

Avaliando Algoritmos para um Elemento Sensor Autônomo

Alexandra da Silva Pereira^{1,2}, Thais R. M. B. Silva¹, Fabrício A. Silva¹,
Antonio A. F. Loureiro², Linnyer B. Ruiz³

¹Instituto de Ciências Exatas – Universidade Federal de Viçosa - *Campus Florestal* (UFV)
35.690-000 - Florestal, MG - Brasil

²Departamento de Ciência da Computação Universidade Federal de Minas Gerais (UFMG)
31.270-010 - Belo Horizonte, MG - Brasil

³Departamento de Informática – Universidade Estadual de Maringá (UEM)
87.020-900 – Maringá - PR - Brasil

alexandra.pereira@ufv.br, thais.braga@ufv.br, fabricio.asilva@ufv.br
loureiro@dcc.ufmg.br, linnyer@gmail.com

Abstract. *Wireless Sensor Networks (WSN) are used in different contexts, including monitoring of climatic aspects in a remote region, actuators for geographic, spatial events and etc. Based on this limitation scenario of WSN a generic model that seeks to use a network computational resources in a more balanced way was developed, with autonomic computing concepts that create a self configurable network. In this paper we develop instances based on this generic model to improve network performance. We have implemented two more sophisticated algorithms and analyze their behavior in experiments performed in real sensor networks, also known as testbeds. The results show that the implementations maintain or improve important aspects such as the number of sent messages, self managed and sensitivity of the application.*

Resumo. *Redes De Sensores Sem Fio (RSSF) são utilizadas em diversos contextos, que podem incluir monitoração de aspectos climáticos em uma região remota, atuadores para eventos geográficos, espaciais e etc. Diante do cenário de limitação de RSSF foi desenvolvido um modelo genérico de Elemento Sensor Autônomo (ESA) que busca utilizar de maneira mais equilibrada os recursos computacionais de um nó sensor, com conceitos de computação autônoma que tornam a rede auto-configurável. Neste trabalho, desenvolvemos instâncias baseadas no modelo genérico para melhorar o desempenho da rede. Implementamos algoritmos de análise e planejamento, e avaliamos seus comportamentos em experimentos executados em redes de sensores reais. Os resultados mostram que as implementações mantêm ou melhoram aspectos importantes como o número de mensagens enviadas, a autoconfiguração e a sensibilidade da aplicação.*

1. Introdução

As Redes de Sensores Sem Fio são um tipo específico de redes sem fio, classificadas como rede *ad-hoc* e formadas por milhares de dispositivos chamados de nós

sensores[Buratti et al. 2009]. Para a formação de uma RSSF, nós sensores são disseminados em um ambiente remoto, geralmente, com a finalidade de monitorar e coletar dados pertinentes a modelagem da rede. Os nós sensores que compõem a rede são conhecidos por possuírem restrições de hardware (i.e., processamento, memória e energia)[Rawat et al. 2014].

Para tornar o gerenciamento de uma RSSF mais inteligente, foi proposto por [Braga 2006] [Braga et al. 2006] Elemento Sensor Autônomo (ESA). O ESA é relevante pela sua estrutura genérica, o que permite incluir e experimentar diversos algoritmos em cada uma de suas etapas. Ele é um modelo desenvolvido com base na proposta de tornar os sistemas de RSSF melhores no aspecto de auto-gestão. O seu funcionamento é cíclico e suas etapas são capazes de prover à rede os seguintes serviços: monitorar, analisar, planejar e executar ações ao longo do ciclo de execução de uma aplicação.

Diante do cenário de restrições em RSSFs e da característica genérica do ESA, a motivação é observar e analisar o comportamento de algoritmos no contexto do ESA. Portanto, implementamos, além da proposta original do ESA, três novas estratégias de análise e planejamento baseadas em meta-heurísticas.

