

Busca de Serviços Baseada em Perfis Sociais dos Objetos em uma Rede SIoT Urbana

Iury Araújo¹, Mikaelly F. Pedrosa¹, Eudisley G. dos Anjos¹,
Fernando Matos¹

¹ Centro de Informática – Universidade Federal da Paraíba

iuryrogerio@ppgi.ci.ufpb.br, mikaelly.felicio@cc.ci.ufpb.br,
[eudisley, fernando]@ci.ufpb.br,

Abstract. *The service interaction provided by objects in IoT networks enables the creation of advanced services to answer application requests. However, the growing number of objects into the IoT network, besides its ad hoc structure, are disturbing some functionalities, such as service discovery. Therefore, when searching for services, the navigability is impaired because the system needs to sweep a great quantity of objects without a previous organization. Social Internet of Things (SIoT) emerged as an alternative to solve several problems faced by IoT through the concept of social networks. In SIoT each object has its own social profile, which contains its characteristics and information, and are organize by relationships. Thus, this research propose a solution for service discovery in a SIoT network. This solution uses the relationships between objects to improve the discovery scalability and considers their social profiles to meet more satisfactorily the requisitions. Simulated results demonstrates the solution performance to answer service requisitions in an urban SIoT network.*

Resumo. *A interação de serviços oferecidos por objetos em redes IoT possibilita a criação de serviços avançados para atender requisições de aplicações. Contudo, o crescimento no número de objetos na rede IoT, além da sua organização ad hoc, acaba dificultando algumas operações, como por exemplo a busca por serviços. Neste caso, ao buscar um serviço, a navegabilidade é prejudicada pois o sistema precisa varrer uma grande quantidade de objetos sem uma organização prévia. A Social Internet of Things (SIoT) surgiu como uma alternativa para solucionar diversos problemas enfrentados pela IoT por meio da utilização do conceito de rede sociais. Na SIoT cada objeto possui um perfil social, contendo suas características e informações, e são organizados através de relacionamentos. Desta forma, este trabalho propõe uma solução para a realização da busca de serviços em uma rede SIoT. Esta solução se utiliza dos relacionamentos entre os objetos para melhorar a escalabilidade da busca e considera os seus perfis sociais para atender de forma mais satisfatória as requisições realizadas. Resultados simulados mostraram o desempenho da solução ao atender requisições por serviços em uma rede SIoT.*

1. Introdução

As áreas urbanas estão cada vez mais populosas o que acarreta no surgimento de diversos problemas relacionados a qualidade de vida, como aumento da poluição, escassez

de recursos, acesso limitado aos serviços públicos, problemas de mobilidade urbana e segurança [Neves et al. 2017]. Dessa forma, existe uma grande procura por soluções inteligentes que, por meio da tecnologia, possam resolver ou diminuir os problemas gerados pela expansão urbana. Muitas dessas soluções são desenvolvidas a partir do conceito de ambientes inteligentes, que pode ser definido como um “pequeno mundo” onde diferentes tipos de objetos inteligentes trabalham para tornar a vida dos seus habitantes mais confortável [Vodanovich et al. 2010]. Dentro desse conceito podemos encontrar diversas áreas de pesquisa como *smart cities*, *smart healthcare* e *smart grid*, que apesar das suas diferenças têm em comum a atuação de diversos objetos inteligentes trabalhando em conjunto para realizar diversas tarefas. Cada vez mais esses objetos adquirem a capacidade de se conectar a Internet, seja para executar ações, adquirir e transmitir dados ou realizar processamento. Essa capacidade facilita a colaboração entre objetos, fornecendo assim uma variedade maior de funcionalidades para as aplicações [Santos et al. 2016].

O cenário descrito é possível através da implementação do paradigma da *Internet of Things* (IoT). A IoT é descrita como a presença pervasiva ao nosso redor de “coisas” ou objetos, que através de esquemas de endereçamento único, estão possibilitados a interagir entre si e cooperar para alcançar um objetivo específico [Atzori et al. 2010]. Esses objetos são os dispositivos tecnológicos do nosso cotidiano, como computadores, *tablets*, celulares, nós sensores, entre outros, que conseguem se conectar a internet. Contudo, o crescente número de objetos conectados é uma preocupação para a IoT. Segundo [Evans 2011], o número de objeto conectados já é maior que a população mundial e deve chegar a 50 bilhões de objetos até 2020. Esse crescimento atinge diretamente o gerenciamento da rede quanto a sua escalabilidade, navegabilidade e a heterogeneidade dos objetos, dificultando a execução de tarefas importantes do paradigma [Jarwar et al. 2018].

O gerenciamento de serviços também é prejudicado com o aumento do número de objetos. Como um objeto pode fornecer um ou mais serviços, o total de serviços na rede pode aumentar vertiginosamente. Tais serviços são abstrações das funcionalidades oferecidas pelos objetos, sem se preocupar com tecnologias ou protocolos. Componentes da arquitetura IoT que lidam com serviços, como os de busca e composição, sofrem diretamente com estes problemas. Como forma de solucioná-los foi proposta a utilização do conceito de redes sociais para organizar os objetos e os seus serviços, surgindo assim um novo paradigma chamado de *Social Internet of Things* (SIoT) [Atzori et al. 2012].

