



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE COMPUTAÇÃO
MESTRADO E DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

MARIA CECILIA REBOUÇAS

UM ESQUEMA DE ESCALONAMENTO DISTRIBUÍDO E
COLABORATIVO PARA GERÊNCIA DE COBERTURA EM REDES
DE SENSORES SEM FIO DENSAS

FORTALEZA - CEARÁ

2012

MARIA CECILIA REBOUÇAS

**UM ESQUEMA DE ESCALONAMENTO DISTRIBUÍDO E
COLABORATIVO PARA GERÊNCIA DE COBERTURA EM REDES
DE SENSORES SEM FIO DENSAS**

Dissertação de Mestrado apresentada à
Coordenação do Programa de Mestrado e
Doutorado em Ciência da Computação, da
Universidade Federal do Ceará, como requisito
parcial para obtenção do grau de Mestre em
Ciência da Computação

Orientador: Prof. Dr. Miguel Franklin de
Castro

Coorientador: Prof. Dr. Gabriel Antoine
Louis Paillard

FORTALEZA - CEARÁ

2012

A000z	<p>REBOUÇAS, M. C..</p> <p>Um Esquema de Escalonamento Distribuído e Colaborativo para Gerência de Cobertura em Redes de Sensores Sem Fio Densas / Maria Cecília Reboúças. 2012.</p> <p>75p.;il. color. enc.</p> <p>Orientador: Prof. Dr. Miguel Franklin de Castro</p> <p>Coorientador: Prof. Dr. Gabriel Antoine Louis Paillard</p> <p>Dissertação(Mestrado e Doutorado em Ciência da Computação) - Universidade Federal do Ceará, Departamento de Computação, Fortaleza, 2012.</p> <p>1. 2. 3. I. Prof. Dr. Miguel Franklin de Castro(Orient.) II. Universidade Federal do Ceará– Mestrado e Doutorado em Ciência da Computação(Mestrado) III. Mestre</p> <p>CDD:000.0</p>
-------	---

MARIA CECILIA REBOUÇAS

**UM ESQUEMA DE ESCALONAMENTO DISTRIBUÍDO E
COLABORATIVO PARA GERÊNCIA DE COBERTURA EM REDES
DE SENSORES SEM FIO DENSAS**

Esta dissertação foi submetida ao Programa de Mestrado e Doutorado em Ciência da Computação (MDCC), como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, outorgado pela Universidade Federal do Ceará.

Aprovada em: __/__/2012

BANCA EXAMINADORA

Prof. Dr. Miguel Franklin de Castro
Universidade Federal do Ceará - (UFC)
Orientador

Prof. Dr. Gabriel Antoine Louis Paillard
Universidade Federal do Ceará - (UFC)
Coorientador

Prof. Dr. José Neuman de Souza
Universidade Federal do Ceará - (UFC)
Membro Interno ao Programa

Prof. Dr. Danielo Gonçalves Gomes
Universidade Federal do Ceará - (UFC)
Membro Externo ao Programa

Prof. Dr. Ângelo Roncalli Alencar Brayner
Universidade de Fortaleza - (UNIFOR)
Membro Externo à Instituição

à cœur vaillant, rien d'impossible

(proverbe français)

AGRADECIMENTOS

Agradeço a minha família por ser a base sólida e confiável de que todo indivíduo necessita. O apoio emocional nos momentos difíceis nos mostra que, apesar das adversidades, estar junto dos seus nos tornam mais fortes e perseverantes.

Aos meus orientadores, professores Miguel Franklin e Gabriel Paillard, pela enorme paciência diante das minhas dificuldades pessoais externas ao mestrado. Acima de tudo, agradeço pelo apoio ao meu trabalho, com seus ensinamentos e experiências. Graças as suas importantes contribuições, foram possíveis a continuidade e a conclusão desta dissertação.

Aos amigos Vinícius Thiago, Sandra Carvalho, Vanessa Viana, Paulo Alexandre, Daniel Martins, Fernando Parente, Carlos Alberto, Diego Victor, Adyson Maia e Diego Cardoso pelo companheirismo, apoio, bem como pela troca de conhecimentos, não somente ao longo do Mestrado, bem como durante todo o período em que estive na UFC.

À professora Rossana Andrade pela oportunidade de fazer parte desta que eu costumo chamar de “família científica”, que é o GREat, primeiro como analista de sistemas e agora como mestrande. Ao professor Danielo Gomes, por suas contribuições de extrema relevância, avaliando meu trabalho desde a sua pré-proposta.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) pelo apoio financeiro concedido sob a forma de bolsa, que viabilizou a elaboração deste trabalho.

Finalmente, ciente de estar omitindo nomes de tantas outras pessoas que contribuiram direta ou indiretamente na realização deste trabalho...aos demais amigos, registro os meus sinceros agradecimentos.

RESUMO

Em muitas aplicações que utilizam uma Rede de Sensores Sem Fio (RSSF), a energia interna dos nós sensores pode se tornar um recurso crítico, limitado. Por conta disso, é importante que a utilização desses sensores seja feita de maneira eficiente, de acordo com a finalidade da aplicação. No caso onde os nós estão densamente implantados com o propósito de monitorar uma área específica, podem ocorrer interseções nas regiões cobertas pelos sensores. Identificando essas redundâncias, os sensores podem, de maneira distribuída e colaborativa, alternar seus estados de atividade de cobertura. Este trabalho busca estabelecer a seleção dos sensores que estarão ativos em um dado instante, utilizando somente informações locais do sensor e de sua respectiva vizinhança. Nesse contexto, este trabalho propõe o DCS² - Esquema de Escalonamento Distribuído e Colaborativo - inspirado em algoritmos de Sistemas Distribuídos. Experimentos realizados simulando cenários de redes densas mostram, comparado com duas outras soluções encontradas na literatura, uma com e outra sem mecanismo de escalonamento, o DCS² propiciou um aumento no tempo de vida da rede e obteve boa taxa de cobertura na RSSF.

Palavras-chave: Rede de Sensores Sem Fio, Escalonamento, Algoritmo Distribuído

ABSTRACT

In many Wireless Sensor Network (WSN) applications the sensor's internal resource energy can become a critical issue. For that reason, according to application requirements, sensors must be employed efficiently. In the case where nodes are densely deployed in order to monitor a certain area, the regions covered by sensors may overlap each other. Identifying these redundancies, so that sensors can schedule their status in coverage activities. This paper aims to establish the selection scheme of sensors that may be activated at a given time. That must be done using only sensor's local information and its respective neighbourhood. In this context, this paper proposes the DCS² algorithm - Distributed and Collaborative Scheduling Scheme - inspired by Distributed Systems algorithms. Experimental tests simulating dense WSN scenarios show that, when compared to two other solutions, with and without scheduling scheme, DCS² succeeded in increasing network lifetime and it achieved a good coverage ratio for dense WSN.

Keywords: Wireless Sensor Network, Scheduling, Distributed Algorithm

SUMÁRIO

LISTA DE FIGURAS	
LISTA DE TABELAS	
LISTA DE SIGLAS	12
LISTA DE ALGORITMOS	14
1 INTRODUÇÃO	15
1.1 Contextualização	15
1.2 Objetivos	16
1.2.1 Objetivo geral	16
1.2.2 Objetivos específicos	16
1.3 Plano de atividades	17
1.4 Organização da dissertação	18
2 REVISÃO BIBLIOGRÁFICA	20
2.1 Autonomia em Redes de Computadores	20
2.1.1 Análise e Projeto de Algoritmos Autônômicos	25
2.1.2 Sensibilidade ao Contexto e Semântica na Comunicação	25
2.1.3 Novos Modelos de Programação para Coordenação e Comunicação	26
2.1.4 Confiabilidade e Segurança	26
2.1.5 Avaliação e Testes	26
2.2 Redes de Sensores Sem Fio	27
2.2.1 Cobertura em RSSF	33
2.3 Escalonamento de Atividade em RSSF	41
2.4 Escalonamento por Inversão de Arestas	42
2.4.1 Sistemas Distribuídos	42
2.4.2 Algoritmo EIA	44
3 ESQUEMA DE ESCALONAMENTO DISTRIBUÍDO: DCS²	48
3.1 Definição do problema	48

3.2	Adaptação do EIA para RSSF	49
3.3	Algoritmo Proposto: DCS²	52
4	EXPERIMENTOS	55
4.1	Descrição do Plano de Testes	55
4.2	Métricas de Avaliação	59
4.3	Análise dos Resultados Obtidos	60
4.3.1	Área Coberta pelos Sensores	60
4.3.2	Energia Residual Média	62
4.3.3	Tempo de Vida da Rede	64
4.4	Conclusão do capítulo	65
5	CONSIDERAÇÕES FINAIS	66
5.1	Conclusões	66
5.2	Contribuições	67
5.3	Trabalhos Futuros	67
	REFERÊNCIAS BIBLIOGRÁFICAS	69

