

Um mecanismo de descoberta sensível ao contexto de objetos inteligentes em Internet das Coisas

Felipe Mota Barreto
Universidade Federal do
Ceará
Fortaleza - CE - Brasil
felipebarreto@great.ufc.br

Windson Viana
Universidade Federal do
Ceará
Fortaleza - CE - Brasil
windson@great.ufc.br

Rossana Maria de Castro
Andrade
Universidade Federal do
Ceará
Fortaleza - CE - Brasil
rossana@great.ufc.br

ABSTRACT

In the Internet of Things vision, smart objects are interconnected providing applications at everyday life places (e.g., house, cars, school, buildings). The number of smart objects will increase every year, which creates a need for contextual discovery services of those devices and their provided services. Web based protocols are widely used by IoT applications to interact with smart objects, among them, the most used is the CoAP protocol. This work presents a discovery mechanism that uses contextual information to select the things that are relevant to the user or to the system itself. To perform the discovery, the mechanism extends the default discovery service from CoAP, adapting it to the new IoT requirements while keeping the Web compatibility provided by CoAP.

Keywords

Internet das Coisas; CoAP; Contexto; Descoberta de Objetos Inteligentes

1. INTRODUÇÃO

Internet das Coisas, do inglês Internet of Things (IoT), é um novo paradigma tecnológico que emerge no cenário já consolidado das redes de comunicações sem fio. A ideia básica desse conceito é a constante e invisível presença, no cotidiano das pessoas, de uma enorme variedade de dispositivos computacionais. Alguns desses dispositivos possuem poder computacional, juntamente com a capacidade de comunicação e certa suficiência energética, e são chamados de Objetos Inteligentes (OI). Exemplos de objetos inteligentes são os sensores de ambiente, as TVs inteligentes, as câmeras de vídeo, as lâmpadas, os *smartphones*, etc. Esses objetos são unicamente endereçáveis, capazes de interagir, trocar dados entre si e ainda cooperar com seus vizinhos para realizar tarefas em comum [1].

Ao longo dos últimos anos, o volume mundial de objetos inteligentes cresceu rapidamente. A previsão é de que no ano de 2020 será alcançada uma marca de 50 bilhões de dispositivos conectados [4]. Grande parte desses dispositivos se-

rão compatíveis com arquiteturas e protocolos Web (HTTP, REST, Services, entre outros). Essa previsão impõe uma discussão sobre as abordagens tradicionais em descoberta de dispositivos e se elas ainda serão adequadas após esse crescimento no número de objetos inteligentes. As abordagens tradicionais tem por principal objetivo descobrir e tornar acessível todos os dispositivos alcançáveis, sejam esses em um escopo local ou global. Entretanto, se o número de dispositivos for muito elevado, uma abordagem mais interessante seria priorizar (ou recomendar) de alguma forma os objetos inteligentes de maior interesse para o usuário, por exemplo, a partir do contexto capturado por seu *smartphone* (e.g, localização, período do dia, histórico, etc.). Além das vantagens diretamente relacionadas ao usuário, com uma descoberta mais seletiva é possível economizar os recursos computacionais desses dispositivos. Dispositivos esses que possuem por natureza limitações de energia da bateria e de memória [6].

A necessidade de uma escolha mais refinada sobre os objetos inteligentes a serem descobertos se torna mais clara em um cenário de Internet das Coisas no qual o número desses objetos é muito grande, como em cidades ou centros comerciais inteligentes. A sensibilidade ao contexto pode ser definida como a capacidade de um sistema de caracterizar o estado atual e as modificações do ambiente no qual o usuário está inserido ou do próprio usuário [12]. Essa sensibilidade ao contexto permite melhorar os serviços existentes de descoberta de recursos nesse tipo de ambiente. A seguir, um cenário motivador de IoT e sensibilidade ao contexto é apresentado para ilustrar essas situações. É comum que pessoas se desloquem da casa ao trabalho diariamente. Ana é uma dessas pessoas. Bastante ligada ao uso de tecnologia, costuma acordar cedo para ir ao trabalho, mas não sem antes conferir a previsão do tempo pela televisão. Com seu *smartphone*, Ana consegue acessar e ligar a televisão, bem como selecionar o canal que está sendo transmitida a previsão. Ela então segue para o trabalho, e usa o *smartphone* para selecionar o noticiário econômico no rádio do carro. Ao chegar no seu destino, Ana prepara a sala de seminários da empresa para uma reunião, novamente utiliza seu *smartphone* para ligar o projetor e as luzes do ambiente, adiantando assim o início da reunião.