A SIoT introduz a utilização de uma rede social para a organização de objetos, onde cada objeto tem potencial para formar “amizade” com outros objetos através de relacionamentos [Atzori et al. 2011]. A partir da rede social gerada é possível gerenciar a escalabilidade e navegabilidade da rede, a heterogeneidade de objetos e a transmissão de dados, como já é realizada pelas redes sociais humanas [Atzori et al. 2014]. Apesar da SIoT herdar funcionalidades da IoT, muitas delas precisam ser adaptadas para lidar com este novo paradigma, como por exemplo a busca de serviços, que é uma das etapas fundamentais para atender aplicações através de serviços compostos para realizar atividades mais complexas. Outro desafio é utilizar os perfis sociais dos objetos como informação útil para realizar funcionalidades na rede SIoT.

Neste trabalho é proposta uma busca de serviços em uma rede SIoT baseada nos perfis sociais (Busca por Perfil Social - BPS). Nesta busca são utilizados os relacionamentos da rede social para melhorar a sua navegabilidade e escalabilidade, levando também

em consideração as características e informações contidas nos perfis sociais dos objetos para atender de forma mais satisfatória as requisições por serviços. Para verificar a eficiência da solução, foram realizados testes com dados de uma rede SIoT Urbana, onde são comparados os resultados entre o BPS e soluções convencionais da literatura.

O restante deste artigo está organizado da seguinte forma: Na Seção 2 é apresentada uma visão geral das pesquisas realizadas na área de SIoT e suas abordagens para busca de serviços. A Seção 3 expõe os conceitos de SIoT e de sua busca de serviços. Na Seção 4 é apresentada a proposta de busca por perfis sociais. A Seção 5 discorre sobre a implementação da proposta, descreve os testes realizados e apresenta e discute os resultados alcançados. Por último, a Seção 6 conclui o artigo e discute trabalhos futuros.

2. Trabalhos Relacionados

Por ser uma área relativamente recente, a maioria dos trabalhos em SIoT focam em questões conceituais do paradigma, como por exemplo, definições de SIoT, áreas de aplicação, infraestruturas necessárias e propostas de arquiteturas. Poucos trabalhos apresentam implementações do paradigma. Ainda sim, alguns trabalhos focam na capacidade social dos objetos, utilizando suas características, informações e o potencial das interações para fornecer soluções para o gerenciamento de objetos e serviços. Os trabalhos apresentados também se preocupam com as funcionalidades da rede SIoT no que concerne à busca de serviços de forma explícita ou implícita.

O SIoTcampus proposto por [Alves et al. 2015] exemplifica a utilização da SIoT para o disseminamento de informações acadêmicas entre usuários de um campus universitário. Este trabalho propõe uma rede exclusiva para objetos inteligentes em que informações de contexto, como localização, agenda, usuário, etc., são utilizadas como critérios para estabelecer relacionamentos entre os objetos. Como a rede social tem relacionamentos dinâmicos, que podem ser modificados através da aquisição de informações do dono do objeto, é possível propagar pela rede informações, eventos e notícias para um número maior de interessados por aquele contexto.

Uma das possibilidades propostas pela SIoT é utilizar a rede social de objetos como ferramenta para determinar a confiabilidade de objetos na rede. A pesquisa realizada em [Brittes et al. 2017] procura construir uma rede dentro de um ambiente médico para que os objetos possam transmitir e receber dados do sistema hospitalar de forma segura, identificando o nível de confiabilidade dos objetos e bloqueando ataques. A confiabilidade é identificada através da criação de um protocolo de gerenciamento de confiabilidade, que utiliza os relacionamentos e informações dos objetos para determinar o nível de confiabilidade dos mesmos. Em [Wang et al. 2016] também é proposto um modelo que verifica a confiabilidade de objetos na SIoT, sendo baseado no trabalho coletivo de objetos que fornecem operações computacionais e de armazenamento. Neste modelo, tarefas são enviadas aos objetos e ao atendê-las, uma recompensa é entregue aos objetos, ajudando assim a criar relacionamentos e estabelecer confiabilidade entre objetos que cooperam com frequência entre si.

A pesquisa desenvolvida em [Nitti et al. 2016] apresenta uma solução para a implementação da busca de serviços da arquitetura SIoT. Utilizando os relacionamentos construídos pelos objetos da rede social é possível criar um método de busca que com um pequeno número de saltos em uma rede implementada como grafo, encontra objetos

que fornecem os serviços requeridos. Isso é realizado utilizando duas propriedades: a primeira é intrínseca a rede social, baseando-se nas amizades dos objetos; a segunda é externa a rede social, baseando-se na similaridade entre o objeto e a requisição.

A pesquisa em [Chen et al. 2015] propõe a criação de um método para a composição de serviços em SIoT utilizando os relacionamentos formados pelos objetos na rede social. O método encapsula objetos heterogêneos em *web services*. Utilizando o estilo RESTful, as aplicações podem requerer acesso de forma homogênea aos objetos, permitindo a colaboração entre eles para compor processos mais robustos. A composição é possível por meio da modelagem dos relacionamentos entre serviços pela rede social, classificando os relacionamentos em três dimensões: localização, tipo e correlação. Ao se requerer um processo complexo, que precisa ser composto por diversos serviços, o componente realiza uma busca em toda rede social através dos três relacionamentos para encontrar os serviços candidatos, esses candidatos depois são avaliados e os melhores são compostos no processo requerido. O trabalho propõe também um algoritmo de busca que deve ser utilizado para encontrar os serviços utilizando o método de busca em largura em grafos.

