

# UMA ABORDAGEM PARA PROVISIONAMENTO AUTOMÁTICO DE SENSORES VIRTUAIS EM NUVENS DE SENSORES

**Marcus Lemos, Carlos Carvalho e Douglas Lopes**

Universidade Estadual do Piauí

{marvin,cgnc}@uespi.br, douglas\_min@hotmail.com

**Raimir Holanda**

Universidade de Fortaleza

raimir@unifor.br

**Ricardo Rabêlo**

Universidade Federal do Piauí

ricardoalr@ufpi.edu.br

**Resumo** – No contexto de nuvens de sensores, o procedimento de provisionamento é essencial, uma vez que é o responsável por selecionar os sensores físicos que serão alocados para compor os sensores virtuais. Os principais trabalhos da literatura consideram a alocação de todos os sensores dentro da região de interesse. Tal abordagem, contudo, pode acarretar sérios problemas, como o desperdício do consumo de energia e o comprometimento da qualidade de serviço das aplicações. Assim, o objetivo deste trabalho é apresentar uma abordagem automática para o provisionamento de sensores virtuais. A abordagem minimiza, ao mesmo tempo, a quantidade de nós selecionados e o consumo total de energia dos nós que compõe a nuvem de sensores. Os resultados de experimentos iniciais demonstram que, dependendo do nível de erro aceitável definido pelo usuário, a abordagem reduziu entre 7% e 46% do consumo de energia, tornando-se uma solução a ser considerada nos cenários de nuvem de sensores.

**Palavras-chave** – Redes de Sensores Sem Fio, Nuvens de Sensores, virtualização.

**Abstract** – The provisioning task is essential in sensor cloud environment, since it is responsible for selecting the physical sensors that will be allocated to compose the virtual sensors. The main works in literature consider the allocation of all sensors within the region of interest. Such approach, however, has serious disadvantages such as waste of energy consumption and commitment of application service quality. The objective of this paper is to present an automatic approach to the provisioning of virtual sensors. The approach minimizes the amount of selected nodes and the total energy consumption of sensor nodes that compose the cloud. Results from initial experiments showed that, depending on the acceptable error level set by the user, the approach reduced between 7 % and 46 % of energy consumption, making a solution to be considered in sensors cloud scenarios.

**Keywords** – Wireless Sensor Networks, Sensor Cloud, virtualization.

## 1. INTRODUÇÃO

De acordo com [1], uma nuvem de sensores pode ser definida como um ambiente computacional, composto por várias Redes de Sensores Sem Fio (RSSF) que podem estar distribuídas em uma grande área

geográfica. É importante destacar que, dentro dessa nuvem, cada RSSF pode ter um provedor(proprietário) diferente e os nós podem ser heterogêneos. Neste caso, os sensores físicos podem ser virtualizados e a atividade de sensoriamento pode ser fornecida como um serviço para os usuários, permitindo que esses últimos possam adquirir/comprar serviços sob demanda. Conforme ilustrado na Figura 1, cada consulta de usuário é repassada à nuvem de sensores, geralmente representada por um *middleware* executado em um servidor de aplicação. O *middleware*, por sua vez, deve analisar os parâmetros definidos pelo usuário (como frequência de sensoriamento, região de interesse, etc.) e verificar se existem sensores físicos capazes de atender tal consulta. Após localizar os sensores físicos, o *middleware* deve decidir, por meio de um processo denominado provisionamento (*Provisioning*) [2], quais devem ser alocados para formar o sensor virtual que será associado à requisição. Não havendo sensores disponíveis para processar a requisição, o usuário deve ser informado. Nesse caso, o usuário pode tentar modificar alguns parâmetros e disparar uma nova consulta.

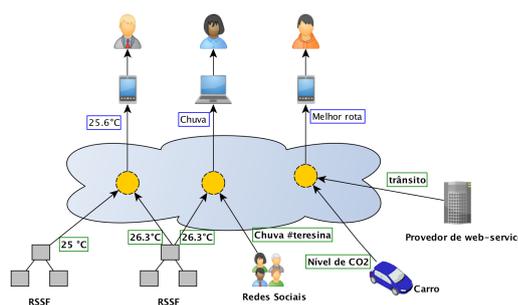


Figura 1: Modelo Conceitual de uma nuvem de sensores

A princípio, o problema de provisionamento de sensores virtuais parece similar ao problema da alocação de sensores, tradicionalmente tratado no contexto de Redes de Sensores Sem Fio [3], [4] e [5]. Contudo, grande parte dos trabalhos considera apenas nós sensores homogêneos (mesmas características físicas, como a quantidade e o tipo de sensores) e uma única aplicação sendo executada nesses mesmos nós (por exemplo, todos os nós possuem sensores de temperatura, umidade e luminosidade). Uma técnica bastante utilizada neste domínio consiste em colocar sensores redundantes dentro da área de interesse e organizar tais sensores em grupos (*cluster*). Dentro de cada grupo estabelecido, os nós periodicamente alternam entre os estados dormindo e acordado (*sleep/wake up*), conforme Figura 2. Entretanto, tal técnica não é viável para o ambiente de Nuvem de Sensores, pois não leva em consideração que podem ocorrer consultas diferentes ao mesmo tempo (por exemplo, consulta 1: requisita temperatura e umidade, e consulta 2: requisita temperatura e luminosidade) e que os nós da nuvem podem ser heterogêneos. Dessa forma, não há como garantir que os nós que estiverem acordados, contenham os sensores requisitados.

