

Arquitetura Baseada em Internet das Coisas para Medição e Monitoramento de Resíduos

Samuel Silva de Oliveira^{1,2}, Janine Kniess²

¹ Departamento de Ciências Exatas e Tecnológicas
Universidade Federal do Amapá (UNIFAP) – Macapá, AP – Brasil

² Programa de Pós-Graduação em Computação Aplicada
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brasil

samuel.oliveira@unifap.br, janine.kniess@udesc.br

Abstract. *This paper presents as a contribution an architecture for measurement and monitoring of waste. The architecture is divided in three layers: Physical devices, Communication interface and Application. The Message Queue Telemetry Transportation protocol (MQTT) and the Constrained Application Protocol (CoAP) were used as platforms to provide the services of Internet of Things (IoT) that monitor collected data from sensors installed in the bins.*

Keywords: *Waste Management, Monitoring, Internet of Things, Communication Protocols, Sensors*

Resumo. *Este artigo apresenta como contribuição uma arquitetura funcional para a medição e monitoramento do nível de resíduos em lixeiras. A arquitetura é dividida em três camadas: Dispositivos físicos, Interface de comunicação e Aplicação. O protocolo Message Queue Telemetry Transportation (MQTT) e o protocolo Constrained Application Protocol (CoAP) foram utilizados como plataformas para prover os serviços de Internet das Coisas (IoT) que monitoram os dados dos sensores instalados nas lixeiras.*

Palavras chave: *Gestão de Resíduos, Monitoramento, Internet das Coisas, Protocolos de Comunicação, Sensores*

1. Introdução

Dentre os grandes desafios urbanos atuais, está o gerenciamento de resíduos sólidos. A alta concentração urbana traz para as cidades a necessidade de uma gestão mais eficiente de resíduos sólidos a fim de alcançar maior economia, menor impacto ambiental e melhor qualidade de vida para os cidadãos. Uma das formas de aumentar a eficiência da gestão de resíduos é monitorar o nível das lixeiras públicas. Dessa forma, seria possível otimizar o serviço de coleta desses resíduos. Porém, medir a quantidade de resíduos depositados em lixeiras por um número significativo de pessoas é uma tarefa muito difícil. Para atingir esse objetivo, [Bouskela 2016] define alguns requisitos que precisam ser alcançados: Os coletores de resíduos sólidos deverão ser conectados por redes sem fio e equipados com sensores que monitoram o volume do resíduo, a umidade e a temperatura; Os dados coletados deverão chegar aos órgãos de limpeza para permitir um melhor planejamento das rotas de coleta, atualizando os motoristas dos caminhões em tempo real em relação aos percursos, resultando na otimização do custo do serviço de gestão de resíduos.

Implantar uma arquitetura que contemple todos os requisitos citados acima pode ser uma tarefa complexa. Uma das tecnologias que facilita o desenvolvimento desta arquitetura é a Internet das Coisas¹. Em Internet das Coisas, tecnologias como redes de sensores sem fio, Bluetooth, WiFi, *Radio-Frequency Identification* (RFID), e outros, estão emergindo para que novos sistemas, tais como o proposto neste artigo, sejam produzidos.

De acordo com o exposto, é possível entender a importância de utilizar os conceitos de Internet das Coisas para que o processo de coleta de lixo seja mais inteligente, através da medição do nível de resíduos em lixeiras e a apresentação dos dados coletados ao usuário. Os órgãos de gerenciamento de resíduos sólidos seriam consideravelmente beneficiados por soluções alternativas que possibilitem a redução dos custos da gestão de resíduos através de novos métodos implementados por tecnologias modernas. Segundo [Anagnostopoulos et al. 2017], a gestão eficaz dos resíduos tem um impacto significativo na qualidade de vida dos cidadãos. A razão é que a eliminação de resíduos reduz os impactos negativos no ambiente e, portanto, na saúde das pessoas.

