

Avaliando Novos Algoritmos para os Módulos de Análise e Planejamento do Elemento Sensor Autônomo

Alexandra da Silva Pereira, Thais Regina de Moura Braga Silva

¹Instituto de Ciências Exatas – Universidade Federal de Viçosa - *Campus Florestal* (UFV)
Rodovia LMG, 818 - km 6
35690-000 - Florestal, MG - Brasil

alexandra.pereira@ufv.br, thais.braga@ufv.br

Abstract. *Wireless Sensor Networks are used in different contexts, including monitoring of climatic aspects in a remote region, actuators for geographic and spatial events as well as applications in military or medical area. However, this particular type of network has a limited power capacity and this directly impacts its processing capacity and the amount of available memory. Based on this limitation scenario a generic model that seeks to use a more balanced way the computational resources of the network 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 and prolong its service life. 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 and the number of packets transferred across the network.*

Resumo. *Redes De Sensores Sem Fio são utilizadas em diversos contextos, que podem incluir monitoração de aspectos climáticos em uma região remota, atuadores para eventos geográficos e espaciais, além de aplicações em área militar ou médica. Entretanto, esse tipo específico de rede possui uma capacidade limitada de energia e isso impacta diretamente sua capacidade de processamento e na quantidade de memória disponível. Diante do cenário de limitação foi desenvolvido um modelo genérico que busca utilizar de maneira mais equilibrada os recursos computacionais da rede, 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 e prolongar sua vida útil. Implementamos dois algoritmos mais sofisticados e analisamos seu comportamento em experimentos executados em redes de sensores reais, mais conhecidos como testbeds. Os resultados mostram que as implementações mantêm ou melhoram aspectos importantes como o número de mensagens enviadas e a quantidade de pacotes trafegados pela rede.*

1. Introdução

As Redes de Sensores Sem Fio(RSSF) são um tipo específico de redes de computadores. Classificadas com rede ad-hoc, elas são formadas por milhares de dispositivos chamados de nós sensores. Para formação de um RSSF, os nós são disseminados em um ambiente

remoto, isolado, ou de difícil acesso a seres humanos, com a finalidade de monitorar e coletar dados relacionados ao ambiente envolvido na modelagem da rede. Os nós sensores que compõem a rede são conhecidos por possuírem grandes restrições de hardware (i.e., processamento, memória e energia).

A literatura já apresenta diversas propostas na computação autônoma (CA) com a finalidade de tornar a rede autogerenciável. Sendo assim, a rede terá maior controle sobre seus próprios recursos, com a capacidade de prover serviços (e.g., autoconfiguração, auto-organização). Esses aspectos tornam a vida útil da rede mais extensa, mesmo com menos interferência de seres humanos para atividade de manutenção ou gerência. Neste cenário de busca do gerenciamento inteligente de RSSFs, a IBM propôs o uso de elementos autônicos. Isso motivou o desenvolvimento de um modelo genérico por [Braga 2006], chamado elemento sensor autônomo (ESA).

Com conhecimento das principais restrições e de muitos trabalhos propostos na literatura, adotamos ESA como módulo para referência de implementação. O consideramos interessante pela sua estrutura genérica, onde podemos 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 RSSFs 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 do mesmo.

Neste trabalho buscamos ambientes reais, onde poderíamos romper as barreiras de ambientes de simulação adotadas massivamente pelos trabalhos da literatura. Com a utilização de testbeds, que são os ambientes reais para RSSF foi possível realizar testes de maior rigor e transparência para modelos propostos em teorias científicas. Através da utilização dessa ferramenta é possível executar as implementações e fazer melhores análises do comportamento do ESA em um ambiente real, de maneira remota, pela Internet.

Diante do cenário de restrições em RSSFs e com um modelo autônomo base bem elaborado, nos sentimos instigados a observar o comportamento de novas implementações do ESA. Isto porque a implementação básica do mesmo apresentou resultados relevantes no funcionamento da RSSF. No presente trabalho apresentamos duas novas instâncias para se integrarem aos serviços providos pelo ESA. As duas novas propostas são baseadas em heurísticas. Na primeira delas incluímos um algoritmo genético (GA) com a finalidade de diversificar o cenário presente na base de conhecimento da aplicação e causar impacto na etapa de análise. A segunda heurística adotada é um ant colony system (ACS) que busca intensificar o planejamento de melhores estratégias de execução para RSSF. Para os dois algoritmos escolhidos, foram feitas análises e levantamento do quanto eles podem impactar a RSSF.

O presente trabalho segue organizado da seguinte forma: a seção 2 descreve o elemento autônomo (EA) e seus conceitos, além de apresentar o ESA e uma breve descrição dos serviços providos por ele em cada etapa do seu ciclo. Na seção 3 há uma apresentação de trabalhos presentes na literatura pertinente ao tema de pesquisa desenvolvido. A seção 4 apresenta os cenários escolhidos, bem como, os algoritmos implementados em cada nova instância do ESA. A seção 5 discute os cenários de cada uma das implementações, quais foram os resultados e como estes resultados podem impactar o

funcionamento da RSSF. Na seção 6 apresentamos, finalmente, expectativas de trabalhos futuros e algumas considerações finais.

