

Uma Avaliação de Desempenho de Protocolos de Camada de Aplicação para Internet das Coisas

Vinícius de Figueiredo Marques¹, Janine Kniess¹

¹Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina (UDESC)

vinicius.marques@edu.udesc.br, janine.kniess@udesc.br

Abstract. *Internet of Things (IoT) aims to bring connectivity for a vast amount of devices and there's a huge field of application. Communication is vital for that goal and efficient network protocols for low-powered devices is a critical issue. This paper presents a behavior analysis of two IoT targeted application protocols and one consolidated web server protocol. Constrained Application Protocol (CoAP), MQ Telemetry Protocol (MQTT) and Hypertext Transfer Protocol (HTTP) respectively. Tests were made to assess the needed response time and the total overhead each protocol causes in a network. These tests were conducted with real hardware sending messages to local and remote servers. The results indicated that IoT protocols usage leads to less network overhead and response time.*

Resumo. *A Internet das Coisas demanda o uso de tecnologias eficientes de comunicação como um elemento chave. Tais tecnologias dependem de protocolos que permitam a comunicação entre dispositivos de sensoriamento e os servidores onde serão armazenados os dados coletados. Este artigo apresenta uma análise de desempenho dos protocolos Constrained Application Protocol (CoAP), MQ Telemetry Transport (MQTT) e Hypertext Transfer Protocol (HTTP). A análise foi realizada em um ambiente real. Os resultados indicam que o uso de protocolos adequados em IoT reduzem a sobrecarga e o tempo de resposta nas aplicações.*

1. Introdução

Internet das Coisas (IoT do inglês *Internet of Things*) é uma área tecnológica emergente que visa permitir que diversos tipos de dispositivos estejam conectados à internet. Tais dispositivos possuem capacidade de sensoriamento e/ou atuação, e devem preferencialmente serem capazes de se comunicarem através de tecnologias sem-fio e com protocolos de rede apropriados. Há diversas aplicações possíveis no contexto de Internet das Coisas, por exemplo, aplicações na saúde, monitoramento ambiental, segurança e transporte [Gubbi et al. 2013].

Deste modo, os protocolos desenvolvidos para redes formadas por dispositivos de baixa capacidade, na qual residem a maioria das aplicações de IoT devem ser eficientes quanto ao consumo de energia, ao tamanho dos pacotes e à sobrecarga de mensagens geradas pelo protocolo na rede.

A Internet utiliza o *Hyper Text Transfer Protocol* (HTTP) como protocolo de comunicação na camada de aplicação do modelo OSI (acrônimo do inglês *Open*

System Interconnection) para transferência de dados no paradigma cliente/servidor na rede mundial de computadores, a World Wide Web. No entanto, segundo [Yokotani and Sasaki 2016] e [Mijovic et al. 2016] quando uma grande quantidade de pacotes pequenos devem ser transmitidos via HTTP, seu uso pode degradar significativamente o desempenho da rede, devido a sobrecarga gerada pelo protocolo HTTP, principalmente, por operar sobre o protocolo TCP na camada de Transporte.

Os protocolos *Constrained Application Protocol* (CoAP) e *MQ Telemetry Transport* (MQTT), ambos para Camada de Aplicação são indicados na literatura [Al-Fuqaha et al. 2015] como os protocolos mais promissores para dispositivos menores, situados em ambientes de IoT.

Neste trabalho, apresenta-se uma análise de desempenho dos protocolos HTTP, COAP e MQTT, com o objetivo de prover direções quanto ao desempenho destes protocolos no âmbito de aplicações de IoT. Os protocolos foram analisados em relação as métricas, sobrecarga, atraso-fim-a-fim e tamanho do pacote. A análise de desempenho foi realizada em um ambiente real, sendo que, os protocolos de camada de aplicação foram implementados em uma plataforma de hardware de IoT.

Este artigo está organizado como segue: Uma fundamentação teórica sobre protocolos na camada de aplicação para IoT é apresentada na Seção 2. Nesta seção, também são apresentados os trabalhos relacionados. A análise de desempenho e os resultados obtidos são descritos na Seção 3. A Seção 4 apresenta as conclusões e direções futuras.