Este artigo apresenta como contribuição uma arquitetura no campo de Internet das Coisas para monitoramento e apresentação do nível de resíduos em lixeiras públicas. O restante deste trabalho está dividido da seguinte forma: Na Seção 2 apresenta-se os trabalhos relacionados; Na Seção 3 demonstra-se a arquitetura para medição de resíduos em lixeiras; Na Seção 4 apresentam-se os aspectos da implementação da arquitetura; e por fim, na Seção 5, as conclusões e possíveis trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção serão discutidos alguns trabalhos da literatura que lidam com o problema do monitoramento do nível de resíduos em lixeiras.

[H. and B. 2016] descrevem um modelo de medição do nível de lixeiras utilizando Arduino Uno, sensor ultrassônico HCSR-04 e um módulo *Global System for Mobile Communications* (GSM) para o envio de mensagens através do Serviço de Mensagens Curtas². Estes dispositivos são definidos pelos autores como escravos. Medições são realizadas nas lixeiras e quando o volume de resíduos chega a um limite pré-determinado, uma mensagem do tipo SMS é enviada a um equipamento denominado Mestre localizado em uma sala de comando. Através do recebimento de um SMS a interface gráfica do sistema proposto é atualizada, informando aos usuários que a lixeira está cheia. Embora esta solução atenda às necessidades de medição do nível de resíduos, a utilização de módulos GSM pode encarecer o projeto. Em larga escala, essa solução pode gerar um custo expressivo.

[Hannan et al. 2011] desenvolveram um sistema baseado em *Radio-Frequency Identification* (RFID) para gestão de resíduos. Outras tecnologias são abordadas, tais como: *Global Positioning System* (GPS), *General Packet Radio Service* (GPRS) e *Geographic Information System* (GIS), juntamente com tecnologias de câmeras do tipo *Red-Green-Blue* (RGB) a fim de compor um sistema de monitoramento inteligente de lixeiras e caminhões de lixo. Câmeras RGB de baixo custo são usadas para o monitoramento das lixeiras, acoplado na parte superior do caminhão de lixo junto com um leitor RFID para obter as imagens. O leitor de RFID e a câmera associada ao módulo de GPS são conectados com o caminhão. Cada lixeira contém uma etiqueta (*tag*) RFID que é lida quando

¹Do Inglês: Internet of Things - IoT

²Do inglês: *Short Message Service* (SMS)

o caminhão se aproxima. Neste momento a câmera captura o nível de resíduos na lixeira, armazena as informações na *tag* RFID e no banco de dados. Outra funcionalidade do sistema é a definição de rotas para os caminhões de lixo através dos dados coletados. Essa abordagem não objetiva o monitoramento remoto do nível das lixeiras, e sim a captação de dados *in loco* que sirvam de indicadores sobre a gestão dos resíduos a fim de alcançar maior eficiência. Além disso, variedade de tecnologias envolvidas pode elevar o custo da sua implantação.

[Mamun et al. 2013] projetaram um sistema de monitoramento de resíduos sólidos no qual redes de sensores sem fio são utilizadas em conjunto com diferentes tecnologias de comunicação, tais como: Zigbee, GSM e GPRS para monitorar o nível dos resíduos sólidos. Neste projeto, o nó sensor é composto por um microcontrolador Waspote, um sensor ultrassônico e uma bateria acoplada. Um algoritmo de otimização foi desenvolvido para que o nó sensor gaste a menor quantidade de energia possível. Os dados coletados pelo nó sensor são enviados primeiramente a um gateway Zigbee e posteriormente a um servidor web através da rede GSM/GPRS. Os dados inseridos no sistemas também são utilizados para a otimização das rotas dos caminhões de lixo. A arquitetura proposta neste artigo baseia-se no uso de nós sensores acoplados em lixeiras compostos por um microcontrolador e um sensor de distância a fim de medir o nível de resíduos.

Os principais benefícios da proposta apresentada neste trabalho estão relacionados com o baixo custo de implantação e facilidade no monitoramento remoto. Os dados coletados pelos nós sensores poderão ser acessados de qualquer lugar e a qualquer momento, pois os mesmos estarão disponíveis na nuvem para a consulta por parte dos usuários. Na Seção seguinte apresenta-se a arquitetura para monitoramento e medição de resíduos.