Consequentemente, ao longo dos anos, têm sido publicadas diversas propostas de *middleware* específicos para ambiente de Nuvens de Sensores. Alguns trabalhos [2], [6], [7] consideram que todos os nós dentro da região de interesse são alocados de forma a enviarem, periodicamente, suas medições ao *middleware* da nuvem da qual faz parte. O procedimento de provisionamento é executado exclusivamente pelo *middleware*, criando sensores virtuais por meio da agregação e/ou correlação das informações recebidas. Segundo [8], tal abordagem possui duas principais desvantagens. A primeira consiste no fato de que uma sobrecarga de mensagens pode inundar a rede, gerando um consumo excessivo de energia dos nós (geralmente alimentados por baterias), comprometendo, assim, o tempo de vida destes e da rede em geral. Além disso, essa sobrecarga de mensagens pode afetar também a Qualidade de Serviço (*Quality of Service - QoS*) das aplicações, ao comprometer o tempo de respostas das consultas disparadas na rede. Assim, o trabalho de [8] propõe uma abordagem diferente, na qual os sensores virtuais podem residir tanto na nuvem quanto nos nós sensores físicos. Dessa forma, os autores alegam que os sensores virtuais podem aproveitar-se da capacidade de processamento local dos nós físicos. Diante disso, seria possível realizar agregação de informações, por exemplo, que seriam, então enviadas ao *middleware*. Contudo, o trabalho deixa em aberto alguns pontos. Não é citado como os nós sensores físicos serão selecionados para compor um determinado sensor vir-

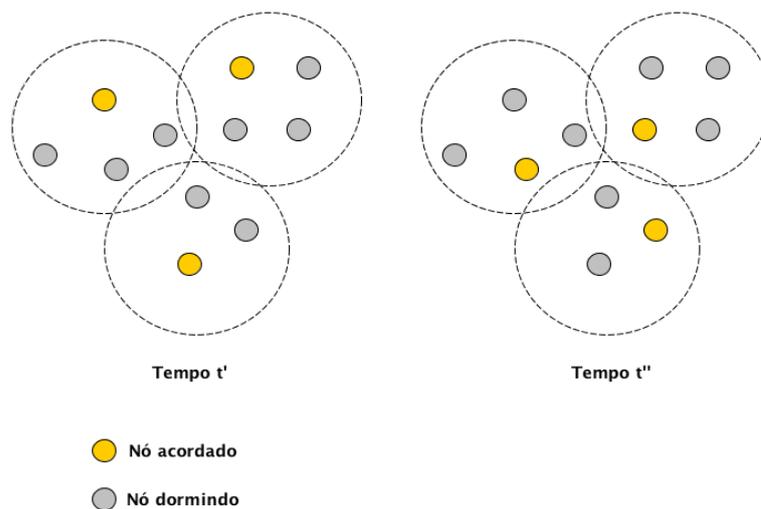


Figura 2: Modo Sleep/Wake up

tual. Além disso, não é tratado como nós físicos de diferentes provedores irão comunicar-se quando forem selecionados para compor um sensor virtual.

Considerando que uma Nuvem de Sensores pode conectar um grande número de nós sensores, provenientes de provedores diferentes, a tarefa de provisionamento pode tornar-se complexa, uma vez que poderá ser necessária a análise de diferentes combinações de nós até encontrar a melhor combinação que possa atender ao usuário e que não comprometa, de forma dispendiosa, os recursos da rede de um determinado proprietário em relação aos outros. Considere, por exemplo, a Figura 3. Ela ilustra um ambiente de Nuvem de Sensores formado por duas redes de sensores de provedores diferentes. Os nós em amarelo correspondem ao provedor *A*, enquanto os nós em cor verde ao provedor *B*. Os nós do provedor *A* possuem apenas sensores de temperatura e os nós do provedor *B* são equipados com sensores de temperatura e umidade. Inicialmente, um determinado usuário requisita informações de temperatura de uma região de interesse (delimitado pelas linhas pontilhadas pretas). Em seguida, outro usuário requisita informações acerca da temperatura e umidade da mesma região de interesse. Nos dois casos, o *middleware* deve selecionar um ou mais nós físicos capazes de responder às consultas. Conforme definido anteriormente, a abordagem *sleep/Wake up* nesse caso não seria adequada, uma vez que requisições diferentes foram disparadas e nem todos os nós são capazes de responder às várias requisições. Assim, fica a cargo do *middleware* analisar qual ou quais nós devem ser necessários para compor os dois nós virtuais, evidenciando, assim, um problema combinatorial.

Conforme mencionado anteriormente, o processo de provisionamento é de extrema importância para o desempenho da nuvem, mas ainda não é abordado de forma clara na literatura [8]. Assim, o objetivo deste trabalho é apresentar uma abordagem para a seleção automática de nós sensores físicos que devem compor os sensores virtuais. A abordagem minimiza a quantidade de nós físicos selecionados, objetivando: (i) atender as requisições solicitadas e (ii) minimizar o consumo de energia dos nós sensores que compõe a nuvem de sensores.

O artigo encontra-se estruturado da seguinte forma: Na Seção 1 foi contextualizado o problema de provisionamento e apontado alguns trabalhos relacionados à esta proposta. Na Seção 2 será apresentado uma visão geral das duas fases que serão consideradas para tratar o problema do provisionamento, enquanto a Seção 3 abordará, com mais detalhes, o funcionamento do ACASIM, responsável pela Fase1. Na Seção 4 serão discutidos os resultados obtidos em simulação. Por último, na Seção 5, tem-se as conclusões com alguns direcionamentos futuros.