2. Fundamentação Teórica

Diferentes grupos trabalham de maneira a desenvolverem protocolos para darem suporte ao desenvolvimento de IoT; grupos como *World Wide Web Consortium* (W3C), *Internet Engineering Task Force* (IETF) e *Institute of Electrical and Electronics Engineers* (IEEE) [Al-Fuqaha et al. 2015]. Para a camada de rede, a IETF já desenvolveu um padrão amplamente discutido na literatura: o 6LoWPAN [Hui and Thubert 2011], um protocolo desenhado para funcionar sobre redes de baixa largura de banda e que tenham suporte ao IPv6. Para a camada de transporte, o protocolo *User Datagram Protocol* (UDP) se mostra mais interessante pela baixa sobrecarga que o protocolo gera na rede.

O protocolo CoAP, desenvolvido pelo *IETF Constrained RESTful Environments* (CoRE) [Bormann et al. 2012] utiliza-se do mesmo estilo organizacional do HTTP: ambos seguem um modelo cliente-servidor, disponibilizam as informações, ou serviços, através de *Uniform Resource Identifier* (URI) e suportam métodos como GET, PUT, POST e DELETE para descrever o tipo de requisição que o cliente envia ao servidor. O CoAP é interoperável com o HTTP; facilitando assim, a comunicação de um sistema web com um sistema de IoT que se utilize do CoAP como protocolo de aplicação [Bormann et al. 2012]. Tendo em vista que o CoAP funciona sobre o protocolo UDP, protocolo este que não garante sucesso na transmissão, o protocolo CoAP disponibiliza um mecanismo para confirmação das mensagens recebidas a nível da camada de aplicação. As mensagens do tipo CON, (*Confirmable*) são reservadas para esse fim. Quando um nó recebe uma mensagem do tipo CON, deve enviar uma mensagem do tipo ACK (*Acknowledgement*) ao emissor. [Shelby et al. 2014].

O MQTT é um protocolo para conectividade máquina-a-máquina (M2M) e possui três modos de transmissão baseados no nível de qualidade de serviço desejado. O QoS de

nível 0 especifica que o destinatário não se compromete em confirmar o recebimento da mensagem ao remetente e nem o remetente se compromete a reenviar a mensagem. QoS de nível 1 garante que a mensagem chegará ao menos uma vez ao destinatário. QoS de nível 2 é o nível mais alto de qualidade de serviço e deve ser utilizado quando a aplicação não pode suportar perda e nem duplicação de mensagens [OASIS 2014].

As mensagens no MQTT são classificadas como tópicos de dois tipos: *Publish* e *Subscribe*. O primeiro tipo define as mensagens que são enviadas por exemplo, por um sensor, enquanto que, a segunda é uma mensagem enviada pelo interessado nos dados coletados pelo sensor. O modelo de funcionamento do MQTT requer a presença de um nó intermediário entre o *publisher* e o *subscriber*, tal mediador é chamado de *broker*. O uso do protocolo TCP na camada de transporte pode causar aumento de *overhead* em aplicações de IoT [Yokotani and Sasaki 2016]. Devido ao fato de que o MQTT utiliza o TCP, seu *overhead* pode ser maior quando comparado ao CoAP.

2.1. Trabalhos Relacionados

Outros estudos já foram realizados com a finalidade de comparar o HTTP com protocolos voltados para IoT. Yokotani e Sasaki (2016) apresentaram um estudo comparativo entre o HTTP e o MQTT. Os autores apontam que a sobrecarga é um requisito essencial para aplicações de IoT. Este estudo conclui que o MQTT tem menor sobrecarga do que o HTTP e que essa sobrecarga é proporcional ao tamanho da mensagem a ser transmitida [Yokotani and Sasaki 2016].

Thangavel et al. (2014) publicaram um estudo comparativo entre o MQTT e o CoAP através de simulações. Os autores mediram a taxa de perda de pacote e o tamanho dos pacotes. A conclusão do trabalho foi de que o MQTT tem uma taxa menor de atraso quando há pouca perda de pacotes, e alto atraso caso contrário [Thangavel et al. 2014].

