

IoT Protocols Comparison for Wireless Sensors Network Applied to Marine Environment Acoustic Monitoring

G. Durante, W. Beccaro, H. E. M. Peres

Abstract—There is a growing interest to analyze acoustic marine environment. Underwater sounds can indicate the natural marine activity, as well as the influence of human activities in the aquatic ecosystems. Several kinds of electronic devices can be used to monitor the marine sound, but normally these devices work autonomously, without communication with other equipment, which makes difficult the real-time monitoring. In addition, the usual equipment is normally expensive and highly complex. Thus, for the design of wireless sensor network (WSN) applied to marine monitoring, it is important to develop systems with good energy autonomy and efficient data processing and communication. However, it is not well established the best WSN communication protocol for applications in hostile environments. Some protocols used for Internet of Things (IoT) applications have emerged recently as alternative for efficient communication between nodes and gateways. This work presents a comparison between two IoT application protocols: CoAP and MQTT-SN. We have verified that the MQTT-SN protocol has some advantages, such as: its latency is 30% lower compared with CoAP, the power consumption is 10% lower than CoAP, and the traffic flow is 2.15 times larger than CoAP, for an architecture with 40 wireless nodes.

Index Terms—Internet of Things, Underwater technology, Machine-to-machine, Acoustic sensors, Wireless sensor networks.

I. INTRODUÇÃO

OCEANOS são ambientes repletos de sons naturais, biológicos ou ambientais, e não-naturais, decorrentes de atividades humanas [1], [2]. Diferentes animais e ambientes marinhos apresentam, assim, “assinaturas” sonoras específicas que permitem definir o conceito de paisagem acústica marinha [3], [4]. Esses sons são indicadores de condições adequadas para o crescimento, alimentação e orientação espacial de diversas espécies marinhas [1]. Existem evidências crescentes de que sons muito intensos associados ao aumento das atividades antropogênicas prejudiquem a comunicação e o reconhecimento de ambientes favoráveis à manutenção do ciclo de vida dos organismos [2].

G. Durante, Departamento de Engenharia de Sistemas Eletrônicos, Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, gbdurante@gmail.com.

W. Beccaro, Departamento de Engenharia de Sistemas Eletrônicos, Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, wesley@lme.usp.br.

H. E. M. Peres, Departamento de Engenharia de Sistemas Eletrônicos, Escola Politécnica da Universidade de São Paulo (USP), São Paulo, Brasil, hperes@lme.usp.br.

Apesar de sua importância para elucidar processos ecológicos marinhos, essas paisagens ainda não puderam ser plenamente caracterizadas para a maioria dos ambientes costeiros e estuarinos devido à lacuna de sistemas eletrônicos adequados às peculiaridades do ambiente marinho.

Para o monitoramento do som nesse ambiente, alguns requisitos são essenciais. O sistema necessita de módulos sensores distribuídos e que não afetem o tráfego de embarcações e de animais. Assim, é necessária a implementação de uma rede de sensores sem fio (RSSF). Além disso, os módulos devem possuir boa autonomia e eficiência energética para garantir sua operação ao longo de dias ou até meses [4], [5]. Em específico, os protocolos de comunicação de dados devem ser robustos para viabilizar a inclusão do contexto de Internet das Coisas (IoT), permitindo a intercomunicação entre os dispositivos do sistema não só para o envio em tempo real dos dados sonoros coletados, mas também para possibilitar análises dos dados coletados ou reconfigurações nos dispositivos do sistema. Para esse caso, convém utilizar protocolos que transmitam a informação coletada sem que haja o controle de erros, ou seja, protocolos não orientados para a conexão [6]-[9].

Existem diversos protocolos para aplicações em IoT que possam atender às necessidades do desenvolvimento de uma rede de sensores em ambiente marinho, destacando-se o *Message Queue Telemetry Transport for Sensor Networks* (MQTT-SN) [10] e o *Constrained Application Protocol* (CoAP) [11]-[13]. Embora alguns trabalhos tenham discutido as diferenças entre os protocolos MQTT e CoAP [14], [15] e, sucintamente, entre o MQTT-SN e CoAP [16], não há uma análise dos principais parâmetros de comunicação e de outras características aplicadas a nós sensores. Para isso, é necessário realizar uma comparação entre seus desempenhos em termos de latência, tempo de tráfego na rede, quantidade de dados transmitidos, consumo de energia dos nós, entre outras características.

Dessa forma, é imprescindível verificar o comportamento de uma rede de sensores sem fio e os meios de transporte de dados nela presentes, garantindo a entrega das características monitoradas. Este trabalho apresenta uma análise comparativa entre os protocolos MQTT-SN e CoAP utilizando o simulador COOJA, para um cenário de monitoração de paisagens acústicas marinhas. Para tanto, foi utilizada a implementação nativa do protocolo CoAP, mas foi necessário implementar o protocolo MQTT-SN e suas camadas no simulador. Dado que

no sistema operacional de rede de sensores Contiki-OS, amplamente utilizado em nós sensores, não há implementação nativa do protocolo MQTT-SN.

