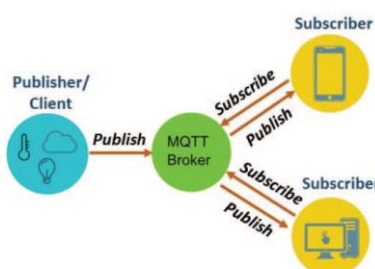
⁵ <https://tago.io/>

acadêmicos, professores e indústrias da região oeste do Paraná a fim de motivar alunos e impulsionar a inovação e a pesquisa por meio da realização de trabalhos de conclusão de curso e desenvolvimento de protótipos.

MATERIAIS E MÉTODOS

MQTT é um protocolo de transporte de mensagens cliente/servidor no conceito de publicação/assinatura como representa a Figura 1, ou seja, uma transmissão parecida com rádio, onde a emissora transmite em uma determinada frequência e os ouvintes sintonizam nela. A ideia é semelhante à usada em redes sociais, onde um usuário “se inscreve” para seguir um determinado usuário e a partir de então passa a visualizar as mensagens daquele usuário em seu próprio perfil. O protocolo é executado sobre o protocolo TCP/IP ou sobre outros protocolos de rede que forneçam conexões bidirecionais ordenadas, sem perda. KODALI & SORATKAL (2016) afirmam que o protocolo é leve e ocupa pouca banda. MQTT é ideal para uso em cenários com recursos restritos, comunicações M2M (Machine-to-Machine), V2V (Vehicle-to-Vehicle) cenários em que são exigidos pacote de dados e/ou largura de banda mínimos.

Figura 1 – Protocolo MQTT



Fonte: Kodali (2017)

MQTT permite níveis de qualidade de serviço, QoS (Quality of Service) para garantia da entrega de uma mensagem. A qualidade do serviço é definida tanto pelo Publisher e pelo Subscriber, impactando diretamente no fluxo de tráfego. São 3 níveis, onde o QoS – 0, “At most once delivery”, ou entrega no máximo uma vez, a mensagem é enviada e não espera um retorno, ideal para conexões estáveis. QoS – 1, “At least once delivery”, ou pelo menos uma entrega, a mensagem é enviada até receber uma confirmação, podendo se duplicar, ideal para ambientes que os dados não podem se perder e não importando a sua duplicação. QoS – 3, “Exactly once delivery”, ou entrega exatamente uma vez, a mensagem é enviada e o mediador confirma o recebimento, encerra a conexão e confirma o encerramento, ideal quando a precisão é crítica.

Eclipse Mosquitto é um broker que implementa o protocolo MQTT. É de código aberto, compatível com plataformas como, Linux, Windows e Raspberry, segundo LIGHT (2017) ele é indicado para o uso em situações onde ocorra troca de mensagens entre dispositivos limitados. Por ser de código aberto, sua integração com serviços da rede LoRa é mais facilitado, uma vez que pode transformar a TTN em um publicador.

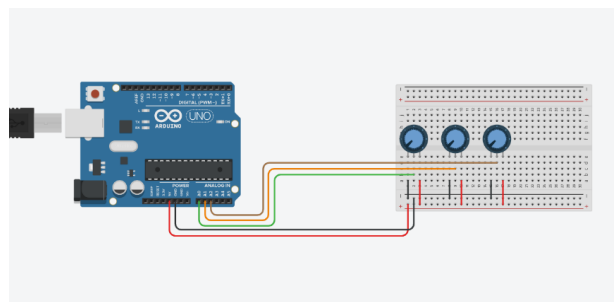
O broker Mosquitto é apenas um intermediador, sendo operado apenas por linha de comando não fornecendo ferramentas visuais completas para a análise dos dados, porém a Eclipse disponibiliza o Paho, que fornece implementação do MQTT em várias linguagens. Isso possibilita a criação de painéis gerenciadores para aplicações de IoT, por exemplo. É possível baixar gratuitamente o Mosquitto, sua instalação é simples em todas as plataformas. Depois de instalado e iniciado ele já estará configurado, podendo testar utilizando aplicativos de teste como o MQTT Box, para publicar em tópicos, por exemplo.

RESULTADOS E DISCUSSÕES

A aplicação desenvolvida para demonstrar o funcionamento do Broker Mosquitto foi um controlador para uma fita de led RGB, a qual é capaz de reproduzir um grande espectro cromático à partir da combinação das cores vermelho, verde e azul. A troca de cores da fita é solicitada através de outro dispositivo. Nota-se que os dois dispositivos estão conectados ao mediador e podem estar em redes diferentes, inclusive se comunicando através da Internet. Com isso um dispositivo envia uma mensagem para que o outro, inscrito no tópico, receba e a interprete.