3. Arquitetura de IoT para Monitoramento de Resíduos

Um modelo de organização é necessário para que a relação entre os componentes de hardware e software envolvidos no projeto sejam definidos. Nesse sentido, propomos um modelo horizontal em camadas para apresentação de todos os componentes da arquitetura. Baseado neste modelo, novos dispositivos podem ser inseridos no ambiente de um modo rápido e seguro.

O modelo de organização proposto é apresentado na Figura 1. As camadas definidas são: Dispositivos físicos, Interface de comunicação e Aplicação. Em uma visão de baixo para cima, têm-se na primeira camada os dispositivos de hardware: Microcontroladores, sensores, pontos de acesso e gateways. Na camada 2 foram alocados os protocolos de comunicação voltados para Internet das Coisas, como o MQTT [MQTT.org 2017] (*utiliza o TCP*) e CoAP [Shelby 2016] (*utiliza o UDP*). Na camada 3 estão, finalmente, os componentes de software com maior nível de abstração, como o Mosquitto [Light 2017], Paho [Eclipse 2016], CoAPthon [Tanganelli et al. 2015] e Copper [Kovatsch 2011]. Mais detalhes serão abordados posteriormente sobre cada um deles.

O modelo em camadas proposto por este trabalho objetiva reduzir a complexidade da arquitetura. O modelo define regras para a comunicação entre as camadas, gerando maior independência de cada nível. Com base nessa organização, é possível introduzir novas funcionalidades em cada camada quando necessário, sem que as demais sejam afetadas. Os detalhes sobre cada nível da arquitetura serão descritos na seção subsequente.

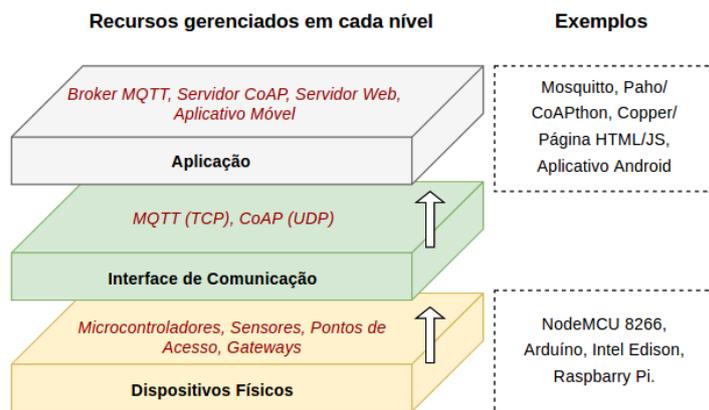


Figura 1. Modelo em camadas do sistema de monitoramento de resíduos

3.1. Camada de Dispositivos Físicos

A composição de um nó sensor, de acordo com [Vieira et al. 2003], é formada pelos seguintes elementos: unidade de comunicação, unidade de energia, unidade de sensoriamento e unidade de computação. No âmbito da arquitetura proposta, a unidade de computação é composta por um microcontrolador, enquanto a unidade de sensoriamento é composta por um sensor de distância, a fim de realizar a medição da distância do sensor até os resíduos nas lixeiras.

Os sensores ultrassônicos geralmente são capazes de medir distâncias de 2cm a 4m com boa precisão. O sensor emite uma onda sonora que é refletida ao encontrar algum obstáculo e conta com um pino ECHO que fica em nível lógico alto quando recebe a reflexão. Dessa forma, é possível determinar a distância do sensor para o obstáculo de acordo com o tempo em que o pino ECHO permaneceu em nível alto após a onda inicial ter sido emitida. A distância pode ser obtida através da fórmula: $d = (t * v)/2$, sendo d a distância que se quer obter, t , o tempo em que o pino ECHO fica em nível lógico alto e v a velocidade do som [ElecFreaks 2017].

Outra possibilidade é a utilização de um sensor de distância infravermelho (IR). O sensor infravermelho possui um módulo transmissor e um módulo receptor. O conceito básico é transmitir o sinal infravermelho em uma direção específica, e então, quando a radiação infravermelha retorna, o sinal é recebido no módulo receptor. Quanto maior a distância do objeto, menor será o sinal recebido [Tsai et al. 2013].