II. ARQUITETURA IOT E PROTOCOLOS DE COMUNICAÇÃO

A. Arquitetura IoT

Existem, na atualidade, diferentes modelos de arquitetura de IoT [17]-[20]. Desde os modelos mais simplificados, com três camadas: camada de aplicação (*Application Layer*), camada de rede (*Network Layer*), e camada de percepção (*Perception Layer*), até modelos com cinco camadas [20], [21]: camada de negociação (*Business Layer*), camada de aplicação, camada de gerenciamento de processos (*Service Management*), camada de abstração de objetos (*Object Abstraction*) e camada de objetos (*Objects*), comparados na Fig.1

A camada *Perception Layer* ou *Objects* inclui os sensores e os atuadores. A parte física dos transdutores é responsável por coletar e processar os dados. Essa camada também contempla protocolos de rede de sensores para garantir a padronização através do conceito de transdutores *plug and play* [20], [21].

A camada *Network Layer* ou *Object Abstraction* é responsável por transferir os dados utilizando canais de comunicação seguros. Dentre as tecnologias dessa camada estão: 3G, *Groupe Special Mobile* (GSM), *Universal Mobile Telecommunications System* (UMTS), WiFi, Bluetooth, *Radio-Frequency IDentification* (RFID), etc.

A camada *Application Layer* na arquitetura de 3 camadas abstrai três blocos na arquitetura de 5 camadas. A camada de *Service Management* é responsável pelo serviço de requisição de dados baseado em endereçamentos. Nessa camada, os pacotes são direcionados de acordo com a demanda de serviços dos nós IoT ou do servidor. Na arquitetura de 5 camadas, a camada de aplicação é responsável por prover serviços de acesso aos dados ao usuário final.

Por último, a camada de *Business Layer* gerencia todas as atividades e serviços do sistema IoT, dentre ela a construção de modelos, gráficos, fluxos de dados, banco de dados, entre outros. Essa camada facilita a construção de sistemas de tomada de decisão baseados na análise e processamento do *Big Data*, uma das principais fronteiras de pesquisa de IoT.

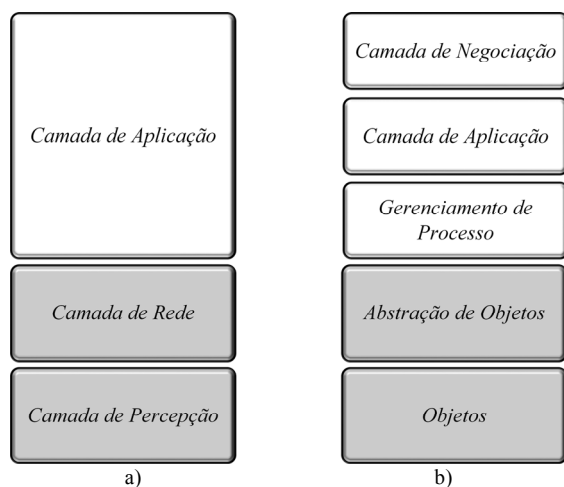


Fig. 1. Comparação entre arquiteturas IoT: a) de 3 e b) 5 camadas.

Existem diferentes protocolos que são utilizados na camada *Application Layer* da arquitetura de IoT. Dentre os principais protocolos estão: *Data Distribution Service* (DDS), CoAP, *Advanced Message Queing Protocol* (AMQP), MQTT, MQTT-SN, *Extensible Messaging and Presence Protocol* (XMPP) e *Hypertext Transfer Protocol* (HTTP). Como dito anteriormente, os protocolos CoAP e MQTT-SN têm se destacado por suas características, descritas a seguir:

B. Constrained Application Protocol (CoAP)

O CoAP define um protocolo de transmissão *web* baseado no conceito de *REpresentational State Transfer* (REST), que facilita a transferência de dados entre clientes e servidores através de HTTP. Esse protocolo foi desenvolvido pelo *Internet Engineering Task Force* (IETF) [12], [13] para aplicações M2M, permitindo a transferência de dados para uso em nós sensores e redes de baixo recurso.

O modelo de protocolo e troca de mensagens foi idealizado para aperfeiçoar o uso de HTTP em redes de sensores sem fio. A organização é feita em duas camadas: camada de transição (*messaging sub-layer*), responsável por gerenciar a troca de mensagens entre os dispositivos e camada de requisição/resposta (*request/response sub-layer*), responsável pelo envio, recebimento e gerenciamento de recursos. As duas camadas permitem que o CoAP forneça mecanismos de confiabilidade e controle de fluxo de dados de forma assíncrona [20]-[24]. O protocolo também possui quatro tipos distintos de mensagens: confirmáveis (*confirmable*), não confirmáveis (*non-confirmable*), reinicializar (*reset*), e reconhecimento (*acknowledgement*).