O cenário apresentado possui três ambientes distintos: a casa, o carro e o local de trabalho. A sensibilidade ao contexto atua em todos eles. Na situação da casa e do carro, embora a tendência seja que o número de dispositivos conectados aumente, pode-se assumir que o volume total ainda

In: Workshop de Teses e Dissertação (WTD), 16., 2016, Teresina. Anais do XXII Simpósio Brasileiro de Sistemas Multimídia e Web. Porto Alegre: Sociedade Brasileira de Computação, 2016. v. 2.

ISBN: 978-85-7669-332-1

©SBC – Sociedade Brasileira de Computação

será controlado. Nesses casos, a sensibilidade ao contexto pode atuar na melhoria do gerenciamento dos dispositivos. No exemplo anterior, dispositivos que não fossem do interesse de Ana naquele momento (televisão e rádio) poderiam ser desligados temporariamente. Já na outra situação, em um ambiente de trabalho, geralmente o número de objetos inteligentes é bem maior, e também compartilhado por vários usuários (microfones, caixas de som, projetores, etc.). Nesse caso, a vantagem de se utilizar de técnicas de sensibilidade ao contexto na descoberta desses objetos está na interação entre usuário e sistema. Para Ana seria difícil conseguir interagir rapidamente com os objetos de interesse caso ela tivesse que procurar dentre os inúmeros que existissem na empresa.

Este trabalho apresenta um mecanismo de descoberta de objetos inteligentes que utiliza informações contextuais para selecionar os dispositivos que atendam ao interesse do sistema ou do próprio usuário. O papel dessa descoberta é fazer com que os objetos inteligentes relevantes se tornem disponíveis ao sistema que utiliza esse serviço, permitindo o acesso e o controle desses objetos de forma automática, sem, ou com a mínima, intervenção do usuário final. A abordagem proposta pode ser aplicada tanto em arquiteturas de descoberta de objetos inteligentes totalmente baseadas em protocolos Web, como em cenários que suportam parcialmente esses protocolos (e.g., CoAP). Para ilustrar a proposta, foi concebido um processo de descoberta que realiza um mapeamento entre interesse do usuário e do sistema e o método final de busca e seleção, que, para este trabalho, são consultas CoAP.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Definição de Contexto

Existem inúmeras definições de contexto na literatura. Uma das primeiras foi proposta por Dey, onde contexto é qualquer informação que pode ser usada para caracterizar a situação de um elemento que é relevante para a interação entre usuário e sistema, incluindo como elementos, o próprio usuário e sistema [5]. A definição que será adotada neste trabalho, proposta em [15], pode ser vista como uma extensão da anterior, removendo a limitação do contexto com a interação entre usuário e sistema. Contexto agora passa a ser um conjunto de informações que podem descrever a situação das entidades (e suas relações) envolvidas em uma ação que seja julgada importante para o sistema (e.g., intenções, anotações, etc.).

O contexto definido em [15] pode ser interpretado como sendo a interseção entre duas regiões, ou zonas. A primeira delas, chamada de Zona de Interesse (ZI) representa o conjunto de entidades que o sistema julgue importante em um determinado instante de tempo, ou seja, quais informações contextuais o sistema tem interesse em obter. A segunda, chamada de Zona de Observação (ZO), são as entidades que podem ser acessadas e obtidas nesse mesmo instante de tempo pela infraestrutura de aquisição de contexto. Contexto então é conjunto de informações que podem ser obtidas e que são importantes para o sistema em dado instante arbitrário de tempo. A Figura 1 exemplifica visualmente essa definição.

Essa definição de contexto possui duas características importantes. O contexto é tanto dinâmico (as informações fornecidas por cada uma das entidades muda com o tempo)

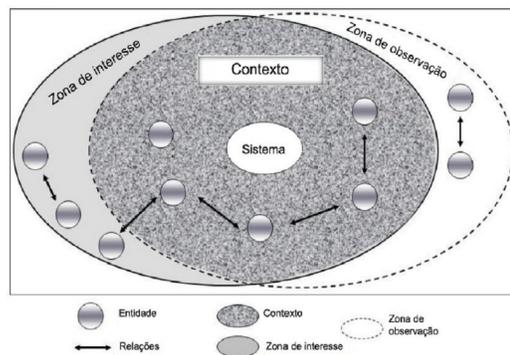


Figura 1: Contexto como sendo a interseção entre as Zonas de Interesse e de Observação [7].

quanto evolutivo (as próprias entidades que compõem o contexto podem mudar) [7]. Isso faz com que a aplicação dessa definição em sistemas móveis e ubíquos seja bastante adequada, pois essas características são inerentes aos tipos de sistemas que fazem parte desse cenário.

