O auto-registro de serviços por dispositivos contribui com a escalabilidade de ambientes de IoT

Caio C. M. Silva, Francisco L. de Caldas, Felipe D. Machado, Rafael T. de Sousa Jr e Fábio L. L. Mendonça.

***Resumo*— Este artigo apresenta uma abordagem de auto-cadastramento de serviços de dispositivos em ambientes de IoT. Em contraponto a trabalhos que utilizam uma abordagem centralizada, a proposta apresentada neste trabalho fundamenta-se na ideia de que o próprio dispositivo deve ser capaz de registrar seus serviços na rede IoT, portanto de maneira distribuída. Também, a proposta emprega conceitos de sensibilidade ao contexto para o próprio dispositivo indicar quais informações devem ser propagadas na rede IoT. Com a proposta desenvolvida em termos de especificação de informações de contexto, desenho de fluxo de registro e interação dispositivos-rede, resultados experimentais são apresentados indicando que a proposta resulta em aumento da escalabilidade de uma rede IoT.**

***Palavras-Chave*— Internet das coisas (IoT), Serviços de IoT, Abstração de Dispositivos, Sensibilidade ao contexto.**

***Abstract*— This paper presents an approach for self-registration of device services within IoT environments. In contrast to studies that use a centralized approach, the proposal presented in this paper is based on the idea that the device itself should be able to register its services in the IoT network, so in a distributed manner. Also, the proposal uses concepts of sensitivity to context for the device itself indicate what information should be propagated in the IoT network. With the proposal developed in terms of context information specification and registration process flow, experimental results are presented indicating that the proposal results in increased scalability of an IoT network.**

***Keywords*— Internet of Things (IoT), IoT Services, Device Abstraction, Sensitivity to Context.**

I. INTRODUÇÃO

A quantidade de dispositivos de internet das coisas (IoT) está em contínuo crescimento, apontando para uma magnitude de milhões de dispositivos nas redes de IoT em um futuro próximo. Tal número, associado à variedade de serviços providos, pode caracterizar um cenário impeditivo para o gerenciamento de tais dispositivos e serviços, caso não sejam utilizadas abordagens avançadas para o seu registro e organização em uma rede IoT.

Tradicionalmente, na literatura, o gerenciamento de serviços em redes IoT é concebido em uma abordagem centralizadora, em que um conjunto de nós na rede tem a responsabilidade de tomar a iniciativa da descoberta dos dispositivos e serviços, além de realizar a gerência dos mesmos. Tal responsabilidade pode também ser atribuída a um servidor com alto poder de processamento e comunicação, por exemplo, uma nuvem computacional.

Ambos os cenários fundamentalmente dificultam a escalabilidade, haja visto que o gerencimento com entidades centralizadoras esbarra no limite quanto à quantidade de recursos ativos disponíveis em um determinado momento. Ademais, tal forma de estruturação desconsidera o crescimento do poder computacional dos próprios dispositivos atuais, que são capazes de realizar diversos processamentos antes de propagar informações em uma rede.

Tendo em vista que um dispositivo pode fornecer um conjunto de serviços, e dado os recursos computacionais presentes nos microcontroladores de hoje, neste artigo é proposta uma abordagem para que o próprio dispositivo possua o controle sobre os serviços que disponibiliza, e tome a iniciativa de atualizar a rede sobre novas informações relativas a cada um de seus serviços, visando aumentar a escalabilidade da rede, além de possibilitar um melhor controle sobre a quantidade de dados trafégados.

O trabalho está organizado da seguinte forma: a seção II traz uma breve discussão sobre os trabalhos relacionados à pesquisa. O processo de auto-registro do dispositivo e o conjunto de informações de contexto utilizadas são apresentados na seção III. Na seção IV são apresentados os resultados experimentais. Finalmente, conclusões e trabalhos futuros são apresentados na seção V.

II. TRABALHOS RELACIONADOS

Nesta seção são apresentados os trabalhos relacionados à pesquisa, abrangendo as áreas de ambientes de IoT e sensibilidade a contexto.