2. Elemento Sensor Autônômico

Dificuldades no desenvolvimento de aplicações RSSFs estão relacionadas à capacidade limitada de recursos computacionais das plataformas de nós sensores. Isso sugere que boas modelagens devem adotadas, uma vez que isso causará impacto durante a execução, e na utilização dos recursos da rede. A sábia utilização dos recursos computacionais pode estender de modo considerável o tempo de vida da RSSF, além de evitar o gerenciamento humano. Pensando nisso, em 2005 a IBM propôs um modelo genérico de EA. O elemento autônômico foi definido com a menor parte de um ambiente autônômico, ele funciona como sistema individual que contém recursos e será capaz de prover serviços para humanos ou para outros possíveis serviços. Inspirado no modelo de EA apresentado pela IBM, foi proposto o ESA, que trabalha sob a ideia de um modelo genérico [Braga 2006] proposto para RSSF.

2.1. Computação Autônômica e Elemento Autônômico

A computação autônômica tem como base conceitos que transformam sistemas computacionais em sistemas auto gerenciáveis. O que os caracteriza como capazes de gerenciar a si próprio com mínima ou nenhuma intervenção humana [Kephart and Chess 2003]. Esse conceito foi desenvolvido com base no funcionamento do sistema nervoso autônomo dos seres humanos. No sistema nervoso há um auto gerenciamento responsável por garantir a estabilidade, e o funcionamento de nossas funções corporais e vitais.

Com base no sistema nervoso são desenvolvidos sistemas computacionais autônômicos. Esses são utilizados para permitir que a rede seja auto gerenciável em diversos contextos de aplicação. O auto gerenciamento é a base da teoria de toda computação autônômica e seu principal intuito é a flexibilização do uso do administrador da rede. Em sistemas autônômicos, o seu foco deixa de ser integralmente com operação e manutenção do sistema e é transferido para tarefas de supervisão e o controle do negócio.

No cenário das RSSF surgiram propostas de modelos e aplicações, onde a rede seja capaz de auto gerenciar seus elementos. A rede também é capaz de auto gerenciar as conexões existentes entre os elementos, além de seus próprios recursos. A ideia principal do EA é possibilitar a independência de um gerente humano. De maneira autônômica dados podem ser armazenados para serem utilizados posteriormente. Este histórico de dados pode garantir um nível de aprendizado de máquina, que é formado a partir de ações executadas no ciclo autônômico.

O comportamento modular e genérico do EA, além de sua capacidade de prover recursos para outros módulos foi o ponto de partida para o modelo genérico do ESA.

2.2. Elemento Sensor Autônômico

O ESA utilizado como base para este trabalho é composto por módulo autônômicos, ou EAs. Para que o ESA funcione de maneira interessante, os módulos trabalham em um contexto de ciclo. O ciclo proposto engloba os seguintes EAs: monitoramento, análise, planejamento e execução[Braga 2006]. A seguir apresentaremos com maiores detalhes o

ESA, seu ciclo de funcionamento, bem como, uma breve discussão do comportamento dos módulos que foram propostos pelo modelo genérico de ESA [Braga 2006].

• Serviço de Monitoração

O serviço de monitoração pode ser projetado de diversas maneiras e depende diretamente do sistema autonômico ao qual ele irá pertencer. Esse serviço monitora parâmetros que em seguida podem ser combinados e armazenados na base de conhecimento. Em RSSFs o contexto das aplicações está fortemente ligado aos recursos envolvidos, isto é, há uma grande limitação de recursos computacionais(i.e., processamento, memória e energia). Logo, aqui também devemos nos preocupar em fornecer um serviço cuidadosamente planejado e gerenciado.

Os contextos para uma RSSF são diversos. Cada um dos contextos exige parâmetros de monitoramento diferentes, cada contexto é modelado independente de um contexto previamente modelado. Por exemplo, um conjunto de parâmetros que é importante para o monitoramento de um RSSF proposta para acompanhar variações da quantidade de uma substância no corpo humano é bastante diferente do conjunto de parâmetros para uma RSSF responsável por ativar um serviço caso exista presença de fumaça em um ambiente.

A responsabilidade do serviço de monitoração é de lidar com determinados parâmetros dentro de um ESA, geralmente dados referentes a aplicação (i.e., dados sensoreados). O interessante é que este seja também seja um EA bem projetado. Monitoramento e sensoreamento de dados irrelevantes consomem quantidades consideráveis de energia e podem, a longo prazo, diminuir a vida útil da rede.

• Serviço de Análise

É responsável por verificar contextos internos e externos pertinentes a um elemento autonômico. Através dos dados de entrada, que são obtidos pelo serviço de monitoração, o serviço de análise é capaz de transformar os dados em informações úteis. Essas informações pré-processadas serão utilizadas pelas etapas seguintes do ciclo. Elas podem representar aspectos importantes no contexto de uma aplicação, além de permitir concluir sobre aspectos de desempenho, produtividade e do cumprimento dos níveis de qualidade ligados a um contexto específico de RSSF.