LISTA DE FIGURAS

Figura 2.1	Relação entre os paradigmas. Fonte: (WANT et al., 2003).	23
Figura 2.2	Componentes básicos de um nó sensor.	27
Figura 2.3	Representação de uma Rede de Sensores Sem Fio (RSSF).	29
Figura 2.4	Cobertura completa para uma área $L \times W$. Adaptada de (HU et al., 2010).	38
Figura 2.5	Dois sensores S_i e S_j , que sobrepõem suas áreas de sensoriamento, conseguem se comunicar entre si uma vez que $R_c \geq 2R_s$.	41
Figura 2.6	Ilustração para o <i>Dining Philosophers Problem</i> .	44
Figura 2.7	Funcionamento do Algoritmo EIA.	46
Figura 3.1	Exemplo - sobreposição de áreas de sensoriamento. Fonte: (MISRA et al., 2011).	49
Figura 3.2	Exemplos de grau de cobertura. Fonte: (ANCUSA; BOGDAN, 2011).	50
Figura 3.3	Exemplo - adaptação EIA para RSSF.	51
Figura 4.1	Porcentagem da área total, coberta pelos sensores - todos os nós implantados ativos, CCMS e DCS ² .	61
Figura 4.2	Energia residual média dos nós da rede.	63
Figura 4.3	Comparação dos tempos de vida da rede com DCS ² , CCMS e com todos os nós ativos.	64

LISTA DE TABELAS

Tabela 4.1	Densidades empregadas nas simulações.	57
Tabela 4.2	Configurações padrão utilizadas no simulador.	58

LISTA DE SIGLAS

BiO4SeL

Biologically-inspired Optimization for Sensor Network Lifetime.

CCMS

Connectivity and Coverage Maintenance Scheduling.

DCS²

Distributed and Collaborative Scheduling Scheme.

DPP

Dining Philosophers Problem.

EB

Estação-base.

EIA

Escalonamento por Inversão de Arestas.

ERA

Escalonamento por Reversão de Arestas.

GPS

Global Positioning System.

GREat

Grupo de Redes de Computadores, Engenharia de Software e Sistemas.

IC

Inteligência Computacional.

MEMS

Micro-Electro-Mechanical Systems.

MIAD

Máquina de Inferência Autônoma Distribuída.

NS-2

Network Simulator - versão 2.

NS-3

Network Simulator - versão 3.

PROST

Protocolo de Roteamento Sensível ao Tempo.

QoS

Qualidade de serviço.

RSSF

Rede de Sensores Sem Fio.

SD

Sistemas Distribuídos.

SEMANTK

SEmantic Middleware for Autonomic seNsor neTworKs.

LISTA DE ALGORITMOS

1	Pseudocódigo do algoritmo EIA.....	45
2	Agrupamento inicial no DCS ² . Fonte: (BARI et al., 2008).....	53
3	Algoritmo DCS ²	54

1 INTRODUÇÃO

Esta dissertação apresenta um mecanismo para gerência de cobertura em Redes de Sensores Sem Fio (RSSFs), realizado por meio do escalonamento de atividade dos nós da rede. Esse escalonamento é feito de forma distribuída e colaborativa, fazendo uso apenas de informações internas contidas em cada sensor, bem como de sua respectiva vizinhança. O objetivo é prolongar o tempo de vida da rede em questão, tendo como cenário RSSFs densas.

A organização deste capítulo obedece a seguinte divisão: na seção 1.1 é descrita a contextualização onde este trabalho está inserido; a problemática da gerência de cobertura é mostrada, apontando, dessa forma, a principal motivação para o desenvolvimento deste trabalho. A seção 1.2 aponta os principais objetivos desta dissertação, seus objetivos geral e específicos. As atividades que compuseram a realização deste trabalho são descritas na seção 1.3. Finalizando, a seção 1.4 expõe como estão estruturados os capítulos subsequentes desta dissertação.

1.1 Contextualização

O crescimento e a diversidade de aplicações em redes de computadores estimulam o vislumbre de um futuro onde dispositivos computacionais poderão estar embutidos em praticamente todos os objetos, bem como em ambientes do cotidiano das pessoas. Para um funcionamento mais eficiente, tais dispositivos podem exigir que sua operação seja feita com mínima intervenção humana, seja em suas atividades individuais, ou na comunicação entre dispositivos.

Considerando esse funcionamento sem intermédio humano, torna-se interessante que os dispositivos sejam dotados de autonomia para gerenciar a realização de suas atividades. Outra questão que pode ser considerada é a possibilidade dos dispositivos cooperarem entre si no gerenciamento de suas atividades, de acordo com os requisitos de aplicação de rede em questão.

Uma RSSF é um exemplo de rede dotada de capacidade de sensoriamento, de processamento e de comunicação, em alguns casos, distribuída e colaborativa. Essas redes são compostas por dispositivos sensores, que são implantados com o propósito de monitoramento de um ambiente específico. As soluções em RSSF dependem de suas aplicações, dentre as quais podem ser destacadas: monitoramento ambiental, controle de qualidade na produção industrial e na agricultura, gestão urbana, aplicações biomédicas, bem como aplicações militares (AKYILDIZ et al., 2002) (SOHRABY et al., 2007).

De acordo com o trabalho apresentado em (ZHU et al., 2011), um problema importante que vem recebendo atenção dos pesquisadores é o problema de cobertura, que está

relacionado a verificar o quão bem os sensores observam o espaço físico onde estão inseridos. Os autores apontam ainda que, em muitas aplicações e cenários, a energia interna de cada um dos nós sensores ¹ se torna um recurso crítico, limitado. Dessa forma, é necessário que a utilização dos sensores seja feita de uma maneira eficiente, de forma a prolongar o tempo de vida da rede.

Dentro desse contexto, esta dissertação aborda a problemática de cobertura, levando em consideração o consumo de energia de cada nó da rede, propondo um mecanismo de escalonamento de atividade para os nós da RSSF. Os objetivos deste trabalho estão presentes na próxima seção.

1.2 Objetivos

A seguir são apresentados os principais objetivos do trabalho desenvolvido nesta dissertação. Os mesmos estão divididos em objetivo geral e objetivos específicos.

1.2.1 Objetivo geral

O objetivo geral deste trabalho é o desenvolvimento de um mecanismo de escalonamento para RSSFs, que seja executado de forma distribuída na rede, fazendo uso da característica colaborativa dos sensores. Esse mecanismo foi concebido para atuar no cenário de RSSFs densas, de maneira a distribuir a execução das tarefas de sensoriamento entre os sensores, visando distribuir o consumo de energia da rede e prover um ganho no seu tempo de vida.

1.2.2 Objetivos específicos

A partir da definição do objetivo geral, percebe-se a existência de alguns objetivos específicos que precisam ser alcançados para, enfim, atingir o objetivo geral. Esses subobjetivos são também definidos a seguir:

- ▶ estabelecer os requisitos que devem ser atendidos pela solução a ser proposta, considerando as necessidades do cenário de cobertura, com foco em economia de energia, em RSSFs densas;
- ▶ propor um esquema de escalonamento de atividades dos sensores que atenda aos requisitos estabelecidos;
- ▶ implementar e analisar os resultados para verificar a validade da solução proposta;

¹ Ao longo desta dissertação, os termos “sensor” e “nó sensor” são utilizados indistintamente.

- relacionar o comportamento da solução sob a ótica de cobertura e consumo de energia, visando identificar onde a solução é mais adequada.

A seção a seguir resume o plano de atividades executadas ao longo do desenvolvimento deste trabalho.

1.3 Plano de atividades

Para alcançar os objetivos definidos na seção anterior, foi estabelecido um plano de atividades, que está resumido a seguir. Essas atividades foram empregadas tanto na abordagem do problema em questão, quanto na concepção e no desenvolvimento da solução apresentada nesta dissertação.

(i) Revisão bibliográfica

Esta atividade compreendeu ao levantamento bibliográfico sobre RSSF, com foco maior na questão de cobertura para essas redes. Foram realizados estudos buscando soluções para RSSFs densas, partindo da premissa de que as soluções deveriam ser descentralizadas e que não possuísem qualquer conhecimento de localização dos sensores. É importante destacar que, como tentativa de manter a revisão bibliográfica a mais atualizada possível, esta atividade foi executada continuamente, ao longo da elaboração desta dissertação.

(ii) Definição do problema e especificação da solução proposta

Essa atividade foi concomitante à revisão bibliográfica, uma vez que envolveu estudo de trabalhos atuais existentes na literatura para definir o problema para o qual se propõe solução. Também foram estudadas soluções de escalonamento em outros domínios, além das RSSF. Assim, foi encontrado o algoritmo de Escalonamento por Inversão de Arestas (EIA), a ser apresentado na seção 2.4, o qual serviu como inspiração para a solução proposta neste trabalho. Dessa forma, foi necessário um estudo mais aprofundado para buscar desenvolver um algoritmo próprio, que se adequasse ao cenário das RSSFs densas.

(iii) Implementação e execução dos experimentos

A solução proposta e os experimentos foram implementados e simulados no simulador de redes Network Simulator, utilizando as linguagens C++ e OTcl. Maiores

detalhes a respeito da implementação se encontram na seção 4.1. Além da solução proposta, foram implementados também: uma solução base, correspondendo a uma RSSF sem qualquer mecanismo de escalonamento e todos os nós ativos; e uma solução escolhida para comparação, o algoritmo CCMS apresentado em (WANG et al., 2010).