A necessidade de protocolos mais leves em aplicações para smartphones foi uma motivação para o trabalho proposto por De Caro et al. (2013). Os autores apresentaram comparações tanto qualitativa (através de descrições textuais das características e funcionalidades oferecidas por cada protocolo avaliado), quanto quantitativa dos protocolos MQTT e CoAP em dispositivos móveis. As conclusões dos experimentos foram de que o CoAP apresentou menor RTT. Os autores também destacaram que o CoAP é o protocolo mais apropriado para o desenvolvimento de aplicações que necessitem de pouca largura de banda e recursos dos dispositivos [De Caro et al. 2013].

O trabalho de Chen e Kunz (2016) apresenta uma avaliação quantitativa de quatro protocolos destinados à aplicações de IoT: MQTT, CoAP, *Data Distribution Service* (DDS) e um outro protocolo desenvolvido pelos autores. Com base na análise de desempenho, concluí-se neste trabalho que o DDS é indicado para as aplicações que requeram a alta confiabilidade. Por outro lado, o CoAP mostrou-se adequado para aplicações que demandam pouca largura de banda [Chen and Kunz 2016]. Os autores ainda apontaram que as implementações do MQTT e DDS são mais robustas e apresentam melhor documentação do que as implementações CoAP.

A discussão apresentada sobre a avaliação de protocolos de camada de aplicação para IoT, reforça a necessidade de novas investigações que contemplem novos protocolos e métricas. Em face dessa conjuntura, neste trabalho é realizada uma avaliação de desempenho de protocolos de camada de aplicação para IoT em um ambiente real. Além

da análise em uma rede local, os protocolos também são avaliados no contexto de redes remotas.

O principal objetivo da comparação apresentada neste trabalho é a experimentação de protocolos populares na literatura através de implementações para um dispositivos real de IoT, o utilizado nos experimentos descritos na Secção 3.1. Espera-se também comparar os resultados obtidos em dois cenários diferentes, um com servidores em rede local e outro com servidores na nuvem.

3. Resultados Computacionais

Nesta seção descreve-se a análise de desempenho realizada bem como, os resultados obtidos com os protocolos HTTP, MQTT e CoAP.

3.1. Ambiente e Métricas de Avaliação

Para a análise experimental dos protocolos foram configurados cenários reais com o microcontrolador NodeMcu rodando uma versão do *firmware* [Community Forum 2014]) que oferece nativamente a implementação do protocolo HTTP. O NodeMcu tem como base o módulo ESP8266 [Espressif 2017], que possui características interessantes para o escopo deste trabalho: baixo custo e alta popularidade na comunidade de desenvolvedores e módulo WiFi integrado compatível com padrão IEEE 802.11 b/g/n e antena embutida.

As implementações do MQTT [Ian 2009] e do CoAP [Nagesh and Patel 2017] utilizadas no microcontrolador não são desenvolvidos pelo mesmo grupo de trabalho do NodeMcu, porém, por outras comunidades *open-source* de maneira independente.

Para a realização dos experimentos e análise de desempenho, foram realizados testes com servidores dentro da rede local na qual os microcontroladores operavam, e com servidores localizados fora da rede local (remotos).

Todos os servidores localizados na rede local configurados para os experimentos rodam sobre a plataforma Java 8. O servidor HTTP utiliza o *framework* Spark [Wendel 2011], o servidor CoAP foi implementado com o *framework* Californium [Californium 2014] e, no servidor MQTT (*broker*) usou-se o Mosquitto [Mosquitto 2014]. Três servidores online gratuitos foram utilizados para experimentos com as transmissões remotas¹.



Figura 1. Topologia física da rede local.

A Figura 1 representa a estrutura física da rede local no qual os testes foram realizados. A topologia foi instalada em um ambiente fechado com cerca de 3 metros quadrados. O módulo ESP8266 foi conectado via Wifi com um roteador que atua como ponto

¹Servidor HTTP: <http://httpbin.org/>, Servidor CoAP: <http://coap.me/about> Servidor MQTT: <http://broker.mqtt-dashboard.com/index.html>

de acesso entre o computador e o microcontrolador. O computador e o microcontrolador foram posicionados em lados opostos do ambiente.

A Figura 2 ilustra a inter-relação dos componentes, e o fluxo de comunicação nos experimentos realizados com os servidores remotos. O microcontrolador transmite mensagens aos servidores através de uma rede sem fio conectada à internet. A escolha da rede sem fio como meio de acesso à internet foi com o objetivo de avaliar os protocolos em um ambiente similar às aplicação de IoT.