Por meio dos trabalhos expostos nessa seção é possível perceber que existem pesquisas preocupadas em endereçar a capacidade social dos objetos, utilizando suas características, informações e o potencial das interações para fornecer soluções para aplicações SIoT, sobretudo dentro dos *smart environments*. Da mesma forma existe uma preocupação crescente em criar funcionalidades da SIoT que se adéquem a utilização da rede social, principalmente para a busca de serviços. Portanto este trabalho apresenta a proposta de um método para a busca de serviços em uma rede SIoT baseado na utilização das informações e características dos perfis sociais dos objetos para melhorar sua eficiência. Diferindo-se de outras soluções por restringir a necessidade de se varrer toda a rede para encontrar os serviços adequados requeridos.

3. Busca de Serviços em uma rede SIoT

A *Social Internet of Things* (SIoT) é um paradigma derivado da IoT que propõe a utilização de redes sociais em conjunto com a infraestrutura já bem estabelecida da IoT. A utilização do conceito de redes sociais em IoT pode levar a grandes vantagens, tais como: (i) a estrutura SIoT pode ser moldada para garantir a navegabilidade da rede, permitindo a eficiência da descoberta de serviços e objetos; (ii) a confiabilidade de objetos podem ser determinadas através da análise da interação com objetos “amigos” e;(iii) modelos utilizados para estudar redes sociais podem ser facilmente aplicados para tratar problemas relacionados a IoT [Atzori et al. 2012]. O conceito formalizado de SIoT determina-o como uma rede social onde cada nó é um objeto inteligente capaz de estabelecer relacionamentos, de forma autônoma, com outros objetos, seguindo regras estabelecidas pelos seus donos [Girau et al. 2017].

Um relacionamento em SIoT é a representação de um vínculo que os objetos podem ter no mundo real, sendo essa representação utilizada para a construção da rede social através da formação de objetos “amigos” que são ligados por relacionamentos. Os trabalhos em SIoT estabelecem cinco tipos de relacionamentos que podem existir entre os objetos [Atzori et al. 2012, Girau et al. 2017, Atzori et al. 2011]: (i) *Parental*: estabelecido entre objetos que pertencem ao mesmo lote de produção, geralmente são objetos

homogêneos, originados no mesmo período e por um mesmo fabricante; (ii) *Co-location*: estabelecido entre objetos que são usados sempre no mesmo local, no caso de sensores, atuadores e objetos de realidade aumentada são considerados os objetos no mesmo ambiente como uma casa ou cidade inteligente; (iii) *Co-work*: estabelecido sempre que objetos cooperam para prover determinada funcionalidade; (iv) *Ownership*: estabelecido entre objetos heterogêneos ou homogêneos que pertencem ao mesmo usuário; e (v) *Social*: estabelecido quando objetos entram em contato, contínuo ou esporádico, porque seus donos entraram em contato durante algum período de suas vidas.

Os objetos SIoT são organizados por meio da rede social de objetos, utilizada em diversos processos do paradigma, como a composição e busca de serviços. A rede social é composta por representações da existência física dos objetos na rede, chamados de perfis sociais. Cada perfil, contém atributos relacionados ao objeto que representa, como sua identificação, os amigos formados pelos seus relacionamentos com outros objetos, os serviços que oferece e suas características e informações [Jadhav and Patil 2016]. As características de um objeto são os dados inerentes a sua constituição e que são geralmente imutáveis, por exemplo seu fabricante, lote de produção, quantidade de núcleos de processamento e de memória primária e secundária, entre outros. Enquanto, que as informações de um objeto são dados sobre o objeto que estão em constante mudança, como capacidade de armazenamento, estado da bateria, confiabilidade do objeto, configurações de segurança, entre outros [Kim et al. 2017].

A busca de serviços geralmente utilizada na literatura, como é o caso de Chen et al. [Chen et al. 2015], utiliza apenas os relacionamentos para procurar na rede por objetos que tenham serviços que atendam às requisições. Devido à peculiaridade de tais relacionamentos, os resultados da busca podem ser diferentes para os mesmos objetos. A Figura 1 apresenta um exemplo de como diferentes relacionamentos afetam a busca em redes SIoT. Cada perfil social é representado por um nó e cada aresta identifica uma relação de amizade entre objetos. São utilizados como exemplos os relacionamentos *Ownership* e *Co-work* (Figuras 1(a) e 1(b), respectivamente). Quando uma requisição é enviada para o componente de busca, ele identifica o objeto que requisitou o serviço (requerente) e o torna o objeto inicial da busca. Isso é possível pois qualquer requerente é um objeto que faz parte da rede social. Em ambas figuras, o requerente é o objeto *D*. A partir do requerente, o método de busca procura saber se os serviços que ele oferece atendem a requisição. Cada cor representa o conjunto de objetos acessados a cada nível de busca. Na Figura 1(a), a partir de *D*, os seus amigos, que são os objetos *B*, *C*, *E* e *G*, são pesquisados. A busca prossegue nos amigos dos objetos anteriores (*A*, *F*, *H*, *I*, *L* e *O*), seguindo assim até a busca ter percorrido todos os objetos na rede. Da mesma forma acontece no grafo da Figura 1(b), onde a busca é realizada utilizando um relacionamento diferente. Neste caso, é possível visualizar que o resultado será diferente, já que não há o relacionamento *Co-work* entre o grupo de objetos *N*, *M* e *I* e o restante da rede.

Em ambos os casos, a busca de serviços é realizada por uma estratégia de busca em largura em grafos, que verifica todos os objetos da rede social para encontrar os serviços que atendem melhor a requisição. Esta abordagem retorna todos os serviços que podem atender à requisição, incluindo diversos serviços que realizam as mesmas atividades ou não atendem requisitos de outras funcionalidades da rede SIoT. Por isso, uma etapa adicional para filtrar os serviços é necessária. Tal etapa pode comprometer o desempenho da

aplicação que realiza a requisição.