• Serviço de Planejamento

O serviço de planejamento recebe os dados pré-processados pelo módulo de análise. Com esses dados o algoritmo é capaz de elaborar uma estratégia ou um novo plano de ações para serem adotadas a cada novo ciclo de execução do EA. O contexto de planejamento de uma estratégia está intimamente ligado ao modelo desenvolvido para a RSSF, são tarefas comuns adotar novos tempos de disseminação e sensoreamento.

• Serviço de Execução

O serviço de execução relaciona-se de maneira direta com a autoconfiguração da rotina de funcionamento da RSSF. Através de aspectos observados nas fases de monitoração e análise do ESA é possível atuar sobre o hardware ou software do elemento gerenciado em questão. Quando é necessário atuar sob o software da rede, reprogramações são feitas, sejam elas em intervalos de sensoreamento ou disseminação.

O objetivo é buscar a reprogramação da rede de maneira adequada e constante, com a finalidade de obter e manter apenas os dados relevantes. Concluímos que esse módulo apresentará melhora nas configurações relacionadas a necessidades que surgem a cada execução do ciclo, e ainda às condições do ambiente ligado ao contexto da RSSF.

- **Base de conhecimento**

A base de conhecimento (BC) é responsável por armazenar dados, políticas, limites, dentre outros [Braga 2006]. Como já dito, os nós sensores possuem limitada capacidade de armazenamento e de bateria, logo todas as operações de leitura e escrita na memória devem ser bem planejadas. Para o ESA é interessante que sejam salvos apenas dados que se apresentem relevantes ou foram considerados necessários em tempo de execução. Com isso é possível manter um determinado equilíbrio entre consumo de memória e de energia. As escritas são feitas apenas para dados que são relevantes para todo o ciclo autônomo. O restante dos dados é descartado por não ter características incomuns a dados já presentes na BC.

A seguir apresentamos a figura 1, que ilustra todo processo descrito até aqui. Nela é possível ver cada umas das etapas citadas e discutidas nesta seção, assim como ver qual é a responsabilidade de cada módulo que compõe o ESA.

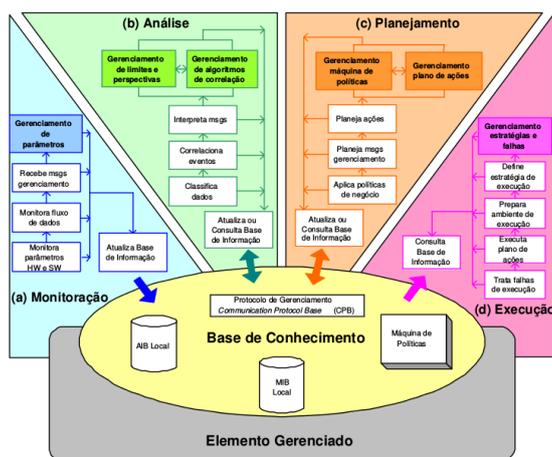


Figura 1. O ESA em detalhes.

O ESA já foi desenvolvido e testado em ambientes de simulação. As execuções em simulação apresentaram resultados que causaram um impacto positivo na RSSF. A característica mais importante do modelo de ESA desenvolvido é a sua natureza generalista, que nos dá a liberdade de testar algoritmos da literatura em diversos módulos e para diversos modelos de RSSF. O problema de tamanha liberdade é a escolha adequada dos algoritmos e quão grande será o impacto destes no consumo da RSSF, uma vez que algoritmos mais robustos e com maior carga de processamento podem ser escolhidos. Essa será a estratégia desse trabalho: analisar o impacto de dois algoritmos mais elaborado nas etapas de monitoração e planejamento em instâncias que tem como base o ESA genérico.

3. Trabalhos Relacionados

[Peng et al. 2014] discutem um modelo que utiliza um automato celular para trabalhar na linha de auto organização da rede. Para isso um algoritmo possui rotinas que mantêm

a cobertura da rede estável e faz com que exista considerável economia de energia na rede. O automato proposto funciona na forma de um grid de dimensões x e y , onde estão dispostos os nós, para o artigo os experimentos foram realizados com $x=y=100$ e 5000 nós. O experimentos para validação foram feitos em simulação com uma rede de sensores homogênea.

Já [Lakshmi and Mohan Rao 2014] propõem um algoritmo com a finalidade de proporcionar autocura baseada em uma técnica de otimização de exame. A ideia é entender o tempo de vida de uma rede de sensores. Neste trabalho foi desenvolvido um algoritmo baseado em colônia de formigas, onde os pacotes buscam por caminhos mais curtos até o ponto de acesso. Foi possível observar que os nós situados em regiões centrais da rede possuem maior nível de energia residual. Vários cenários de simulação foram observados e testados em ambiente NS2, como esperado o atraso aumentou a medida que o número de nós da rede aumentou, mas houve economia de energia como esperado.