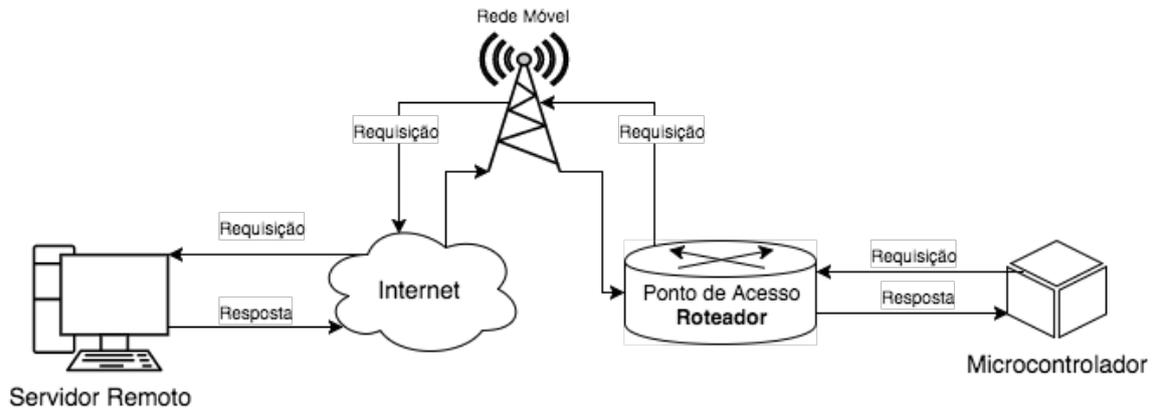


Figura 2. Topologia física com servidores remotos.

O desempenho dos protocolos CoAP, MQTT e HTTP foi avaliado em relação as métricas: *Round Trip Time* (RTT), sobrecarga e tamanho dos pacotes. O RTT foi calculado através da medição do tempo necessário para a mensagem de confirmação de recebimento de pacote ser recebido pelo microcontrolador. A sobrecarga foi calculada através da relação entre a quantidade de mensagens de controle sobre o total de mensagens transmitidas em cada experimento e por cada protocolo. O tamanho dos pacotes foi medido em *bytes*.

Em cada experimento, um número de 100 requisições foram enviadas do microcontrolador para o servidor. O servidor deve retornar uma resposta ao microcontrolador. O campo de carga útil dos pacotes de requisição é constituído de uma cadeia de caracteres que representa um valor de dados de *Global Positioning System* (GPS), comum em aplicações de IoT no campo de transporte [Fabiano Baldo 2017].

A ferramenta de análise de redes Wireshark² foi aplicada para permitir a visualização dos pacotes capturados e fornecer dados estatísticos acerca do tamanho e quantidade de pacotes.

Os protocolos avaliados possuem diferenças fundamentais e características próprias conforme descrito na Seção 2. Por exemplo, o protocolo HTTP envia ao remetente uma mensagem de confirmação via o protocolo de transporte TCP. A confirmação no protocolo CoAP ocorre com as mensagens do tipo ACK posto que, o protocolo de transporte empregado pelo COAP é o UDP. Para o MQTT definiu-se um QoS de nível 0, sendo a confirmação é responsabilidade do TCP.

Na Figura 3, apresenta-se uma análise do RTT(*Round Trip Time*) dos protocolos,

²<https://www.wireshark.org/>

HTTP, COAP e MQTT na rede local.

O RTT mínimo e médio foram avaliados neste experimento. Desse experimento chegou-se a algumas constatações: que o RTT médio do protocolo CoAP foi menor em comparação ao HTTP e MQTT. Este resultado justifica-se pelo fato de que o protocolo UDP é aplicado na camada de transporte, não sendo necessário o estabelecimento da conexão como no TCP. Além disso, o UDP é mais rápido por possuir um cabeçalho do segmento menor que o TCP. Outra justificativa para este resultado seria o fato de que o tamanho (em *bytes*) do cabeçalho do protocolo CoAP é menor do que o MQTT.

A Figura 4 apresenta os RTTs mínimos e médios (medidos em segundos) no qual as mensagens foram enviadas aos servidores remotos.