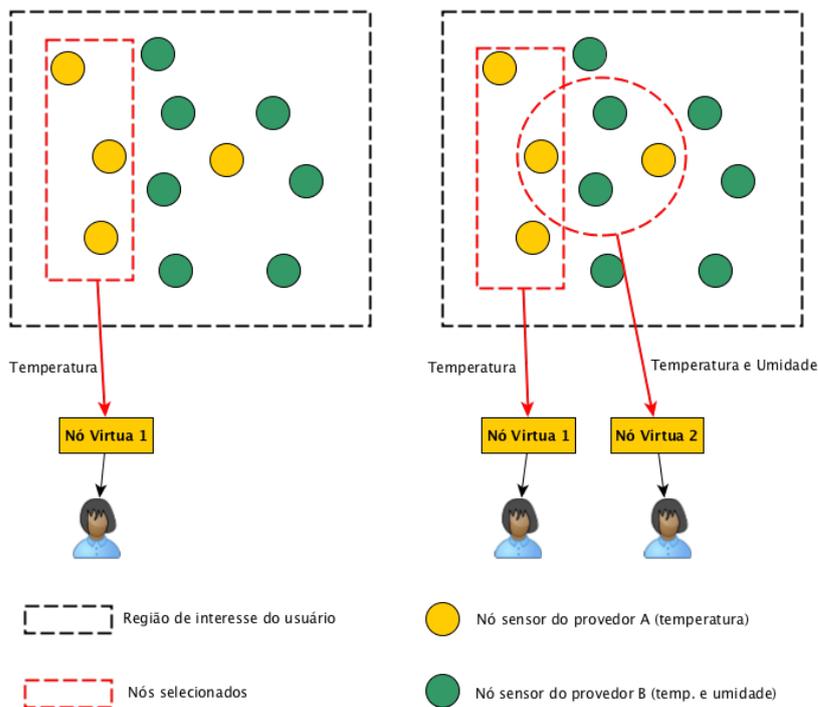


Figura 3: Exemplo de provisionamento

## 2 ABORDAGEM DE PROVISIONAMENTO PROPOSTA

Neste trabalho, considerou-se que o processo de provisionamento é executado pelo *middleware* responsável por gerenciar a nuvem de sensores. Além disso, por possuir uma abordagem centralizada, informações de todos os nós que compõem a RSSF são armazenadas em uma base de dados local. Esta base é atualizada sempre que uma nova RSSF é adicionada à nuvem ou quando nós são desativados ou adicionados em uma RSSF já existente. O *middleware* não precisa ser consciente do protocolo de roteamento utilizado pelas redes, uma vez que o *sink* de cada RSSF será o responsável pela comunicação com os nós. Partindo dessas premissas, o *middleware*, ao realizar a criação de um determinado sensor virtual, informará os respectivos nós selecionados (por meio dos *sinks* de cada RSSF). Uma vez selecionados, os nós responderão de acordo com os parâmetros recebidos.

Na abordagem proposta, foram definidas duas fases necessárias ao provisionamento. Na fase 1, é realizado, por um algoritmo denominado ACASIM (*algoritmo de clusterização adaptativo baseado na similaridade de dados*), o agrupamento dos nós sensores físicos que estão dentro da região de interesse do usuário. Os nós físicos são agrupados com base na similaridade dos valores medidos por seus sensores. Consequentemente, garante-se que os valores das medições de todos os nós de um determinado *cluster* estão próximos (dentro de uma margem aceitável definida pelo administrador da rede). Além disso, esse processo de agrupamento é dinâmico (adaptativo), pois os *clusters* podem ser recriados à medida que os valores das medições dos nós dentro de um determinado *cluster* comecem a ultrapassar um limiar de erro aceitável.

Na literatura, encontram-se diversos trabalhos relacionados a agrupamento de nós sensores em RSSF [9], [10], [11], [12], [13] e [14]. O principal ponto em comum entre estes trabalhos consiste no fato de que os grupos criados contêm nós próximos entre si. As medições dos nós que compõem cada grupo podem ser agregadas, geralmente por um nó especial denominado *cluster-head*, e transmitidas ao nó *sink*. Entretanto, a proximidade física não garante que as medições dos nós de um determinado grupo estejam correlacionadas de forma que seja possível a agregação. Considere por exemplo, dois nós que estejam separados por uma parede. Um nó pode estar localizado dentro de uma sala com temperatura baixa, devido ao funcionamento de um ar-condicionado, enquanto o outro nó pode estar ao ar-livre, durante um dia quente.

Durante a execução da Fase 2, um subconjunto de nós físicos de cada *cluster* são selecionados para

compor o sensor virtual responsável por atender determinada requisição. Esse procedimento irá minimizar a quantidade de nós escolhidos, de forma que a requisição seja atendida e o consumo de energia dos nós sensores minimizada.