A Fig. 2 apresenta a arquitetura de uma rede com protocolo CoAP, onde os nós sensores e dispositivos clientes concentram os dados no servidor CoAP e, posteriormente, enviam seus dados para o CoAP *proxy*, mediante requisições.

C. Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN)

O MQTT-SN estende o uso do MQTT para rede de sensores sem fio usando o *User Datagram Protocol* (UDP) ao invés do *Transmission Control Protocol* (TCP), tendo em vista que nessas redes é comum a falha do enlace devido à mobilidade dos nós sensores ou dos dispositivos clientes. Esta abordagem se torna mais eficaz no transporte de dados entre sensores e concentradores de dados.

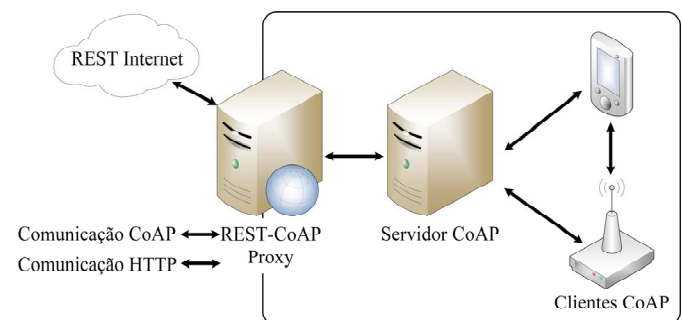


Fig. 2. Ambiente do protocolo CoAP.

O principal conceito de funcionamento do MQTT e do MQTT-SN é a comunicação centralizada de dados onde os recursos são transferidos via mensagens publicação/assinatura (*publish/subscribe*), possibilitando maior simplicidade, flexibilidade e menor banda de transmissão de dados.

A Fig. 3 apresenta a arquitetura do MQTT-SN onde os nós com baixo recurso rodam instâncias clientes e seus dados são publicados no MQTT *broker*.

A comunicação pode ser feita de forma transparente através do uso adicional de um nó MQTT-SN *gateway*. Nessa condição, apenas algumas alterações de sintaxe são feitas e a conexão entre o MQTT-SN *gateway* e o *broker* é realizada através do protocolo TCP. Outra forma de realizar a comunicação é a forma agregada, quando apenas uma conexão é feita com o MQTT *broker* [11].

As mensagens que são transportadas pelo protocolo MQTT-SN possuem um nível de qualidade de serviço associada (*QoS*) e uma identificação (*topic id*). O protocolo define três níveis de *QoS*: o nível 0, em que o remetente envia a mensagem apenas uma vez e nenhuma nova tentativa é executada, e os níveis 1 e 2, nos quais o protocolo garante que nenhuma mensagem de publicação será perdida, envolvendo, assim, um processo de reconhecimento de um ou dois passos, respectivamente.

Algumas diferenças estão presentes entre a versão do MQTT em comparação com o MQTT-SN, dedicada à rede de sensores e dispositivos com baixo consumo de energia e pequena banda de transmissão. A principal diferença é o uso de UDP, o que retira qualquer necessidade de recebimento de mensagens de confirmação, típicas no TCP. Além disso, podem ser destacadas: a quebra da mensagem CONNECT em mais mensagens, o envio do *topic id* ao invés de *topic name* nas mensagens PUBLISH, o procedimento de DISCOVERY para os nós que não possuem configuração prévia de *gateway* e, por último, um procedimento para manter os nós em estado de dormência (*sleep mode*).

Destaca-se que o MQTT-SN suporta múltiplos *gateways*, função comumente usada em redes de sensores sem fio devido a possíveis falhas de *link*. Diferentemente do CoAP, no qual a comunicação é feita de forma ponto a ponto (“um para um”), a comunicação no MQTT-SN é feita em forma *multicast*. A Tabela 1 apresenta outras comparações entre esses protocolos enfatizando os parâmetros: padrão de envio de mensagens, protocolo de transporte, confiabilidade, tamanho do cabeçalho das mensagens e vantagens.

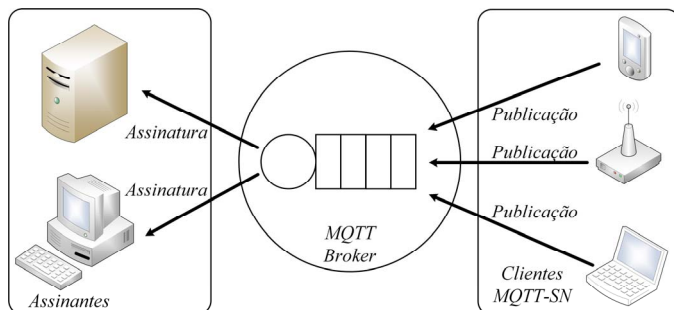


Fig. 3. Funcionamento do protocolo MQTT-SN.