2.2 Internet das Coisas

Devido ao avanço na indústria de sensores, tais componentes estão se tornando cada vez mais baratos, menores e poderosos. Como resultado, a utilização desses sensores já atingiu uma escala considerável, além de que esse número deve crescer ainda mais rapidamente nos próximos anos. Esse grande número de sensores criará uma grande massa de dados, ou *big data*. Esses dados podem não ter nenhum valor caso não sejam analisados, interpretados e o mais importante, entendidos. É por isso que os desafios de sensibilidade ao contexto que eram aplicáveis aos paradigmas de computação ubíqua continuarão existindo, agora em IoT [14].

Embora Internet das Coisas não possua uma definição única, a grande maioria das definições encontradas na literatura giram em torno de uma mesma ideia. Nessa de que em um cenário de IoT, objetos inteligentes estão interconectados de forma a permitir a troca de dados entre eles mesmos e o ambiente, enquanto reagem a eventos criados com ou sem a intervenção humana. Outro aspecto importante é a utilização de padrões para a comunicação desses objetos inteligentes, buscando assim atingir a interoperabilidade entre os mais diversos dispositivos.

Quando analisada pelo ponto de vista de descoberta de objetos inteligentes em IoT, a definição de contexto adotada pode ser usada como base conceitual dessa descoberta. As entidades que compõem o sistema podem ser vistas como os OI que formam esse ambiente IoT. Com isso, a lista de dispositivos relevantes ao sistema, e que podem ser acessados em um dado momento, pode ser vista como a interseção entre os objetos de interesse do usuário e os OI que fazem parte da zona de observação.

2.3 Descoberta de Objetos Inteligentes

Descoberta de serviços é um aspecto conhecido do desenvolvimento de sistemas distribuído. Crasso define essa descoberta como o processo de encontrar os serviços adequados para resolver uma determinada tarefa em particular [3]. Descoberta de objetos inteligentes, para um cenário de Internet das Coisas, pode ser comparada à descoberta de serviços,

possuindo diversas características comuns. Porém, é preciso levar em consideração requisitos que são específicos desse novo paradigma de IoT. Guinard lista alguns desses requisitos específicos para o processo de descoberta de objetos inteligentes [8].

1. **Mínimo *Overhead* no serviço de descoberta:** Como a maioria dos objetos inteligentes do mundo real são dispositivos com baixo poder computacional, existe uma necessidade de utilização de paradigmas de descoberta menos complexos, que não tragam muita sobrecarga ao dispositivo.
2. **Mínimo esforço no registro de dispositivos:** Um dispositivo deve ser capaz de anunciar seus serviços a um servidor de registros através da rede. Esse processo precisa ser realizado sem nenhuma intervenção humana. A quantidade de informações necessárias para esse registro também deve ser bem reduzida.
3. **Suporte a busca contextual e dinâmica de dispositivos:** Os algoritmos de busca devem ir além de uma simples busca por palavras-chave. Eles devem levar em consideração dados dinâmicos do contexto do usuário, como localização.
4. **Suporte a alocação de serviços sob demanda:** Os serviços ofertados por cada objeto inteligente devem ser ativados ou desativados sob demanda, evitando assim a má utilização dos recursos.

A proposta deste trabalho busca estender o mecanismo padrão de descoberta do CoAP de modo a atender o terceiro requisito, busca contextual de objetos inteligentes.

2.4 CoAP

O *Constrained Application Protocol* (CoAP) é um protocolo de troca de mensagens focado especificamente em dispositivos limitados computacionalmente. CoAP foi projetado para ser uma alternativa ao HTTP para aplicações *machine-to-machine* (M2M), como por exemplo sistemas de automação residencial.

CoAP dá suporte a uma comunicação entre aplicações e objetos inteligentes seguindo o paradigma requisição/resposta. Possui um serviço de descoberta já implementado, baseado no conceito de diretórios de dispositivos. Sua estrutura é baseada no HTTP, facilitando assim a integração com os recursos disponíveis na WEB. Porém, diferente do HTTP, o CoAP cumpre alguns requisitos específicos para dispositivos com limitações computacionais, como o baixo *overhead* na troca de mensagens.