A Figura 4a ilustra um exemplo de nuvem de sensores composta por duas RSSF de provedores diferentes (uma rede representada por círculos amarelos e a outra por triângulos verdes). Inicialmente, cada nó corresponde a um grupo individual. O *middleware*, baseado nos parâmetros da requisição de um determinado usuário, dispara uma consulta na nuvem. Em seguida, cada nó transmite suas informações para os nós sinks de suas respectivas redes (4b). Estes, por sua vez, devem ser configurados para enviar tais informações ao *middleware* responsável pelo gerenciamento da nuvem de sensores, representado, na Figura 4, pelo nó quadrado cinza-escuro. É possível também que as RSSF sejam configuradas para transmitir as informações diretamente ao *middleware*, sendo uma decisão de projeto do proprietário da rede. O *middleware*, depois de um determinado período recebendo as medições dos nós que compõe a nuvem, executa o ACASIM, responsável pela agrupamento dos nós (Fase 1). Por meio deste algoritmo, os nós são agrupados baseados na similaridade das suas medições. Isto significa que grupos podem conter nós que estejam separados fisicamente. Após a execução do ACASIM, o *middleware* registra a formação de dois grupos (Figura 4c). Em seguida, o *middleware* define um subconjunto de nós (Fase 2) que ficarão responsáveis por enviar as medições. Tais nós ficarão ativos, enquanto os demais poderão entrar em modo de economia de energia. Todos esses procedimentos estão resumidos na Figura 5.

Neste trabalho, será detalhando apenas o funcionamento da Fase 1, executado pelo algoritmo ACASIM. Em trabalhos futuros, será abordado o funcionamento da Fase 2.

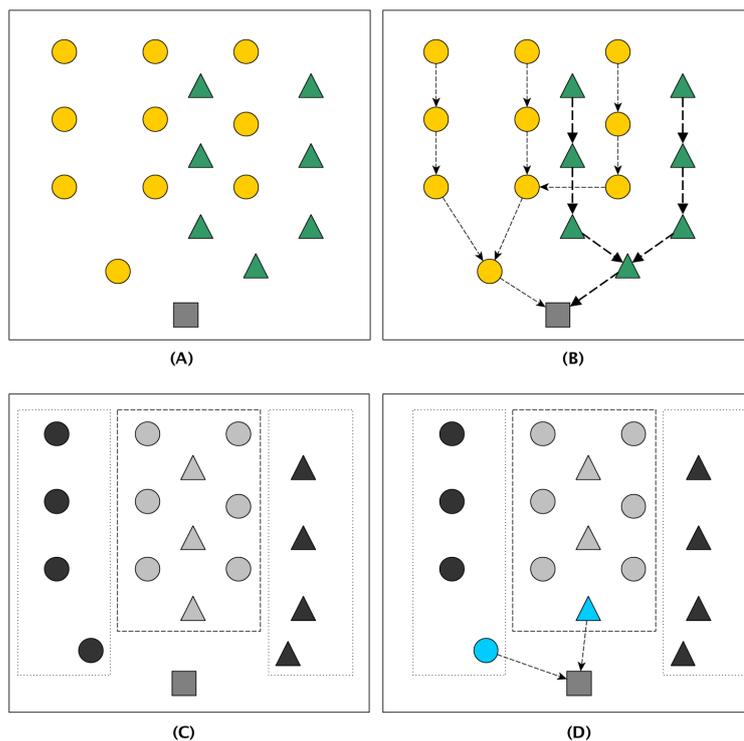


Figura 4: Exemplo do funcionamento básico do ACASIM

Uma característica importante do ACASIM é a sua adaptabilidade às variações dos padrões das medições dos nós que estão nos *clusters*. O algoritmo trabalha em conjunto com o ASLR [15] e, por isso, todos os nós da RSSF recebem um modelo que representa as medições (gerado pelo ASLR) dos nós que estão no seu grupo. Ao realizar uma nova medição, o nó sensor compara este valor com o gerado pelo modelo. Caso a diferença seja acima de um limiar parametrizado, o nó envia um sinal ao *middleware*. Este, ao receber o sinal, executa novamente o ACASIM, para que os grupos possam ser recriados.

A principal diferença do ACASIM em relação aos demais trabalhos da literatura é o fato dos *clusters* criados não agruparem os nós pela proximidade física, mas sim pela similaridade dos valores medidos.