No trabalho proposto por [Jabeur and Sahli 2015] propõe modelos para redes de sensores sem fio baseadas em aspectos da natureza e são chamados de bioinspirados. Nele a rede formada pelos nós sensores é descrita como análogo a um ecossistema. Conforme este ecossistema passa por metamorfoses a rede evolui e essa evolução resolve problemas que são recorrentes em RSSF. Dos modelos definidos podemos citar, a etapa da RSSF como um ecossistema e os sensores como organismos vivos. Após essa etapa um mapa de características de organismos vivos para redes de sensores sem fio, por exemplo, excreção como eliminação de dados que não mais interessantes para a rede, nutrição como a obtenção de dados, dentre outros aspectos. Com o modelo definido uma nova arquitetura é apresentada para RSSF, entretanto não existem ainda implementações concretas para este trabalho. Toda o modelo proposto mostrou-se bastante interessante e pode trazer melhorias para aplicações em redes de sensores sem fio.

Os autores [Sadouq et al. 2014] fazem a proposta de um framework que trabalha com a rede de sensores como um cluster, onde vários nós formam um cluster. Os nós possuem 3 funções pré definidas em cada cluster (i.e., sensorear, rotear dados e ponto de acesso). O principal objetivo era ter uma rede mais sistêmica, robusta e escalável. A topologia do modelo foi apresentada como um grid onde vários nós são divididos em grupos. Em cada um desses grupos nós existe um nó mestre, responsável por rotear essa informação entre as redes, vários nós escravos, responsáveis por sensorear dados. Apenas os mestres podem enviar dados a um ponto de acesso, o que eles chamam de comunicação externa. Como apenas um nós faz essa tarefa, isso já proporciona uma grande economia de energia, fazendo com que menos nós enviem mensagens, uma vez que a maioria dos nós se comunica com o mestre e ele trata de repassar mensagens ao ponto de acesso. Testes do modelo foram implementados em ambiente NS2.

Na proposta discutida por [Abdelaal 2014], um modelo foi desenvolvido com o intuito de melhorar as técnicas existentes em RSSF para economizar ainda mais energia. Os três principais objetivos são: agregar dados para o sensoriamento tornar-se mais barato, otimização de hardware e autoadaptações preditivas. A primeira técnica, que visa a compreensão de dados explorou algoritmos relacionados a transformações Fuzzy, criando o chamado FuzzyCAT, responsável por fazer análises dos dados de temperatura coletados. Resultados apresentados pelo trabalho dizem que a etapa de compressão conseguiu consumir 58 vezes menos. Na etapa de otimização de hardware um refinamento foi feito

no padrão IEEE 802.15.4 que está na camada física. Já as autoadaptações preventivas aplicadas a aplicação para que a execução seja em conformidade com comportamentos observados ao longo do funcionamento da rede.

[Nguyen et al. 2014] propuseram um algoritmo genético para trabalhar na RSSF. O objetivo era obter um algoritmo responsável pela autoconfiguração e otimização da rede. A rede é definida como um cluster, onde 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. Com isso há uma variação entre quem se comunica com o ponto de acesso e quais são os clusters-head, esse mecanismo faz com que a rede tenha um período de vida maior. Os experimentos para validação foram feitos em simulação como um número de nós entre 100 e 400, os resultados obtidos elevam cerca de 30% da vida útil da rede.

O trabalho proposto neste artigo difere dos citados aqui 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 autocura, autocontrole e autoconfiguração. Partindo de uma implementação fiel ao modelo proposto por [Braga 2006] de ESA, criamos 2 novas instâncias. Essas instâncias utilizam algoritmos mais sofisticados, que atuam nos EAs de monitoração e de planejamento que compõem o ESA. A expectativa é compensar o consumo de energia para executar a rotina de instruções dos algoritmos com melhoras consideráveis em número de pacotes trafegados na rede, relevância dos dados sensoreados e número de mensagens disseminadas.

4. Novas Instâncias para o ESA

O trabalho proposto para este artigo é a investigação do comportamento de algoritmos em novas instâncias do ESA. O ESA é composto por módulos autônomos, que são a menor parte de um ambiente autônomo. No funcionamento do ESA aplicamos a ideia de ciclo autônomo, onde existem módulos autônomos específicos para monitoração, análise, planejamento e execução. As etapas citadas são responsáveis respectivamente, por, monitorar modificações e anormalidades no ambiente, analisar os dados caso existam modificações, planejar as ações diante de ocorrências das duas etapas anteriores e execução das operações necessárias e já planejadas. Esses algoritmos, em conjunto com o ESA, podem melhorar significativamente a utilização dos recursos de energia e memória em uma RSSF. Todos os experimentos foram executados em testbeds, ferramentas com capacidade de viabilizar a execução de aplicações em ambientes reais. Existe uma RSSF real onde podem ser executados experimentos remotamente via conexão com a internet. Através dos dados gerados, métricas importantes no contexto das RSSF serão analisadas para comparações entre os comportamentos das novas instâncias.