3.2. Camada de Interface de Comunicação

Com o propósito de realizar o monitoramento do nível de resíduos nas lixeiras, é preciso definir uma forma de enviar os dados coletados pelo nó sensor para uma nuvem computacional. Esta é a tarefa da Camada de Interface de Comunicação.

Na Internet, o protocolo *Hipertext Transfer Protocol* (HTTP) é amplamente utilizado para envio e recebimento de dados seguindo o modelo requisição/resposta. Porém, o protocolo HTTP não é adequado para a Internet das Coisas quando uma grande quantidade de pacotes pequenos devem ser transmitidos, pois seu uso em aplicações de IoT pode degradar significativamente o desempenho da rede [Karasiewicz 2013]. Para esse fim,

existem dois protocolos que têm sido amplamente utilizados em ambientes de IoT: *Message Queue Telemetry Transport* (MQTT) e *Constrained Application Protocol* (CoAP).

O MQTT é um protocolo máquina-a-máquina³ de mensagens para sensores e dispositivos móveis, projetado para lidar com baixa largura de banda, alta latência e instabilidade na comunicação [Torres et al. 2016]. O protocolo MQTT define três níveis de Qualidade de Serviço⁴ que vai de 0 (nenhuma garantia de entrega) a 2 (garantia de entrega sem duplicidade). A qualidade de serviço (QoS) define com qual nível de persistência as mensagens serão entregues ao destinatário [Torres et al. 2016]. As mensagens podem ser enviadas com qualquer um dos três níveis de QoS.

O transporte de mensagens no protocolo MQTT é do tipo *publicador/subscritor*⁵ e é útil para conectar dispositivos de baixa capacidade em locais remotos. Na esfera da arquitetura proposta, o publicador é representado pelo microcontrolador (Camada de Dispositivos Físicos) e o subscritor é representado pela página web/aplicativo (Camada de Aplicação).

O paradigma *publicador/subscritor* utiliza o conceito de tópicos para processar as mensagens, sendo que cada mensagem é enviada para um determinado tópico. Nesse sentido, o microcontrolador não envia mensagens diretamente ao servidor web, mas a um *broker* que gerencia as mensagens pelos seus tópicos. O *broker* é um software da Camada de Aplicação na arquitetura, responsável por receber, enfileirar e enviar as mensagens recebidas dos microcontroladores para a interface web. Existem diversas implementações de *brokers* MQTT disponíveis atualmente. Utilizou-se o *broker* Mosquitto [Light 2017] neste trabalho.

Outra possível solução para o transporte de dados da Camada de Dispositivos Físicos para a Camada de Aplicação na arquitetura é a utilização do Protocolo CoAP. Este protocolo é utilizado em Internet das Coisas para estabelecer a comunicação entre dispositivos com baixo poder computacional [Santos et al. 2016]. Sua arquitetura interna é baseada no paradigma *REpresentational State Transfer* (REST) que acessa a recursos por meio de *Uniform Resource Identifiers* (URIs).

O protocolo CoAP utiliza o acesso à métodos similares aos do HTTP, tais como: GET, POST, PUT e DELETE. Porém, o protocolo de transporte utilizado pelo CoAP é o UDP, o tornando mais adequado para ambientes de IoT [Florea et al. 2017]. As mensagens CoAP são divididas em quatro categorias: *confirmable*, *non-confirmable*, *reset* e *acknowledgement*. Embora o UDP seja um protocolo de transporte não confiável, é possível obter maior nível de confiabilidade através da utilização de mensagens CoAP do tipo *confirmable*.

No contexto da arquitetura, o microcontrolador envia mensagens CoAP para um Servidor CoAP (Camada de Aplicação) através do método POST. A partir de então, é possível que um cliente CoAP (Camada de Aplicação) requisiute a mensagem através do método GET, obtendo o valor da distância aferida pelo nó sensor.

³Do Inglês: *Machine-to-Machine* (M2M)

⁴Do Inglês: *Quality of Service* (QoS)

⁵Do inglês: *publish/subscribe*

3.3. Camada de Aplicação

A Camada de Aplicação concentra os componentes de software utilizados para apresentar os dados do nível de resíduos aos usuários. Dentre os softwares, citam-se o Broker Mosquitto, Paho, CoAPthon, Copper, uma interface web e um aplicativo móvel.