(iv) Análise dos resultados obtidos

Após a realização dos experimentos, foram coletados os dados provenientes das simulações e, de acordo com as métricas estabelecidas na seção 4.2, foram gerados os gráficos que constam na seção 4.3. Dessa forma, também foram feitas as análises e discussões dos resultados obtidos, as quais constam também na seção 4.3.

1.4 Organização da dissertação

O conteúdo restante desta dissertação está organizado nos capítulos subsequentes, da seguinte forma:

Capítulo 2

Apresenta a revisão bibliográfica realizada, destacando as definições, principais características e aplicações na área de RSSF. É abordada também a questão de cobertura em RSSF, com foco nas soluções existentes, mais precisamente as que envolvem escalonamento de atividade dos nós.

O capítulo é composto também de uma breve introdução sobre o sistemas distribuídos, contexto onde está inserida a solução EIA, cujo algoritmo também é apresentado neste capítulo.

Capítulo 3

Inicialmente, apresenta a definição do problema para o qual este trabalho propõe solução. Depois descreve como foi realizada a adaptação do EIA para RSSF, explicando os detalhes da solução DCS₂. Por fim, apresenta a descrição dos detalhes do algoritmo proposto.

Capítulo 4

Este capítulo descreve como se deram os experimentos. Inicialmente, é descrito o plano de testes que foi elaborado para a realização dos experimentos, apresentando as configurações da rede, bem como os parâmetros que foram utilizados na ferramenta de simulação. Em seguida são definidas as métricas a serem consideradas para avaliação do desempenho do algoritmo proposto. Por fim, o capítulo apresenta os resultados obtidos nos experimentos, bem como é feita uma análise sobre esses resultados.

Capítulo 5

Neste capítulo, são feitas as considerações relevantes oriundas da análise dos resultados obtidos com os experimentos realizados. Este capítulo também aponta as principais contribuições deste trabalho.

Além das considerações a respeito dos resultados e contribuições alcançados, as perspectivas de trabalhos futuros, a partir do estado atual deste trabalho. Para concluir o capítulo, são relacionados e discutidos possíveis caminhos de evolução do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta a revisão bibliográfica dos assuntos abordados na concepção e desenvolvimento desta dissertação. Um levantamento sobre autonomia em redes de computadores é apresentado na seção 2.1, relacionando os principais paradigmas nessa área. A seção 2.2 descreve RSSFs, apresentando sua definição, componentes, características e exemplos de aplicações na área. A seção apresenta ainda problemas encontrados na literatura. Na seção 2.3 são apresentados trabalhos que abordam mecanismos para a problemática de cobertura em RSSF. Por fim, na seção 2.4, é feita uma breve introdução sobre sistemas distribuídos, situando o contexto onde se insere o algoritmo EIA, que foi a inspiração para o algoritmo proposto nesta dissertação.

2.1 Autonomia em Redes de Computadores

Conforme considerado na seção 1.1, pode existir o interesse em inserir autonomia aos dispositivos e aplicações em redes de computadores, retirando das mãos dos administradores e usuários parte da complexidade inerente ao seu gerenciamento. Para isso, é relevante compreender alguns paradigmas que conjecturam avanços nessa direção. Dessa forma, faz-se necessário identificar quais áreas ainda podem ser exploradas sob o ponto de vista de buscar melhores soluções aos problemas em questão. A seguir, serão descritos alguns desses paradigmas.

O paradigma da Computação Autônoma (*Autonomic Computing*), visionada por (KEPHART; CHESS, 2003), aponta a importância do desenvolvimento de sistemas sob a ótica da autonomia. Esses sistemas poderiam controlar a si próprios, reduzindo a complexidade das mãos de seus administradores e usuários. Outro paradigma também existente compreende ao paradigma da Computação Proativa (*Proactive Computing*), o qual se sobrepõe à computação autônoma em alguns aspectos. Contudo, aquela enfatiza a transição dos sistemas interativos atuais para ambientes proativos, estes capazes de antecipar necessidades ou comportamentos.

Uma visão geral da relação entre computação autônoma e computação proativa é apresentada em (WANT et al., 2003), onde os autores analisam as características de cada um desses paradigmas. Inicialmente, analisando a computação autônoma, esta pode ser caracterizada através dos oito princípios que a regem. A seguir são descritos esses princípios, conforme estudo e compilação a partir de (HORN, 2001), (KEPHART; CHESS, 2003) e (BANTZ et al., 2003):

Autoconhecimento (*Self-Awareness*)

O sistema deve conhecer a si próprio, deve ter ciência dos componentes que possibilitam sua existência, bem como deve possuir uma identidade própria. Deve ainda conhecer seu estado atual e todas as suas conexões com outros sistemas.

Autoconfiguração e Autorreconfiguração (*Self-Configuration and Self-Reconfiguration*)

A inicialização e configuração do sistema devem ocorrer de maneira automática, bem como o sistema deve ser capaz de se configurar e reconfigurar de forma a melhor se adequar às mudanças no ambiente onde ele está inserido.

Auto-otimização (*Self-Optimization*)

Ao longo do tempo, o sistema deve ser capaz de analisar seu próprio funcionamento e tomar decisões buscando melhor otimizá-lo. Para executar essa função, o sistema necessita monitorar as métricas associadas as suas funções, de maneira a escolher apropriadamente a ação de otimização.

Autorreparação (*Self-Healing*)

O sistema deve ser capaz de identificar problemas ou detectar iminência de problema. Depois, deve ser capaz de diagnosticar o problema identificado, para então utilizar os mecanismos de reconfiguração e encontrar a melhor alternativa de reparar, se possível, o problema.

Autoproteção (*Self-Protection*)

O sistema deve ser capaz de perceber uma ameaça, identificá-la e se proteger contra diversos tipos de ataques. O objetivo é manter a segurança e integridade do sistema como um todo.

Autoavaliação (*Self-Evaluation*)

O sistema deve ter conhecimento do ambiente ao seu redor, bem como do contexto que permeia as suas atividades. Dessa forma, o sistema deve atuar de acordo com o ambiente ao seu redor, buscando interagir da melhor maneira possível com seus vizinhos. Autoavaliação se assemelha à auto-otimização, mas neste caso está mais relacionada à interação com o ambiente externo.

Autodescoberta (*Self-Discovery*)

Como apresentado anteriormente, o sistema deve ser capaz de gerenciar a si próprio, de maneira independente. Contudo, o sistema deve interagir, coexistir e colaborar com outros sistemas. Para isso, ele deve permitir acessibilidade perante outros sistemas, bem como ser capaz de descobrir formas de interação com estes.

Autoprevisão (*Self-Prediction*)

Quando utilizado, o sistema deve ser capaz de pressupor as necessidades de seu usuário e, de maneira transparente, alocar os recursos necessários. Essa, por sua vez, é considerada uma das tarefas mais críticas, que é antecipar a otimização dos recursos necessários, ao mesmo tempo que mantém a complexidade oculta para o usuário.

Face à busca por reduzir complexidade e prover escalabilidade nos sistemas de redes de computadores, muitas vezes as atenções se voltam para o desenvolvimento de sistemas capazes de garantir sua própria manutenção de forma automática, diante de cenários diversos. Esses sistemas são comumente descritos pelas características apresentadas acima, que são muitas vezes referenciadas como propriedades auto-* (*self-* properties*). Essas propriedades estão intrinsecamente relacionadas aos paradigmas discutidos nesta seção.

No trabalho proposto em (BERNS; GHOSH, 2009), os autores apresentam uma definição formal para os termos (auto-*) acima descritos, com a finalidade de eliminar possíveis ambiguidades entre os conceitos. No trabalho, os autores definem ainda duas novas características: *Self-Immunity* e *Self-Containment*. Maiores detalhes dessas definições podem ser encontrados no referido artigo, embora nesta dissertação foi optado por deixar esses termos sem tradução, no momento, evitando assim qualquer perda do real sentido dos mesmos ao fazê-lo.

Já a construção de sistemas sob o ponto de vista da computação proativa também é orientada por princípios básicos, alguns destes descritos abaixo, de acordo com (WANT et al., 2003):

Conectar ao mundo físico (*Connecting with the physical world*)

Uma vez que as infraestruturas computacionais estejam conectadas via rede ou mesmo um conjunto de servidores próximos conexos, o sistema resultante constitui-se de um ambiente virtual, que permite criar, processar e armazenar informações. Para isso, o mundo físico precisa estar equipado de tal maneira que os sistemas sejam capazes de perceber o ambiente, bem como atuar sobre ele.

Antecipação (*Anticipation*)

Antecipação é fundamental no paradigma de computação proativa. Para um sistema ser realmente proativo, é necessário certo grau de predição do futuro. Com o objetivo de alcançar essa certa predição, as pesquisas nessa direção fazem uso de informações como:

Contexto - O sistema pode fazer uso de informações de localização, recursos disponíveis na vizinhança, entre outras, para modificar o comportamento da aplicação.

Análise estatística - A utilização de métodos estatísticos tem se expandido, no lugar da aplicação de abordagens tradicionais, estas que por sua vez fazem uso de métodos determinísticos. Alguns exemplos são os Métodos de Estatística Multivariada, Modelos de *Markov*, Métodos Heurísticos e técnicas Bayesianas.