TABELA I
ANÁLISE COMPARATIVA DOS PROTOCOLOS
COAP E MQTT-SN [21], [22], [26]

| PARÂMETROS DO PROTOCOLO | CoAP | MQTT-SN |
|------------------------------|--|--|
| Padrão de envio de mensagens | Similar ao HTTP utilizando método <i>Request/response</i> (<i>client-server</i>) | <i>Publish/subscribe</i> utilizando <i>message broker</i> |
| Protocolo de Transporte | UDP | UDP |
| Confiabilidade | Possui recurso para verificar o transporte de mensagens utilizando mensagens do tipo CON. | Possui recurso para verificar o transporte de mensagens utilizando o recurso <i>QoS</i> 1 e 2. |
| Cabeçalho das mensagens | 4 bytes, sendo os dois primeiros com indicação da versão do protocolo e os últimos ligados ao tipo de mensagem CON, NON, ACK ou RST. | 2 bytes, apresentando as informações do tamanho e o tipo da mensagem. |
| Vantagens | Protocolo rápido e eficiente devido ao uso do UDP. Interoperável com o protocolo HTTP. | A arquitetura do tipo <i>broker</i> facilita o gerenciamento da rede. |

III. MATERIAIS E MÉTODOS

A. Arquitetura da Rede Proposta

A rede de sensores proposta usa a camada PHY e MAC do IEEE802.15.4, o sistema operacional adotado foi o CONTIKI-OS [24] e a ferramenta de simulação foi o *software* COOJA [25] que acelera o desenvolvimento e testes de aplicativos, permitindo a simulação de um único nó sensor sem fio ou uma rede completa desses nós em um computador comum sem o *hardware* do nó do sensor. A execução da aplicação, comunicação de rede e periféricos podem ser simulados em tempo real para uma grande quantidade de nós. Entre os tipos de simulação estão presentes quatro tipos de modelos de propagação de rádio: *Unit Disk Graph Medium* (UDGM) *Distance Loss* onde a interferência é considerada porém os pacotes são transmitidos e recebidos com probabilidade de sucesso, *UDGM Constant Loss*, onde o alcance de transmissão é determinado por uma circunferência ideal e o sinal alcança todo o perímetro, *Directed Graph Radio Medium* (DGRM) onde a razão de sucesso é determinada de forma assimétrica em cada link por meio de parametrização dos delays de propagação e *Multi-path Ray-tracer Medium* (MRM), no qual a potência de recepção é determinada usando a fórmula de Friis e os obstáculos impõem atenuações nos sinais dos sensores [26].

O usuário pode interagir com os nós usando uma interface gráfica de usuário (GUI), que permite: a inserção de nós sensores, modificar entradas de sensores, inserir perturbações e mobilidade na rede. A simulação do conjunto de instruções executa os códigos de *firmware* usando ferramentas de compilação cruzada e um simulador de microcontrolador externo. O método de simulação usado foi *UDGM Distance Loss*, a camada de roteamento usada foi a *Routing Protocol for Low power and Lossy Networks* (RPL) [12], que faz o gerenciamento de rotas da rede 6LoWPAN [13]. Para evitar qualquer interferência no envio dos dados, não foi configurada a camada *Radio Duty Cycle* (RDC) no CONTIKI.

O *software* desenvolvido foi dividido em três partes: MQTT-SN *Client*, MQTT-SN *Gateway* e MQTT *Broker*. A *camada de aplicação Client* e *Gateway* rodam no sistema operacional CONTIKI-OS e a *camada de aplicação Broker* roda nativamente no sistema operacional Linux, onde a plataforma de testes foi instalada. Como ponte entre o ambiente de simulação COOJA e o sistema operacional Linux, foi necessário usar os utilitários TUNSLIP, que faz um tunelamento IP via serial e o TAYGA, que faz a tradução do IPv6 para IPv4, uma vez que o *serviço de gerenciamento* MQTT instanciado no servidor utiliza apenas a pilha de protocolos IPv4. Para os testes do CoAP, todos os utilitários estavam presentes dentro do ambiente COOJA.

B. Descrição das Simulações Realizadas

Os testes foram realizados no simulador COOJA usando dois tipos de nós, criados com a base de *hardware* WISMOTE. O primeiro tipo de nó simula os sensores acústicos na rede e o segundo simula o *gateway*, que fará a conexão com outra rede para envio dos dados. Para os testes do protocolo MQTT-SN, foram usados os parâmetros $QoS = 1$ (comunicação consiste em duas mensagens PUBLISH e PUBACK), porta 1884 UDP, envio de mensagens de 10 em 10 segundos com até 4 tentativas de registro de tópico entre os nós sensores e o servidor MQTT. Para os testes do protocolo CoAP foram usados os parâmetros de porta 61616 UDP e requisições de 10 em 10 segundos, considerando mensagens que garantam confirmação (CoAP CON e ACK).