As instâncias desenvolvidas ao longo deste trabalho foram focadas em autoconfiguração, porque até o momento todo conteúdo referente a este assunto encontrado na literatura e as propostas já apresentadas e implementadas para o ESA tem tido a autoconfiguração como estratégia principal. A aplicação possui a finalidade de monitorar a temperatura na região do testbed e ao executar instâncias do ESA. A autoconfiguração se encarrega de re-programar a rede nos momentos onde mais existem variações na grandeza sensoreada. Buscamos dois algoritmos que pudessem contribuir de maneira positiva para

a auto configuração da rede. Logo, poderemos fazer comparações entre as instâncias mais básicas do ESA e as duas novas instâncias desenvolvidas com algoritmos mais complexos e que exigem maior processamento.

Nesta fase do trabalho implementamos dois algoritmos diferentes para trabalhar em conjunto com o ESA. Um algoritmo é baseado em um algoritmo genético(GA), e outro em técnica de enxame, o ant colony system(ACS). Tais algoritmos foram escolhidos por apresentarem natureza aleatória. O esperado é que a cada execução isso pudesse contribuir e ajudar o ESA na sua tarefa de auto configuração buscando valores não explorados, ou ainda, aceitando configurações menos ideais no contexto geral para obter melhores estratégias em auto configurações seguintes.

Nas duas implementações, os algoritmos foram aplicados a EAs diferentes. O GA trabalha nas etapas de monitoração e análise de forma mais significativa. O indivíduos reais são gerados para povoar 50 % da BC, com o objetivo de diversificar as amostras existentes, portanto a função objetivo busca os indivíduos com maior desvio padrão. Na instância seguinte o ACS trabalha de maneira intensa nas etapas de análise e processamento do ESA. Este algoritmo busca caminhos para os quais seja mais interessante auto configurar a rede.

4.1. Algoritmo Genético e o ESA

Os algoritmos genéticos utilizam conceitos provenientes do princípio de seleção natural para abordar uma série ampla de problemas, em especial de otimização. Robustos, genéricos e facilmente adaptáveis, consistem de uma técnica amplamente estudada e utilizada em diversas áreas [Lucas 2002]. Principalmente por ser um algoritmo genérico e adaptável, o escolhemos para uma implementação no contexto do elemento sensor autônomo. Na instância implementada em conjunto do algoritmo genético temos todas as etapas que caracterizam com um algoritmo genético: avaliação, seleção, cruzamento, mutação, atualização e finalização.

O algoritmo genético foi desenvolvido com base nas teorias de evolução dos pesquisadores naturalistas Charles Darwin e Alfred Russel Wallace. Darwin e Charles Babbage, um dos fundadores da computação, eram amigos e desde o início do desenvolvimento existia uma desconfiança da ligação entre biologia e computação. Para surpresa de ambos a máquina analítica desenvolvida por Babbage confirmou a suspeita de ligação entre as áreas, o que os deixou bastante orgulhosos e felizes. Entretanto, esses fatos tiveram maior impacto na comunidade após os anos 1900, quando pesquisadores de genética moderna criaram um o princípio básico de Genética Populacional onde a variabilidade entre indivíduos de uma população é determinada por sua reprodução sexuada, onde ocorre mutação e recombinação genética. Este princípio foi desenvolvido durante os anos 30 e 40, por biólogos e matemáticos de importantes centros de pesquisa. Nos anos 50 e 60, muitos biólogos começaram a desenvolver simulações computacionais de sistemas genéticos. Entretanto, foi John Holland quem começou, seriamente, a desenvolver as primeiras pesquisas no tema. Holland foi gradualmente refinando suas idéias e em 1975 publicou o seu livro *Adaptation in Natural and Artificial Systems*, hoje considerado a Bíblia de Algoritmos Genéticos. Desde então, este algoritmo vêm sendo aplicado com sucesso nos mais diversos problemas de otimização e aprendizado de máquina [Carvalho 2009].

A instância do ESA com o algoritmo genético se concentra no EA responsável por

monitoração e na BC. Para que a aplicação tenha uma perspectiva de análise mais interessante para planejamento e execução, o algoritmo genético preenche 50 % da BC com indivíduos reais. A característica completamente aleatória do algoritmo busca gerações de dados com maior desvio padrão para povoar a BC.

A figura 2 apresenta um pseudo código do AG implementado, onde a população é composta por variáveis que representam temperaturas reais. Cada indivíduo possui um conjunto de temperaturas, na etapa de seleção a estratégia adotada é manter indivíduos com que possuem o maior desvio padrão. A técnica de cruzamento escolhida é a conhecida como cruzamento com um ponto de corte, onde um novo indivíduo recebe material genético de um dos pais até o ponto de corte e do outro pai do ponto de corte em diante. Como taxa para mutação é aplicada uma perturbação de 1.5 % a cada indivíduo, que após a operação a operação terá valor igual a 98.5 % do valor original.

```
ALGORITMOGENETICO()
1  t = 0
2  criar população P(t);
3  for cada indivíduo i de P(t)
4      avaliar aptidão individuo(i)
5  Fim-para
6  while condição de parada não satisfeita
7      t := t + 1;
8      Selecionar população P(t) de P(t - 1);
9      Aplicar operadores de cruzamento sobre P(t);
10     Aplicar operadores de mutação sobre P(t);
11     Avaliar P(t);
12 Fim-enquanto
```

Figura 2. Pseudo-código GA.