Tratamento de dados - Os sistemas de computação proativa podem tirar vantagem do aumento da capacidade de armazenamento e transmissão de dados, para agilizar a obtenção de dados necessários aos usuários. Por outro lado, a técnica empregada deve garantir confiabilidade aos dados providos pelo sistema.

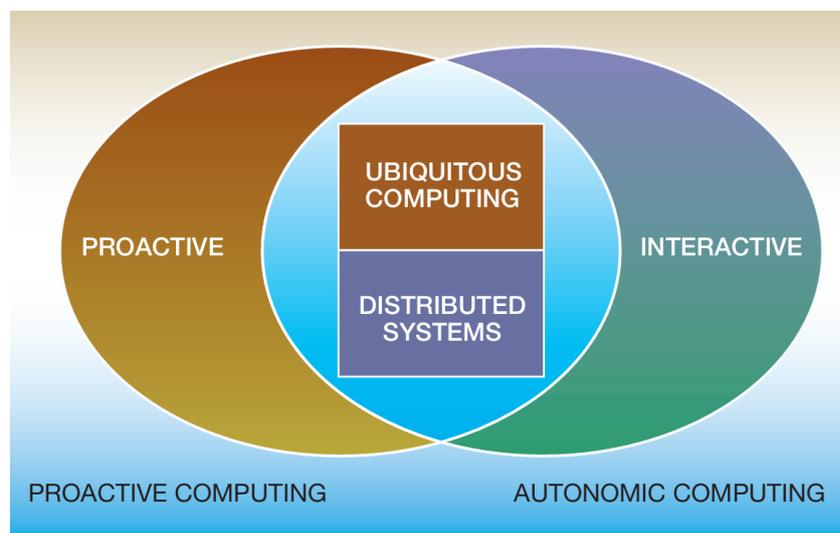


Figura 2.1: Relação entre os paradigmas. Fonte: (WANT et al., 2003).

O trabalho de (WANT et al., 2003) apresenta ainda a relação entre os dois paradigmas, ilustrada na figura 2.1. É possível perceber a sobreposição dos dois paradigmas supracitados, sobreposição essa necessária para a obtenção de sistemas ubíquos e autonômicos.

Posterior a esses dois paradigmas, surgiu a visão de Computação Orgânica (*Organic Computing*), que está igualmente relacionada aos Sistemas Distribuídos (SD), introduzida em (SCHMECK, 2005). Neste caso, o termo “orgânico” é referente à organização. Com sistemas dotados da capacidade de se auto-organizarem, de maneira a adaptarem-se às mudanças e restrições do ambiente, esse paradigma afirma que é necessário garantir que o processo de auto-organização não resulte em comportamentos indesejados ou imprevistos para o sistema. Apesar da relativa semelhança com os paradigmas anteriores, a computação orgânica não busca sistemas totalmente autonômicos. Segundo outro trabalho, (SCHMECK et al., 2010), os autores afirmam que os seres humanos devem sempre ter controle do sistema em sua vizinhança, fato que é definido pelos autores como auto-organização controlada (*controlled self-organization*). Apesar da concepção diferenciada, os autores apontam que a definição de diversos graus de liberdade para a auto-organização corresponderiam a uma das formas de se obter esse controle definido por eles.

Discutidos esses paradigmas, é pertinente comentar essa questão de auto-organização em sistemas distribuídos, levantando pontos importantes de como alcançá-la, mantendo em mente o cenário de dinamicidade e escalabilidade desses sistemas. Dessa forma, tomando como base os estudos feitos sobre os paradigmas apresentados nesta seção, este trabalho estende-se mais um pouco, e visa focar na Comunicação Autonômica (*Autonomic Communication*), a qual distingue-se da computação autonômica em alguns aspectos. O desenvolvimento de infraestruturas de rede e de comunicação, dotadas de capacidades de autogerenciamento, autoconfiguração e autorregulação – referenciadas coletivamente como comunicação autonômica – é uma área de pesquisa considerável, bem como de interesse da indústria. Esse paradigma faz uma analogia ao sistema nervoso autônomo humano, o qual é responsável por regular as atividades corporais visando manter o equilíbrio corporal. Por essa analogia, a comunicação autonômica busca simplificar o gerenciamento de funções complexas de comunicação entre sistemas, bem como visa reduzir a necessidade de intervenção manual nesse gerenciamento (DOBSON et al., 2006). De uma maneira geral, a computação autonômica está mais orientada para softwares de aplicação e gerenciamento de recursos computacionais. Por outro lado, a comunicação autonômica deposita grande parte do seu foco em serviços e sistemas distribuídos, bem como no gerenciamento de recursos de rede, sejam estes nos níveis de infraestrutura ou de usuário (QUITADAMO; ZAMBONELLI, 2008). De maneira geral, ambas, computação e comunicação autonômica, compartilham objetivos semelhantes, uma vez que as pesquisas nessas áreas buscam as já referenciadas características de autonomia.

A comunicação autonômica apoia-se em diversas áreas para seu desenvolvimento, como pode ser exemplificado, segundo (DOBSON et al., 2006):

- ▶ Projeto de protocolos
- ▶ Gerenciamento de redes

- ▶ Inteligência artificial
- ▶ Computação ubíqua
- ▶ Teoria dos jogos
- ▶ Semântica
- ▶ Biologia
- ▶ Sistemas sensíveis ao contexto (*context-aware systems*)
- ▶ Redes de Sensores
- ▶ Segurança

Dentre os temas de estudos e pesquisas envolvendo comunicação autonômica, (DOBSON et al., 2006) apresenta uma divisão em cinco grandes domínios, os quais concentram desafios e tecnologias emergentes nessa área. São eles: Análise e Projeto de Algoritmos Autonômicos; Sensibilidade ao Contexto e Semântica na Comunicação; Novos Modelos de Programação para Coordenação e Comunicação; Confiabilidade e Segurança; Avaliação e Testes. Esses domínios são brevemente caracterizados a seguir.

2.1.1 Análise e Projeto de Algoritmos Autonômicos

Haja vista a característica distribuída e fortemente mutável dos sistemas autonômicos, seja sob a ótica da computação ou comunicação autonômica, as soluções tradicionais para gerenciamento de redes podem não se tornar praticáveis em diversos casos. Abordagens de gerenciamento utilizando algoritmos centralizados, ou que partam do pressuposto de que há um conhecimento prévio da rede podem não se adequarem a um cenário intensamente dinâmico. Por essas razões, para adequarem-se às tecnologias emergentes, algumas soluções tradicionais precisam ser renovadas. De sistemas inicialmente centralizados, para quase ou completamente distribuídos. A análise e o projeto de novos algoritmos face a essa nova tendência deve vir acompanhada de características de colaboração entre os elementos do sistema, bem como da confiabilidade na comunicação.

2.1.2 Sensibilidade ao Contexto e Semântica na Comunicação

A comunicação autonômica implica em um forte grau de autogerenciamento e auto-otimização. Para prover isso, torna-se necessário investigar a temática de sensibilidade ao contexto, tendo em vista as propriedades da rede. Entidades de software e componentes de rede

podem ser utilizados para coletar informações de contexto relacionadas à presença, localização, identificação, bem como perfis de usuários e serviços. Outra característica desejável é o comportamento auto-adaptativo, onde um sistema autônomo deve ser capaz de analisar o contexto onde ele está inserido e definir o seu comportamento relativo a esse contexto analisado. Um desafio na área de semântica está em como representar as informações contextuais, e como inferir acerca dessas informações.

2.1.3 Novos Modelos de Programação para Coordenação e Comunicação

De maneira semelhante aos algoritmos, os modelos de programação também precisam ser renovados para se adaptarem às tecnologias emergentes da comunicação autônoma. Torna-se necessária a utilização de novos métodos para o desenvolvimento focado em autonomia. Métodos que busquem demonstrar as propriedades auto-* já citadas anteriormente.

2.1.4 Confiabilidade e Segurança

Consiste essencialmente nas propriedades e serviços de segurança: integridade, confidencialidade, autenticação, confiabilidade, reputação, identificação e criptografia. É dedutível que com o aumento da complexidade e escalabilidade dos sistemas, maiores sejam também as possibilidades de falhas de segurança desses sistemas. Nesse contexto, os sistemas autônomos necessitam implementar políticas de segurança para permitir que os outros componentes do sistema funcionem adequadamente.

2.1.5 Avaliação e Testes

Os mecanismos de testes e avaliações permitem comparar sistemas autônomos entre si, bem como com abordagens tradicionais. O uso de simulações que permitam avaliar a escalabilidade e as reações do sistema diante de situações de maior complexidade também é uma técnica interessante. Outra técnica é o desenvolvimento de plataformas de testes (*testbeds*), que possibilitam avaliar e testar soluções em ambientes semelhantes a uma situação real.

Realizado esse levantamento sobre autonomia em redes de computadores, este trabalho aborda agora um tipo especial de rede, que é o objeto principal do estudo aqui apresentado. Dessa forma, a seção a seguir introduz os estudos realizados sobre RSSF.