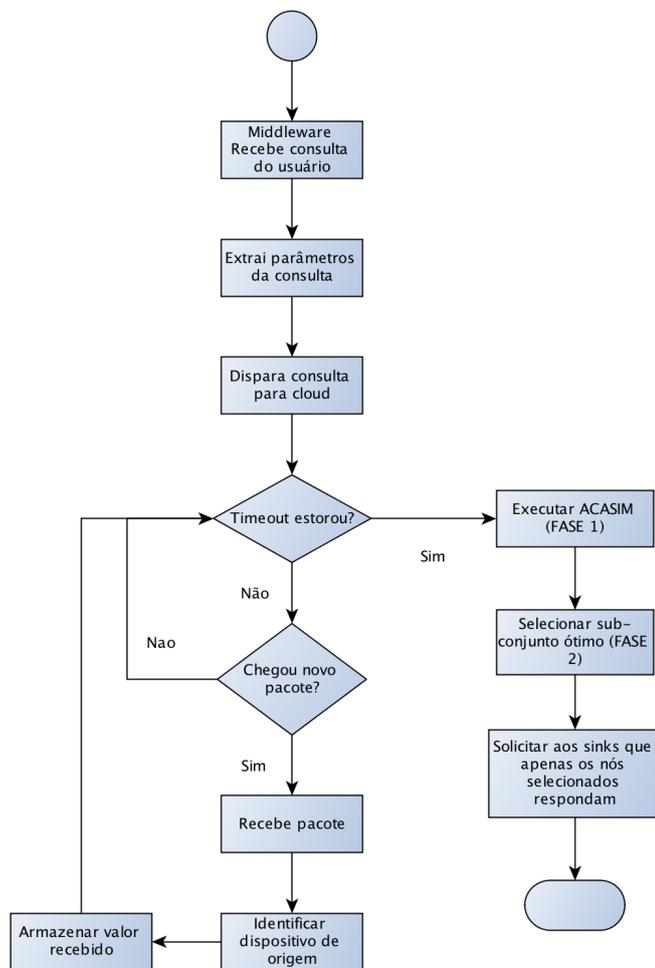


Figura 5: Funcionamento do algoritmo de provisionamento proposto

Essa abordagem traz a vantagem que o *middleware* poderá escolher apenas um nó de cada *cluster* (ou um pequeno subconjunto de nós) e mesmo assim, ter medições que representem todos os nós da região de interesse. Considere, por exemplo, a Figura 4c. Tem-se um cluster (*A*) formado por nós com a cor cinza-claro e outro cluster (*B*) formado por nós com a cor cinza-escuro. Além disso, cada cluster é formado por nós de proprietários diferentes. Percebe-se que o cluster *B* possui nós que estão fisicamente distantes um dos outros (em comparação com cluster *A*). Entretanto, uma vez que o ACASIM garante que os nós dentro de um cluster possuem os valores de suas medições dentro de uma margem de erro aceitável (conforme será visto na Sub-seção 3), a escolha de apenas um nó sensor (ou um sub-conjunto de nós sensores) será suficiente para que se tenha uma visão completa de toda a região monitorada pelo cluster.

### 3 ACASIM-ALGORITMO DE CLUSTERIZAÇÃO ADAPTATIVO BASEADO NA SIMILARIDADE DE DADOS

O fluxograma de funcionamento básico do ACASIM é ilustrado na Figura 6. Inicialmente, o *middleware* responsável pelo ACASIM recebe uma nova consulta vinda da aplicação do usuário da RSSF. Em seguida, o *middleware* verifica (da sua base de dados local) todos os nós sensores físicos que estão ativos na região de interesse especificada na consulta. Neste momento, cada nó sensor é tratado como um *cluster* e os seus identificadores são adicionados em uma lista denominada *lista\_clusters*. Uma outra lista, chamada de *clusters\_comparados*, é criada vazia e tem como objetivo, armazenar todos os *clusters* que já foram comparados. Em seguida, o algoritmo entra em um laço de repetição, no qual a condição de igualdade é a existência de ao menos dois *clusters* que ainda não foram comparados. Caso a condição seja verdadeira, o algoritmo calcula, baseado na distância euclidiana, quais os dois *clusters* (denominados *A* e *B*) estão

mais próximos. De  $A$  e  $B$  são calculados a média ( $\bar{X}$ ) e o desvio padrão ( $\sigma$ ). O valor de  $\sigma$  abaixo de um  $LIMIAR$  definido pelo administrador do *middleware* significa que os valores das medições dos nós sensores estão próximos entre si (o suficiente de forma a serem representados pela  $\bar{X}$ ), fazendo com que os dois *clusters* sejam unidos em um novo *cluster*  $C$ . Em seguida,  $C$  é adicionado na *lista\_clusters*, enquanto um conjunto representado por  $(A, B)$  é adicionado na lista *clusters\_comparados*. Tal processo é repetido até que não existam mais *clusters* a serem comparados. Após a definição dos *clusters*, o *middleware* seleciona os nós sensores (de cada *cluster*) que devem responder a consulta do usuário. O *middleware* gera um modelo das medições, baseado no algoritmo ASLR, que é enviado aos nós sensores das RSSF. Os nós sensores comparam o valor de cada nova medição com o valor gerado pelo modelo. Caso haja uma diferença significativa, o nó sensor então envia um alerta para que o *middleware* possa reconstruir os grupos.