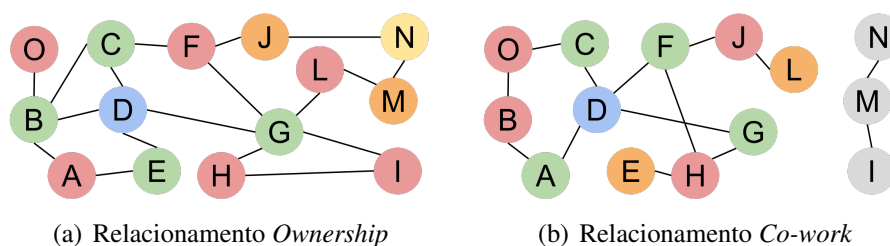


Figura 1. Grafos representando relacionamentos da rede SIoT.

4. Proposta

O método de busca BPS proposto neste trabalho utiliza as características e informações contidas nos perfis sociais dos objetos como critérios adicionais de comparação para verificar se os serviços de um objeto, e ele próprio, atendem aos requisitos da requisição. O BPS também introduz uma nova condição de parada para busca em largura, uma vez que todos os serviços requisitados sejam encontrados, independentemente da quantidade de objetos visitados, a busca é interrompida. Além disso, como na estratégia de busca em largura, o BPS também finaliza sua execução quando todos os objetos da rede já foram pesquisados. Desse modo, o método garante que uma resposta satisfatória para a requisição possa ser encontrada em menos tempo, balanceando a qualidade dos serviços encontrados e o tempo de busca, porém deixando aberta a possibilidade do pior cenário, que acontece quando não é possível encontrar na rede todos os serviços para atender tal requisição.

A Figura 2 apresenta a mensagem de requisição que é recebida pelo BPS, a qual é formada por três atributos: (i) Objeto Requerente (OR), contendo a identificação do objeto que requisitou a busca; (ii) uma Lista de Serviços (LS), contendo os serviços requeridos e que serão buscados pelo BPS; e (iii) a Lista de Requisitos (LR), contendo requisitos que serão verificados e deverão ser atendidos pelas características e informações dos objetos, como por exemplo, tipo de tecnologia de transmissão, quantidade de memória, confiabilidade do objeto, dentre outros.

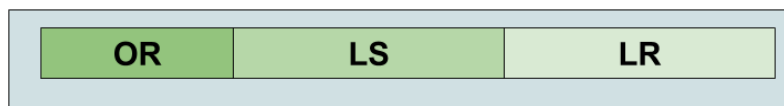


Figura 2. Requisição de busca de serviços e seus atributos.

A Figura 3 ilustra o processo BPS em uma rede com sete objetos que recebe uma requisição por busca de serviços. Como todo objeto que requer uma busca de serviços também faz parte da rede social de objetos, o BPS utiliza o perfil social do requerente na rede como o objeto inicial da busca. Com os outros atributos da requisição o BPS começa a verificar se os objetos amigos de *A* possuem os serviços requisitados (Figura 3(b)). O objeto inicial *A* é marcado como não usável, pois, ou não possui os serviços buscados ou não atende todos os requisitos impostos, mesmo que, tenha o serviço requerido. Ao verificar os objetos *B* e *D*, BPS percebe que os dois tem serviços requeridos e que eles

atendem todos os requisitos, portanto suas identificações são adicionadas a uma lista com o possível resultado. Como não foram encontrados todos os serviços requeridos a busca é expandida para os objetos *F* e *C* (Figura 3(c)), que possuindo os serviços que faltavam e atendendo aos requisitos, completam a execução da busca.

Como exemplo podemos considerar um objeto que pertence a uma rede SIoT. Nessa rede, os objetos possuem como uma das suas características, os tipos de conexões sem fio que podem realizar, como bluetooth, 3G, GSM e Wi-Fi, sendo permitido possuir qualquer combinação dessas quatro conexões. Esse objeto possui somente a capacidade de conectar-se por Wi-Fi, enquanto que a requisição de busca pede por serviços de objetos que podem se comunicar via Wi-Fi e 3G, mesmo que esse objeto tenha um serviço que atenda a requisição ele não será considerado, pois não atende aos requisitos mínimos da busca. Esse tipo de abordagem facilita a manipulação dos resultados da busca por outras funcionalidades da rede SIoT, como a composição de serviços, pois garante que os serviços encontrados não precisam passar por uma nova avaliação de checagem de requisitos.