2.2 Redes de Sensores Sem Fio

Os sistemas Micro-Eleto-Mecânicos - *Micro-Electro-Mechanical Systems* (MEMS) - são formados pela junção de componentes mecânicos, eletrônicos, sensores, e atuadores. MEMS permitem o desenvolvimento de dispositivos inteligentes, integrando a capacidade computacional da microeletrônica, o poder de percepção dos microssores e a habilidade de controle dos microatuadores (PERLMUTTER; ROBIN, 2012). Com a proliferação da tecnologia de MEMS, as redes formadas por elementos sensores têm ganhado bastante atenção em todo o mundo.

De uma maneira geral, sensores são pequenos dispositivos dotados de componentes de sensoriamento, reduzida capacidade de processamento e armazenamento de dados. Uma vez que os nós sensores possuem memória limitada e, em certas circunstâncias, são implantados em locais de difícil acesso, eles são dotados de capacidade de comunicação sem fio para realizar transferência dos dados coletados por eles.

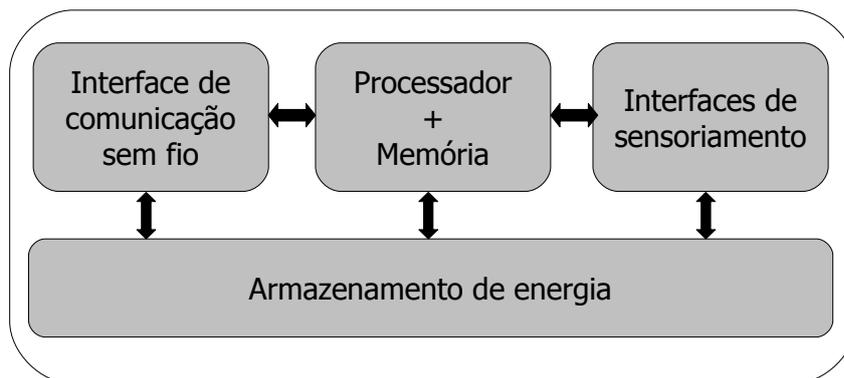


Figura 2.2: Componentes básicos de um nó sensor.

A figura 2.2 ilustra os componentes básicos de um nó sensor. A interface de comunicação sem fio é responsável pela transmissão dos dados coletados, bem como pela troca de mensagens e encaminhamentos realizados pelos sensores. Além das unidades de processamento e memória, a interface de sensoriamento realiza as tarefas de coletar dados percebidos no ambiente. Sensores mecânicos, térmicos, biológicos, químicos, óticos ou ainda magnéticos podem ser acoplados à interface de sensoriamento. Esses componentes são alimentados pelo elemento de armazenamento de energia. Na maioria dos casos, a bateria é a fonte principal de energia. Contudo, podem ser adicionadas fontes secundárias de energia, como painéis solares que captam energia do ambiente (VOIGT et al., 2003), (GAKKESTAD; HANSSSEN, 2011).

Além dos componentes apresentados na figura 2.2, atuadores podem ser incorporados aos sensores, dependendo da aplicação e do tipo de sensor utilizado. Um atuador é um dispositivo eletro-mecânico que pode ser usado para controlar diferentes componentes em um sistema. No caso de um nó sensor, os atuadores podem agir ajustando parâmetros do sensor, movendo o sensor de posição, ou monitorando o consumo de energia do nó sensor (YICK et al., 2008).

Dentro desse contexto, os sensores são utilizados para monitorar fenômenos físicos, tais como: temperatura, umidade, vibrações, luminosidade, velocidade, aceleração, eventos sísmicos. Baseado nas informações obtidas a partir dos dados captados pelos sensores, podem ser disparadas ações de controle sobre ao ambiente que está sendo monitorado.

Com o advento das tecnologias sem fio (*wireless*), os dispositivos sensores passaram a se conectar entre si, transmitindo as informações coletadas, formando uma rede de comunicação de sensores. Nesse contexto, uma RSSF é composta por uma certa quantidade de nós sensores, os quais podem estar densamente implantados dentro da área onde ocorre o fenômeno ou relativamente próximo a ela (AKYILDIZ et al., 2002). Dessa forma, uma RSSF consiste de um número de nós sensores (da ordem de dezenas a milhares) que trabalham conjuntamente para monitorar uma região e obter dados sobre o ambiente.

Além dos sensores, existem uma ou mais estações-base (EBs), para a(s) qual(is) os sensores encaminham os dados coletados. A EB, então, funciona como uma entidade que agrega esses dados, gerando informações relevantes a partir deles.

Muitas das RSSF são redes onde a comunicação é feita através de múltiplos saltos. Contudo, os protocolos para essas redes são diferenciados dos protocolos para redes cabeadas ou até mesmo redes *Wi-Fi*. A necessidade de soluções diferenciadas surge das restrições específicas dos dispositivos (sensores) e dos ambientes onde eles estão normalmente inseridos. Por exemplo, alguns dispositivos sem fio não realizam detecção de colisão, bem como sua memória reduzida pode não comportar o armazenamento de grandes tabelas de roteamento (STANKOVIC, 2008).

A figura 2.3 representa uma RSSF composta por vários nós sensores e uma EB. A figura mostra também o encaminhamento dos dados coletados por um sensor até a EB.

Considerando o gasto de energia dispensado em tarefas de sensoriamento e processamento dos dados, é importante analisar também a energia gasta para a tarefa de envio dos dados. Para executar essa tarefa, todos os nós sensores poderiam ter raio de alcance suficientemente grande para atingir a EB diretamente. Esse caso inviabiliza a longa duração da rede, pois os sensores logo consumiriam toda a sua bateria, na tentativa de alcançar a EB por meio do aumento das suas potências de transmissão. Na prática, esse envio de dados é realizado de outra maneira. Os sensores repassam os dados entre eles, por um ou mais saltos, até chegar à

EB. Esse encaminhamento dos dados pode ser visualizado na figura 2.3, por meio das linhas tracejadas que passam de um sensor para o outro. Essas linhas representam o caminho que uma mensagem de dados, ou de controle, poderia percorrer partindo de um sensor até chegar à EB existente na rede.

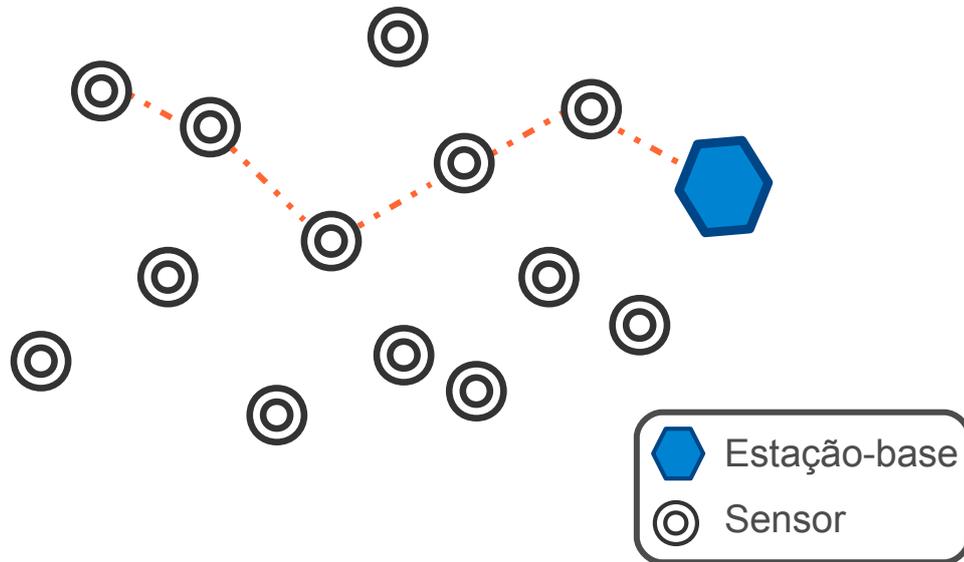


Figura 2.3: Representação de uma RSSF.

Quanto à infraestrutura das RSSFs, de acordo com (YICK et al., 2008), estas podem ser divididas em dois tipos: estruturadas e não-estruturadas.

RSSF Estruturada

Em uma rede estruturada, há um planejamento prévio para o posicionamento de todos ou parte dos sensores da rede. Esse planejamento é realizado antes dos sensores serem implantados no ambiente, onde a localização de cada sensor já é conhecida a priori. Neste caso, os sensores já estão predeterminados a ficarem em posições fixas, antes mesmo de serem implantados na área em questão (KAR; BANERJEE, 2003). Isso pode ser aplicado quando existem condições favoráveis, por exemplo, o ambiente é propício ou acessível. Uma vantagem nas redes estruturadas é que podem ser implantados um número reduzido de sensores, reduzindo por sua vez o custo de gerenciamento e de manutenção da rede.