3.3.1. Gerenciamento e Visualização de Mensagens MQTT e CoAP

Na arquitetura, o *broker* Mosquitto é utilizado para o gerenciamento das mensagens MQTT. Ele fornece implementações de servidor e cliente compatíveis com padrões do protocolo MQTT [Light 2017]. [Torres et al. 2016] constatou em seus testes que o Mosquitto apresentou maior desempenho comparado a outros *brokers* nos quesitos utilização da CPU e consumo de memória. Tais resultados motivaram a sua utilização e avaliação neste trabalho.

Nos experimentos apresentados na Seção 4, um publicador foi adicionado no microcontrolador que publica as leituras da distância do sensor aos resíduos dentro de uma lixeira em um tópico previamente definido. Para visualizar estes dados, é necessário que um cliente MQTT atue como subscriber no referido tópico para receber as informações enviadas pelo publicador. Para que esses dados estejam disponíveis em um meio de comunicação interoperável, utilizou-se o Paho [Eclipse 2016].

Através do Paho é possível criar um cliente MQTT em diversas linguagens de programação, tais como: Python, Java, Javascript, GoLand, C e .NET [Eclipse 2016]. Neste trabalho, foi utilizado um cliente MQTT em Javascript numa página web capaz de subscrever a quaisquer tópicos. Na Figura 2, ilustra-se a sequência de funcionamento do *broker* Mosquitto em conjunto com o Paho permitindo a visualização das mensagens publicadas pelo nó sensor nos tópicos pré-definidos através da Web.

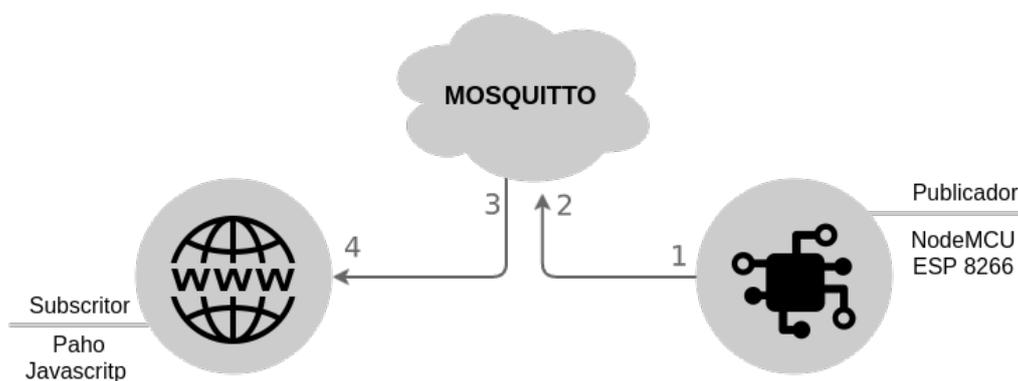


Figura 2. Funcionamento do Broker MQTT. Fonte[Elaborado pelos autores]

Conforme descrito na Figura 1, o protocolo CoAP pode ser utilizado no lugar do MQTT para receber as mensagens enviadas pelos microcontroladores instalados nas lixeiras. Para receber as mensagens CoAP enviadas pelos microcontroladores, utilizou-se o CoAPthon. O CoAPthon é uma biblioteca desenvolvida na linguagem Python, e segundo [Tanganelli et al. 2015], o CoAPthon oferece aos desenvolvedores uma interface de programação simples para que o CoAP seja explorado como um protocolo de

comunicação em rápidas prototipações e implantações de sistemas de Internet das Coisas. Para a exibição ao usuário final, o Copper [Kovatsch 2011] foi adotado. Através do Copper, é possível utilizar os recursos do servidor CoAP para o gerenciamento da comunicação entre dispositivos de forma simples e intuitiva.

4. Análise Experimental

Com o objetivo de avaliar a viabilidade da arquitetura apresentada anteriormente, desenvolveu-se um protótipo para o monitoramento de resíduos em lixeiras. Para o desenvolvimento do protótipo, utilizou-se o microcontrolador NodeMCU ESP8266 [NodeMCU 2014] e o sensor ultrassônico HCSR-04 [ElecFreaks 2017].