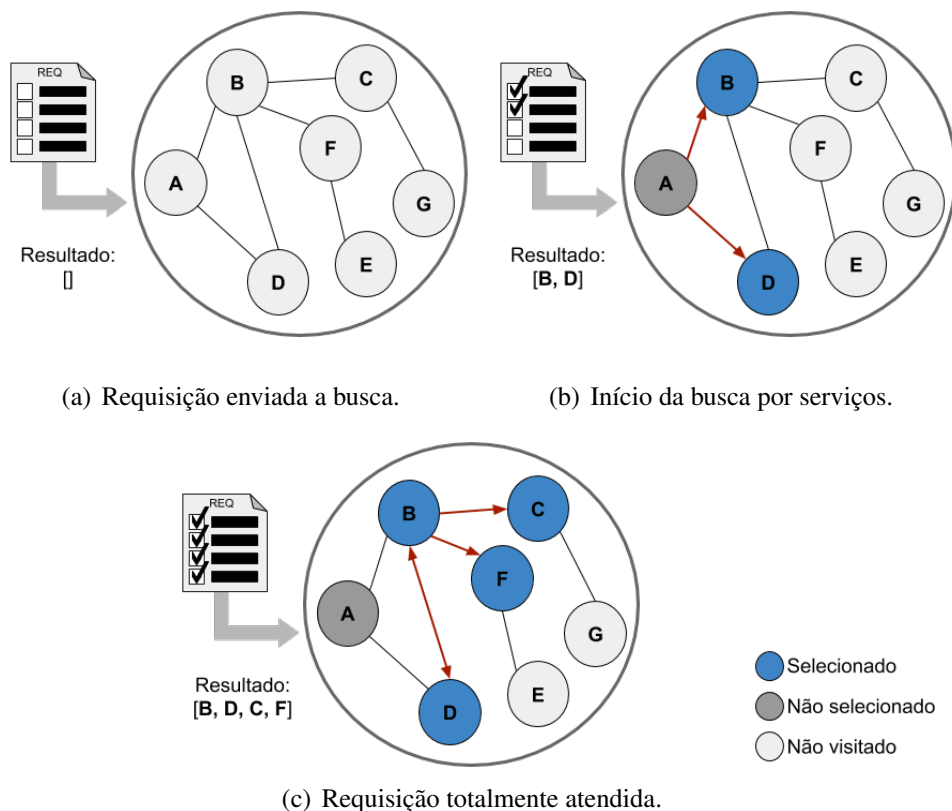


Figura 3. Funcionamento do método BPS.

O Algoritmo 1 descreve o método BPS para busca de serviços considerando os perfis dos objetos. A requisição é tratada como três blocos separados, (i) lista serviços requisitados (*required_service_list*) que representa a lista contendo os serviços devem ser encontrados; (ii) Lista de requisitos (*requirement_list*) que contém todos os requisitos mínimos que os objetos devem possuir; e (iii) Objeto requerente (*requester*) que será utilizado como ponto inicial da busca na rede SIoT. Além da requisição o algoritmo também

recebe como entrada a rede social *social_network* que contém todos os perfis sociais dos objetos na rede

Algoritmo 1 Busca de serviços Baseada em Perfis Sociais (BPS)

Entrada: *required_service_list*, *requirement_list*, *requester*, *social_network*

Saída: *service_list*

Require: *requester* deve ser um objeto pertencente à rede social

```
1: search_list.adiciona(social_network.recupera(requester))
2: while search_list não é vazia do
3:   head ← search_list.removeHead()
4:   if required_service_list foi atendida then
5:     break
6:   else
7:     if head.visualizado = true then
8:       continue
9:     else if head atende required_service_list and head atende requirement_list
10:      then
11:        service_list.adiciona(head.services())
12:      end if
13:    end if
14:    head.visualizado ← true
15:    search_list.adiciona(social_network.recuperaAmigos(head))
16: end while
17: return service_list
```

Inicialmente, o BPS adiciona o perfil social do objeto requerente a *search_list* e em seguida inicia o laço buscando serviços por toda a rede até que a *required_service_list* tenha sido completamente atendida ou *search_list* esteja vazia. Para cada objeto retirado da lista (*head*) é verificado se o perfil social já foi pesquisado (linha 7). Em caso positivo, verifica-se o próximo objeto. Caso contrário, é verificado se um ou mais serviços da *head* correspondem aos serviços requeridos em *required_service_list* e se aquele objeto atende os requisitos mínimos dispostos na *requirement_list* (linha 9). Os serviços que atenderem ambas as condições são adicionados à *service_list* (linha 10). O BPS marca então o perfil social contido em *head* como já visualizado e adiciona os seus amigos à *search_list* (linha 14), iniciando novamente o processo. Quando o algoritmo termina sua execução por uma das duas condições de parada os serviços encontrados são entregues como resposta da requisição. O resultado final do BPS permite três estados diferentes, (i) não encontrar nenhum serviço que atenda à requisição, (ii) encontrar somente alguns serviços, atendendo então parcialmente a requisição, ou (iii) encontrar todos os serviços de forma a atender por completa à requisição.

5. Ambiente de teste e avaliação

O método BPS foi implementado em Python e testado utilizando o *dataset* de uma rede SIoT urbana disponibilizado por [Atzori et al. 2016], o qual foi criado no contexto do projeto SmartSantander, executado nas cidades de Belgrade, Guildford, Lübeck and Santander. Porém, o único *dataset* disponível é o da cidade de Santander.

O *dataset* é composto por 16216 dispositivos (objetos da rede social) de diversas tecnologias. Ele fornece informações sobre os relacionamentos da rede social e os serviços que cada objeto oferece. Os serviços são representados por números que indicam algum serviço que o objeto pode prover, sendo que cada objeto pode possuir até oito tipos de serviços diferentes de um total de dezesseis distintos. Como o *dataset* não trabalha com os serviços sendo entidades únicas, com sua própria identificação, o BSP foi implementado com a alteração de que ao invés de retornar os serviços encontrados, são retornados os objetos que os provem. O *dataset* apresenta os relacionamentos dos objetos através de matrizes com valores *booleanos* que representam a existência, ou não, de uma “amizade” entre os objetos. O relacionamento *ownership* foi utilizado para realizar os testes deste trabalho. A Figura 4 ilustra a organização de uma matriz de relacionamento onde o número um identifica uma amizade entre dois objetos. Por exemplo os objetos (A,C) e (E,F) são amigos pois a interseção das suas colunas e linhas possuem o valor um.