RSSF Não-estruturada

Uma rede não-estruturada é composta por uma densa coleção de nós sensores. Neste

caso, os sensores podem ser implantados na área de maneira aleatória (*random deployment*), não previamente determinada (MEGERIAN; POTKONJAK, 2002). Nos casos onde o ambiente é inacessível ou inóspito para o ser humano, os sensores podem ser lançados de um avião, por exemplo. Uma vez realizada a implantação dos sensores, a rede então é deixada para realizar suas atividades de monitoramento e comunicação de maneira autônoma. Nas RSSF não-estruturadas existe uma dificuldade maior na sua manutenção, seja para gerenciar conectividade ou detectar falhas, por exemplo. Essa dificuldade é resultante da densidade da rede, do número maior de nós sensores implantados.

Para o caso das redes estruturadas, um número reduzido de nós podem ser implantados em posições específicas e prover cobertura para a rede, ao passo que a implantação aleatória dos sensores pode resultar em regiões não cobertas.

Os nós de uma RSSF são geralmente alimentados por baterias. No caso onde os sensores são implantados em locais inóspitos, ou de difícil acesso para o ser humano, pode ser inconveniente ou impossível a recarga de bateria. Por outro lado, a RSSF precisa ter um tempo de vida longo o suficiente para cumprir os requisitos da sua respectiva aplicação.

Por se tratar de um conceito pertinente quando se aborda RSSF, é apropriado dedicar algumas linhas para discutir e definir o conceito de "tempo de vida" da rede. Este conceito será adotado ao longo deste trabalho.

O tempo de vida da rede é uma das métricas mais importantes para avaliar o desempenho de uma RSSF (DIETRICH; DRESSLER, 2009). O consumo de energia desempenha papel fundamental na definição dos limites de utilidade da rede, pois uma RSSF só consegue desempenhar o papel que lhe foi designado se a rede estiver viva. O tempo de vida da rede está fortemente relacionado com os tempos de vida de cada um dos sensores. Segundo (AKYILDIZ et al., 2002), o consumo de energia de um nó sensor está predominantemente ligado às atividades de sensoriamento, comunicação e processamento de dados. Como já discutido anteriormente, em muitos casos não é possível recarregar as baterias dos sensores após implantados. Essa impossibilidade desencadeou os estudos acerca do tempo de vida das redes, uma vez que estas não podem durar indefinidamente.

Dentre as definições para tempo de vida de uma RSSF, listadas e discutidas em (DIETRICH; DRESSLER, 2009), este trabalho adota a seguinte definição.

- O tempo de vida da rede é definido como o tempo decorrido até o primeiro nó da rede esgotar sua energia interna.

Haja vista essa restrição de energia nos sensores, é importante que hajam esforços no nível de *hardware*, que visem diminuir o consumo associado às tarefas executadas pelos

sensores. Essa é uma vertente importante e que deve ser acompanhada de soluções no nível lógico também, com soluções que busquem otimizar a questão da energia. Dessa forma, um dos desafios em RSSF é desenvolver mecanismos que visem reduzir o consumo de energia dos sensores, prolongando assim o tempo de vida da rede.

As aplicações em RSSF são diversas, permeando domínios como: monitoramento ambiental (CASTILLO-EFFER et al., 2004) (WERNER-ALLEN et al., 2006), aplicações militares (SIMON et al., 2004) (YICK et al., 2005), monitoramento de seres vivos (GAO et al., 2005) (LORINCZ et al., 2004) e aplicações domiciliares. Para fins militares, aplicações de RSSF são utilizadas para monitoramento de condições de terreno, rastreamento de tropas inimigas e detecção de ataques por terra ou mar. Aplicações para fins ambientais compreendem o monitoramento de florestas, que objetiva a detecção de incêndios florestais. Existem ainda os monitoramentos de seres vivos, monitoramento meteorológico, bem como o monitoramento de edifícios e outras construções. Os avanços no *hardware* dos sensores já permite a utilização desses dispositivos para monitoramento do corpo humano, coletando informações como: pressão sanguínea e temperatura corporal (YICK et al., 2008).

Dentre os trabalhos encontrados na literatura, existem abordagem para RSSFs em seus diversos aspectos. Destaque para os recentes trabalhos desenvolvidos pelo Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat), os quais serão comentados a seguir.

O trabalho proposto em (GARCIA et al., 2012) aborda energia e segurança, uma vez que propõe um sistema de monitoramento passivo eficiente em energia para RSSF. Este trabalho compõe à tese de doutorado do autor principal, cujo trabalho já se encontra em andamento.

Outro trabalho, desenvolvido por (CARVALHO et al., 2012) e (CARVALHO, 2012b) propõe a utilização da metodologia Seis Sigma (*Six Sigma*) (WANG, 2008) para a análise e melhoria de soluções em redes de sensores.

Uma Máquina de Inferência Autônoma Distribuída (MIAD) para RSSF foi desenvolvida por (CAMPOS, 2010). A MIAD proposta utiliza lógica *fuzzy* distribuída (MARIN-PERIANU; HAVINGA, 2007) com objetivo de que os sensores autoconfigurem seus intervalos de sensoriamento e disseminação de informações, bem como para que atuem na agregação distribuída de contexto.

O protocolo *Biologically-inspired Optimization for Sensor Network Lifetime* (BiO4SeL) ou Otimização de Tempo de Vida em Redes de Sensores Biologicamente Inspirada, foi proposto em (RIBEIRO, 2009) e (RIBEIRO; CASTRO, 2010). Consiste em uma solução de roteamento autônoma, biologicamente inspirada, que faz uso de técnica de Inteligência Computacional (IC), em especial Colônia de Formigas (BING et al., 2007). Seu objetivo é maximizar o tempo de vida de RSSF.

O trabalho de (OLIVEIRA, 2011) propõe uma ampliação ao protocolo BiO4SeL, para o cenário de redes densas. São apresentadas mais duas versões do protocolo: BiO4SeL v2 e BiO4SeL v3. A versão v2 corresponde à versão original do protocolo, acrescida de uma heurística de escalonamento dos sensores. Nessa heurística, aqueles nós que possuem pouca energia residual e que são constantemente utilizados, são temporariamente desligados no escalonamento. Já na versão v3 ocorre uma extensão da versão v2, incluindo no processo de escalonamento os sensores que nunca foram utilizados nas tarefas de roteamento, obtendo ganho maior no prolongamento do tempo de vida da rede.

Outro trabalho abordando a temática de segurança em RSSF é apresentado em (CAVALCANTE et al., 2012). Neste trabalho, o autor avalia o desempenho de mecanismos de segurança, visando identificar a provisão de requisitos como: confidencialidade, integridade e autenticação. O objetivo dessa avaliação de desempenho é apontar indicadores que auxiliem na escolha dos mecanismos de segurança mais apropriados para RSSF.

Em (NETO, 2009) é proposto um Protocolo de Roteamento Sensível ao Tempo (PROST), que tem como objetivo permitir que as RSSF possam realizar suas atividades de monitoramento em ambientes críticos, onde há a necessidade de leituras com restrições de tempo real.

No domínio de computação ubíqua, o capítulo de livro apresentado por (NETO et al., 2010) discute acerca dos avanços das RSSFs. Os autores analisam ainda, como esses avanços podem ser empregáveis na criação e desenvolvimento de ambientes inteligentes, característica importante no contexto da computação ubíqua.

Sob a ótica do desenvolvimento de aplicações colaborativas e que englobem aspectos da computação autônoma, um *middleware* semântico para RSSFs autônomas é apresentado por (ROCHA, 2012). O trabalho intitulado SEMantic Middleware for Autonomic seNsor neTworkS (SEMANTK) estabelece a especificação de relacionamentos semânticos entre sensores vizinhos. Os resultados obtidos pelo SEMANTK proporcionaram uma melhoria no consumo de energia, se comparado à técnicas tradicionais de clusterização que não utilizam semântica.

Na área de redução de dados, o trabalho de (CARVALHO, 2012a) propõe um método baseado em correlação espacial e temporal multivariada para buscar melhoria na precisão da predição de dados, bem como na redução de dados em si. O trabalho realizou simulações e comparou a utilização de funções de regressão linear simples e regressão linear múltipla, a fim de verificar o desempenho de sua proposta. O método visa, de uma maneira geral, reduzir a quantidade de informações a serem enviadas para a EB, reduzindo também o volume de dados trafegados na rede.

Como é possível perceber, RSSF podem ser utilizadas para uma gama diversa de aplicações, onde cada aplicação apresenta requisitos e características bem específicos. Traba-

lhos interessantes de levantamento de aplicações podem ser encontrados também em (AKYILDIZ et al., 2002) (YICK et al., 2008) (ANASTASI et al., 2009).

Em (KO et al., 2003), os autores afirmam que para reduzir a probabilidade de que um número insuficiente de sensores sejam depositados em uma determinada área, geralmente os sensores são implantados a uma densidade maior do que se espera ser necessário. Dessa forma, os dispositivos sensores podem tirar proveito dessa elevada densidade para preservar sua energia interna, por meio do mecanismo de tempo compartilhado (*time-sharing*), estabelecendo períodos para cada um dos sensores realizarem suas atividades de sensoriamento.