O limite de variação para geração de indivíduos foi determinado com base em dados sensoreados no testbed e ficou definido em 15°C, que varre toda a variação de temperaturas observadas em prévios experimentos. As temperaturas são geradas pelas etapas do GA e seguem como parte da BC até o fim do experimento. Os dados gerados pelo algoritmo são utilizados ao longo de toda execução, em conjunto com as temperaturas sensoreadas e armazenadas na BC, para que o ESA prossiga com as etapas de análise, planejamento e execução.

4.2. Ant Colony System e o ESA

O ACS surgiu em 1992 fruto da tese do pesquisador italiano Marco Dorigo. Para a instância seguinte optamos por integrar o ESA a um algoritmo de colônia de formigas. Este algoritmo é uma meta-heurística, que retrata o comportamento forrageiro das formigas. Quando caminha em busca de alimento, uma formiga deixa um rastro pelo caminho, chamado de feromônio. Na possibilidade de seguir por vários caminhos, uma formiga pode, para decidir entre eles, considerar a quantidade de feromônio depositado anteriormente por outras formigas em cada um dos caminhos. Um caminho com muito feromônio é mais atrativo que outro com menos. Com o decorrer do tempo, essa substância química sofre evaporação. O que se nota na natureza é que, após um determinado tempo, a maioria das formigas decide trilhar um caminho mais curto entre o seu ninho e a fonte de alimento de acordo com [Becceneri et al. 2010].

Na instância implementada com o ACS, a heurística foi aplicada a etapa no EA responsável pelo planejamento da melhor estratégia a ser adotada pela rede. O algoritmo trabalha sob uma matriz de adjacência que representa quais as alternativas de tempo para disseminação existem para re-planejamento da rotina da rede. Para escolher qual o tempo será mais adequado, o algoritmo trabalha com uma função de probabilidade de transição que pode ser vista na figura 3.

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

Figura 3. Função de Probabilidade de Transição.

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

Pensando neste contexto do funcionamento básico do algoritmo de colônia de formiga, o integramos a na etapa de planejamento do ESA. O algoritmo trabalha principalmente sob a ideia de um reforço positivo, portanto colocamos essa ideia no contexto do ESA e da RSSFs, para que houvesse um reforço positivo sempre que a aplicação notar que algo relevante deve ser coletado. A ideia do reforço positivo está diretamente ligada a como serão os intervalos de disseminação na RSSFs. O funcionamento para o ESA é estabelecido na ideia de que enquanto o sensoramento retornar dados relevantes, será reforçada determinada estratégia, e a partir do momento em que essa rotina torna-se desinteressante há um reforço negativo, ou seja, evaporação de feromônio, para evitar que estratégias que não são mais interessantes continuem sendo adotadas.

5. Implementação e Resultados

Com as instâncias implementadas, todos os experimentos foram executados em um ambiente real, chamado testbed. Para apresentar os resultados, foram feitas análises de métricas importantes no contexto de RSSFs. As métricas coletadas neste trabalho foram: consumo de energia, atraso, jitter, número de mensagens enviadas, número de mensagens perdidas, número de pacotes enviados e número de pacotes perdidos.

5.1. Indriya Testbed

O Indriya é um testbed que está localizado na escola de computação da Universidade Nacional de Singapura(NUS), em Singapura, na Ásia. Sua topologia é homogênea, composta apenas por nós sensores da família TelosB. Este testbed foi aberto a comunidade científica em dezembro de 2009, a partir de quando foi possível solicitar contas. Assim como a maioria dos testbeds citados, ele possui como objetivo pesquisas em ambientes RSSF, protocolos de comunicação, design de sistemas e aplicações. O Indriya é um testbed que possui topologia homogênea. A rede é formada por 139 nós TelosB distribuídos ao longo de 3 andares da NUS. No prédio de ciência da computação os nós sensores estão distribuídos em entradas de salas de seminário, auditórios, laboratórios e também nos corredores como é possível ver na figura Z.

5.2. Cenários Implementados

Todas as aplicações foram executadas durante 15 minutos, com pelo menos 10 execuções, para obtenção de dados mais consistentes e concisos. A topologia da rede é a definida por padrão na plataforma do testbed, e é homogênea. A plataforma de nó sensor utilizada neste trabalho é o TelosB, inicialmente desenvolvida por um grupo de pesquisadores da UC Berkeley (Universidade da Califórnia em Berkeley) e atualmente é fabricada e comercializada pela empresa Memsic. Além da família TelosB, hoje a Memsic é fabricante de algumas das mais importantes famílias de nós, como Mica, Lotus e Iris.

Um nó telosB possui as seguintes características de hardware e software:

- **Transceptor:** Rádio transceptor 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.
- **Alimentação:** 2 baterias AA.
- **Sensores disponíveis:** luz, infravermelho, umidade e temperatura.
- **Processador:** Texas TI MSP430 com frequência de 8MHz.
- **Memória:** 1MB de memória não-volátil, 1024Kb de memória de programa e 10kB de RAM.
- **Dimensões:** 6,5 cm x 3,1 cm x 6,0 cm. O telosB pesa 23 gramas sem as baterias.
- **Sistema Operacional:** TinyOS.