		Objetos					
		A	B	C	D	E	F
Objetos	A	0	0	1	0	0	1
	B	0	0	0	1	0	0
	C	1	0	0	0	1	0
	D	0	1	0	0	0	0
	E	0	0	1	0	0	1
	F	1	0	0	0	1	0

Figura 4. Matriz de relacionamento do *dataset*.

Para atestar a eficiência do BPS, foram utilizadas duas métricas: tempo de busca e escalabilidade da solução. O tempo de busca é o tempo que o algoritmo leva para responder a requisição, sendo calculado a partir do momento que o algoritmo começa a consultar a rede social para encontrar serviços até o momento em que a lista de objetos encontrados seja retornada. Já a escalabilidade avalia o comportamento da solução à medida que aumenta o número de objetos na rede. Os testes foram realizados em uma máquina com processador i7-8750H hexa-core de 2666 MHz e 16GB de RAM DDR4.

O BPS foi comparado com o algoritmo proposto por Chen et al. [Chen et al. 2015], que utiliza o método de busca em largura. Para testar o tempo de busca, um arquivo foi criado contendo todas as requisições que foram realizadas durante os testes. A utilização do arquivo garante que as requisições que foram realizadas no método BPS eram idênticas às realizadas no algoritmo de Chen. Cada linha do arquivo representa uma requisição, contendo as seguintes informações: o objeto requerente, os serviços que se desejam encontrar e os requisitos que serão utilizados. Esta última informação é utilizada apenas pelo método BPS.

Cada atributo da requisição foi gerado aleatoriamente obedecendo determinados valores mínimos e máximos. O objeto requerente é obrigatoriamente um objeto pertencente a rede. Os serviços desejados são gerados a partir de uma lista de dezesseis serviços

possíveis do *dataset*. Dentre estes, são escolhidos quatro para formar os serviços desejados de cada requisição, de forma que um serviço já escolhido não possa ser escolhido novamente. Para gerar os requisitos, foi necessário primeiro criar os perfis sociais para os objetos do *dataset*, definindo quais características e informações seriam associadas ao perfil e tornariam-se seus atributos, esses, podem ser observados na Tabela 1. O próximo passo foi preencher os atributos contidos nos perfis sociais, seguindo os valores permitidos na Tabela 1. Finalmente, os requisitos foram gerados e adicionados à requisição, sendo esses baseados nos atributos dos perfis sociais. Valores iniciais aleatórios foram utilizados para cada requisito, respeitando determinados limites impostos na Tabela 1. No total, 2380 requisições foram criadas para os testes.

Tabela 1. Atributos dos perfis sociais e requisitos da requisição.

Atributos	Valor Mínimo	Valor Máximo
Memória primária	256 MB	4 GB
Memória secundária	0 MB	512000 MB
Núcleos de processamento	1	8
Conectividade	1	5
Capacidade de armazenamento	15%	75%
Estado da bateria	10%	100%
Confiabilidade do objeto	45	95

O teste de escalabilidade deve verificar como se comporta a solução ao aumentar o número de objetos da rede. Para realizá-lo, foi necessário dividir a rede em diversos recortes que representam redes de tamanhos variados, sendo elas compostas de quinhentos, mil, dois mil, quatro mil, doze mil e dezesseis mil objetos. Esses recortes precisam obedecer a distribuição de probabilidade dos tipos de objetos presentes no *dataset* inicial. A Tabela 2 apresenta as porcentagens de cada tipo de dispositivo.

Tabela 2. Distribuição dos tipos de dispositivos no *dataset*.

Tipo de Dispositivo	Porcentagem
<i>Smartphone</i>	22.45%
Computador Pessoal	20.72%
Carro	13.57%
Impressora	13.07%
Tablet	9.87%
<i>Smart Fitness</i>	5.43%
Estacionamentos	4.17%
Sensores Residenciais	3.70%
Iluminação Pública	3.12%
<i>Smartwatch</i>	1.23%
Transporte	0.88%
Sensores de Ambiente ou de clima	0.86%
Pontos de Interesse	0.58%
Alarmes de Segurança	0.23%
Indicador	0.06%
Caminhão de Lixo	0.04%

Uma vez garantido que os recortes tinham a mesma distribuição do conjunto de dados original, foi necessário lidar com outro problema causado pelos recortes. A aleatoriedade de escolha dos objetos dificulta que objetos que tenham relacionamentos entre

si sejam escolhidos juntos, principalmente para recortes de tamanhos pequenos. Esta situação prejudica a navegabilidade e dificulta a execução da busca, pois é possível que existam diversos objetos ou grupos de objetos isolados na rede. Como solução, foi necessário acrescentar manualmente novos relacionamentos, de forma que todos os perfis sociais tenham pelo menos um relacionamento com outro objeto do mesmo recorte, dessa forma garantindo a navegabilidade da rede.

5.1. Resultados

Foram realizadas dez execuções para cada recorte da rede utilizando cada um dos algoritmos BPS e Chen. Para cada teste a mediana do tempo de busca das 2380 requisições foi calculada, a mediana foi utilizada como forma de evitar que *outliers* impactassem os resultados. Em seguida foi calculada a média de todas as medianas dos testes realizados para cada recorte. A Tabela 3 apresenta a mediana do tempo de busca, em segundos, de cada recorte. Analisando os valores, é possível perceber que o algoritmo BPS consegue responder as requisições de todos os recortes em um tempo expressivamente mais baixo que o algoritmo de Chen. Este comportamento pode ser visualizado principalmente nos recortes 4000 e 8000, onde o tempo de busca do BPS é menor 93,98% e 94,32% do que Chen, respectivamente.