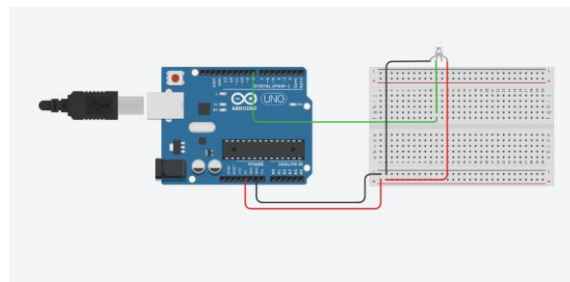
Foi utilizado para o experimento, dois microcontroladores, um ESP32 conectado na rede via Wi-Fi e um Arduino conectado por meio de rede cabeada, utilizando um Shield Ethernet. No Arduino foi conectado a fita de led. No ESP32 há três potenciômetros. Para representação das ligações eletrônicas, foi usado o simulador Tinkercad. Porém alguns dispositivos não estão presente no simulador. No lugar da fita de led, foi utilizado um led RGB convencional, e no lugar do ESP32, um Arduino. A Figura 2 representa o ESP32 e seu esquema, já a Figura 3 o Arduino.

Figura 2 – Microcontrolador 1 com Potenciômetros



Fonte: Autoria própria (2020).

Figura 3 – Microcontrolador 2 com LED RGB

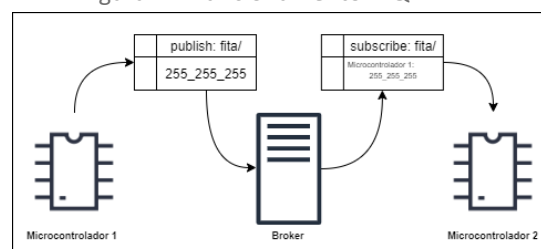


Fonte: Autoria própria (2020).

Para a exibição das cores, deve ser passados três parâmetros, os quais são números entre 0 e 255, respeitando a ordem RGB – “Red, Blue, Green”, Vermelho, azul e verde, respectivamente. O valor passado implica diretamente na intensidade, podendo formar mais de 16 milhões de cores. A manipulação da fita foi feita através da biblioteca NeoPixel da Adafruit, e por potenciômetros que estão responsáveis por variar os valores do RGB, gerando novas cores a cada interação.

Os microcontroladores se comunicam através da rede. Com isso é possível estabelecer uma comunicação entre eles, utilizando o protocolo de mensagem MQTT. Conforme ilustra a Figura 4, um dispositivo publica a mensagem em um tópico e o outro previamente inscrito recebe ela.

Figura 4 – Funcionamento MQTT



Fonte: Autoria própria (2020).

Para obter e enviar as mensagens, os microcontroladores utilizam a biblioteca PubSubClient, que cria a conexão com o broker, gerencia as publicações e inscrições. Na Figura 5, é obtido os valores dos potenciômetros, feito uma divisão para deixar entre os valores da fita de led, concatenados e publicados no tópico “fita”.

Figura 5 – Código do Microcontrolador 1

```

void loop() {
  //Cria a conexão
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  //Divisão para ficar dentro do intervalo 0-255 (cores rgb), pode ser utilizado a função map.
  //Esp32 lê valores analógicos até 4095, arduino até 1023
  r = analogRead(pot_r)/16;
  g = analogRead(pot_g)/16;
  b = analogRead(pot_b)/16;
  //r = analogRead(pot_r)/4; //Utilização no arduino
  //g = analogRead(pot_g)/4;
  //b = analogRead(pot_b)/4;

  //coloca os valores inteiros em um array de char separados por "_"(R_G_B)
  char buf[100];
  sprintf(buf, "%d_%d_%d", r, g, b);

  //verifica se houve alteração nos dados
  if (buf_save != String(buf)) {
    //publica os dados no broker, no canal fita
    client.publish("fita", buf);
  }
  delay(300);
}
  
```

Fonte: Autoria própria (2020).

Figura 6 – Conexão com o Broker e inscrição no tópico

```
//Função padrão da biblioteca
void reconnect() {
  // Estabelece conexão com o broker
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Cria um ID
    String clientId = "Client";
    clientId += String(random(0xffff), HEX);
    //Cria um a conexão
    if (client.connect(clientId.c_str())) {
      //Se inscreve no tópico fita, para obter as informações lá postadas.
      client.subscribe("fita");
    }
    else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
    }
  }
}
```

Fonte: Autoria própria (2020).

Já o microcontrolador 2, utiliza uma função de “callback”, gerenciada pela biblioteca, que retorna o valor obtido das inscrições da aplicação, conforme a Figura 7. O valor é separado, organizado e enviado para a fita de led. O funcionamento é demonstrado no vídeo⁶.