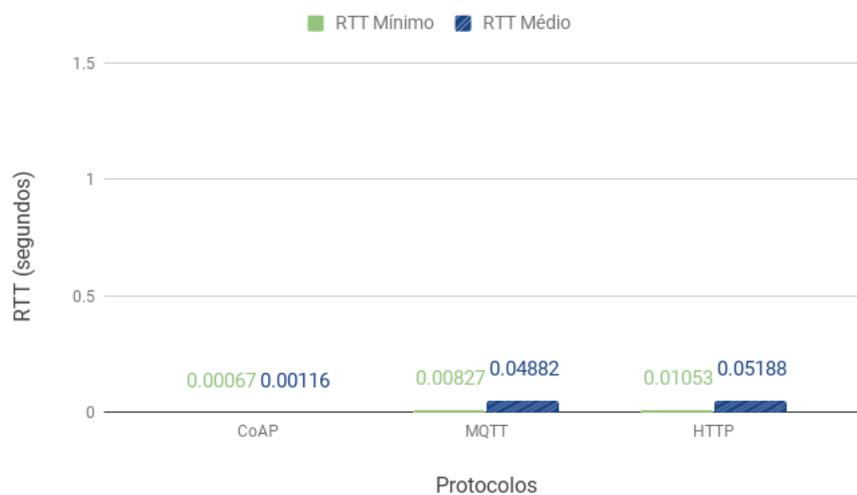


Figura 3. Análise dos protocolos em relação ao RTT na rede local.

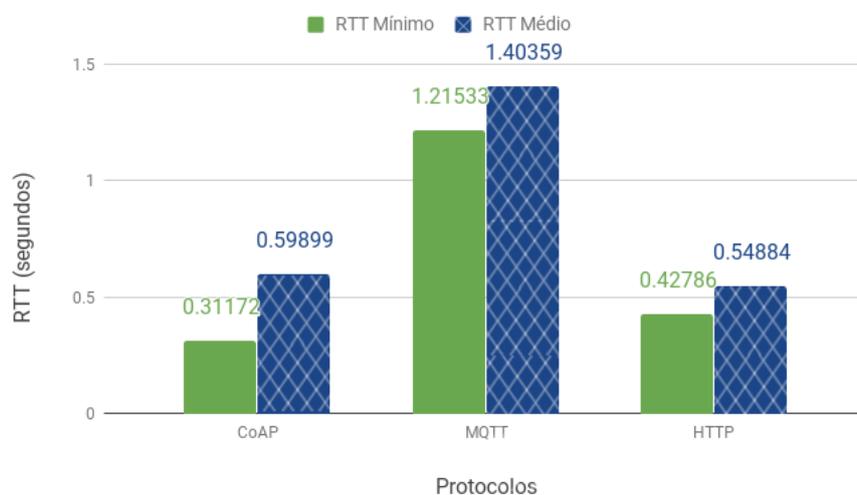


Figura 4. Análise dos protocolos em relação ao RTT na rede remota.

Os resultados obtidos sugerem que o protocolo CoAP apresenta um melhor desempenho em relação ao atraso fim-a-fim no contexto de IoT. O RTT médio para o pro-

tolocolo MQTT ilustrado na Figura 3 e na Figura 4 mostraram uma diferença significativa entre os testes local e remoto. O RTT médio no acesso aos servidores foi de 1,426s enquanto que no acesso ao servidor na rede local foi de 0,048s. Justifica-se esse comportamento a fatores como, disponibilidade do servidor remoto e grande quantidade de retransmissões devido à perda de pacotes.

Na Figura 5, apresenta-se o desvio padrão calculado sobre o RTT médio em ambos os cenários de configuração, local e remoto. Na rede local, observou-se que o CoAP apresentou menor desvio. Este resultado é devido ao fato de que na rede local não ocorreram picos de atrasos na transmissão, gerando um RTT médio linear. Neste experimento, identificou-se que o MQTT apresentou menor desvio padrão em relação ao HTTP. Porém nos testes com os servidores remotos, o HTTP teve o menor desvio padrão quando comparado ao MQTT e CoAP. Esse resultado sugere que há a necessidade de melhores servidores aptos para suportar esses novos protocolos destinados a aplicações de IoT.