Para todas as estratégias, foram feitas análises e levantamentos do quanto eles podem impactar a RSSF. O objetivo é avaliar a capacidade da rede de monitorar eventos mais rapidamente e mais precisamente (i.e., com mais informações), com o intuito de prover mais detalhes aos responsáveis por atuar nos casos de ocorrências de eventos, como um incêndio, por exemplo.

O presente trabalho segue organizado da seguinte forma. Na Seção 2 é feita uma apresentação de trabalhos presentes na literatura que se relacionam com o tema de pesquisa desenvolvido. A Seção 3 apresenta os cenários escolhidos, bem como, os algoritmos implementados. A Seção 4 discute os cenários de cada uma das implementações, quais foram os resultados e como estes resultados podem impactar o funcionamento da RSSF. Na Seção 5 apresentamos, finalmente, expectativas de trabalhos futuros e algumas considerações finais.

2. Trabalhos Relacionados

A computação autônoma aplicada às RSSFs é um tema que vem sendo explorado na literatura nos últimos anos. [Peng et al. 2014] discutem um modelo que utiliza um autômato celular para auto-organização da rede. Para isso, o algoritmo possui rotinas que mantêm a cobertura da rede estável, isto é, um algoritmo que verifica periodicamente a conectividade com os vizinhos, caso ele tenha menos de 2 vizinhos ativos, ele sai do modo *stand-by*. Os experimentos para validação foram feitos em simulação com uma rede de sensores homogênea.

Os autores [Sadouq et al. 2014] fazem a proposta de um arcabouço que trabalha com a rede de sensores como um cluster formado por vários nós. Os nós possuem 3 funções pré definidas: sensorar, rotear dados e atuar como ponto de acesso. O principal objetivo é ter uma rede mais sistêmica, robusta e escalável.

Na proposta discutida por [Abdelaal 2015], um modelo foi desenvolvido com o intuito de melhorar as técnicas existentes em RSSFs para economizar ainda mais energia. Os três principais objetivos são: agregar dados para o sensoriamento tornar-se mais barato, otimização de hardware e auto-adaptações preditivas.

O trabalho de [Jabeur et al. 2015] propõe modelos para redes de sensores sem fio baseadas em aspectos da natureza e são chamados de bio-inspirados. Nele, a rede formada pelos nós sensores é descrita como análoga a um ecossistema. Conforme este ecossistema passa por metamorfoses a rede evolui, e essa evolução resolve problemas que são recorrentes em RSSF. Todo o modelo proposto mostrou-se bastante interessante e pode trazer melhorias para aplicações em redes de sensores sem fio.

Esses aspectos da natureza também são explorados em outros trabalhos. Já [Lakshmi and Mohan Rao 2014] propõem um algoritmo com a finalidade de proporcionar auto-cura baseada em uma técnica de otimização de enxame. A ideia é estender o tempo de vida de uma rede de sensores. Neste trabalho foi desenvolvido um algoritmo baseado em colônia de formigas, no qual os pacotes buscam por caminhos mais curtos até o ponto de acesso. [Nguyen et al. 2014] propuseram um algoritmo genético para atuar na auto-configuração e otimização de uma RSSF. A rede é definida como um cluster, existe comunicação hierárquica entre os nós, e os nós que podem se comunicar com o ponto de acesso. O algoritmo genético trabalha para encontrar os melhores líderes, esses serão definidos como clusters e os nós dos quais cada cluster é responsável.

O trabalho proposto neste artigo difere dos anteriores por apresentar uma modelagem genérica e bem dinâmica no contexto das RSSF. O ESA genérico estabelece rotinas para controle autônomo da rede, e trabalha sob rotinas de auto-cura, auto-controle e auto-configuração. Partindo de uma implementação fiel ao modelo proposto por [Braga 2006], foram criadas instâncias que utilizam algoritmos mais sofisticados, que atuam na análise e no planejamento que compõem o ESA. O objetivo é fazer com que eventos ocorridos na rede sejam mais rapidamente e mais precisamente monitorados pelos nós sensores, fazendo com que os responsáveis pela atuação possam agir de maneira mais apropriada.