Figura 7 – Função de Callback

```
//Função que obtém as mensagens do broker que vc se inscreveu
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }

  String pl = String((char *)payload);

  String topico = String(topic);

  //verifica o topico
  if(topico == "fita"){
    Serial.println("entrou_fita");
    //Salva os dados obtidos do broker em variaveis
    String part01 = getValue(pl, '_', 0);
    String part02 = getValue(pl, '_', 1);
    String part03 = getValue(pl, '_', 2);

    //Converte o valor para inteiro
    int r = part01.toInt();
    int g = part02.toInt();
    int b = part03.toInt();

    //Seta a cor na fita
    colorWipe(strip.Color(r, g, b), 10);
  }
}
```

Fonte: Autoria própria (2020).

O objetivo não é apenas mudar as cores, mas mostrar como o Broker Mosquitto é capaz de mediar informações entre microcontroladores e outros dispositivos. É possível fazer a integração dele com a TTN de maneira facilitada. Para isso é preciso ter uma aplicação em funcionamento, obter a sua região, seu “AppID” e seu “AppKey” e executar o comando, conforme a Figura 8. A TTN oferece toda documentação para comunicação com um broker MQTT, dando exemplos de inscrições e publicações através da rede LoRa.

⁶ <https://youtu.be/F0Jp1KbxfI4>

Figura 8 – Comando de assinatura de mensagens

```
mosquitto_sub -h <Region>.thethings.network -t "+/devices/+/events/activations" -u "<AppID>" -P "<AppKey>" -v
```

Fonte: The Things Network⁷ (2020)

O Mosquitto se torna ainda mais interessante, uma vez que é possível fazer sua integração com várias linguagens utilizando a biblioteca PAHO. Isso viabiliza a criação de painéis de controle onde os dados podem ser visualizados graficamente, pode ter controles, tabelas, botões entre outras possibilidades programáveis. Com esta integração e com a TTN torna possível a criação de dashboards específicas para aplicações.

CONCLUSÃO

O broker Mosquitto se torna uma ferramenta muito útil para aplicações de IoT. É de fácil instalação, código aberto, gratuito, gera várias possibilidades de implementação e facilita a comunicação entre dispositivos dentro de uma rede. Sua integração é ampla, diversos microcontroladores conseguem se comunicar através dele, além de computadores e smartphones. Atualmente, a rede LoRaWAN do câmpus Toledo faz o uso de plataformas abertas que permitem acesso somente as suas funções básicas. Sendo que algumas, como a ThingsBoard e Tago.io não permite a integração com a rede LoRaWAN ou uso de suas funções especiais sem o pagamento de assinatura. O broker Mosquitto surge como possibilidade de desenvolvimento de uma solução própria para o desenvolvimento de aplicações IoT nos projetos desenvolvidos pelo grupo.

AGRADECIMENTOS

Agradeço ao Professor Edson por confiar e dar a oportunidade de estar neste projeto e a Universidade Tecnológica Federal do Paraná – Campus Toledo, pela bolsa de extensão – PROREC.

⁷ <https://www.thethingsnetwork.org/docs/applications/mqtt/quick-start.html>

REFERÊNCIAS

Kodali, R. K., & Gorantla, V. S. K. (2017). Weather tracking system using MQTT and SQLite. 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). doi:10.1109/icatccT.2017.8389134. Disponível em: <https://ieeexplore.ieee.org/document/8389134> Acesso em: 28 ago. 2020.

Andrade, R. O. and Yoo, S. G. (2019). A comprehensive study of the use of lora in the development of smart cities. Applied Sciences, 9(22):4753. Doi: <https://doi.org/10.3390/app9224753>. Disponível em: <https://www.mdpi.com/2076-3417/9/22/4753>. Acesso em: 31 ago. 2020.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. doi:10.1016/j.future.2013.01.010. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0167739X13000241> Acesso em: 28 ago. 2020.

KODALI, R. K., & SORATKAL, S. (2016). MQTT based home automation system using ESP8266. 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). doi:10.1109/r10-htc.2016.7906845. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7906845> Acesso em: 29 ago. 2020.

Light, (2017). Mosquitto: server and client implementation of the MQTT protocol. Journal of Open Source Software, 2(13), 265, doi:10.21105/joss.00265 LENCAR, L. H. Disponível em: <https://joss.theoj.org/papers/10.21105/joss.00265>. Acesso em: 31 ago. 2020

MURATORI, J.R. Os desafios do mercado de Automação Residencial. 2013. Disponível em: https://www.aecweb.com.br/cont/a/os-desafios-do-mercado-da-automacao-residencial_8192. Acesso em: 28 ago. 2020.