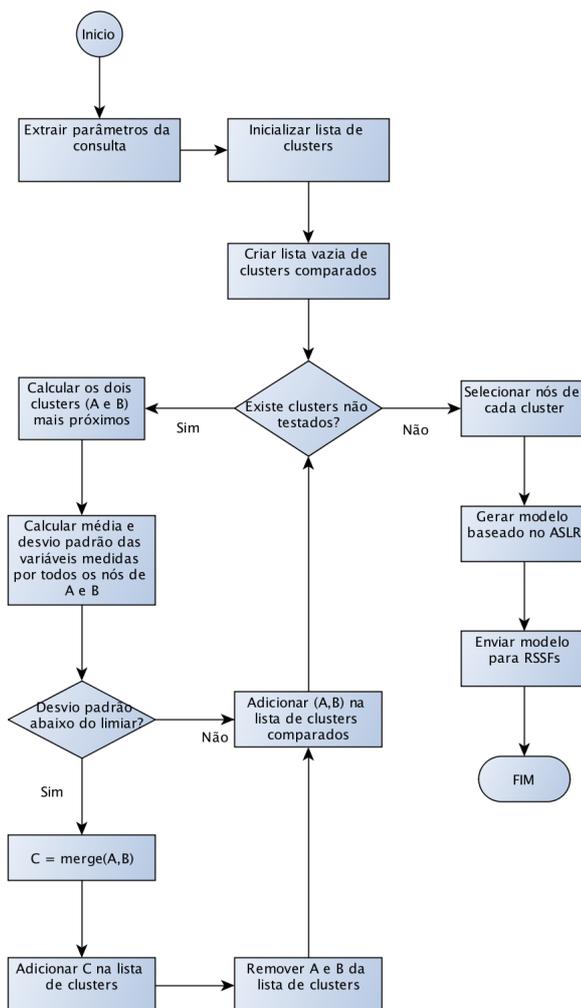


Figura 6: Fluxograma do funcionamento básico do ACASIM

A Figura 7 ilustra, de forma reduzida, como ocorre este processo iterativo de agrupamento. Na Figura há 5 nós sensores e a similaridade das suas medições é representada pela posição relativa na imagem (não correspondendo, necessariamente, à localização física). Assim, quanto mais próximos os dois nós sensores, mais similares são suas medições. Inicialmente, os grupos correspondem aos nós sensores individuais (Figura 7a). Em seguida, o ACASIM calcula que  $A$  e  $B$  são os dois grupos mais próximos e que o desvio padrão de suas medições está abaixo do limiar definido. Assim,  $A$  e  $B$  são unidos para formar um único grupo (Figura 7b). Na Figura 7c, este grupo também é unido com  $C$ , pois o desvio padrão das medições de  $A$ ,  $B$  e  $C$  também ficou abaixo do limiar. Como  $D$  e  $E$  são os dois grupos mais próximos e o desvio padrão das medições também ficou abaixo do limiar, eles são unidos para formar um novo grupo (Figura 7d). Na Figura 7e, o ACASIM tenta unir os grupos formados pelos nós sensores  $A$ ,  $B$  e  $C$  com o grupo formado

pelos nós sensores  $D$  e  $E$ . Entretanto, como o desvio padrão das medições de todos os nós sensores acima do limiar, os grupos não foram unidos (Figura 7f).

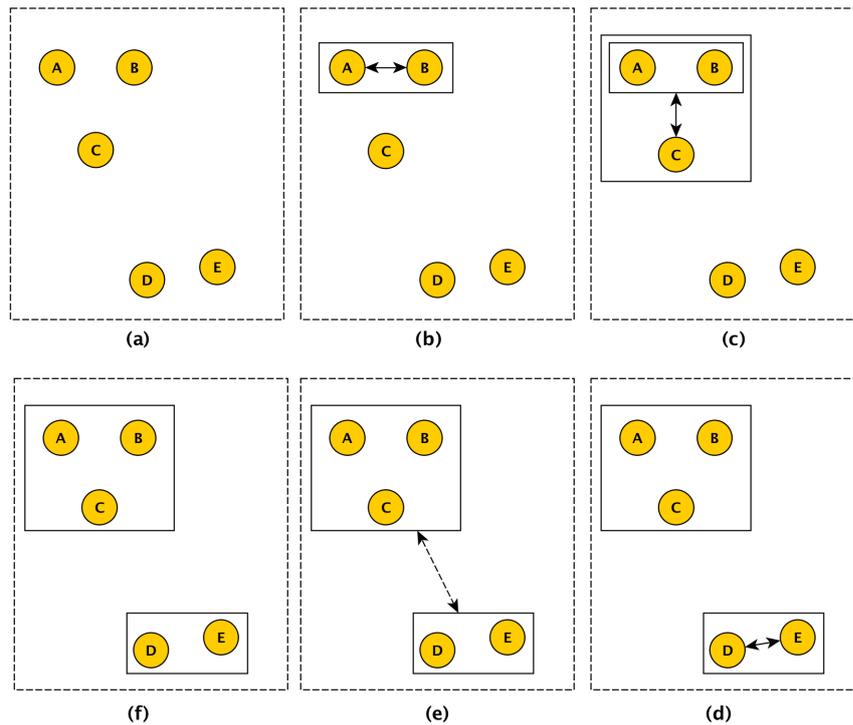


Figura 7: Funcionamento básico do ACASIM

## 4 RESULTADOS E DISCUSSÕES

A linguagem R<sup>1</sup> foi utilizada para simular uma RSSF baseada em dados reais, obtidos do projeto Intel, Berkeley Lab Data<sup>2</sup>. Os dados disponibilizados consistem nas medições de 54 nós sensores, medidos à cada 31 segundos, entre os dias 28 de Fevereiro e 04 de Abril de 2004. Neste trabalho, levou-se em conta apenas as primeiras 1000 medições das variáveis temperatura e umidade e que apenas o nó com mais energia de cada clusters irá transmitir as medições. Em trabalhos futuros, serão analisadas outras variáveis para a seleção do subconjunto ótimo de nós. Cada nó possui comunicação direta com o nó *sink*, assim, não foi definido nenhum protocolo de roteamento.