Caio C. M. Silva, Francisco Lopes, Felipe D. Machado, Rafael T. de Sousa Jr e Fábio L. L. Mendonça, Laboratório de Tecnologias da Tomada de Decisão, Faculdade de Tecnologia, Universidade de Brasília, DF, Brasil, E-mails: caio.silva, francisco.lopes, felipe.duarte, rafael.desousa, fabio.mendonca@redes.unb.br.

Alguns trabalhos de pesquisa em IoT adotam uma abordagem orientada a serviço para atender os requisitos de topologias de rede desconhecidas e dinâmicas. Enquanto alguns projetos focam na abstração dos dispositivos na forma de serviços de rede [1] [2] [3] [4], outros dedicam atenção às abstrações de informações/dados e suas integrações com os serviços [5] [6] [7].

Um desafio comum para as diversas soluções supracitadas é o desconhecimento da topologia utilizada, o que é usualmente tratado com processos de *discovery* em grande parte baseados nas abordagens tradicionais para descoberta de serviços/dispositivos utilizadas na Internet, ou em ambientes ubíquos e redes de sensores e atuadores sem fio [8] [9].

Por exemplo, SOCRATES [4] provê a descoberta em dois níveis, designados como nível do dispositivo e nível do serviço, neles utilizando ou o WS-Discovery para serviços web (*web services* - WS) ou um mecanismo de descoberta RESTful (para serviços *representational state transfer* - REST). Nos citados trabalhos, também é possível identificar que a abordagem para descoberta dos serviços disponíveis na rede é realizada sob responsabilidade de uma entidade centralizadora.

Na proposta do presente artigo, é defendida a ideia de que um dispositivo deva realizar o seu auto-registro em uma rede IoT, dando a oportunidade a cada dispositivo de atuar de forma independente na rede, realizando requisições somente para os nós com os quais haja interesse de interação. Para que seja possível habilitar a operação de auto-registro, cada dispositivo deve ser capaz de entender o contexto computacional em que está inserido. Segundo [10], o contexto computacional pode ser entendido como o conjunto de todos os aspectos técnicos relacionados às capacidades e recursos computacionais.

Mais especificamente, o contexto computacional possui dois objetivos principais, o primeiro sendo expressar a heterogeneidade que normalmente está presente em ambientes de IoT, inclusive as diferentes capacidades computacionais dos dispositivos e os aspectos de conectividade [11] [12]. Em segundo lugar, o contexto objetiva considerar os diferentes recursos que um dispositivo de IoT encontra enquanto está em *roaming* [13]. Isso posto, a utilização de informações de contexto permite que o dispositivo possa realizar processamentos para reduzir a quantidade de dados comunicados [14] sem prejuizo ao funcionamento da rede IoT.

Assim, a sinergia da abordagem de auto-registro do dispositivo com a consideração de aspectos de sensibilidade ao contexto favorece a escalabilidade da rede IoT, dado que tem o efeito resultante de diminuir o tráfego de dados na rede, além tornar desnecessária uma entidade centralizadora para descoberta dos dispositivos e serviços.

III. Proposta de Auto-registro de Serviços

Nesta seção é apresentada a proposta de auto-registro de serviços pelos dispositivos em ambientes de IoT, sendo apresentadas as seguintes contribuições, e as respectivas subseções onde são tratadas no artigo. Primeiramente é discriminado o conjunto de informações de contexto que habilitam o processo de auto-registro, servindo para estabelecer a identificação e os serviços do dispositivo em uma rede IoT, conforme detalhado na subseção III-A. Adicionalmente, é definido do fluxo de informações para o registro de serviços em uma rede IoT, cujos passos são apresentados na subseção III-B.

