

## Concepção e Avaliação de Gateways Nativos para o Middleware EXEHDA

LEONARDO DA ROSA SILVEIRA JOÃO<sup>1</sup>; HUBERTO KAISER FILHO<sup>2</sup>; LUCAS XAVIER<sup>3</sup>; PATRÍCIA T. DAVET<sup>4</sup>; TAINÁ R. CARVALHO<sup>5</sup>; ADENAUER C. YAMIN<sup>6</sup>

*Universidade Federal de Pelotas*

{<sup>1</sup>ldrsojao, <sup>2</sup>hkaiser, <sup>3</sup>lmdsxavier, <sup>4</sup>ptdavet, <sup>5</sup>trcarvalho, <sup>6</sup>adenauer}@inf.ufpel.edu.br

### 1. INTRODUÇÃO

A Internet das Coisas, do inglês (*Internet of Things - IoT*), vem ganhando destaque como uma abordagem para promover a ubiquidade das soluções computacionais no mundo real, principalmente em relação aos aspectos infraestruturais. Nesta perspectiva, a infraestrutura computacional interage com o ser humano, provendo informação de qualquer “coisa”, a todo o momento, independente de localização, constituindo um ambiente altamente distribuído, heterogêneo, dinâmico, móvel, mutável e com forte interação entre homem e máquina [Perera et al. 2013].

O middleware EXEHDA (*Execution Environment for Highly Distributed Applications*) vem sendo desenvolvido pelo LUPS (*Laboratory of Ubiquitous and Parallel Systems*) para uso em frentes de pesquisa que tratam desafios da IoT [Souza et al. 2015] [Davet et al. 2015]. Dentre os desafios, são priorizados aqueles que dizem respeito à ciência de contexto e ao tratamento da heterogeneidade decorrente da diversidade de tecnologias de hardware e software presentes nos ambientes típicos da IoT.

Considerando a necessidade de obter informações contextuais provenientes do ambiente, o Subsistema de Adaptação e Reconhecimento do Contexto do EXEHDA emprega uma organização baseada em três tipos de equipamentos: Servidores de Contexto, Servidores de Borda e Gateways. Os Gateways coletam as grandezas físicas diretamente do meio, tratando a heterogeneidade dos diversos tipos de sensores, tanto em aspectos de hardware, como de protocolo de comunicação e transferem estas informações de forma normalizada aos Servidores de Borda. Por sua vez, os Servidores de Borda transmitem os dados recebidos aos Servidores de Contexto, os quais realizam as etapas de armazenamento e processamento.

O trabalho desenvolvido nesta Blosa de Iniciação Científica tem como objetivo geral, contribuição a aquisição de dados contextuais, auxiliando na aquisição de dados pelo Servidor de Borda.

Considerando a necessidade de otimização do consumo energético, os Gateways tem como característica serem dispositivos IoT com poder computacional limitado, o que introduz diversos desafios quando da sua concepção. O objetivo central deste artigo é relatar os esforços de concepção e avaliação dos Gateways Nativos projetados no LUPS para emprego no middleware EXEHDA.

De forma mais específica, o trabalho desenvolvido contemplou os seguintes objetivos: (i) sistematização de conceitos sobre os protocolos de comunicação estudados, (ii) implementação desses protocolos, a fim de transportar dados dos equipamentos monitorados, (iii) realização de testes para verificar o tempo médio e desvio padrão no transporte de informações.

## 2. METODOLOGIA

Os esforços de estudo e pesquisa desta Bolsa de Iniciação Científica contemplaram uma revisão bibliográfica sobre do vários tópicos relacionados a pesquisa. Dentre estes tópicos destacaríamos: conceitos e tecnologias da IoT, a arquitetura de software do middleware EXEHDA e conceitos em Ciência de Contexto.

Em particular destacaríamos a sistematização dos conceitos e funcionalidades referentes aos protocolos de comunicação MQTT (*Messaging Queue Telemetry Transport*) e o CoAP (*Constrained Application Protocol*), cuja implementação nos hardwares empregados pelo LUPS foi avaliada neste trabalho.