3. Soluções

Buscamos dois algoritmos que pudessem contribuir de maneira positiva para a autoconfiguração da rede. Logo, poderemos fazer comparações entre as instâncias mais básicas do ESA e as três novas instâncias desenvolvidas com algoritmos mais complexos e que exigem maior processamento. Foram implementados dois algoritmos diferentes para trabalhar em conjunto com o ESA. O primeiro é baseado em um algoritmo genético (GA), o segundo é uma implementação de colônia de formigas (ACS) e o terceiro uma composição com os dois algoritmos anteriores.

O objetivo dos algoritmos implementados é atuar na autoconfiguração dos parâmetros de sensoriamento dos ESAs. Em outras palavras, ambos os algoritmos devem ajustar os intervalos de sensoriamento e de disseminação dos dados sensoreadas de acordo com as medidas coletadas. Neste caso, quando os valores sensoreados estiverem dentro de um valor considerado fixo (i.e., sem ocorrência de eventos), é possível aumentar os intervalos para economizar recursos. Por outro lado, quando os valores sensoreados apresentarem mudanças significativas, os intervalos devem ser reduzidos para que os detalhes dos eventos não sejam perdidos.

3.1. Algoritmo Genético (GA)

Os algoritmos genéticos (GAs) utilizam conceitos provenientes do princípio de seleção natural para abordar uma série ampla de problemas, em especial de

otimização [Whitley 1994].

A nova instância do ESA com o algoritmo genético se concentra na monitoração e reproduz grandezas sensoreadas, para que a aplicação tenha uma perspectiva de análise mais interessante para as etapas de planejamento e execução.

```

1 AlgoritmoGenetico()
2  $t=0$ ;
3 cria populacao P(t);
4 for  $t \leftarrow 1$  to  $P(t)$  do
5   | avalia aptidão indivíduo(t)
6 end
7 while condição de parada não satisfeita do
8   |  $t=t+1$ ;
9   | Selecionar população P(t) de P(t-1);
10  | Aplicar operadores de cruzamento sobre P(t);
11  | Aplicar operadores de mutação sobre P(t);
12  | Avaliar P(t);
13 end

```

Algorithm 1: Pseudo-código GA.

O algoritmo 1 apresenta um pseudo-código do GA implementado, no qual a população é composta por variáveis que representam os dados sensoreados pela aplicação de RSSF. Para essa implementação, a geração inicial foi determinada por dados reais sensoreados em uma RSSF real. Com isso, a linha 3 do pseudo-código representa uma função que tem a responsabilidade de percorrer um vetor com dados sensoreados e representar cada indivíduo, que é representado por uma média de 3 três ocorrências de sensoreamento que foram observadas em experimentos prévios. Na linha 5, temos uma função responsável por fazer o cálculo da aptidão, que avalia o quão bom um indivíduo é.

Na linha 7, entramos em um laço que irá determinar quando finalizar o algoritmo, de acordo com o número de gerações estabelecidas. Neste laço, são executadas as etapas de seleção, cruzamento, mutação e avaliação. Na função de seleção, representada pela linha 9, é utilizado o resultado da aptidão para estabelecer quais serão os indivíduos que farão parte do cruzamento, no caso implementado, os indivíduos, ou dados, com maior variação (i.e., maior desvio padrão) foram considerados mais valiosos. Na linha 10, a chamada de cruzamento é responsável por executar o procedimento com um ponto de corte, onde o filho recebe características genéticas dos pais escolhidos. Em seguida, a linha 11 indica a função responsável pela mutação que causará variabilidades no indivíduo, hora com perturbações de 101,15%, hora com perturbações de 98,85%, valores esses escolhidos para não causar variações muito significativas. Por fim, uma nova chamada a avaliação, que determina a aptidão e escolhe quais indivíduos farão parte das próximas iterações.