No protótipo, o microcontrolador NodeMCU ESP8266 corresponde tanto a unidade de computação quanto a unidade de comunicação sem fio do nó sensor. A unidade de sensoriamento é composta pelo sensor ultrassônico HCSR-04. A unidade de energia, por sua vez, foi composta por uma bateria de polímero de lítio⁶ de 36000 miliampere-hora (mAh). Utilizou-se um ponto de acesso sem fio (IEEE 802.11 a/b/g/n) para conectar o nó sensor à internet visando o transporte dos dados coletados para a nuvem. A Figura 3 mostra o nó sensor acoplado em uma lixeira.



Figura 3. Protótipo do Sistema de Monitoramento de Resíduos

Com a finalidade de oferecer ao usuário soluções para a visualização dos dados do monitoramento, uma página web e um aplicativo móvel para Android foram desenvolvidos. A página web e o aplicativo móvel são baseados em HTML5 [Berjon et al. 2014] e contêm um subscritor MQTT implementado em Javascript (Paho). A Figura 4 ilustra a interface da página web. Na página/aplicativo é possível subscrever a um tópico MQTT ou a uma raiz de tópicos. Para cada tópico subscrito, uma barra de progresso é exibida correspondente ao valor da medição da lixeira (de 0% até 100%).

Nos casos em que o protocolo CoAP for utilizado, os dados do monitoramento podem ser visualizados através do Copper. O Copper utiliza o endereço do Servidor CoAP seguido do nome do recurso empregado para o envio das mensagens. Dessa forma é possível requisitar ao Servidor CoAP as mensagens enviadas pelo microcontrolador.

4.1. Análise de Desempenho do MQTT e CoAP

Foram realizados testes com o MQTT e o CoAP para as seguintes métricas: *Round-Trip Time* (RTT), sobrecarga e tamanho dos pacotes. Nos experimentos utilizou-se uma amostragem de 2000 mensagens enviadas ao *broker* MQTT com QoS 0 através de uma rede

⁶Do Inglês: *Lithium polymer battery* (LiPo)

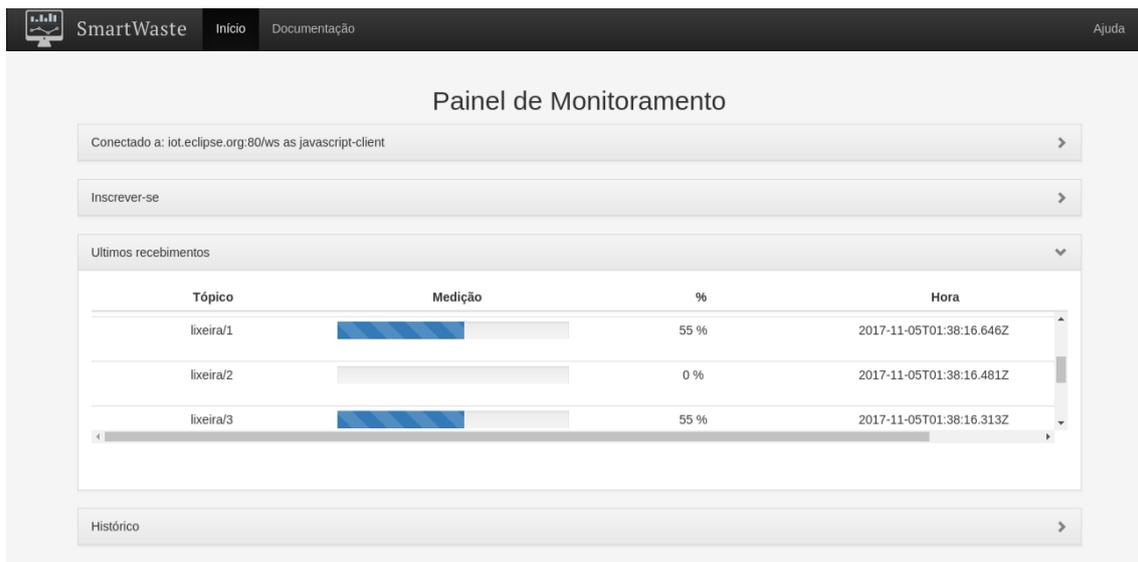


Figura 4. Página Web para monitoramento das lixeiras

local *Wi-Fi* (IEEE 802.11 a/b/g/n). A mesma amostragem foi utilizada para as mensagens CoAP do tipo CON.