Clientes se utilizam de servidores CoAP para acessar o serviço de descoberta. Uma vez conhecendo o endereço do servidor, o cliente pode acessar a lista de dispositivos mantida por ele. Cada objeto inteligente é representado por uma URI, seguindo o formato especificado pelo *Constrained RESTful Environments* (CoRE). Essa especificação também é responsável pelo serviço de descoberta presente no CoAP. A busca por objetos inteligentes pode ser realizada com a utilização de filtros, que são parâmetros adicionados a *string* de consulta por dispositivos, enviada ao respectivo diretório de recursos. O CoRE também define um conjunto de atributos que representam os recursos presentes em um diretório. Os principais atributos são: *Resource Type*, responsável por

identificar a função de um determinado recurso (temperatura, luminosidade, impressora, etc.); *Interface Description*, indica os métodos que podem ser utilizados para a comunicação com esse recurso (*GET*, *POST*, etc); *Context Type*, representa o formato dos dados fornecidos pelo recurso.

Como o CoAP já é um protocolo bastante utilizado em IoT e possui inúmeras implementações disponíveis, para as mais diversas plataformas, isso faz com que exista uma grande quantidade de produtos que seguem suas especificações. Com base nisso, criar soluções que também utilizem o CoAP pode garantir a interoperabilidade com essa gama imensa de dispositivos já ativos. Embora essa interoperabilidade não possa ainda ser estendida para IoT em geral, garantir a compatibilidade com o CoAP é uma estratégia que potencializa a interoperabilidade.

3. OBJETIVOS E CONTRIBUIÇÕES ESPERADAS

No grupo de pesquisa no qual este trabalho de mestrado é desenvolvido¹, está sendo concebido um *framework* para acesso, controle e gerência de objetos inteligentes em IoT. Um dos elementos chave do *framework* é o módulo de descoberta de OI. A ideia inicial do trabalho é, portanto, propor um mecanismo de descoberta que utilize técnicas de sensibilidade ao contexto para melhorar a interação entre usuário e sistema. Logo, a principal contribuição do trabalho é a seguinte:

- Utilizar o contexto do usuário e dos objetos inteligentes, em conjunto, para apresentar o conjunto de objetos inteligentes disponíveis para serem acessados e controlados. Dessa forma é possível que o mecanismo de descoberta priorize a interação do usuário com os dispositivos de interesse, melhorando a qualidade dessa interação e a experiência de uso do sistema.

O mecanismo de descoberta sensível ao contexto proposto deverá ser agregado ao *framework* geral para criação de aplicações em IoT. A versão final do trabalho deverá agregar ainda mais contribuições, como formas de aquisição e apresentação do interesse do sistema e a criação de um mecanismo de descoberta extensível e desacoplado ao CoAP. Com isso, o *framework* pretende auxiliar na criação de aplicações do tipo controle universal de IoT, onde os OI são encontrados e controlados de forma unificada e transparente por meio de tecnologias e protocolos baseados na Web.

4. ESTADO ATUAL DO TRABALHO

Para atender o requisito de descoberta contextual de recursos, apresentado em [8], o processo de descoberta de objetos inteligentes proposto é dividido em 8 etapas. (1) Aquisição do contexto do usuário; (2) Inferência dos objetos inteligentes de interesse; (3) Representação e identificação dos OI de interesse; (4) Criação de *strings* de consulta CoAP que representam o interesse; (5) Busca por OI que satisfaçam os filtros da consulta CoAP; (6) Aquisição da lista de objetos inteligentes presentes no ambiente e que são relevantes ao usuário; (7) Representação dos OI relevantes e disponíveis. (8) Apresentação dos OI ao usuário. Todo esse processo pode ser visualizado na Figura 2.

¹Grupo de Redes de Computadores, Engenharia de Software e Sistemas - GREat - www.great.ufc.br