O consumo total de energia da rede será utilizado como métrica para avaliar o desempenho da abordagem proposta. Para o cálculo desse consumo, utilizou-se o modelo de energia abordado em [16], [17] e [18]. Considera-se a taxa de transmissão do nó de  $0,26\mu s/bit$ , sendo a corrente elétrica que flui pelo nó ao receber um pacote de  $7,0mA$  e ao transmitir de  $21,5mA$ . Assim, definiu-se o seguinte modelo [18]:

- $Q_{Transmissao} = 3 * 21,5mA * (0,26 * 10^{-6}s/bit * 288bits) = 0,48375mJ/mensagem$
- $Q_{Recepcao} = 3 * 7,0mA * (0,26 * 10^{-6}s/bit * 288bits) = 0,1575mJ/mensagem$
- $Q_{ouvir} = 3 * 7,0mA * (0,26 * 10^{-6}s/bit * 16bits) = 0,00875mJ/mensagem$

onde

- $Energia\ Dissipada\ (Q) = Voltagem * Corrente\ Eletrica * Tempo$ , sendo

<sup>1</sup><http://www.r-project.org/>

<sup>2</sup><http://db.csail.mit.edu/labdata/labdata.html>

- $Tempo = TaxaDeTransmissao * TamanhoDaMensagem.$

Considerando que todos os nós sensores transmitiram ao *sink* as 1000 primeiras medições, o consumo total de energia sem a utilização do ACASIM foi de 26.122,50. Foram realizadas 8 simulações com os seguintes valores de *LIMIAR*: 0.1, 0.3, 0.5, 1.0, 2.0, 3.0, 4.0, 6.0. A Figura 8 ilustra o consumo em *mJ* para cada um dos *LIMIARES*. Pela imagem, percebe-se que há uma correlação negativa entre as variáveis *LIMIAR* e *ENERGIA*, ou seja, à medida que o valor de *LIMIAR* aumenta, o consumo total de *ENERGIA* diminui. A Figura 9 ilustra essa economia em percentual. Pela análise dos gráficos, percebe-se que o ACASIM conseguiu reduzir entre 7% e 46% o consumo total de energia. À medida que o valor de *LIMIAR* cresce, o número de nós físicos por grupo também aumenta, ocasionando a diminuição da quantidade de grupos. Como apenas um subconjunto de nós de cada grupo responde, há uma diminuição na quantidade de transmissões de pacotes na rede, provocando, conseqüentemente, a redução no consumo de energia.

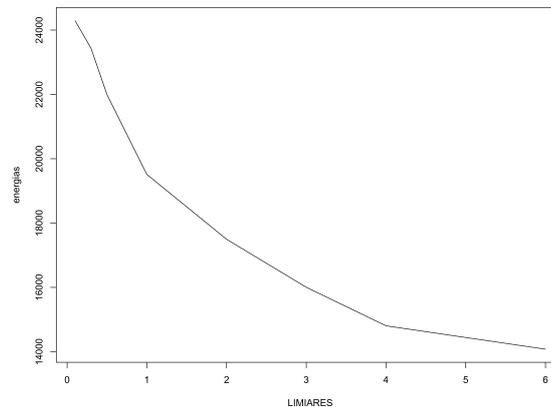


Figura 8: Consumo de energia

Considerando que apenas um nó sensor de cada *cluster* transmite suas medições e que essas medições representam as medições de todos os outros nós sensores do *cluster*, calcula-se o Erro Quadrático Médio (EQM) de cada *cluster*. Ou seja, o quanto as medições (denominado  $X_j$ ) dos outros nós sensores diferem da medição do nó sensor que está transmitindo (denominado  $X_T$ ). Para cada valor  $L$  de *LIMIAR* calculou-se o EQM corresponde ao somatório dos erros das medições dos nós sensores pertencentes a cada um dos  $n$  clusters gerados durante a simulação. Assim, o valor do EQM pode ser expresso por:

$$EQM^L = 1/n \sum_{i=1}^n 1/m \sum_{j=1}^m ||X_j - X_T||^2$$

A Figura 10 representa o  $EQM^L$  para cada valor de *LIMIAR*. Verifica-se uma tendência de aumento do  $EQM^L$  à medida que o valor de *LIMIAR* aumenta.

Pelos resultados obtidos, verifica-se que quanto maior o consumo de energia, menor o EQM. Assim, tem-se que a escolha do valor ideal de  $L$  dependerá de uma decisão de projeto. Ou seja, deve-se analisar o quanto se aceita de precisão de medição x consumo de energia.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou uma abordagem para o provisionamento automático de sensores virtuais em ambiente de nuvens de sensores. Por meio de um algoritmo denominado ACASIM, *clusters* de nós sensores, baseado na similaridade de suas medições, são formados. Em seguida, subconjunto de nós sensores são selecionados, de forma a compor os sensores virtuais responsáveis pelas consultas dos usuários.

Testes iniciais demonstraram uma redução de até 46% no consumo de energia, indicando a viabilidade do algoritmo proposto. Como trabalhos futuros, pretende-se: (i) realizar uma comparação entre o ACASIM