Tabela 3. Mediana do tempo de busca, em segundos, para cada requisição nos recortes de teste.

Recorte (nº de objetos)	Chen (s)	BPS (s)
500	0,02594487	0,01072197
1000	0,10345836	0,01405903
2000	0,38549267	0,03829563
4000	1,37108796	0,08257948
8000	4,28259313	0,2434388
12000	0,56687746	0,00898465
16000	0,00587981	0,00399698

Algumas surpresas foram observadas durante a comparação dos tempos de busca entre os algoritmos, sendo essa comparação apresentada na Figura 5. Entre os recortes de 500 a 8000 é possível perceber que existe um crescimento no tempo médio necessário para que uma requisição seja respondida, sendo esse comportamento já esperado com o aumento de objetos na rede social. Entretanto, nos recortes 12000 e 16000 temos um efeito totalmente reverso com uma diminuição busca do tempo de brusca. Por exemplo, no recorte de 12000, o algoritmo de Chen tem uma queda de tempo na razão de 86,76%, enquanto que o BPS tem uma redução de tempo de 96,30%.

Uma das possibilidades para esse comportamento é que a adição de novos objetos criou uma rede muito mais conectada, tornando-se assim mais fácil para os algoritmos alcançarem objetos que anteriormente tinham poucos relacionamentos e estavam mais isolados. Outra possibilidade é que a aleatoriedade usada para criar relacionamentos e para escolher objetos da rede tenham tornado o seu funcionamento mais complexo. Um grande indício seria o fato de que a maior parte das alterações realizadas na rede foram realizadas no recortes menores, de 500 a 8000, enquanto que poucas alterações foram efetuadas em 12000 e ainda menos no recorte 16000. Sendo este o caso, esses resultados podem ser um claro testemunho da eficiência das soluções SIoT, pois quanto menor for a alteração na rede, provida por [Atzori et al. 2016], melhor é a sua eficiência.

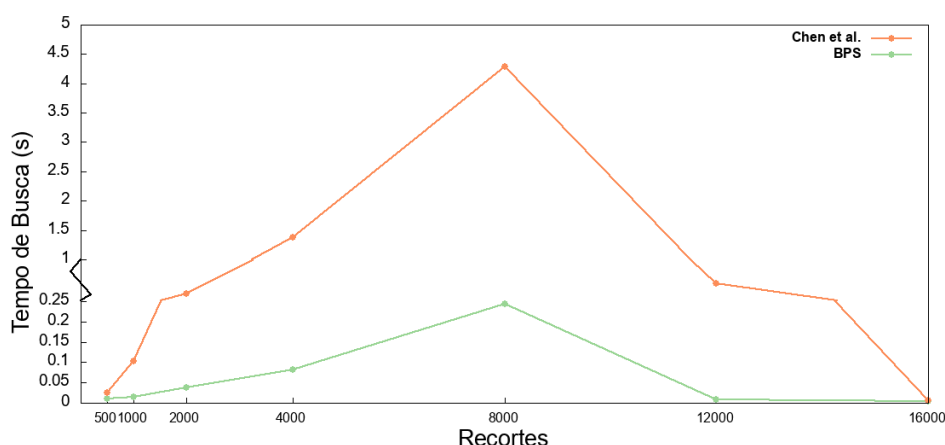


Figura 5. Gráfico comparativo do comportamento do tempo de busca nos algoritmos Chen et al. e BPS.

Ao comparar a execução dos algoritmos por meio da análise da Figura 5 é fácil perceber como o BPS consegue manter o tempo de busca muito mais baixo que o algoritmo Chen et al., sendo geralmente 94,03% mais rápido e não demorando mais que, em média, 0,25 segundos para responder a uma requisição. Outro aspecto importante a ser observado e analisado é o comportamento dos algoritmos em relação a escalabilidade da rede. Obviamente 16000 objetos ainda é um número bem pequeno para testar a escalabilidade desses algoritmos em uma rede que hoje já comporta bilhões de objetos. Entretanto, indícios que o algoritmo BPS é escalável podem ser observados através do crescimento controlado do tempo de busca em relação a expansão do número de objetos na rede e também da queda expressiva dele em recortes maiores.

Em contraste ao BPS, Chen et al. tem um crescimento de tempo de busca bem acentuado até o recorte 8000, onde logo em seguida, cai para valores bem próximos aos do BPS. Apesar de existir uma sugestão que o Chen et al. não é escalável, os resultados observados mostram que é necessário verificar o comportamento desse algoritmo com um número ainda maior de objetos por recorte para ter um veredito final. Entretanto, é possível afirmar que a solução proposta neste trabalho por meio do algoritmo BPS mostrou-se mais escalável que o método proposto por Chen et al. [Chen et al. 2015]. Outra informação fundamental indicada por meio dos testes apresentados na Figura 5 é a necessidade de verificar como a organização dos perfis sociais e seus relacionamentos afetam os algoritmos de busca em uma rede SIoT.

A Tabela 4 apresenta a média de objetos retornados na busca em cada recorte. É possível observar que o número de objetos retornados por Chen é muito maior que o do BPS (em média 99,09% mais objetos). Apesar do método de Chen poder encontrar sempre os melhores serviços para cada requisição, ainda é necessária uma etapa adicional para escolher quais objetos, dentre os encontrados, serão utilizados para satisfazer a requisição, o que pode impactar no desempenho da aplicação que está realizando tal requisição. Já no BPS não é necessária esta etapa adicional, uma vez que o método retorna, assim que encontra, os objetos necessários para atender, por completo, a requisição. Dessa forma, aliado a um tempo de busca mais rápido, o BPS mostra que consegue atender mais satisfatoriamente as requisições de busca de serviços por levar em consideração o perfil social

dos objetos da rede.