O termo “rede densa” é comumente utilizado no contexto das redes RSSFs para representar uma elevada concentração de nós em uma determinada área. Contudo, esse termo pode causar o equívoco de se pensar que a população em torno de todos os sensores é alta, o que nem sempre pode ser verdadeiro (AKL et al., 2011). Considerando o caso onde a implantação dos sensores obedece uma distribuição uniforme, então todos os sensores terão aproximadamente a mesma quantidade de vizinhos. Contudo, nos casos onde a implantação não obedece uma distribuição uniforme, alguns sensores podem ter mais vizinhos ao seu redor do que outros. Dessa forma, a concentração de sensores poder ser muito elevada em uma região, bem como pode ser reduzida em outra. Neste trabalho a densidade é uma função do número de sensores por unidade de área.

Ao abordar a construção de RSSF eficientes em energia, uma das questões fundamentais é a questão da gerência de cobertura de sensoriamento (ANASTASI et al., 2009). A problemática da cobertura em RSSF será analisada na seção a seguir.

2.2.1 Cobertura em RSSF

Um problema relevante em RSSF e que tem recebido crescente interesse de pesquisa é o problema de cobertura. Cobertura é uma importante métrica de desempenho para RSSF. Haja vista que o objetivo é conseguir que cada região do espaço de interesse esteja dentro da área de sensoriamento de pelo menos um sensor. Esse problema está centrado na seguinte pergunta:

Quão bem os sensores observam o espaço onde estão implantados?

Segundo (MEGUERDICHIAN et al., 2001), o conceito de cobertura é uma medida de Qualidade de serviço (QoS) acerca da função de sensoriamento, bem como está estreitamente relacionada com o consumo de energia. Conforme já foi discutido, em alguns casos é possível obter energia do ambiente externo à rede (WANT et al., 2005). Contudo, em muitas aplicações a cenários, os nós da RSSF são depositados no ambiente de maneira completamente aleatória,

bem como a obtenção de energia externa nem sempre pode ser possível. Dessa forma, a busca por redução no consumo e uso eficiente de energia são pontos importantes para o projeto de sistemas baseados em RSSF.

Inicialmente é pertinente conceituar algumas propriedades relacionadas à cobertura em RSSF, tais como: características dos nós, modelos de sensoriamento e métricas de avaliação. A seguir são comentadas essas propriedades, de acordo com (ZHU et al., 2011).

► Características dos nós

* Nós fixos

Um nó fixo possui a habilidade de coletar dados de sensoriamento, enviar ou receber mensagens, processar esses dados e mensagens, e realizar outros tipos de atividade em uma RSSF estática. Em geral, os sensores não se movem após implantados. Vale destacar que, neste trabalho, são utilizados nós fixos.

* Nós móveis

Um nó móvel pode realizar as mesmas atividades dos nós fixos, mas também possuem certa mobilidade. No caso onde a falha de um sensor qualquer ocasiona um “buraco”, ou seja, uma região não coberta na rede, um nó móvel pode se deslocar para a região descoberta e passar a realizar o roteamento, ou mesmo a tarefa de sensoriamento.

* Cobertura e k -cobertura

Seja $S = \{s_1, s_2, \dots, s_n\}$ um dado conjunto de sensores, em uma área bidimensional D . Cada sensor s_i , ($i = 1, 2, \dots, n$), está localizado na coordenada (x_i, y_i) dentro de D , e possui raio de sensoriamento r_i . Um ponto qualquer dentro de D é dito coberto por s_i se ele está dentro do raio de sensoriamento de s_i . Por sua vez, um ponto qualquer em D é dito estar k -coberto, se ele está dentro do raio de sensoriamento de pelo menos k sensores, sendo k um inteiro qualquer. Nesta dissertação é considerado, a título de implementação da solução proposta, o caso onde $k = 1$.

* Conectividade

Supondo que dois sensores quaisquer s_1 e s_2 estão localizados internamente em D . s_1 e s_2 são ditos conectados se eles podem se comunicar entre si.

► Modelos de sensoriamento

De maneira geral, existem dois modelos de sensoriamento principais: Modelo de

sensoriamento binário, ou Booleano; Modelo de sensoriamento probabilístico. Esses modelos são descritos a seguir.

* **Modelo de sensoriamento Booleano**

Este é o modelo mais simples, onde cada sensor possui um raio de sensoriamento fixo. Outra característica desse modelo é que um sensor só é capaz de sensoriar ou detectar eventos dentro do seu raio de sensoriamento. Assim, a área onde o sensor poder atuar possui a forma de um círculo de raio r , onde a detecção é 0 ou 1, implicando na detecção ou não de evento (MEGERIAN et al., 2002).

* **Modelo de sensoriamento probabilístico**

Em ambientes reais, a detecção dos sensores é imprecisa, portanto pode ser estendido o modelo de disco Booleano, para expressar essas imprecisões em termos probabilísticos. Neste modelo, a probabilidade de percepção de um dado a ser sensoriado ou a detecção de um evento decresce à medida que a distância aumenta (FAN; JIN, 2010).

► **Algoritmos centralizados/distribuídos**

Uma vez que os sensores são implantados, um algoritmo é executado para determinar como esses sensores irão operar sobre o ambiente onde estão inseridos. Quando se refere à atividade de cobertura, o esquema de cobertura define como se comportará esse algoritmo. Em uma RSSF, um esquema de cobertura pode operar de maneira centralizada ou distribuída.

* **Algoritmo centralizado**

No caso dos algoritmos centralizados, os algoritmos de gerência de cobertura são executados em um nó central. Neste caso é necessário que cada nó sensor transfira suas informações para esse nó concentrador. Um exemplo de solução centralizada é apresentada em (CHAMBERLAND; VEERAVALLI, 2003). Neste trabalho, os autores empregam um nó coletor de dados centralizado, que funciona como a EB. Esse nó coleta os dados e também é responsável por determinar quais sensores serão desativados, de maneira a conservar energia e preservar a k -cobertura. Em (ZHU et al., 2000) é feito uso também de um nó central coletor de dados para reunir informações oriundas dos outros nós e, então, decidir quais sensores serão colocados no modo dormindo (*sleep*).

*** Algoritmo distribuído**

Os algoritmos distribuídos, por outro lado, são executados nos nós por toda a rede. Dessa forma, o algoritmo executado permite que cada nó sensor decida seu próprio modo de funcionamento, tendo como base as informações colhidas apenas dos nós de sua vizinhança. Exemplos de soluções distribuídas podem ser encontradas em (RAGHUNATHAN et al., 2002) e (HILL et al., 2000). Comparando com os algoritmos centralizados, os algoritmos distribuídos conseguem reduzir o consumo de energia na comunicação de dados, uma vez que as comunicações são em geral à curtas distâncias, dentro da vizinhança do nó. Essa redução reflete em um aumento da energia para processamento de dados nos sensores. Por outro lado, o preço que isso resulta é a relativa redução na precisão do dado sensoreado (ZHU et al., 2011).

Como em muitos casos as RSSFs possuem topologia dinâmica e necessitam acomodar um grande número de nós sensores, os algoritmos e protocolos desenvolvidos para esses casos devem ser distribuídos, de maneira a melhor acomodar uma arquitetura escalável para essas redes (CARDEI; WU, 2006). Portanto, de uma maneira geral, algoritmos distribuídos são mais adequados para redes densas, de larga escala.

► Métricas de avaliação para algoritmos de gerência de cobertura

A forma de avaliar o desempenho dos algoritmos para gerência de cobertura é de considerável importância, pois permite inferir a usabilidade e a eficácia da rede. A seguir, são considerados alguns dos fatores principais para esse tipo de avaliação.

*** QoS da cobertura**

A QoS da cobertura decide o cumprimento da tarefa principal da rede, ela reflete como a rede percebe o ambiente físico. Neste trabalho, essa métrica é considerada por meio da medição da área total média coberta pelos sensores.

*** Número de nós ativos**

Uma vez que a rede consiga atender seus respectivos requisitos de cobertura, quanto menor o número de nós ativos, mais efetiva está sendo a gerência de cobertura. Pela ótica de consumo, reduzir o número de nós ativos reflete positivamente também no desempenho do consumo de energia da rede.

*** Associação ou não com a localização dos nós**

Para que o esquema de gerência de cobertura seja capaz de inserir ao seu algoritmo informação de localização dos nós, há uma certa dependência de uma infraestrutura

externa. Essa dependência diz respeito, por exemplo, ao uso de *Global Positioning System* (GPS) ou algum outro mecanismo de localização. Essa infraestrutura geralmente apresenta um custo relativamente elevado, bem como pode inserir um maior consumo de energia na rede (ZHU et al., 2011). Ainda existem questões relacionadas à precisão das informações de localização, que por sua vez podem afetar o comportamento do algoritmo. Por essas questões que é sensato considerar um algoritmo para gerência de cobertura, que não envolva informações de posicionamento dos sensores, mais vantajoso para redes de larga escala.

* **Eficiência em energia**

Algoritmos para gerência de cobertura exigem que o consumo de energia para uma simples tarefa de sensoriamento seja o menor possível. Contudo, esse requisito não é único, uma vez que o algoritmo deve também manter o equilíbrio de energia da rede, para a execução de uma série de tarefas de sensoriamento. Neste trabalho, essa métrica é medida sob a forma da energia residual média dos sensores da rede, que é descrita na seção 4.2.