O protocolo de comunicação MQTT, foi desenvolvido em 1999 pela IBM (International Business Machines). A estrutura é baseada em um modelo de *publish/subscribe*, onde os clientes publicam (publish) informações que podem ser acessadas por outros clientes (subscribe) interessados [Carissimi A. (2016)].

O encapsulamento do PDU (*Protocol Data Unit*) do protocolo MQTT é realizado por meio do TCP, o que significa que o cabeçalho dos dados são enviados utilizando a área de dados do TCP. Existe outra versão para implementação deste protocolo, MQTT Sensor Network (MQTT-SN), a sua PDU é encapsulada pelo protocolo UDP.

Segundo [Jaffey 2014], o protocolo segue o modelo cliente/servidor. Os dispositivos estão conectados a um *broker*, tanto na produção de informações quanto na leitura das mesmas. O *broker*, é responsável por distribuir as mensagens entre os clientes.

O outro protocolo trabalhado, o CoAP, foi desenvolvido por um grupo de trabalho do IETF (*Internet Engineering Task Force*), chamado CoRE (*Constrained RESTful Environments*), por volta de 2010.

O CoAP é um padrão IETF, descrito no RFC7252, adaptando os conceitos do HTTP para dispositivos com restrições computacionais e de largura de banda. Baseado nesses dispositivos limitados, o CoAP emprega uma pilha de protocolos simples, baseada em UDP sobre IP [Martins I. R. 2013].

O mesmo trabalha seguindo o modelo de cliente/servidor, que disponibiliza os recursos por meio da URL. Onde os clientes podem acessar essas informações usando métodos pré-definidos como GET, PUT, POST e DELETE. No quesito segurança, o CoAP optou por usar DTLS (*Datagram Transport Layer Security*), equivalente ao uso de chaves RSA de 3072 bits, tornado o protocolo seguro [Carissimi A. (2016)].

Em relação a escolha do hardware para a concepção dos Gateways Nativos do EXEHDA, foram considerados os seguintes aspectos: baixo consumo energético, a possibilidade de operação sem fio (wi-fi) e empregar uma plataforma de software disseminada, o que contribui para o suporte de um maior número de sensores e atuadores.

Considerando os principais hardwares disponíveis no mercado: Arduíno, Raspberry, BeagleBone e outras plataformas de prototipação. A escolha recaiu sobre o SOC (*System On Chip*) ESP8266-12 que possui comunicação wi-fi integrada, podendo ser programado através da interface de programação do Arduino, além da Linguagem Lua através de um firmware denominado nodeMCU [Kurniawan 2015].

Em função do poder computacional reduzido da ESP8266-12, o protocolo de comunicação implementado tem impacto significativo no desempenho global do Gateway, construído com este processador.

Por fim, um cenário de testes foi concebido com o intuito de testar os protocolos de comunicação COAP e MQTT, considerando a transmissão de dados entre os módulos de recepção e transmissão, levando em consideração o tempo médio gasto para uma determinada quantidade de repetições realizada, priorizando uma avaliação do desempenho do Gateway concebido, considerando os protocolos mais usuais em IoT.

### 3. RESULTADOS E DISCUSSÃO

Foram empregados dois equipamentos para os testes: (i) como Servidor de Borda foi empregado um equipamento dotado de um processador core i5 5200U de 2.2 GHz e turbo boost de 2.7 GHz, 8 GB de RAM DDR3L e HD 1TB 5400 RPM, com distribuição Linux (Ubuntu 15.04) e (ii) um Gateway desenvolvido pelo grupo de pesquisa (Gateway Nativo) equipado com um processador ESP8266-12 [Kurniawan 2015].

No que diz respeito ao software, para os testes com o CoAP foi empregado no Servidor de Borda uma implementação em C do CoAP (libcoap), por sua vez, para os testes com o MQTT foi empregado no Servidor de Borda o software Mosquitto versão 1.4.3.

No Gateway Nativo, os protocolos CoAP e MQTT foram programados na Linguagem Lua, empregando as bibliotecas disponibilizadas pela comunidade que usufrui do hardware.