Tabela 4. Média do número de objetos retornados por requisição.

Recorte (nº de objetos)	Chen	BPS
500	344	3
1000	679	5
2000	1340	8
4000	2459	17
8000	3502	37
12000	3789	42
16000	4102	39

6. Conclusão

Este trabalho propôs uma solução para a busca de serviços em uma rede SIoT mais eficiente, utilizando as características e informações contidas no perfil social do objeto para atender mais satisfatoriamente as requisições de busca. Para isso um método de busca de serviços, chamado de BPS, foi proposto a partir da modificação de um algoritmo de busca em largura em grafo. O BPS adiciona uma nova condição de parada para melhorar a eficiência da busca e insere a utilização das características e informações dos perfis sociais dos objetos para verificar a qualidade dos serviços encontrados.

Testes simulados utilizando o *dataset* de uma rede SIoT urbana constataram que o algoritmo BPS realiza uma busca de serviços, 94,03% em média, mais rápida que método proposto na literatura. Analogamente, os resultados desses testes também indicam uma boa adaptabilidade do BPS para lidar não só com o aumento do número de objetos na rede, mas também com a complexidade da sua organização. Os testes também indicaram que existe a necessidade de estudar como os relacionamentos entre os objetos estão sendo formados na rede social, sendo um dos pontos que podem levar a melhorias na eficiência e eficácia dos algoritmos de busca de serviços em SIoT.

Como trabalhos futuros, novos testes serão realizados para determinar o comportamento do BPS com requisições mais variadas, contendo um número maior de serviços requeridos e também uma variação maior das características apresentadas. Outro teste que deverá ser essencialmente realizado é a verificação do comportamento dos algoritmos de busca ao lidar com os diversos tipos de relacionamentos possíveis em uma rede SIoT, identificando se existe uma ordem de busca específica entre esses relacionamentos que beneficie a busca de serviços. Além dos testes é possível focar esforços na construção de outras funcionalidades da rede SIoT que dependem diretamente da busca de serviços, como métodos de gerenciamento da rede social e composição de serviços. Outro trabalho futuro pode ser realizado por meio da identificação de outras estratégias de busca de serviços utilizadas na *Internet of Things* e verificar a possibilidade de adaptá-las para a busca em SIoT, incluindo métodos que usem inteligência artificial para isso.

Referências

Alves, T. M., da Costa, C. A., da Rosa Righi, R., and Barbosa, J. L. V. (2015). Exploring the social internet of things concept in a univeristy campus using nfc. In *Computing Conference (CLEI), 2015 Latin American*, pages 1–12. IEEE.

- Atzori, L., Carboni, D., and Iera, A. (2014). Smart things in the social loop: Paradigms, technologies, and potentials. *Ad Hoc Networks*, 18:121–132.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Atzori, L., Iera, A., and Morabito, G. (2011). Siot: Giving a social structure to the internet of things. *IEEE communications letters*, 15(11):1193–1195.
- Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012). The social internet of things (siot)—when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608.
- Atzori, L., Nitti, M., and Marche, C. (2016). Social internet of things. In <http://www.social-iot.org/index.php?p=downloads>.
- Brittes, M. P., Schneider Jr, B., and Wille, E. C. (2017). Trustworthiness management through social relationships in internet of medical things. *Journal of Communication and Information Systems*, 32(1).
- Chen, G., Huang, J., Cheng, B., and Chen, J. (2015). A social network based approach for iot device management and service composition. In *Services (SERVICES), 2015 IEEE World Congress on*, pages 1–8. IEEE.
- Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011):1–11.
- Girau, R., Martis, S., and Atzori, L. (2017). Lysis: A platform for iot distributed applications over socially connected objects. *IEEE Internet of Things Journal*, 4(1):40–51.
- Jadhav, B. and Patil, S. (2016). Wireless home monitoring using social internet of things (siot). In *Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on*, pages 925–929.
- Jarwar, M., Kibria, M., Ali, S., and Chong, I. (2018). Microservices in web objects enabled iot environment for enhancing reusability. *Sensors*, 18(2):352.
- Kim, J. E., Fan, X., and Mosse, D. (2017). Empowering end users for social internet of things. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 71–82. ACM.
- Neves, A. R. d. M., Sarmanho, K. U., Nascimento Jr, F. C., and Meiguins, B. S. (2017). Iniciativa smart campus: um estudo de caso em progresso na universidade federal do par. In *Anais do I Workshop de Computao Urbana (COURB 2017)*, volume 1. SBC.
- Nitti, M., Pilloni, V., and Giusto, D. D. (2016). Searching the social internet of things by exploiting object similarity. In *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*, pages 371–376. IEEE.
- Santos, B. P., Silva, L., Celes, C., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. (2016). Internet das coisas: da teoria  prtica. *Minicursos SBRC-Simpsio Brasileiro de Redes de Computadores e Sistemas Distribuidos*.
- Vodanovich, S., Sundaram, D., and Myers, M. (2010). Research commentary—digital natives and ubiquitous information systems. *Information Systems Research*, 21(4):711–723.
- Wang, K., Qi, X., Shu, L., Deng, D.-j., and Rodrigues, J. J. (2016). Toward trustworthy crowdsourcing in the social internet of things. *IEEE Wireless Communications*, 23(5):30–36.