O Wireshark [Orebaugh et al. 2007] foi utilizado para capturar os pacotes decorrentes da comunicação CoAP e MQTT. Com base na análise dos resultados, observou-se que o RTT aferido nas mensagens MQTT foi de 22,15 ms, considerando algumas retransmissões TCP. Com o CoAP, o RTT foi em média 3,86 ms. Constatou-se ainda que o CoAP gerou pacotes de 55 bytes, enquanto os pacotes MQTT têm em média 82,9 bytes.

Com relação à sobrecarga, identificou-se que as mensagens MQTT representaram 99,6% do tráfego da rede, enquanto as mensagens CoAP representaram 71,6%. Este resultado justifica-se pelo fato de que, para cada mensagem CoAP enviada, dois pacotes são trafegados: CON e ACK. No caso do MQTT, para cada mensagem enviada, quatro pacotes são criados: Connect Command, Connect Ack, Publish Message e Disconnect Req, além dos pacotes TCP de sincronismo. Nos gráficos da Figura 5, ilustram-se o RTT (a), tamanho médio dos pacotes (b) e sobrecarga (c).

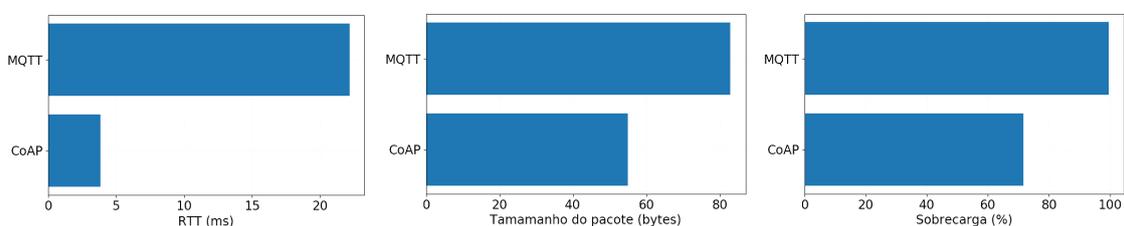


Figura 5. Avaliação de desempenho dos protocolos MQTT e CoAP

Com base nos resultados obtidos, é possível concluir que ambos os protocolos analisados têm características vantajosas para Internet das Coisas. Embora o protocolo MQTT gere um nível considerável de sobrecarga, é possível utilizá-lo facilmente na troca de mensagens muitos-para-muitos com bom desempenho. O protocolo CoAP, por sua vez, pode ser utilizado com maior vantagem em dispositivos com maior restrição de

comunicação e energia, principalmente por operar sobre o protocolo UDP na camada de transporte e gerar pacotes menores que o MQTT.

5. Conclusão e Trabalhos Futuros

Com as tecnologias descritas neste trabalho, identificou-se que é possível tornar a gestão de resíduos mais inteligente, otimizando o processo de coleta dos resíduos e consequentemente, reduzindo custos, o impacto ambiental e melhorando a qualidade de vida dos cidadãos. Em média, o custo dos equipamentos de cada nó sensor utilizado no protótipo foi de aproximadamente R\$ 50,00. Nós sensores formados por outros componentes, como Arduíno e Sensor Infravermelho também possuem custo similar. O baixo custo do protótipo torna ainda mais fácil sua implantação.