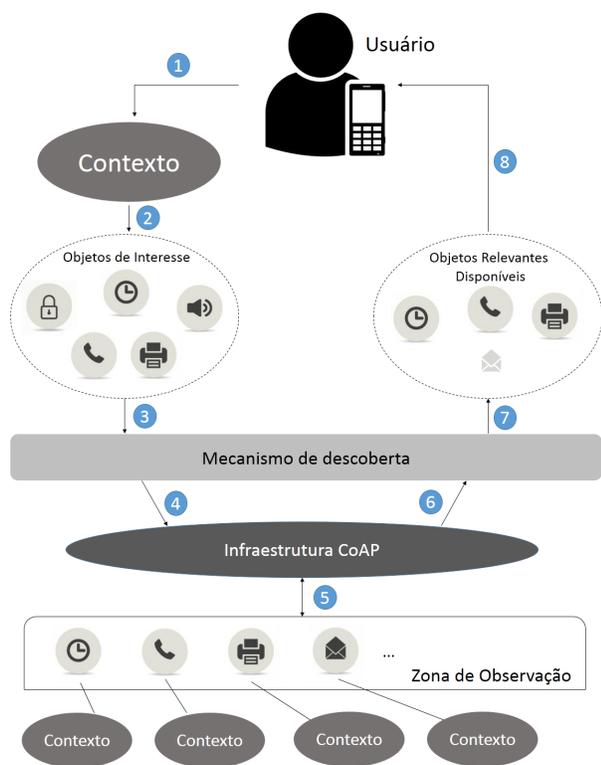


Figura 2: Visão geral do processo de descoberta proposto de objetos inteligentes.

Na primeira etapa (1), as informações que descrevem o contexto atual do usuário (e.g., localização, situação, atividade, etc.) são obtidas e utilizadas para a identificação de quais OI são de interesse do usuário em um dado momento (2). A Figura 2 trás um exemplo de interesse que inclui os seguintes recursos: Controle de acesso (controle das portas); Horário atual; Alarme sonoro; Dispositivo móvel; Impressora. Esses OI são representados através de uma única *String* que agrega informações contextuais e de identificação desse dispositivo (3). Como os OI utilizam o CoAP como protocolo de comunicação e interação, pode-se utilizar ferramentas já disponíveis nesse protocolo, como as *strings* de consulta CoAP, geradas na etapa (4). Nesse caso, um exemplo de consulta ao servidor CoAP seria: `GET /.well-known/core?rt="access-control time alarm mobile-device printer"`.

A referência aos OI que se registraram no servidor CoAP (5) e que atenderam às restrições expressas na consulta realizada é retornada ao mecanismo de descoberta (6). Antes de gerar a lista final de objetos relevantes e disponíveis (7), é preciso verificar as informações dinâmicas do contexto do usuário e dos OI, como localização (e.g., Usuário e OI estão na mesma sala da casa). Por fim essa lista final é apresentada ao usuário (8), que por sua vez poderá interagir com esses OI e realizar operações do cotidiano mais facilmente.

Embora este trabalho ainda esteja em andamento, o que faz com que várias etapas não estejam finalizadas ou possam sofrer grandes alterações. Pode-se dar outro exemplo, um pouco mais complexo, do uso do processo de descoberta proposto, como visto na Figura 3. Para isso será utilizado

parte do cenário motivador apresentado na Seção 1. Inicialmente o usuário, ao acordar, deseja interagir com a televisão para visualizar a previsão do tempo para o dia. Nesse caso, a primeira tarefa do sistema é obter essa informação através do módulo de aquisição de contexto. Esse módulo é responsável pela detecção da situação (usuário acordou) e fornecimento das informações contextuais suficientes para caracterizar seu interesse naquele momento. Essas informações são o histórico do usuário, utilizado para identificar a recorrência no uso da TV, e a localização *indoor*, usada para identificar onde o usuário costuma utilizar essa TV.

Com base nessas duas informações, o Módulo de Caracterização de Interesse constrói uma forma simples de representação. Uma lista de *Strings* contendo informações separadas por "." é criada, de forma hierárquica, para essa representação. No exemplo em questão, deve-se representar o interesse do usuário em acessar (*control*) a TV (*tv*), que está no ambiente (*ambient*) ao seu redor. Com isso forma-se a *String* `control.ambient.tv`. Com base na mesma lógica, precisa-se também representar o contexto (*context*) em que esta TV está inserida, que é o quarto (*quarto*) do usuário, logo criou-se a *String* `context.ambient.tv.quarto`.

Com essa representação do interesse, o Serviço de Descoberta pode realizar um mapeamento entre interesse e *String* de consulta CoAP. Cada campo das *Strings* de interesse é mapeado para campos que definem e descrevem os objetos inteligentes que seguem a especificação CoRE. Os campos *control* e *ambient* dizem que o objeto deve ser capaz de atuar no ambiente, o que pode ser representado por um filtro no campo *Interface-Description* do OI, limitando seu valor para apenas *actuator*. Já o campo *tv* da *String* de interesse, representa o tipo, ou classe, do objeto inteligente, e pode ser mapeado para o campo *Resource-Type*, aceitando apenas OI que possuam esse atributo igual a *tv*. Com isso é possível gerar a requisição para o Servidor CoAP em busca de todas as TVs que podem ser controladas pelo usuário.