3.2. Algoritmo Baseado em Regras (ESA-REGRAS)

As regras foram implementadas para que sejam executadas as seguintes reprogramações na etapa de planejamento:

- **Condição:** Caso o intervalo de sensoriamento esteja muito alto.

Ação: Diminuir intervalo até o limite de 1 segundo.

- **Condição:** Caso o intervalo de sensoriamento esteja muito baixo.
Ação: Aumentar intervalo até o limite de 10 segundos.
- **Condição:** Caso o intervalo de disseminação esteja muito alto.
Ação: Diminuir intervalo até o limite de 1 segundo.
- **Condição:** Caso o intervalo de disseminação esteja muito baixo.
Ação: Aumentar intervalo até o limite de 20 segundos.

3.3. Algoritmo de Colônia de Formigas (ACS)

O ACS surgiu em 1992 como fruto da tese do pesquisador italiano Marco Dorigo. Para a instância seguinte, optamos por integrar o ESA ao ACS. Este algoritmo retrata o comportamento forrageiro das formigas. Quando caminha em busca de alimento, uma formiga deixa um rastro pelo caminho, chamado de feromônio. O feromônio é responsável por estabelecer uma comunicação entre as formigas, e cada formiga deposita ao longo do caminho para indicar o quão interessante foi o caminho que ela escolheu. Com o decorrer do tempo, essa substância química passa por um processo chamado evaporação, que é exatamente a perda de feromônio. [Dorigo et al. 2006].

$$\rho_{i,j}^k = \frac{(\tau_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum_{I \in N_i^k} (\tau_{i,j})^\alpha (\eta_{i,j})^\beta}$$

Figura 1. Função de Probabilidade de Transição.

Na instância implementada com o ACS, a heurística foi aplicada na etapa responsável pelo planejamento da melhor estratégia de disseminação a ser adotada pela rede. O algoritmo trabalha sob uma matriz de adjacência que representa a vizinhança factível, ou seja, quais as alternativas de variação nos intervalos de disseminação existem para replanejamento da rotina da rede a partir de determinado intervalo de tempo. Para escolher qual o tempo será mais adequado, o algoritmo trabalha com uma função de probabilidade de transição que pode ser observada na Figura 1, onde:

$\tau_{i,j}$: Representa o feromônio ligado à aresta i,j .

α e β : São valores determinados aleatoriamente que representam a influência do feromônio e da informação heurística.

N_i^k : Representa a vizinhança factível da formiga.

É importante dizer que o feromônio é de natureza aleatória e para o desenvolvimento deste trabalho, ele está configurado para receber valores que variam de 0 a 1, o que pode causar variações de 0% a 100%. Os pesos α e β também são de natureza aleatória e variam entre 0 e 1. A vizinhança factível representa as variações permitidas em intervalos de disseminação, como ilustrado pela Figura 2.

No grafo de exemplo da Figura 2, temos nos vértices o atual intervalo de disseminação, em segundos. As arestas representam a vizinha factível, isto é, quais as possibilidades novas de auto-configuração a partir de um estado, a configuração é passível de

incrementar ou decrementar o tempo da aplicação de acordo com o que seja mais interessante para a configuração da aplicação (e.g., diminuir na ocorrência de eventos e aumentar caso contrário). Eles levam em consideração os dados sensoreados e sua vizinhança para fazer o cálculo de qual estratégia deve ser adotada.

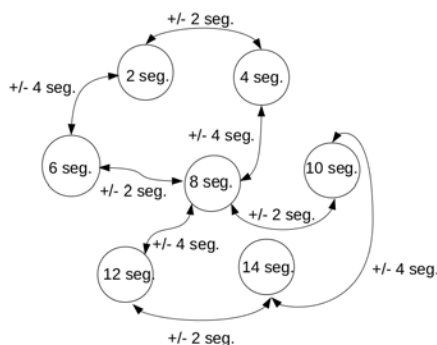


Figura 2. Grafo que ilustra intervalos de disseminação e possibilidades.