Foram testados 1, 5, 10, 15, 20, 30 e 40 nós com alcance direto ao *gateway* e distribuídos ao seu redor em forma de elipse a fim de verificar a entrega dos dados para os servidores MQTT e CoAP bem como outras características de transmissão/recepção.

A topologia estrela foi escolhida, pois é tipicamente encontrada em rede de sensores para monitoramento de parâmetros marinhos [26], [27]. Como exemplo de aplicação, pode-se implementar o *gateway* em uma plataforma de petróleo, cuja a fonte de energia não se esgota, e os nós sensores possuem alcance direto ao sistema de monitoramento. Foi utilizado o comando CoAP POST e MQTT-SN PUBACK para realizar o envio das mensagens de texto.

A Fig. 4 mostra a interface gráfica do simulador COOJA. Os resultados foram coletados através da própria interface gráfica que disponibiliza as informações de cada nó instanciado na rede com precisão de microssegundos.

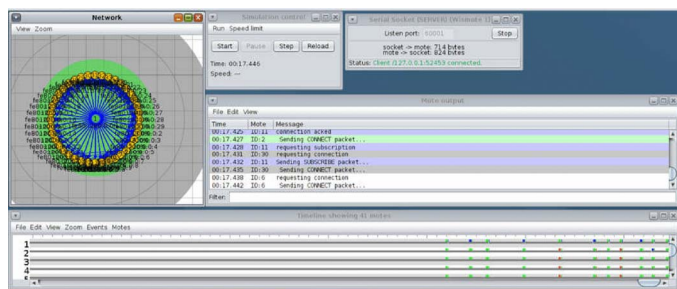


Fig. 4. Interface gráfica da ferramenta de simulação COOJA com a arquitetura de rede de sensores proposta para aplicações marinhas.

IV. RESULTADOS E ANÁLISE

A. Análise de Desempenho dos Protocolos

Os dois protocolos foram avaliados comparando-se os seguintes indicadores de desempenho em função da quantidade de nós sensores: latência, tempo de tráfego na rede, quantidade de pacotes trafegados, quantidade de *bytes* trafegados e consumo de energia.

A Fig. 5 apresenta as curvas de latência para cada protocolo. Compreende-se a latência [15], *Round Trip Time* (RTT), como o tempo entre o início da comunicação até o momento de recepção da mensagem pelo servidor de aplicação, considerando o processamento do protocolo.

O protocolo MQTT-SN apresentou latência de 36 ms na entrega da mensagem para um nó. Levou-se em consideração a latência entre as mensagens PUBLISH e PUBACK. Pode-se verificar um acréscimo de, aproximadamente, 2 ms a cada nó adicionado, levando em consideração o processamento do servidor MQTT.

Para os testes do protocolo CoAP, a latência foi de, aproximadamente, 231 ms para a resposta do dispositivo considerando novamente um único nó após a requisição do servidor. Dessa maneira, o protocolo CoAP é 195 ms mais lento que o MQTT-SN. Essa diferença de valores se mantém aproximadamente a mesma para qualquer quantidade de nós na rede. Sendo assim, foi possível verificar que o MQTT-SN é viável para a aplicação proposta e apresenta menor latência na entrega dos dados. Pode-se sugerir que o menor tempo de latência do protocolo MQTT-SN ocorre devido ao menor tamanho do pacote necessário para enviar mensagens entre os nós e o *gateway*.

A Fig. 6 mostra o tempo de tráfego na rede para os dois protocolos. Pode-se notar que ambos os protocolos apresentam praticamente o mesmo desempenho com relação ao tempo de tráfego na rede. Isso ocorre, pois ambas as pilhas de protocolos utilizam a mesma camada de roteamento (RPL), demonstrando que o tempo de atraso se deve exclusivamente ao processamento dos protocolos tanto nos nós transmissores como no *gateway*.

Durante os testes uma mesma mensagem foi enviada dos nós sensores para o *gateway*. A mesma mensagem consiste numa frase concatenada com os últimos dois *bytes* do endereço IPv6 do nó sensor. Quando se compara a quantidade de pacotes trafegados (Fig. 7) o MQTT-SN apresenta um tráfego 2,15 vezes maior que o CoAP, para qualquer quantidade de nós. Isso se deve à fragmentação de mensagem realizada pelo MQTT-SN. O protocolo CoAP, por ser baseado no protocolo HTTP, encaminha mensagens mais longas, pois possui métodos que garantem a entrega. O MQTT-SN encaminha mensagens mais curtas, utilizando menor banda.