O Servidor CoAP mantém o registro de todos os OI disponíveis no ambiente, isso inclui até os que não fazem parte do interesse do usuário naquele momento. Ao recebimento da requisição feita em busca das TVs, o servidor aplica os filtros sobre os campos *Resource-Type* e *Interface-Description* e retorna a lista de OI que atendem à requisição. Nesse exemplo são retornadas as duas referências para as TVs do quarto e da sala. Não é possível realizar uma consulta para identificar, ainda no Servidor CoAP, qual é a televisão do quarto especificamente. Isso porque a informação contextual do OI, que é dinâmica, não fica disponível no serviço de descoberta do Servidor CoAP. Para ter acesso a essas informações, clientes CoAP devem realizar uma operação de GET nesses OI e receber seu contexto no *Payload* da mensagem.

O Serviço de Descoberta, ao receber a lista com as duas TVs disponíveis, gera uma requisição GET para cada uma delas, e identifica qual a TV localizada no quarto com as informações recebidas no *Payload* da mensagem. A referência do OI correto, e relevante, ao usuário naquele momento é por fim repassada para o gerenciador da interface gráfica da aplicação, que é responsável por exibir na tela do *smartphone* o OI respectivo, além de permitir que o usuário o controle.

Toda a infraestrutura CoAP (IC), composta pelo servidor CoAP e os objetos inteligentes compatíveis com o protocolo, pode ser replicada dentro de um mesmo ambiente. É possível existir vários servidores CoAP que gerenciam recursos específicos, classificados com base na função ou na locali-

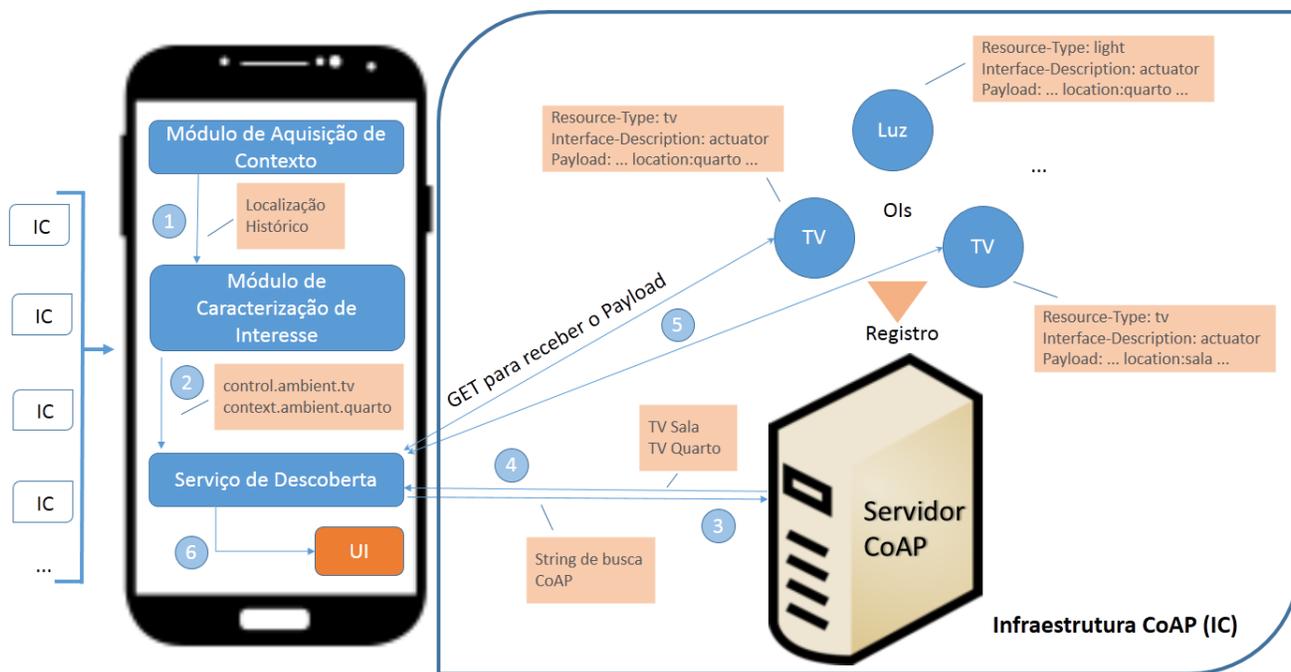


Figura 3: Exemplo de utilização do processo de descoberta proposto em um cenário real de IoT.

zação por exemplo. A Figura 3 mostra essa possibilidade do mecanismo de descoberta interagir com várias outras infraestruturas CoAP disponíveis, que, por sua vez, possuem elementos similares aos da IC expandida apresentada na figura.