Pensando neste contexto do funcionamento básico do algoritmo de colônia de formiga, o integramos na etapa de planejamento do ESA. O algoritmo trabalha sob a ideia de um reforço positivo, que deposita ou evapora feromônios e portanto colocamos essa ideia no contexto do ESA e das RSSFs. A ideia do reforço positivo está diretamente ligada a como serão os intervalos de disseminação na RSSF. O funcionamento para o ESA é estabelecido na ideia de que enquanto o sensoramento retornar dados relevantes, será reforçada determinada estratégia de distribuição de informações sensoreadas com o objetivo de entregar mais detalhes sobre o evento para os interessados, e a partir do momento em que essa rotina torna-se desinteressante, há um reforço negativo, ou seja, evaporação de feromônio, para evitar que dados não tão relevantes sejam transmitidos com muita frequência. Com isso, haverá variações no intervalo em que as mensagens são disseminadas na rede.

3.4. Algoritmo Genético e de Colônia de Formigas

Em um outro cenário, implementamos uma aplicação que trabalha com as duas estratégias apresentadas anteriormente: GA e ACS. As mesmas configurações dos algoritmos foram mantidas. O ponto principal deste cenário é analisar o comportamento de dois algoritmos atuando em conjunto no ESA.

4. Implementação e Resultados

Todos os experimentos foram executados no ambiente real (i.e., *Testbed*) chamado Indriya [Doddavenkatappa et al. 2011], para que os resultados fossem coesos e consistentes. Para apresentar os resultados, foram feitas análises de métricas importantes para a detecção e detalhamento de eventos em uma RSSFs, sendo: quantidade de mensagens recebidas pelo ponto de acesso (PA), quantidade de mensagens enviadas pelos nós sensores ao longo do tempo, e tempo gasto para que a primeira notificação chegue ao PA. Vale destacar que, como os nós sensores do Indriya estão constantemente ligados à uma fonte de energia, não foi possível fazer análises confiáveis e coerentes sobre o consumo de energia nos nós sensores.

4.1. Cenários Implementados

Todas as aplicações foram executadas durante 10 minutos, para a obtenção de dados mais consistentes e concisos. Para todas as implementações, trabalhamos com linguagens de programação C e nesC, além de adotar TinyOS como o sistema operacional utilizado na plataforma de nós sensores.

Um nó telosB, utilizado nos experimentos, possui as seguintes características de hardware e software. O transceptor, utiliza um rádio TPR2420CA que opera em faixa de frequência de 2400 MHz a 2483.5 MHz com capacidade de transmissão de 250 kbps. O alcance está entre 75 a 100 metros para ambientes externos e de 20 a 30 metros para ambientes internos. Possui taxas de consumo de 23mA em modo de recepção, 21 μ A em modo ocioso e 1 μ A para modo sleep. O TelosB utiliza um processador Texas TI MSP430 com frequência de 8MHz e tem disponível os sensores de temperatura, luz, infravermelho e umidade. Em relação à memória, possui 1MB de memória não-volátil, 1024Kb de memória de programa e 10kB de RAM.

Para avaliar os algoritmos propostos para o ESA, foram implementados 5 cenários:

Básico (NAIVE): Não implementa nenhuma estratégia do ESA, sendo utilizada como base de comparação;

Algoritmo Baseado em Regras (ESA_REGRAS): implementa a solução que altera os parâmetros de acordo com regras definidas;

Algoritmo Genético (ESA_GA): implementa a solução que altera os parâmetros de acordo com o algoritmo genético;

Algoritmo de Colônia de Formiga (ESA_ACS): implementa a solução que altera os parâmetros de acordo com o algoritmo genético;

Algoritmos GA e ACS (ESA_GA_ACS): implementa as duas soluções, GA e ACS, em conjunto.