A Fig. 8 apresenta a quantidade de *bytes* trafegados. O protocolo MQTT-SN transmite a mesma mensagem utilizando somente 70% da quantidade de *bytes* do protocolo CoAP. Isso é esperado, pois o CoAP possui maior número de *bytes* para encaminhar os cabeçalhos e as mensagens. Dessa forma, o MQTT-SN realiza o envio da mesma informação com menor quantidade de dados.

Com relação ao consumo de energia, a Fig. 9 mostra que o consumo do MQTT-SN é levemente melhor, sendo aproximadamente 10% menor que o consumo do CoAP. Como mencionado anteriormente, as características do MQTT-SN, entre elas menor tráfego de *bytes* e menor latência, resultam no menor consumo de energia dos nós.

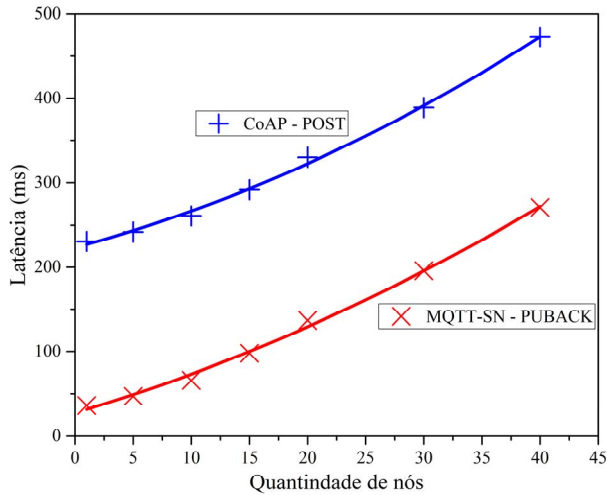


Fig. 5. Resultados de latência utilizando os protocolos MQTT-SN e CoAP.

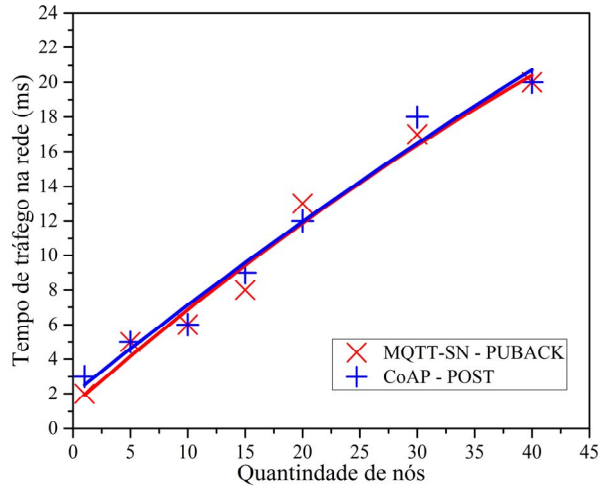


Fig. 6. Comparação entre o tempo de tráfego na rede (ms) da arquitetura MQTT-SN e CoAP.

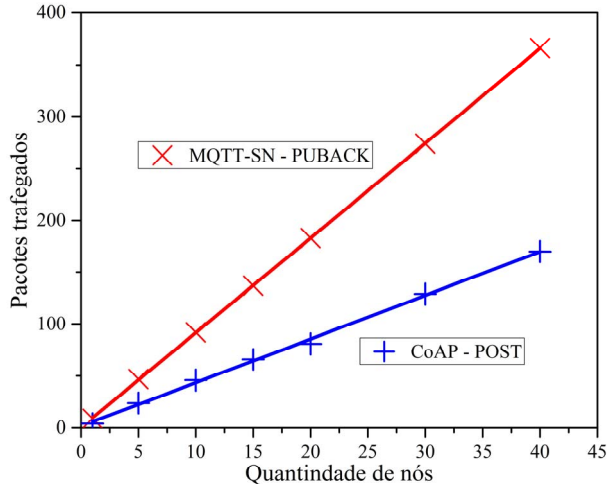


Fig. 7. Comparação entre a quantidade de pacotes trafegados utilizando os protocolos MQTT-SN e CoAP.

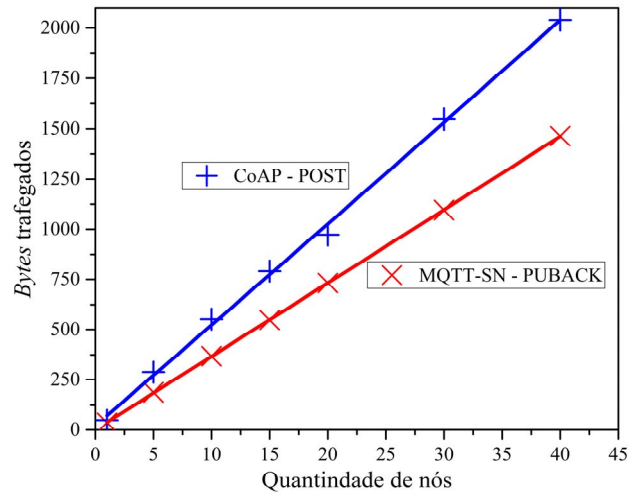


Fig. 8. Quantidade de *bytes* trafegados utilizando os protocolos MQTT-SN e CoAP.