As regras de mapeamento utilizadas nesse exemplo ainda não foram especificadas em sua completude, sendo implementadas apenas as necessárias para a criação desse cenário. Outro aspecto importante é a utilização do *Payload* para a descrição dos elementos contextuais dinâmicos do OI. O formato do *Payload* deve seguir alguma especificação, de modo a permitir a interpretação dos valores descritos nele, para isso é utilizado o atributo *Content-Type* do CoRE.

5. TRABALHOS RELACIONADOS

Nesta seção, serão discutidos trabalhos relacionados ao tema de descoberta e descrição de objetos inteligentes em um cenário de IoT. Foi dada preferência aos trabalhos que propõem algum tipo de serviço ou arquitetura de descoberta de dispositivos, bem como algum modelo de descrição com características específicas para IoT.

Liu propõe uma arquitetura de descoberta distribuída de dispositivos focada em Internet das Coisas. Cada dispositivo é considerado um nó de uma rede P2P e é capaz de realizar o processo de registro, bem como ajudar no serviço de descoberta geral [11]. Por trazerem uma abordagem totalmente distribuída, os autores eliminam os possíveis problemas causados pela mobilidade e volatilidade, intrínsecas em cenário de IoT, para um elemento centralizador. Por outro lado, como cada dispositivo deve ser capaz de realizar o registro, isso pode ser um outro problema para dispositivos estritamente limitados.

Digcovery [10] é uma proposta de descoberta global de objetos inteligentes, que se utiliza de uma infraestrutura centralizada onde dispositivos podem se registrar. Para o

acesso a essa infraestrutura, foi desenvolvido um serviço para dispositivos móveis, que permite a descoberta e o acesso aos objetos inteligentes. Características contextuais de geolocalização são levadas em consideração durante a fase de descoberta. Neste trabalho, tem-se o *smartphone* como ponto de acesso do usuário aos objetos inteligentes.

Cirani apresenta uma arquitetura auto-configurável e escalável de descoberta de serviços. No trabalho, é proposta uma topologia P2P com a utilização de diversos *gateways*. Esses são responsáveis por manter a lista de objetos inteligentes presentes na rede. Esses *gateways* são baseados nos servidores CoAP cujos clientes podem realizar uma operação GET para a URI `/.well-know/core` para receber a lista de dispositivos disponíveis [2].

DiscoWoT é um serviço de descoberta de objetos inteligentes proposto em [13], e que é possível definir estratégias de descoberta em tempo de execução por meio de interfaces RESTful². Esse serviço de descoberta se utiliza de *Microformats* e *Microdata* em conjunto com outras tecnologias WEB para representar semanticamente os objetos inteligentes.

Ishaq propõe um mecanismo de descoberta baseado em CoAP e DNS. Traduções entre CoAP e HTTP são disponibilizadas, permitindo a descoberta de qualquer objeto inteligente compatível com o padrão *IPv6*. Clientes se conectam ao serviço de descoberta através do endereço do servidor CoAP, e a interação com os dispositivos se dá através de interfaces RESTful [9].

A Tabela 1 apresenta um comparativo entre os trabalhos apresentados nesta seção. Analisando os resultados, é possível notar que a sensibilidade ao contexto ainda não é amplamente utilizada nas soluções de descoberta de serviços para IoT. Com relação aos trabalhos relacionados, apenas dois apresentaram elementos contextuais, mas nenhum combinou

²RESTful *web services* são aqueles que seguem o modelo arquitetural REST.