As tabelas 1 e 2 apresentam respectivamente os resultados dos testes de tempo médio dos pacotes enviados e o desvio padrão deste tempo. Pelos resultados da tabela 1, constatou-se que o Gateway Nativo empregando a ESP8266-12 apresentou melhor resultado empregando o CoAP, como pode ser visto a diferença de tempos entre os protocolos.

**Tabela 1. Tempo médio de envio dos pacotes**

Quantida de pacotes enviados	Tempo Médio(s)	
	CoAP	MQTT
30	0.2266	1.6321
300	1.7769	16.2005
3000	17.1876	162.3159
30000	172.8077	1653.2164

Um motivo para que ocorra essa situação de diferença nos tempos entre os protocolos, tem por base o protocolo de transporte utilizado, o CoAP basea-se no UDP sobre IP, enquanto o MQTT basea-se no TCP sobre IP. Por não implementar controle de sequenciamento no envio de mensagens, o protocolo UDP possui um melhor desempenho em relação ao protocolo TCP. Considerando a perspectiva de que a comunicação entre o Servidor de Borda e o Gateway aconteça através de rede local, onde o controle da sequencia sobre o envio e recepção dos pacotes não é prioritário, o uso de UDP mostra-se promissor.

O desvio padrão, apresentado na tabela 2, relata pouca variação entre repetições das execuções, mesmo que o protocolo CoAP apresente o desvio padrão maior, tempo de envio torna-se melhor em relação ao MQTT. Pode-se inferir pelos resultados mostrados que mesmo com o aumento significativo do número de repetições o valor médio tende a variar pouco.

**Tabela 2. Desvio padrão do tempo médio de envio dos pacotes**

Quantida de pacotes enviados	Desvio Padrão(s <sup>2</sup> )	
	CoAP	MQTT
30	0.0112224	0.0003149
300	0.0392909	0.0089945
3000	1.2360724	0.0147329
30000	19.2378341	2.8873984

#### 4. CONCLUSÕES

Este trabalho explorou critérios para concepção dos Gateways Nativos para o EXEHDA, além de apresentar uma comparação do desempenho do mesmo empregando os protocolos de comunicação CoAP e MQTT, avaliando o tempo médio dos pacotes enviados e o desvio padrão deste tempo, discutindo os resultados obtidos.

Como trabalhos futuros pretende-se realizar outros testes em relação ao consumo de banda e perda de pacotes, viabilizando uma melhor avaliação destes protocolos no uso dos Gateways Nativos.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- Davet, P., Kaiser Filho, H., João, L., Xavier, L., Carvalho, T., Fleischmann, A., Yamin, A., Lopes, J., and Souza, R. (2015). **Consciência de contexto na IoT: uma arquitetura distribuída e escalável**. In Proceedings of the V SBESC. CPS.
- Jaffey, T. (2014). **Mqtt and coap, iot protocols**. Disponível em: [http://www.eclipse.org/community/eclipse\\_newsletter/2014/february/article2.php](http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php).
- Kurniawan, A. (2015). **Sparkfun ESP8266 Thing Development Workshop**. PE PRESS, Canada.
- Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2013). **Context aware computing for the internet of things: A survey**. Communications Surveys & Tutorials, IEEE, 16(1):414–454.
- Souza, R. S., Lopes, J. L. B., Davet, P. T., Garcia, C. G., Gadotti, G. I., Yamin, A. C., and Geyer, C. F. R. (2015). **Context awareness in ubicomp: an iot oriented distributed architecture**. In Proceedings of 2015 IEEE International Conference on Electronics, Circuits, and Systems. IEEE.
- Martins I. R., Z. J. L. (2013). **Estudo dos protocolos de comunicação mqtt e coap para aplicações machine-to-machine e internet das coisas**. Revista Técnologica de Fatec Americana, 1(1):64–86.
- Carissimi A. (2016). **Internet das Coisas, middlewares e outras coisas**. XVI Escola Regional de Alto Desempenho, São Leopoldo, Brasil.