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. A título de organização, facilidade de manutenção e reuso dos códigos, os mesmos foram implementados com tipos abstratos de dados, tornando-se bibliotecas em linguagem C que se integram ao código de aplicação para RSSFs, esse em linguagem nesC.

Ao longo dos experimentos utilizamos 105 nós sensores. As entradas de dados para o testbed são padrão: um arquivo de aplicação .exe, um .class que funciona com um cabeçalho e determina quais campos ou valores, serão transmitidos pelos pacotes até o ponto de acesso(PA). Em todos os experimentos o PA está localizado em um região periférica da rede. Logo após a execução do experimento o arquivo .dat é disponibilizado em um banco de dados MySQL, ele contém todas as informações relacionadas ao experimento e os dados detalhados de informações contidas nos pacotes.

A rotina da aplicação RSSF programada coleta a temperatura no ambiente do testbed em intervalos programados que variam na faixa de 1 a 10 segundos e disseminam mensagens em um tempo que está na faixa de 2 a 20 segundos. O intervalo de tempo não é fixo, uma vez que o propósito principal é a característica de auto configuração da rede.

O gráfico exibido pela figura 4 representa o consumo médio de energia por nó e a média de consumo geral da rede. Através de uma breve análise podemos concluir que a heurística mais elaborada, de fato, consumiu mais energia. Mas se observarmos o gráfico

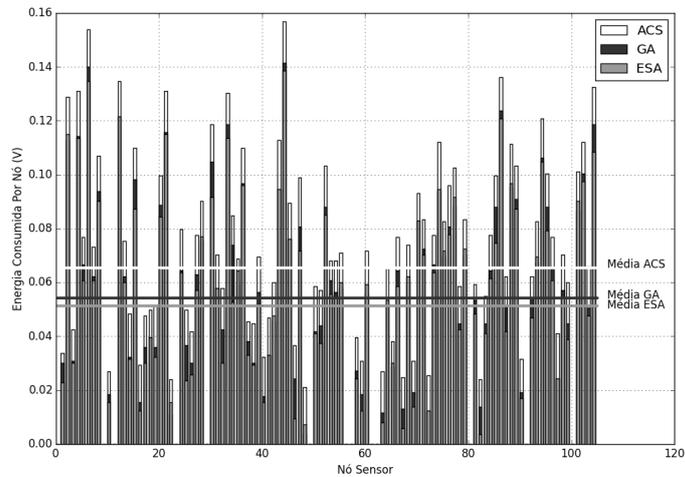


Figura 4. Gráfico de Consumo de Energia Por Nó e Média Geral de Consumo

referente ao número de mensagens enviadas, na figura 5, o ACS conseguiu fazer com a que a rede enviase menos mensagens, o que é uma característica interessante. O objetivo era de fato conseguir fazer com a que rede enviase menos mensagens, mesmo conhecendo o processamento maior que seria necessário para executar os algoritmos mais sofisticados. Também foram calculadas as médias para cada cenário, e estas estão representadas no gráfico por linhas tracejadas sobre as barras. O desvio padrão para o consumo está entre 0.0013s e 0.0015s o que foi considerado satisfatório.

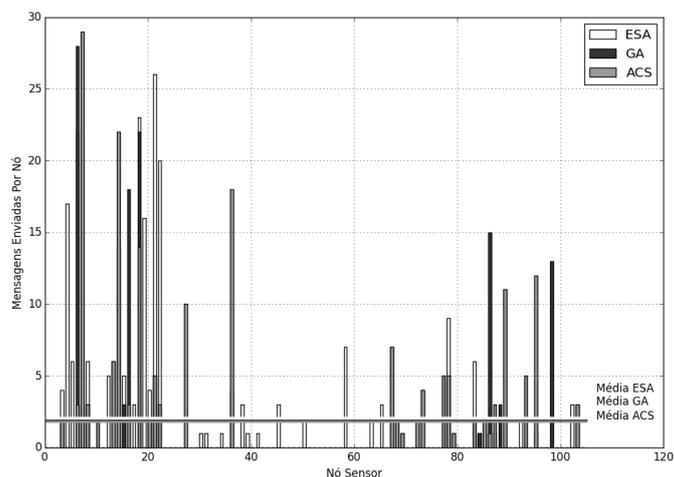


Figura 5. Gráfico de Mensagens Enviadas Por Nó e Médias

A figura 6 apresenta um gráfico onde é possível observar o número de pacotes trafegados pelos nós durante o experimento. Neste aspecto conseguimos uma pequena melhora no desempenho com as duas novas instâncias. O ACS conseguiu uma melhora em relação ao número de pacotes do ESA genérico, mas o GA por sua vez não apresentou tanta melhora, comportando-se de maneira parecida ao ESA genérico. As médias também