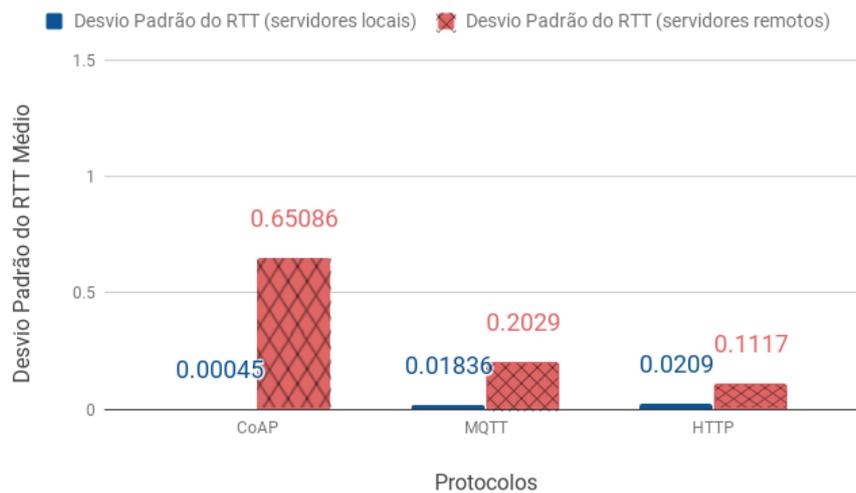


Figura 5. Desvio padrão calculado sobre o RTT médio.

No gráfico, ilustrado na Figura 6, mediu-se a sobrecarga, ou seja, o número de mensagens de requisição gerada por cada protocolo. A sobrecarga foi avaliada na rede local. A sobrecarga com o CoAP foi de 7,8%, enquanto que a do MQTT foi de 50,0% e do HTTP, 42,2%.

Com base em uma análise nos pacotes MQTT transmitidos, verificou-se que o MQTT transmitiu mais pacotes a nível de camada de aplicação, se comparado com o HTTP.

No gráfico da Figura 7 avaliou-se o tamanho médio dos pacotes transmitidos por cada protocolo. Verificou-se que o CoAP apresenta os pacotes mais leves com apenas 55 bytes. O MQTT gera pacotes menores quando comparado com o HTTP. Com base nesse resultado, conclui-se então que o CoAP e o MQTT são mais adequados para aplicações de ambientes restritos como no IoT.

Os experimentos computacionais realizados em um ambiente real demonstraram que o CoAP tem melhor desempenho em relação à sobrecarga e RTT médio na rede local

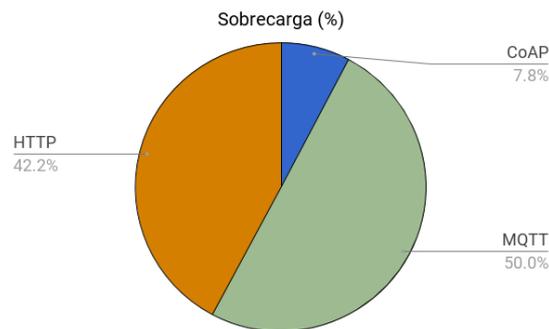


Figura 6. Análise da sobrecarga dos protocolos.

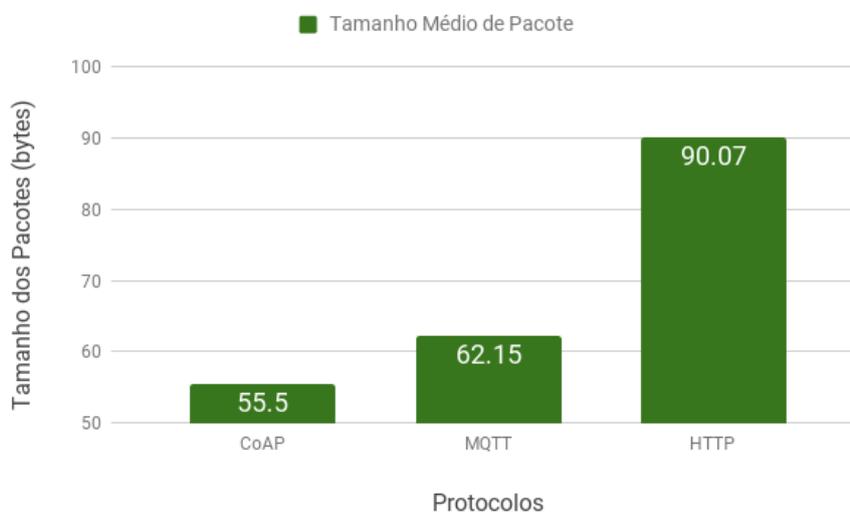


Figura 7. Análise do tamanho dos pacotes.