A. Informações de contexto

Nesta subseção são apresentadas as informações de contexto relevantes para permitir o registro de serviços em um ambiente de IoT. Cada conjunto de informações foi organizado como um "documento" que deve ser enviado pelo dispositivo para a rede IoT. Foram definidos três tipos de documentos: **documento de identidade**, **documento de registro** e **documento de atualização**.

É importante ressaltar que os documentos foram definidos utilizando o padrão JSON, dado que esse formato utiliza uma menor quantidade de bytes para representar uma informação se comparado com outras abordagens como, por exemplo, XML [15].

1) Documento de identidade: O documento de identidade possui as informações relativas ao contexto computacional do dispositivo IoT.

O dispositivo IoT deve ter a capacidade de obter essas informações, através de um algoritmo operante em seu próprio contexto, e compô-las em um documento de identidade, um exemplo do qual sendo apresentado no trecho de código em formato JSON 1. As informações de **Chipset ID**, **Processor ID** e **Mac address**, são compostas e utilizadas pela rede IoT para gerar um ID único do dispositivo para a rede. Já o **host name** é utilizado pela rede para enviar informações ao dispositivo quando necessário, e para permitir (ou negar) envios de requisições do dispositivo para a rede.

JSON 1
EXEMPLO DE DOCUMENTO DE IDENTIDADE PARA DISPOSITIVOS IOT

```
{id_cps : 4C4C4544004731108047B4C04F4C3232,
 id_prc : 51 06 04 00 FF FB EB BF,
 hd_srl : W761TTGL, driver : ethernet,
 mac    : 74:e6:e2:ce:23:6d, host : rpy-iot}
```

As informações sobre as tecnologias de comunicação são utilizadas pela rede IoT para

que seja possível identificar a tecnologia mais adequada para realizar a troca de informações, além de permitir uma maior disponibilidade do dispositivo. Aqui é importante ressaltar que por princípio uma rede IoT deve ser capaz de comunicar uma determinada informação independentemente da tecnologia de comunicação utilizada. Como discutido em [16], ambientes de IoT devem suportar diferentes formas de comunicação de maneira transparente para os clientes da rede.

O dispositivo deve primeiramente enviar o documento de identidade para a rede IoT, recebendo em resposta um *id token* que será utilizado nas interações seguintes. Nestas, o dispositivo deve informar, conjuntamente com o *idtoken*, o endereço físico (mac) para que seja possível identificar por qual interface de comunicação a requisição foi enviada.

2) Documento de registro: O dispositivo IoT deve informar quais são os serviços que ele deseja oferecer para a rede, realizando o registro dos serviços através do documento de registro. Neste, cada serviço é caracterizado por quatro informações nome, tipo da informação, unidade da informação e descrição. Um exemplo desse documento encontra-se no trecho de código JSON 2.

JSON 2
EXEMPLO DE DOCUMENTO DE REGISTRO PARA DISPOSITIVOS IOT

```
{id_token : 11768768, mac : 74:e6:e2:ce:23:6d,
 services:[{name : get_temp, type : float,
 unit : celsius, desc : temp sensor}]}
```

Como um determinado dispositivo pode possuir a capacidade de oferecer mais de um serviço, no documento de registro deve estar inserida uma lista com todos os serviços que o dispositivo deseja oferecer para o middleware. Além da coleção de serviços, é necessário que seja inserido no documento o *id token* válido e *mac* do dispositivo. O *mac* deve ser enviado para que seja verificado se um determinado dispositivo trocou sua interface de rede, podendo essa troca ser caracterizada como um comportamento suspeito.

Na resposta ao envio do documento de registro, a rede vai informar o identificador de cada serviço cadastrado, bem como o tempo de envio das informações para cada serviço requisitado ao dispositivo.

3) Documento de atualização: Este terceiro documento serve ao envio das informações coletadas pelo dispositivo, ou seja, para cada serviço cadastrado, o dispositivo deve enviar as pertinentes informações coletadas. Por exemplo, caso o dispositivo tenha cadastrado um serviço para informar a temperatura, outro para informar a humidade e outro para enviar informações de autenticação de RFID, tal dispositivo deve usar um documento de atualização para enviar a temperatura atual, a humidade a atual e se houve alguma tentativa de autenticação por RFID. Um exemplo de um documento de registro encontra-se no trecho de código JSON 3.