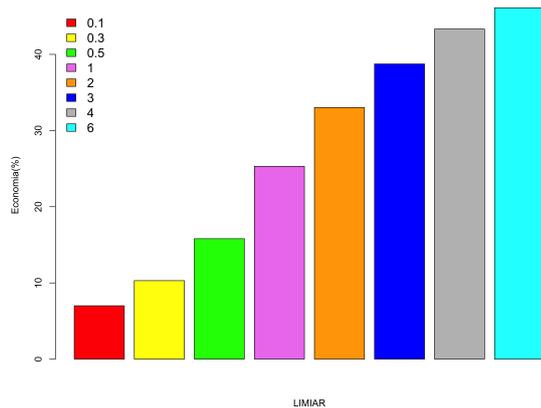


Figura 9: Percentual de economia de energia

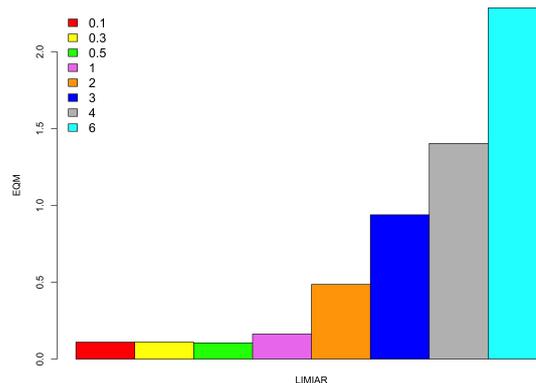


Figura 10: Erro quadrático médio

e outros algoritmos de clusterização para virtualização de sensores, existentes na literatura, (ii) realização simulações em ambientes mais complexos, levando-se em conta outras variáveis tal como o protocolo de roteamento, (iii) desenvolver um algoritmo para realizar a escolha ótima de nós sensores que irão compor o sensor virtual e (iv) implementar o algoritmo em uma plataforma de hardware e avaliar o seu desempenho em um ambiente real para sua validação em campo.

## AGRADECIMENTOS

Este trabalho foi parcialmente financiado pela Universidade Estadual do Piauí (Uespi) e o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) sob o processo No 482271/2013-2.

## REFERÊNCIAS

- [1] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi and M. A. Hossain. “A Survey on Sensor-Cloud: Architecture, Applications, and Approaches”. *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–18, 2013.
- [2] S. Madria, V. Kumar and R. Dalvi. “Sensor Cloud : A Cloud of Virtual Sensors”. *Software, IEEE*, vol. 31, no. 2, 2014.
- [3] Y. Lin, J. Zhang, H.-H. Chung, W. Ip, Y. Li and Y.-H. Shi. “An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks”. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 3, pp. 408 – 420, 2011.
- [4] A. Chamam and S. Pierre. “On the Planning of Wireless Sensor Networks: Energy-Efficient Clustering under the Joint Routing and Coverage Constraint”. *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1077–1086, 2009.
- [5] A. Boukerche, X. Fei and R. B. de Araujo. “An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange.” *Computer Communications*, vol. 30, no. 14-15, pp. 2708–2720, 2007.
- [6] A. M. Ortiz, D. Hussein, S. Park, S. N. Han and N. Crespi. “The Cluster Between Internet of Things and Social Networks: Review and Research Challenges”. *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 206–215, June 2014.
- [7] V. C. Emeakaroha, K. Fatema, P. Healy and J. P. Morrison. “Analysis and Architecture for a Generic Cloud-based Sensor Data Management Platform”. *Sensors & Transducers Journal*, vol. 185, no. 2, 2015.

- [8] I. Santos, L. Pirmez, F. Delicato, S. Khan and A. Zomaya. “Olympus: The Cloud of Sensors”. *Cloud Computing, IEEE*, vol. 2, no. 2, pp. 48–56, Mar 2015.
- [9] F. Gielow, G. Jakllari, M. Nogueira and A. Santos. “Data similarity aware dynamic node clustering in wireless sensor networks.” *Ad Hoc Networks*, vol. 24, pp. 29–45, 2015.
- [10] R. Marin-Perianu, C. Lombriser, P. Havinga, H. Scholten and G. Tröster. “Tandem: A Context-Aware Method for Spontaneous Clustering of Dynamic Wireless Sensor Nodes”. In *The Internet of Things*, edited by C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern and S. Sarma, volume 4952 of *Lecture Notes in Computer Science*, pp. 341–359. Springer Berlin Heidelberg, 2008.
- [11] N. D. Pham, T. D. Le, K. Park and H. Choo. “SCCS: Spatiotemporal clustering and compressing schemes for efficient data collection applications in WSNs”. *International Journal of Communication Systems*, vol. 23, no. 11, pp. 1311–1333, 2010.
- [12] N. C. Devi, S. Prabeela, V. Palanisamy and K. Baskaran. “Distance Functions For Clustering in Wireless Sensor Networks”. *International Journal of Wireless Communications and Networking*, vol. 3, no. 1, pp. 73–78, 2011.
- [13] A. Ribas, J. Colonna, C. Figueiredo and E. Nakamura. “Similarity clustering for data fusion in Wireless Sensor Networks using k-means”. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–7, June 2012.
- [14] V. Sharma, K. R. Sharma and S. Sharma. “Simulated Annealing Based Neural Network for Dynamic Clustering In Wireless Sensor Network”. *International Journal Of Engineering Development And Research*, vol. 3, no. 1, pp. 41–45, 2014.
- [15] V. Bezerra, M. C. Júnior, O. Valéria, C. D. Neto, L. Leal, M. Lemos, C. G. Carvalho, J. B. Filho, R. Holanda and N. Agoulmine. “A Quality-Aware and Energy-Efficient Context Management Framework for Ubiquitous Systems”. In *28th International Conference on Advanced Information Networking and Applications*, AINA '14, Washington, DC, USA, 2014. IEEE Computer Society.
- [16] M. V. d. Lemos, L. B. Leal and R. H. Filho. “Detecção de Intrusão em Redes de Sensores Sem Fio utilizando uma abordagem colaborativa e cross-layer”. In *XXVII Simpósio Brasileiro de Redes de Computadores*, 2009.
- [17] D. Schmidt, M. Krämer, T. Kuhn and N. Wehn. “Energy modelling in sensor networks”. *Advances in Radio Science*, vol. 5, pp. 347–351, 2007.
- [18] A. P. R. D. Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz and H. C. Wong. “Decentralized intrusion detection in wireless sensor networks”. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pp. 16–23, New York, NY, USA, 2005. ACM.