Ao longo dos experimentos, utilizamos 105 nós sensores. A aplicação implementada para todos os cenários coleta periodicamente (de 1 a 10 segundos) a temperatura do ambiente, e dissemina, também periodicamente (de 1 a 20 segundos), os valores coletados após a última disseminação. O intervalo de tempo de sensoriamento e disseminação é fixo em 5 segundos apenas para o cenário chamado básico (*NAIVE*). Para os demais cenários, esses valores são iniciados em 5 segundos, mas variam ao longo do tempo de acordo com a ocorrência de eventos na rede.

Para tornar possível avaliar a atuação dos cenários durante ocorrência de eventos, foram gerados eventos virtuais aleatoriamente na rede, fazendo com que aproximadamente 10 nós sensores "monitorem" valores bem superiores de temperatura, indicando, por exemplo, a ocorrência de um incêndio. Essa estratégia se fez necessária pois no *Test-bed* utilizado em um ambiente fechado, as temperaturas não variam significativamente. A Figura 3(a) representa os instantes de tempo que ocorreram eventos para os 5 cenários. Essa figura será referenciada na análise dos resultados para que seja possível verificar o momento de atuação dos ESAs implementados.

4.2. Resultados

Nessa seção discute-se algumas análises obtidas pelos dados do arquivo de saída fornecidos pelo *Testbed*.

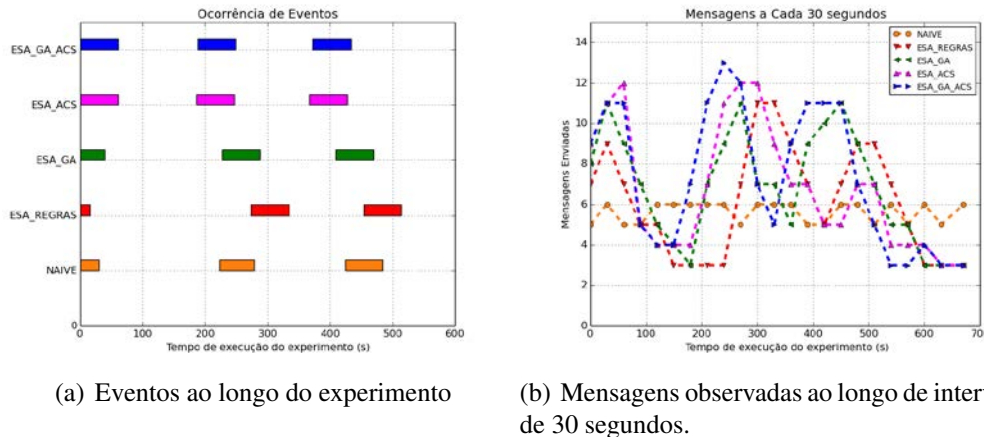


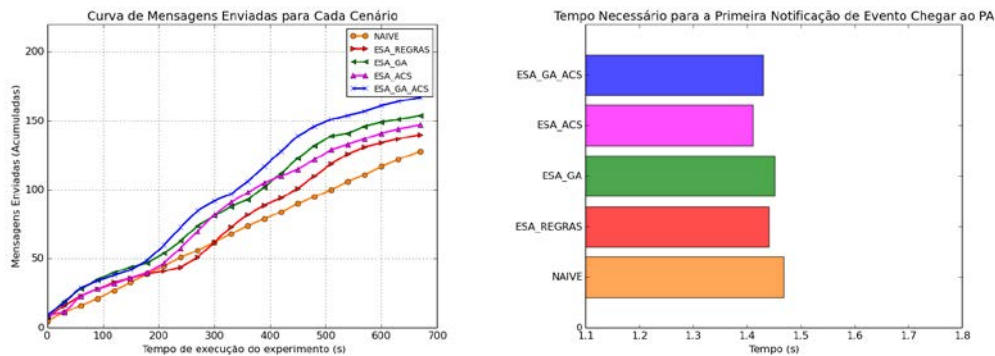
Figura 3. Eventos Virtuais e Mensagens ao Longo do Tempo.