JSON 3
EXEMPLO DE DOCUMENTO DE ATUALIZAÇÃO PARA DISPOSITIVOS IOT

```
{id_token : 11768768, mac : 74:e6:e2:ce:23:6d
 services:[{id_serv : 001, value: 24}]}
```

O dispositivo saberá quando deve enviar uma informação relativa a um serviço, pois a rede IoT terá fornecido esse parâmetro na operação precedente de registro do serviço.

B. Processo de registro

O processo de registro entre o dispositivo e a rede IoT, com a separação de seus componentes e intereções, é apresentado na figura 1.

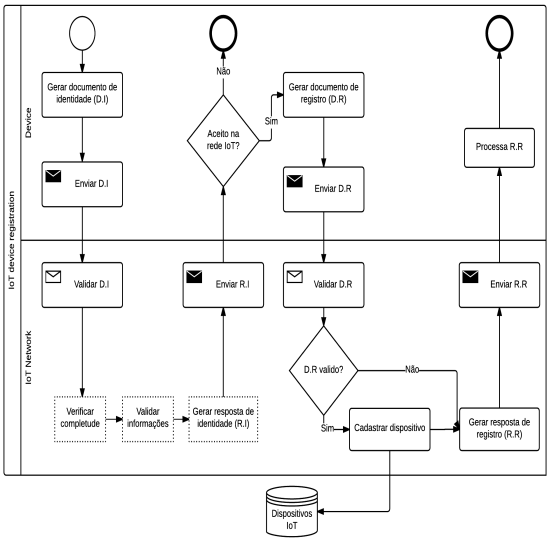

Fig. 1. Processo de auto-registro de serviços para dispositivos IoT

O processo se inicia com o próprio dispositivo gerando o documento de identidade, o que requer capturar suas informações, inseri-las no documento de identidade e enviar este último para a rede IoT. A rede recebe o documento e realiza o processo de validação (PV) do documento, em três fases fundamentais:

- **Verificar completude:** Nesta fase é verificado se todas as informações necessárias para gerar a identidade do dispositivo estão disponíveis no documento. Caso alguma informação não tenha sido enviada, o dispositivo não será aceito na

rede IoT. Avaliar a completude é um dos requisitos fundamentais para se garantir a qualidade de contexto [17].

- **Validar as informações:** É verificada a validade de cada informação, processo este que ocorre através de definições de políticas de contexto [14]. Cada informação possui um conjunto de políticas que a definem como valida ou não.
- **Gerar resposta de identidade:** A resposta de identidade contém um código informando se a operação foi bem sucedida ou não, na forma de uma mensagem especificando o código e o *id token* caso o dispositivo tenha sido aceito na rede. Um exemplo dessa resposta é apresentado no trecho de código JSON 4.

JSON 4
EXEMPLO DE RESPOSTA PARA DOCUMENTO DE IDENTIDADE

```
{code : 200, id_token : 11768768}
```

Tendo a rede IoT gerado e enviado a resposta para o dispositivo IoT, este será responsável por interpretar o documento e avaliar se foi aceito ou não pela rede. Caso não tenha sido aceito o dispositivo deve iniciar todo o processo novamente. Porém, se o documento de identidade foi validado pela rede, o dispositivo deve assumir o *id token* como seu identificador único, e gerar o documento de registro, este informando a especificação de cada um dos serviços que o dispositivo deseja realizar na rede IoT. Após gerar o documento de registro, o dispositivo deve enviá-lo à rede, que, ao receber o documento, realiza o respectivo processo de validação. Caso o documento de registro seja válido, os serviços são registrados em um banco de dados do middleware distribuído de IoT. Caso contrário, o pedido é desconsiderado, sendo gerada uma resposta negativa para o dispositivo.