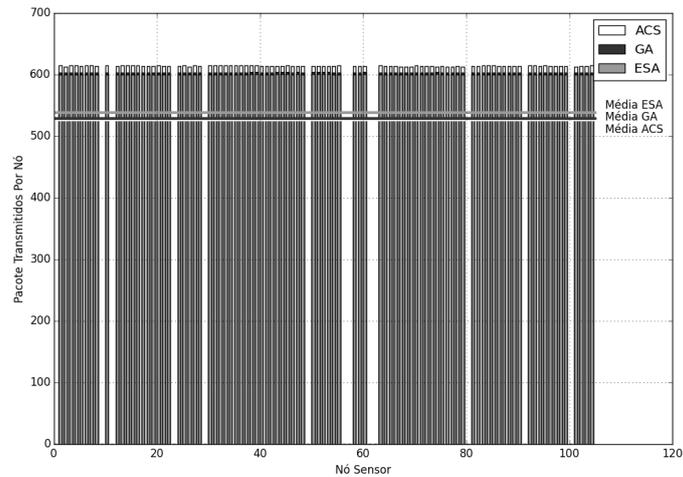


Figura 6. Pacotes Trafegados Por Nó e Médias

podem ser observadas nos gráficos, e o desvio padrão calculado para este caso está entre 3 e 15 pacotes, que representa um volume pequeno se comparado ao número total de pacotes para cada experimento, que ultrapassa 50 mil.

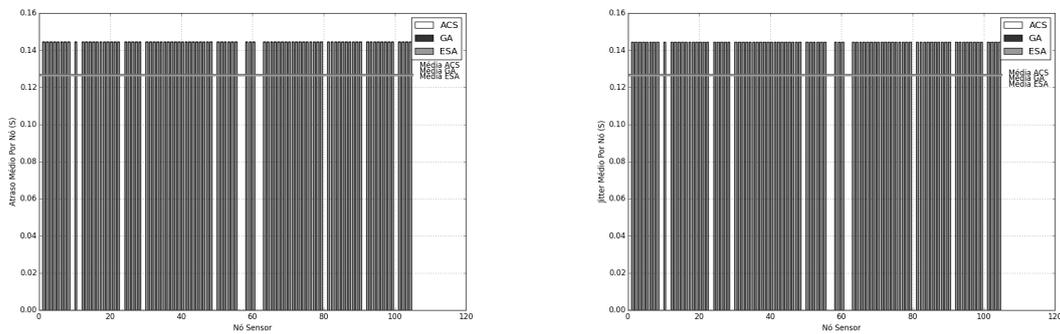


Figura 7. Atraso e Jitter de Rede

O gráfico exibido pela figura 7 representa o atraso da rede e também o jitter de rede. Nele podemos perceber que o atraso na rede manteve-se constante e isto causou pouco impacto no jitter. É possível observar um jitter na faixa de 0.14 s de maneira constante durante a execução do experimento. O que é uma característica importante e relevante no contexto de redes de computadores e RSSF.

6. Conclusão e Trabalhos Futuros

Como objetivo deste trabalho, implementamos duas novas instâncias do ESA genérico, uma dessas instâncias com um AG e outra com um ACS. Ambos foram desenvolvidos e executados em testbeds, o que deu ao trabalho uma característica mais consistente devido aos experimentos serem executados em um ambiente real que representa uma RSSF. As duas instâncias foram avaliadas sob a perspectiva do impacto causado no funcionamento

da rede, com discussão de métricas pertinentes ao consumo, a quantidade de mensagens trocadas, o tráfego de pacotes e o jitter da rede. O comportamento das instâncias foi bastante interessante porque, de fato, os algoritmos mais elaborados conseguiram fazer com que as etapas de análise e execução se tornassem mais elaboradas. Por isso temos um número menor de mensagens enviadas nas duas instâncias, e isso é o que faz a maior diferença no aspecto do consumo. 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.

Referências

- Abdelaal, M. (2014). Distributed techniques for energy conservation in wireless sensor networks.
- Becceneri, J. C., Stephany, S., Velho, H. F. d. C., and Neto, A. J. d. S. (2010). Otimização por colônia de formigas (ant colony optimization). In *Disponível em mtc-m19.sid.inpe.br/col/sid.inpe.br/mtc-m19@80/2010/01.20.19.27/doc/cap7.pdf*. [Online; Acessado em Outubro de 2015].
- 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*.
- Carvalho, A. P. d. L. F. d. (2009). Algoritmos genéticos - um breve histórico. In *Disponível em www.icmc.usp.br/andre/research/genetic/*. [Online; Acessado em Novembro de 2015].
- Jabeur, N. and Sahli, N. (2015). Enabling cyber physical systems with wireless sensor networking technologies, multiagent system paradigm, and natural ecosystems.
- Lakshmi, C. 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*.
- Lucas, C. D. (2002). Algoritmos genéticos: uma introdução. In *Disponível em www.inf.ufrgs.br/alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf*. [Online; Acessado em Outubro de 2015].
- 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*.
- Sadouq, Z., El Mabrouk, M., and Essaïdi, M. (2014). Conserving energy in wsn through clustering and power control. In *Information Science and Technology (CIST), 2014 Third IEEE International Colloquium in*.