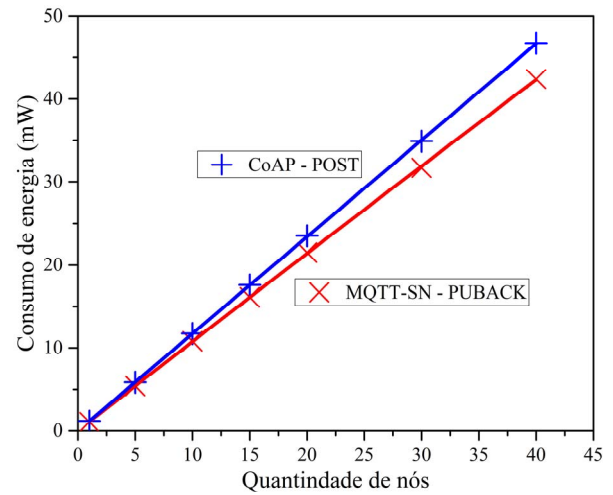


Fig. 9. Comparação entre o consumo de energia dos nós sensores utilizando os protocolos MQTT-SN e CoAP.

A comparação entre os protocolos pode ser sintetizada na Tabela II. Nessa tabela são detalhados os resultados obtidos para 40 nós.

| PARÂMETROS DO PROTOCOLO | MQTT-SN | CoAP | COMPARATIVO |
|---------------------------------|---------|------|--|
| Latência da rede (ms) | 271 | 473 | MQTT-SN é 200 ms mais rápido comparado ao CoAP. |
| Tempo de tráfego na rede (ms) | 20 | 20 | Ambos os protocolos possuem aproximadamente o mesmo tempo de tráfego na rede. |
| Pacotes trafegados | 366 | 170 | O MQTT-SN fragmenta a informação em maior número de pacotes, cerca de 2,15 vezes. |
| Bytes trafegados | 1464 | 2040 | O MQTT-SN transmite a mesma mensagem com apenas 70% dos <i>bytes</i> utilizados no CoAP. |
| Consumo de energia dos nós (mW) | 42,4 | 46,7 | O MQTT-SN apresenta consumo de energia 10% menor do que o CoAP. |

V. CONCLUSÕES

Por meio de testes de simulação foi possível comparar os protocolos MQTT-SN e CoAP com relação a: latência para a entrega dos dados, tempo de tráfego na rede, tráfego de pacotes e de bytes e consumo energético. Para realizar essa simulação, foi implementado o protocolo MQTT-SN dentro do sistema operacional Contiki e do simulador COOJA. Conforme os resultados obtidos, o protocolo MQTT-SN apresentou, em geral, desempenho melhor que o CoAP. O MQTT-SN mostrou latência da ordem de 200 ms menor que o CoAP, mesmo trafegando uma quantidade maior de pacotes. Também apresentou consumo energético 10% menor que o CoAP. Dessa forma, o protocolo MQTT-SN se mostra como uma alternativa mais adequada para implementação em aplicações IoT para redes de sensores sem fio em ambientes hostis, em especial, para monitoramento acústico marinho.