Trabalhos Relacionados	CoAP	Elementos Contextuais	Tipo de Contexto
Liu et al.	✓	-	-
Digcovery	-	✓	Usuário
Cirani et al.	✓	-	-
DiscoWoT	-	✓	Objeto Inteligente
Ishaq et al.	✓	-	-
Proposta	✓	✓	Usuário + OI

Tabela 1: Comparativo entre os trabalhos relacionados.

informações contextuais do usuário e do objeto inteligente em um único serviço. Nota-se também a tendência de utilização do CoAP como base para a descoberta, o que pode ser justificado tanto pela interoperabilidade quanto pelo fato desse protocolo já possuir um serviço de descoberta interno, que pode ser reutilizado. A proposta deste trabalho busca combinar a utilização de elementos contextuais e o protocolo CoAP, de modo a permitir realizar a descoberta usando informações contextuais tanto do usuário quanto do OI.

6. CONSIDERAÇÕES FINAIS

Este trabalho apresenta uma proposta de um mecanismo de descoberta de objetos inteligentes, sensível ao contexto, para o cenário de Internet das Coisas. A proposta utiliza como base o serviço de descoberta já oferecido pelo protocolo CoAP. Para garantir e satisfazer requisitos específicos para a descoberta de dispositivos em IoT, é proposta uma melhoria na obtenção do contexto final do sistema. O mecanismo proposto tenta oferecer ao usuário uma lista de dispositivos disponíveis que são relevantes, o que, de modo geral, melhora a interação entre usuário e sistema.

Como trabalho futuro, é possível citar a utilização desse mecanismo de descoberta sensível ao contexto para permitir uma maior economia de recursos. Para isso é preciso utilizar a informação sobre a lista de objetos relevantes disponíveis para, de fato, desabilitar ou reduzir o número de mensagens trocadas pelos objetos inteligentes que não façam parte dessa lista. Além disso, formas de representação e obtenção do contexto do usuário e dos OI é outro ponto que será abordado futuramente.

7. REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Comput. Netw.*, 54(15):2787–2805, Oct. 2010.
- [2] S. Cirani, L. Davoli, G. Ferrari, R. Léone, P. Medagliani, M. Picone, and L. Veltri. A scalable and self-configuring architecture for service discovery in the internet of things. *IEEE Internet of Things Journal*, 1(5):508–521, Oct 2014.
- [3] M. Crasso, A. Zunino, and M. Campo. Easy web service discovery: A query-by-example approach. *Science of Computer Programming*, 71(2):144 – 164, 2008.
- [4] E. Dave. The internet of things: how the next evolution of the internet is changing everything. *CISCO white paper*, 2011.
- [5] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.*, 16(2):97–166, Dec. 2001.
- [6] P. A. Duarte, F. A. Gomes, F. M. Barreto, W. Viana, and F. M. Trinta. Locomconfigurator: Uma ferramenta para modelagem visual de configurações de um middleware de suporte à aplicações conscientes de contexto. In *Proceedings of the 20st Brazilian Symposium on Multimedia and the Web, WebMedia '14*, 2014.
- [7] P. A. Duarte, L. F. M. Silva, F. A. Gomes, W. Viana, and F. M. Trinta. Dynamic deployment for context-aware multimedia environments. In *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web, WebMedia '15*, pages 197–204, New York, NY, USA, 2015. ACM.
- [8] 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. *IEEE Transactions on Services Computing*, 3(3):223–235, July 2010.
- [9] I. Ishaq, J. Hoebeke, J. Rossey, E. D. Poorter, I. Moerman, and P. Demeester. Facilitating sensor deployment, discovery and resource access using embedded web services. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 717–724, July 2012.
- [10] A. J. Jara, P. Lopez, D. Fernandez, J. F. Castillo, M. A. Zamora, and A. F. Skarmeta. Mobile digcovery: A global service discovery for the internet of things. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 1325–1330, March 2013.
- [11] M. Liu, T. Leppänen, E. Harjula, Z. Ou, M. Ylianttila, and T. Ojala. Distributed resource discovery in the machine-to-machine applications. In *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 411–412, Oct 2013.
- [12] M. E. F. Maia, A. Fonteles, B. Neto, R. Gadelha, W. Viana, and R. M. C. Andrade. Locom - loosely coupled context acquisition middleware. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, pages 534–541, New York, NY, USA, 2013. ACM.
- [13] S. Mayer and D. Guinard. An extensible discovery service for smart things. In *Proceedings of the Second International Workshop on Web of Things, WoT '11*, pages 7:1–7:6, New York, NY, USA, 2011. ACM.
- [14] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014.
- [15] W. Viana, A. D. Miron, B. Moisuc, J. Gensel, M. Villanova-Oliver, and H. Martin. Towards the semantic and context-aware management of mobile multimedia. *Multimedia Tools Appl.*, 53(2):391–429, June 2011.