A Figura 3(a) representa os eventos virtuais que aconteceram ao longo da execução dos experimentos para todos os cenários. Consideramos que por aspectos naturais de redes de computadores, como assincronicidade e paralelismo, não foi possível fazer com que os eventos fossem rigorosamente idênticos. Mesmo assim, foi possível observar o funcionamento do ESA nos cenários propostos e implementados, uma vez que ocorreram algumas variações bruscas nas temperaturas coletadas e analisadas.

A Figura 3(b) ilustra a quantidade de mensagens enviadas por todos os nós sensores dentro de um intervalo de 30 segundos. Em outras palavras, esses valores indicam que para cada intervalo de 30 segundos, a quantidade de mensagens que cada cenário enviou. Ao analisarmos em conjunto as figuras 3(a) e 3(b), é possível notar que os intervalos dos eventos gerados estão diretamente relacionados aos intervalos de maior envio de mensagens para cada cenário. Um detalhe que deve ser ressaltado é que a aplicação *NAIVE* manteve-se com uma quantidade de envios aproximadamente constante durante todo o experimento. Por outro lado, as implementações do ESA alteraram os intervalos de sensoriamento e disseminação durante a ocorrência dos eventos, elevando assim a quantidade de mensagens enviadas e, conseqüentemente, os detalhes dos eventos enviados aos responsáveis. Em geral, o cenário *ESA_GA_ACS* se mostrou mais atuante que os demais, enviando mais mensagens quando necessário. O cenário *ESA_REGRAS*, por outro lado, se mostrou menos atuante, enviando menos detalhes sobre o evento. Os cenários *ESA_GA* e *ESA_ACS* apresentaram resultados intermediários. Como observação geral, ficou caracterizado que as implementações do ESA levaram a uma melhor monitoração dos eventos, e que algoritmos elaborados são importantes para que a auto-configuração seja efetiva.

A Figura 4(a) apresenta as curvas de mensagens acumuladas por cada cenário ao longo dos 10 minutos de execução dos experimentos. É possível observar variações nas inclinações das curvas, e relacionar esses fatos diretamente aos eventos causados pelos sensores virtuais (veja Figura 3(a)). Praticamente em sincronia com o evento, há um aumento gradual do número de mensagens recebidas em tal intervalo de tempo, com

exceção do cenário *NAIVE*, em que a inclinação continua constante ao longo do tempo. Vale destacar também que o cenário *ESA_GA_ACS* levou a um maior número de mensagens, notificando em mais detalhes a ocorrência dos eventos ao PA. Por outro lado, o cenário *ESA_REGRAS* enviou um número menor de mensagens, enquanto os cenários *ESA_GA* e *ESA_ACS* transmitiram valores intermediários. Mais uma vez, vale destacar a importância de algoritmos mais elaborados, que são mais sensíveis à detecção e disseminação dos eventos, enviando assim mais informações sobre os mesmos. Também é importante comentar que, ao final dos eventos, é notada uma redução na inclinação das curvas, mostrando que os cenários que implementam o ESA se readaptaram com o fim dos eventos.



(a) Mensagens acumuladas durante o experimento. (b) Tempo médio de notificação de uma mensagem ao PA.

Figura 4. Total de mensagens enviadas (acumulado) e Tempo médio de notificação do evento no PA.

Por fim, foi feita uma análise do tempo necessário para que a primeira notificação da ocorrência de um evento chegue ao PA da rede. O gráfico apresentado na Figura 4(b) mostra um tempo médio, com base em 3 rodadas de experimentos, que pode ser observado para cada um dos cenários. Para essa métrica, o cenário *ESA_ACS* se mostrou mais eficiente, entregando a primeira notificação do evento mais rapidamente que as outras estratégias. A estratégia mais elaborada *ESA_GA_ACS* obteve o segundo melhor resultado, seguida pelas estratégias *ESA_REGRAS*, *ESA_GA* e *NAIVE*. Como nos resultados anteriores, essa métrica demonstra a importância de se adotar um ESA em RSSFs com algoritmos elaborados.