O documento de resposta de registro contém um código informando se a operação foi realizada com sucesso ou não, e em caso de sucesso, uma lista de serviços que discrimina, para cada serviço informado pelo dispositivo, o nome do serviço, o id do serviço e o *refreshing time*. O nome na lista é o mesmo nome provido pelo dispositivo, o id do serviço representa o identificador único do serviço gerado pela rede, e o *refreshing time* é o intervalo de tempo requisitado ao dispositivo para envio das informações relativas ao serviço. O trecho de código JSON 5 apresenta um exemplo de resposta para documento de registro.

JSON 5
EXEMPLO DE RESPOSTA PARA REGISTRO DE IDENTIDADE

```
{code : 200, services:[{name : get_temp,
id_serv : 001, timing : 60}]}
```

Por fim, após receber uma resposta de sucesso da rede IoT, o dispositivo está registrado na rede com seus serviços associados e, a partir desse momento, passa a enviar os documentos de atualização dentro da especificação de *refreshing time* fornecida pela rede.

IV. EXPERIMENTOS E ANÁLISE DOS RESULTADOS

Com a finalidade de validar a proposta, foram realizadas duas implementações (denominadas **ativa** e **passiva**) do middleware de IoT, aqui representado por um servidor. Na abordagem ativa, o servidor é responsável pela descoberta dos serviços. Já na abordagem passiva, são os dispositivos que devem enviar ao servidor as informações sobre os serviços. A escalabilidade do sistema foi avaliada através da verificação do atraso no tempo de resposta do servidor à medida que o número de dispositivos cresce. O ambiente experimental é apresentado na tabela I.

Para realizar os testes, foi utilizada a ferramenta JMeter. O cenário de teste se inicia com um dispositivo e, no segundo seguinte, é criado um segundo dispositivo e assim, continuamente, até o limite de trezentos mil dispositivos. Cada dispositivo envia informações no intervalo de [1, 2] segundos. Para cada requisição, é captado o tempo de resposta, tanto para a abordagem passiva quanto para a ativa. A figura 2 apresenta o valor médio do tempo de resposta à medida que cresce o número de dispositivos na rede.

Fig. 2. Relação entre número de dispositivos e tempo de resposta

Os resultados indicam que a medida que o número de dispositivos cresce na rede o tempo de resposta aumenta para ambas as abordagens. Porém, a abordagem ativa possui um aumento de até 42% no tempo de resposta em relação a abordagem passiva. De fato, a diferença no tempo médio de resposta entre duas abordagens está no intervalo de [-1.2, 1358] milissegundos, ou seja, a abordagem ativa foi até 1,3 segundos mais lenta para responder uma mesma requisição, em média.

TABELA I
ESPECIFICAÇÃO DO AMBIENTE EXPERIMENTAL

Nó	**SO**	**Processador**	**Memoria**	**Armazenamento**	**Simulador**
Dispositivos	Ubuntu 14.04	Intel i5 2.5GHz	4GB	1TB	JMeter
Middleware	CentOS 6	Intel XEON 2.79GHz	32GB	4TB	Apache

Além disso, a medida que o sistema é escalado, pode ser percebido um aumento gradativo na diferença entre as duas abordagens. Os experimentos indicaram que para cenários com mais de trinta mil dispositivos a aboradgem ativa tem um perda de 38% no tempo de resposta, no melhor dos casos. Assim, tal abordagem pode ser vista como impeditiva para ambientes IoT onde se estima uma quantidade de dispositivo na magnitude de milhões.

Dessa forma, pode ser argumentado que a utilização de uma abordagem onde o dispositivo informa ao servidor suas informações, serviços e seu estado atual, isto é, realiza seu auto-registro, pode favorecer a escalabilidade de middlewares para internet das coisas.

V. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo propõe uma abordagem para auto-registro de serviços em ambientes de IoT. A proposta argumenta que dispositivos de IoT devem ser capazes de identificar seus próprios serviços e enviá-los a rede. Para tanto, os dispositivos devem enviar documentos de identificação, registro e atualização, documentos estes que possuem informações de contexto sobre o ambiente em que o dispositivo se encontra.

Resultados experimentais coletados de duas implementações indicam que a abordagem proposta, isto é, com o dispositivo enviando informações para a rede IoT, favorece a escalabilidade da rede, tendo em vista que o tempo de resposta das requisições é até 42% menor em relação a abordagem onde tal iniciativa cabe a uma entidade centralizadora, à medida em que a quantidade de dispositivos na rede aumenta.

Para trabalhos futuros é sugerido um refinamento do modelo de contexto, para caracterizar todo o ambiente em que um dispositivo de IoT está inserido. Além disso, objetiva-se realizar processamentos de filtragem no próprio dispositivo para que a quantidade de informações por ele veiculadas na rede seja reduzida.

REFERÊNCIAS

[1] M. Eisenhauer, P. Rosengren, and P. Antolin, "Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems," in *The Internet of Things*. Springer, 2010, pp. 367–373.

[2] W. Zhang and K. M. Hansen, "Semantic web based self-management for a pervasive service middleware," in *Self-Adaptive and Self-Organizing Systems, 2008. SASO'08. Second IEEE International Conference on*. IEEE, 2008, pp. 245–254.

[3] M. Presser, P. M. Barnaghi, M. Eurich, and C. Villalonga, "The sensei project: integrating the physical world with the digital world of the network of the future," *Communications Magazine, IEEE*, vol. 47, no. 4, pp. 1–4, 2009.

[4] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *Services Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 223–235, 2010.

[5] D. Massaguer, B. Hore, M. H. Diallo, S. Mehrotra, and N. Venkatasubramanian, "Middleware for pervasive spaces: Balancing privacy and utility," in *Middleware 2009*. Springer, 2009, pp. 247–267.

[6] J. Honkola, H. Laine, R. Brown, and O. Tyrkko, "Smart-m3 information sharing platform," in *The IEEE symposium on Computers and Communications*. IEEE, 2010, pp. 1041–1046.

[7] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Mobile Data Management, 2007 International Conference on*. IEEE, 2007, pp. 198–205.

[8] F. Zhu, M. W. Mutka, and L. M. Ni, "Service discovery in pervasive computing environments," *IEEE Pervasive computing*, no. 4, pp. 81–90, 2005.

[9] E. Meshkova, J. Riihijärvi, M. Petrova, and P. Mähönen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," *Computer networks*, vol. 52, no. 11, pp. 2097–2128, 2008.

[10] G. Chen, D. Kotz *et al.*, "A survey of context-aware mobile computing research," Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, Tech. Rep., 2000.

[11] A. Bartolini, M. Ruggiero, and L. Benini, "Visual quality analysis for dynamic backlight scaling in lcd systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*. European Design and Automation Association, 2009, pp. 1428–1433.

[12] S. Ceri, F. Daniel, M. Matera, and F. M. Facca, "Model-driven development of context-aware web applications," *ACM Transactions on Internet Technology (TOIT)*, vol. 7, no. 1, p. 2, 2007.

[13] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. IEEE, 1994, pp. 85–90.

[14] C. Silva and M. Dantas, "Quality-aware context provider: A filtering approach to context-aware systems on ubiquitous environment." in *WiMob*, 2013, pp. 422–429.

[15] C. C. de Melo Silva, H. G. C. Ferreira, R. T. de Sousa Júnior, F. Buiati, and L. J. G. Villalba, "Design and evaluation of a services interface for the internet of things," *Wireless Personal Communications*, pp. 1–38, 2016.

[16] H. Kopetz, "Internet of things," in *Real-time systems*. Springer, 2011, pp. 307–323.

[17] Y. Kim and K. Lee, "A quality measurement method of context information in ubiquitous environments," in *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, vol. 2. IEEE, 2006, pp. 576–581.