Como perspectivas de trabalhos futuros, sugere-se que seja utilizado um banco de dados para armazenar os dados históricos das lixeiras. No cenário atual, as mensagens MQTT estão sendo enviadas pelos nós sensores e o monitoramento pode ser realizado em tempo real. Porém, a análise de dados históricos pode ser interessante para a geração de indicadores relacionados à quantidade de resíduos descartados e a frequência da coleta. Uma possível linha de pesquisa é a utilização de algoritmos baseados em Inteligência Artificial, como Redes Neurais, Algoritmos Genéticos ou Programação Genética, para a realização de análises nos dados históricos visando identificar padrões e realizar previsões futuras que colaborem com a elaboração de um plano eficiente para o gerenciamento de resíduos. Sugere-se ainda que seja incluído um módulo de GPS em cada nó sensor para que através da localização de cada lixeira uma rota seja criada a fim de otimizar o serviço de coleta.

Referências

- Anagnostopoulos, T., Zaslavsky, A., Kolomvatsos, K., Medvedev, A., Amirian, P., Morley, J., and Hadjiefthymiades, S. (2017). Challenges and opportunities of waste management in IoT-enabled smart cities: A survey. *IEEE Transactions on Sustainable Computing*, pages 1–1.
- Berjon, R., O'Connor, T., Faulkner, S., Leithead, T., Pfeiffer, S., Hickson, I., and Navara, E. D. (2014). HTML5. W3C recommendation, W3C. <http://www.w3.org/TR/2014/REC-html5-20141028/>.
- Bouskela, M. (2016). The road toward smart cities: Migrating from traditional city management to the smart city. Technical report.
- Eclipse (2016). Paho - open source messaging for m2m. <http://www.eclipse.org/paho/>. Acessado em: 18/06/2017.
- ElecFreaks (2017). Ultrasonic ranging module hc - sr04. <http://www.micropik.com/PDF/HCSR04.pdf>. Acessado em: 18/06/2017.
- Florea, I., Rughinis, R., Ruse, L., and Dragomir, D. (2017). Survey of standardized protocols for the internet of things. In *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, pages 190–196.
- H., K. M. and B., S. S. (2016). A novel approach to garbage management using internet of things for smart cities. *International Journal of Current Trends in Engineering Research*, 2(5):348–353.

- Hannan, M., Arebey, M., Begum, R., and Basri, H. (2011). Radio frequency identification (RFID) and communication technologies for solid waste bin and truck monitoring system. *Waste Management*, 31(12):2406–2413.
- Karasiewicz, C. (2013). Why http is not enough for the internet of things. https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en. Acessado em: 10/10/2017.
- Kovatsch, M. (2011). Demo abstract: Human-coap interaction with copper. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–2.
- Light, R. A. (2017). Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, 2(13).
- Mamun, M. A. A., Hannan, M. A., Hussain, A., and Basri, H. (2013). Wireless sensor network prototype for solid waste bin monitoring with energy efficient sensing algorithm. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 382–387.
- MQTT.org (2017). Mqtt. https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_http_is_not_enough_for_the_internet_of_things?lang=en. Acessado em: 10/10/2017.
- NodeMCU (2014). Nodemcu documentation. <https://nodemcu.readthedocs.io/en/master/>. Acessado em: 18/06/2017.
- Orebaugh, A., Ramirez, G., Beale, J., and Wright, J. (2007). *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Syngress Publishing.
- Santos, B. P., Silva, L. A. M., Celes, C. S. F. S., Neto, J. B. B., Peres, B. S., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. A. F. (2016). Internet das coisas: Da teoria à prática.
- Shelby, Z. (2016). Rfc 7252-the constrained application protocol (coap). internet engineering task force (ietf)(2014). Acessado em: 10/10/2017.
- Tanganelli, G., Vallati, C., and Mingozi, E. (2015). Coapthon: Easy development of coap-based iot applications with python. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 63–68.
- Torres, A. B. B., Rocha, A. R., and de Souza, J. N. (2016). Análise de desempenho de brokers mqtt em sistema de baixo custo. In *Anais do XXXVI congresso da sociedade brasileira de computação*. Sociedade Brasileira de Computação.
- Tsai, C. H., Bai, Y. W., Lin, Y. W., Lin, K. S., Cheng, L. C., and Jhang, R. J. R. (2013). Using an ir sensor module to reduce the standby power consumption of a pc monitor. In *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–6.
- Vieira, M., Coelho, C., da Silva, D., and da Mata, J. (2003). Survey on wireless sensor network devices. In *EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696)*. IEEE.