5. Conclusão e Trabalhos Futuros

Neste trabalho, implementamos e avaliamos novas instâncias de um ESA, com o objetivo de monitorar eventos em uma RSSF. As soluções foram implementadas em um ambiente real com mais de cem nós sensores, que tornaram os resultados coesos e consistentes. Foram avaliadas métricas que indicam a qualidade da monitoração durante a ocorrência de evento na rede. Em geral, pode-se concluir que as implementações de algoritmos elaborados para o elemento sensor autônomo levam a uma monitoração mais rápida e detalhada dos eventos. Como trabalhos futuros, esperamos implementar novas instâncias com base no ESA genérico. Essas novas instâncias podem trazer conjuntos de algoritmos das mais diversas áreas que são relevantes na literatura e que podem causar impacto positivo na RSSF. Também pretendemos avaliar outras configurações de experimentos.

6. Agradecimento

Este trabalho contou com o apoio do CNPq (573.738/2008-4 INCT NAMITEC) e CAPES.

Referências

- Abdelaal, M. (2015). **Distributed Techniques for Energy Conservation in Wireless Sensor Networks**. In *DcSensorNets*.
- Braga, T. R., Silva, F. A., Ruiz, L. B., Nogueira, J. M. S., and Loureiro, A. A. (2006). **Design and Evaluation of an Autonomic Sensor Element**. In *1st Latin American Autonomic Computing Symposium, LAACS*.
- Braga, T. R. d. M. (2006). **Um Elemento Sensor Autônomo Para Redes De Sensores Sem Fio**. In *Dissertação de Mestrado, Departamento de Ciência da Computação da Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil*.
- Buratti, C., Conti, A., Dardari, D., and Verdone, R. (2009). **An overview on wireless sensor networks technology and evolution**. volume 9, pages 6869–6896. Molecular Diversity Preservation International.
- Doddavenkatappa, M., Chan, M. C., and Ananda, A. L. (2011). **Indriya: A low-cost, 3D wireless sensor network testbed**. In *Testbeds and Research Infrastructure. Development of Networks and Communities*, pages 302–316. Springer.
- Dorigo, M., Birattari, M., and Stützle, T. (2006). **Ant colony optimization**. In *Computational Intelligence Magazine, IEEE*, volume 1, pages 28–39. IEEE.
- Jabeur, N., Sahli, N., Zeadally, S., and Yasar, A.-U.-H. (2015). **Enabling Cyber Physical Systems with Wireless Sensor Networking Technologies, Multiagent System Paradigm, and Natural Ecosystems**. volume 2015. Hindawi Publishing Corporation.
- Lakshmi, C. B. and Mohan Rao, S. (2014). **Bio-inspired self-healing routing to improve lifetime of wireless sensor networks**. In *Communication and Network Technologies (ICCNT), 2014 International Conference on*, pages 134–138.
- Nguyen, T.-T., Shieh, C.-S., Horng, M.-F., and Dao, T.-K. (2014). **A Genetic Algorithm with Self-Configuration Chromosome for the Optimization of Wireless Sensor Networks**. In *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*.
- Peng, M., Xu, K., Yu, Q., Jiang, W., Leng, S., and Mao, Y. (2014). **Cellular automata self-organization algorithm for Wireless Sensor Network**. In *Communication Problem-Solving (ICCP), 2014 IEEE International Conference on*.
- Rawat, P., Singh, K. D., Chaouchi, H., and Bonnin, J. M. (2014). **Wireless sensor networks: a survey on recent developments and potential synergies**. volume 68, pages 1–48. Springer.
- Sadouq, Z., El Mabrouk, M., and Essaaidi, M. (2014). **Conserving energy in WSN through clustering and power control**. In *Information Science and Technology (CIST), 2014 Third IEEE International Colloquium in*.
- Whitley, D. (1994). **A genetic algorithm tutorial**. In *Statistics and computing*, volume 4, pages 65–85. Springer.