REFERÊNCIAS

- [1] E. R. Vivas, B. L. Lopez, "Construction, calibration, and field test of a home-made, low-cost hydrophone system for cetacean acoustic research," *Proceedings of Meetings on Acoustics*, vol. 11, 2011.
- [2] M. O. Lammers, R. E. Brainard, W. W. L. Au, T. A. Mooney, K. B. Wong, "An ecological acoustic recorder (EAR) for long-term monitoring of biological and anthropogenic sounds on coral reefs and other marine habitats," *The Journal of the Acoustical Society of America*, vol. 123, 2008.
- [3] R. S. Sousa-Lima, T. F. Norris, J. N. Oswald, D. P. Fernandes, "A review and inventory of fixed autonomous recorders for passive acoustic monitoring of marine mammals," *Acoustics in Underwater Geosciences Symposium (RIO Acoustics)*, 2013.
- [4] S. M. Wiggins, J. A. Hildebrand, "High-frequency Acoustic Recording Package (HARP) for broadband, long-term marine mammal monitoring," *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies*, 2007.
- [5] J. J. Martinez, J. R. Myers, T. J. Carlson, Z. D. Deng, J. S. Rohrer, K. A. Caviggia, "Design and Implementation of an Underwater Sound Recording Device," *Sensors (Basel)*, vol. 11, no. 9, pp. 8519-35, 2011.
- [6] C. Alexandre Gouvea da Silva, E. Leonardo dos Santos, A. Christian Krainski Ferrari and H. Tertuliano dos Santos Filho, "A Study of the Mesh Topology in a ZigBee Network for Home Automation Applications," *IEEE Latin America Transactions*, vol. 15, no. 5, pp. 935-942, May 2017.
- [7] D. C. Yacchirema Vargas and C. E. Palau Salvador, "Smart IoT Gateway for Heterogeneous Devices Interoperability," *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3900-3906, Aug. 2016.
- [8] P. D. Godoy, R. Cayssials and C. Garcia Garino, "A WSN Testbed for Teaching Purposes," *IEEE Latin America Transactions*, vol. 14, no. 7, pp. 3351-3357, July 2016.
- [9] D. Adorno Gomes and D. Bianchini, "Interconnecting Wireless Sensor Networks with the Internet Using Web Services," *IEEE Latin America Transactions*, vol. 14, no. 4, pp. 1937-1942, April 2016.
- [10] Z. Shelby, K. Hartke, C. Bormann, "Constrained Application Protocol (CoAP)," Internet Engineering Task Force (IETF), "Internet-Draft, draft-ietf-core-coap-18, 2013.
- [11] OASIS Standard, "MQTT v3.1.1 Protocol Specification," 2014.
- [12] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," *Internet Engineering Task Force (IETF)*, "Internet-Draft, draft-ietf-roll-rpl, 2012.
- [13] N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," *Internet Engineering Task Force (IETF)*, "Internet-Draft, draft-ietf-6lowpan-format, 2007.
- [14] D. Thangavel, X. Ma, A. Valera, H. X. Tan and C. K. Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, 2014, pp. 1-6.
- [15] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino and G. Reali, "Comparison of two lightweight protocols for smartphone-based sensing," *2013 IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Namur, 2013, pp. 1-6.
- [16] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, H. Hashim, "A Comparison of Lightweight Communication Protocols in Robotic Applications," *Procedia Computer Science*, vol. 76, pp. 400-405, 2015.
- [17] L. Atzori, A. Iera and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, pp. 2787-2805, 2010.
- [18] R. Khan, S. U. Khan, R. Zaheer and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," *Frontiers of Information Technology (FIT)*, 2012 10th International Conference On, 2012, pp. 257-260.
- [19] P. Lopez, D. Fernandez, A. J. Jara and A. F. Skarmeta, "Survey of internet of things technologies for clinical environments," *Advanced Information Networking and Applications Workshops (WAINA)*, pp. 1349-1354, 2013.
- [20] D. Yang, F. Liu and Y. Liang, "A survey of the internet of things," *Proceedings of the 1st International Conference on EBusiness Intelligence (ICEBI2010)*, 2010, pp. 358-366.
- [21] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann and K. K. Leung, "A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities," *Wireless Communications, IEEE*, vol. 20, pp. 91-98, 2013.
- [22] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang and W. Liu, "Study and application on the architecture and key technologies for IOT," *2011 International Conference on Multimedia Technology (ICMT2011)*, Hangzhou, 2011, pp. 747-751.
- [23] M. Wu, T. J. Lu, F. Y. Ling, J. Sun and H. Y. Du, "Research on the architecture of internet of things," *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Chengdu, 2010, pp. V5-484-V5-487.
- [24] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," *29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 455-462.
- [25] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "CrossLevel Sensor Network Simulation with COOJA," *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, Tampa, FL, 2006, pp. 641-648.
- [26] D. Fanton, "Connecting the IoT MQTT and CoAP protocols," *IO Hub*, [Online]. Available: <http://www.logicsupply.com>. [Accessed: 17-Oct-2017]
- [27] C. Albaladejo, P. Sánchez, A. Iborra, F. Soto, J.A. López, R. Torres. "Wireless Sensor Networks for Oceanographic Monitoring: A Systematic Review," *Sensors (Basel)*, vol. 10, no. 7, pp. 6948-6968, 2010.
- [28] G. Xu, W. Shen, X. Wang, "Applications of Wireless Sensor Networks in Marine Environment Monitoring: A Survey," *Sensors (Basel)*, vol. 14, no. 9, pp. 16932-16954, 2014.



Gabriel Durante received his B.Sc. degree in Electrical Engineering from Fundação Instituto Tecnológico de Osasco (FAC-FITO) His main research interests are related to IoT, sensors networks, digital signal processing, embedded systems, and machine learning.



Wesley Beccaro received his B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from the Escola Politécnica da Universidade de São Paulo (USP). He is currently a researcher at the Department of Electronic

Systems Engineering in the same institution and his research interests are related to control systems, embedded systems and digital signal processing.



Henrique E. M. Peres received his M.Sc. and Ph.D. degrees both in Electrical Engineering from the Escola Politécnica - Universidade de São Paulo, Brazil. He is currently a researcher at the Department of Electronic Systems Engineering in the same institution, and his research interest is related to MEMS, integrable sensors and nanomaterials.