e remota. O HTTP tem o maior tamanho de pacote, porém, o HTTP mostrou-se mais estável em relação ao CoAP e MQTT no acesso aos servidores remotos.

4. Conclusão

Neste trabalho, realizou-se uma análise de desempenho dos protocolos de camada de aplicação comumente usados em IoT. O protocolo CoAP se mostra preferível para se economizar energia dos dispositivos e largura de banda, uma vez que gerou a menor sobrecarga e tamanho de pacotes. Algo que já era esperado dado ao fato de se utilizar o protocolo UDP na camada de transporte em conjunto com as análises encontradas nos trabalhos relacionados.

As implementações dos protocolos para servidores e clientes nos microcontroladores já se encontram estáveis para o NodeMcu. Contudo, o resultado de RTT nos testes com os servidores remotos indicou a necessidade de se fornecer servidores online que ofereçam suporte a estes novos protocolos para IoT.

Pode-se concluir então que as implementações dos protocolos próprios para IoT (MQTT e CoAP) encontradas na comunidade *open-source* apresentaram bons resulta-

dos nos experimentos. Porém, há a necessidade de melhores servidores online para os serviços que utilizem tais protocolos.

Referências

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376.
- Bormann, C., Castellani, A. P., and Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2):62–67.
- Californium, E. I. (2014). Californium. <https://www.eclipse.org/californium/>.
- Chen, Y. and Kunz, T. (2016). Performance evaluation of iot protocols under a constrained wireless access network. In *Selected Topics in Mobile & Wireless Networking (MoWNeT), 2016 International Conference on*, pages 1–7. IEEE.
- Community Forum, E. (2014). Esp8266 core for arduino. <https://github.com/esp8266/Arduino>.
- De Caro, N., Colitti, W., Steenhaut, K., Mangino, G., and Reali, G. (2013). Comparison of two lightweight protocols for smartphone-based sensing. In *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on*, pages 1–6. IEEE.
- Espressif (2017). Espressif esp8266. <http://espressif.com/en/products/hardware/esp8266ex/overview>.
- Fabiano Baldo, Rafael Stubs Parpinelli, F. B. V. B. R. (2017). Trajectorypatternmining - proposta de um arcabouço computacional para a descoberta de padrões em trajetórias de objetos móveis usando tecnologias nosql.
- Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.
- Hui, J. and Thubert, P. (2011). Rfc 6282 compression format for ipv6 datagrams over ieee 802.15. 4-based networks. sep-2011.
- Ian, T. (2009). ESPCoAP Arduino Library thingtronics. <https://github.com/Imroy/pubsubclient>.
- Mijovic, S., Shehu, E., and Buratti, C. (2016). Comparing application layer protocols for the internet of things via experimentation. In *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on*, pages 1–5. IEEE.
- Mosquitto, E. I. (2014). Mosquitto. <https://github.com/eclipse/mosquitto>.
- Nagesh, P. and Patel, L. (2017). ESPCoAP Arduino Library thingtronics. <https://github.com/automote/ESP-CoAP>.

- OASIS, S. (2014). Mqtt version 3.1.1. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.
- Shelby, Z., Hartke, K., and Bormann, C. (2014). Rfc 7252 the constrained application protocol (coap). *Internet Engineering Task Force (IETF)*.
- Thangavel, D., Ma, X., Valera, A., Tan, H.-X., and Tan, C. K.-Y. (2014). Performance evaluation of mqtt and coap via a common middleware. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE.
- Wendel, P. (2011). Spark framework. <https://github.com/perwendel/spark>.
- Yokotani, T. and Sasaki, Y. (2016). Comparison with http and mqtt on required network resources for iot. In *Control, Electronics, Renewable Energy and Communications (ICCEREC), 2016 International Conference on*, pages 1–6. IEEE.