* **Sobrecarga (*overhead*) de comunicação**

Em geral, a maior fonte de consumo de energia de um nó sensor está presente na tarefa de transmissão de dados (ZHU et al., 2011). Dessa forma, o algoritmo de gerência de cobertura deve ser capaz de reduzir esse custo no processo de comunicação dos sensores.

* **Escalabilidade da rede**

O algoritmo para gerência de cobertura deve ser capaz de se adaptar à dinâmica de mudança de topologia, bem como à diferentes escalas da RSSF. Essa última métrica é avaliada neste, de maneira a verificar o comportamento do algoritmo proposto para vários níveis de densidade da rede.

A subseção a seguir aborda a relação da RSSF com a área de monitoramento onde os sensores são implantados. Alguns dos fatores relevantes para o projeto de RSSF, que influenciam na cobertura da rede, também são analisados a seguir.

RSSF e sua área de monitoramento

Seja $S = \{s_1, s_2, \dots, s_N\}$ um conjunto de N sensores depositados em uma área de tamanho $L \times W$. Haja vista que cada sensor s_i , com $i = 1, 2, \dots, N$, monitora uma determinada

região, configura-se uma cobertura completa quando toda a área está sob monitoramento dos sensores.

A figura 2.4 ilustra uma cobertura completa para uma área $L \times W$. Os sensores s_i são representados pelos pontos escuros da figura. Cada sensor cobre uma região dentro do seu raio de alcance, que possui tamanho r . Os círculos em volta dos sensores representam a região de sensoriamento respectiva de cada sensor. É possível perceber pela figura que toda a área em questão está sob monitoramento dos sensores, pois qualquer ponto interno da área está dentro da região de sensoriamento de pelo menos um sensor.

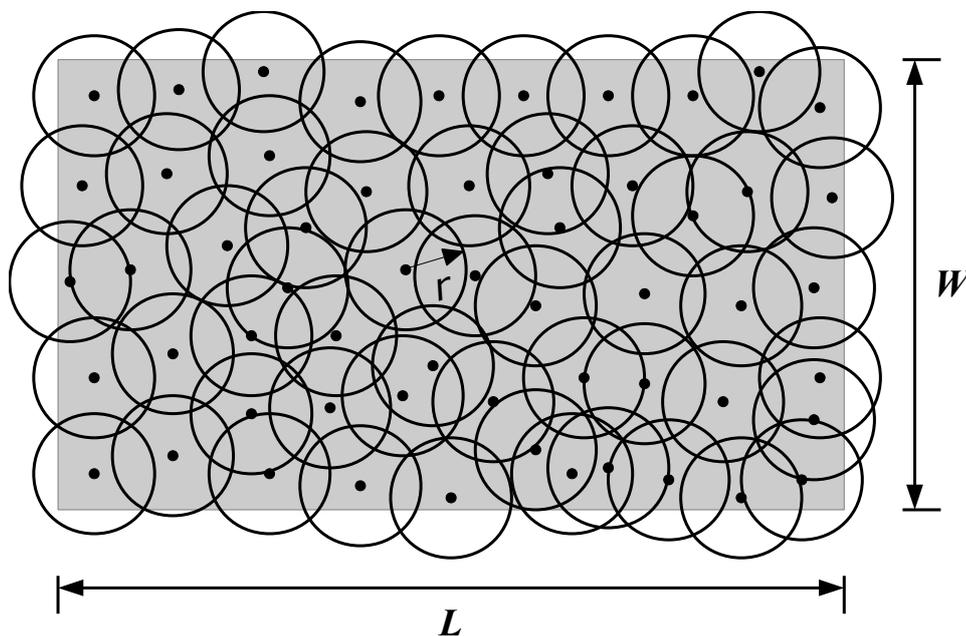


Figura 2.4: Cobertura completa para uma área $L \times W$. Adaptada de (HU et al., 2010).

Por outro lado, é possível perceber que diversos pontos dentro da área são cobertos por mais de um sensor. Nestes casos, pode ser necessário que apenas um sensor esteja cobrindo um dado ponto. Pode acontecer ainda que os sensores redundantes alternem entre si a tarefa de cobrir essa região de interseção de cobertura. Dependendo dos requisitos da aplicação, a solução de cobertura deve ser projetada de modo a atender esses requisitos, e manter a rede operando pelo tempo desejado.

Dependendo dos requisitos da aplicação, existe uma série de fatores que devem ser analisados para o projeto de esquemas de cobertura, fatores estes que afetam o desempenho de uma RSSF. A seguir, uma breve descrição de alguns desses fatores, considerados importantes, de acordo com (FAN; JIN, 2010) e (WANG, 2011).

Estratégia de distribuição (*Deployment strategy*)

Em alguns casos, os sensores podem ser distribuídos na área de interesse de maneira determinística. A disposição dos sensores é estabelecida previamente, como o caso de uma distribuição em grade, por exemplo (FAN; JIN, 2010).

Já em outros casos, os sensores podem ser distribuídos na área de interesse de maneira aleatória. Este tipo de distribuição dos sensores pode ser a única opção em casos como o de ambientes remotos ou adversos, em casos de desastres ou aplicações militares, ou ainda quando a RSSF é extremamente densa. Geralmente nesses casos, os sensores são lançados pelo ar, seja por meio de uma aeronave, que espalha os sensores na área em questão, ou ainda, os sensores podem ser lançados por artilharia.

O modelo de distribuição aleatória mais utilizado é o modelo de distribuição uniforme, onde cada nó sensor possui igual probabilidade de cair em qualquer local da área em questão, e independe dos outros nós sensores (WANG, 2011).

Grau de cobertura (*Coverage degree*)

Refere-se a como um ponto(região) qualquer da área de interesse é coberto. Se uma região é coberta por k sensores, então essa região pode tolerar a falha de até $k - 1$ sensores e continuar sendo coberta pela RSSF. O grau de cobertura pode ser tratado como um dos requisitos de cobertura da aplicação.

Taxa de cobertura (*Coverage ratio*)

Também pode ser considerada como requisito de cobertura da aplicação. Ela mede quanto da área, ou quantos alvos(*targets*) dentro da área, estão cobertos pelos sensores. Por exemplo, se 8 dos 10 *targets* estão cobertos, então a taxa de cobertura é de 80% (WANG, 2011). Como visto anteriormente, na figura 2.4, uma cobertura completa refere-se a uma taxa de cobertura de 100%.

De maneira similar existe também o caso de cobertura parcial, onde o requisito de cobertura da aplicação permite que nem todos os pontos da área estejam cobertos por sensores.

Escalonamento de atividade (*Activity scheduling*)

Ao comentar sobre grau de cobertura, exemplificamos o caso onde um ponto ou região é coberto por um ou mais sensores. Se uma região coberta por um sensor s pode também

ser coberta por outros sensores, o sensor s pode ser considerado como redundante. Neste caso, tal sensor pode ser temporariamente desligado, economizando sua energia.

O objetivo fundamental do escalonamento de atividade é decidir quais sensores estarão ativos, e por quanto tempo (WANG, 2011). Isso é feito de modo que os requisitos de cobertura da aplicação sejam atendidos e o tempo de vida da rede seja prolongado.

Características do algoritmo (*Algorithm characteristics*)

Algoritmos para cobertura podem operar tanto de forma centralizada quanto distribuída. Nos algoritmos centralizados, existe um controlador central, que pode ser um nó da rede ou uma entidade externa. Os dados obtidos pelos sensores são enviados para esse controlador, que é o responsável por tomar as decisões e retornar os resultados para os nós sensores. Já no caso dos algoritmos distribuídos, a tomada de decisão é realizada localmente em cada sensor, que utiliza apenas informações de sua vizinhança.

Até então foram discutidas essencialmente questões relacionadas ao raio de cobertura dos sensores. Contudo, principalmente na tarefa de transmissão de dados, a conectividade também é um fator determinante. Na próxima seção será discutida a relação entre cobertura e conectividade, definindo também as condições que são empregadas nesta dissertação.

Relação entre cobertura e conectividade

Conforme discutido em (WANG et al., 2003), a conectividade requer apenas que a localização de qualquer nó ativo esteja dentro do raio de comunicação de um ou mais sensores também ativos, de tal maneira que esses nós possam se conectar. Por outro lado, os autores definem também que a cobertura requer que todos os pontos dentro da área de interesse estejam dentro do raio de sensoriamento de pelo menos um nó sensor ativo. Dessa forma, é possível analisar a relação entre esses dois raios inerentes aos sensores.

Dentro dessa temática de integrar comunicação e sensoriamento, (WANG et al., 2003) e (ZHANG; HOU, 2005) provaram o seguinte resultado:

Se uma região convexa é completamente coberta por um conjunto de nós, o grafo de comunicação composto por esses nós é conexo quando $R_c \geq 2R_s$.

Isso significa dizer que, sob a condição de que $R_c \geq 2R_s$, uma RSSF necessita apenas ser configurada para garantir a cobertura. Segundo o resultado obtido pelos autores, essa

condição permite garantir a cobertura satisfaz também a garantia de conectividade. Se o raio de comunicação estabelecido for muito grande, a comunicação de rádio pode estar sujeita a excessivas interferências. Portanto, se o raio de transmissão pode ser ajustado, uma boa abordagem para garantir conectividade é configurá-lo como sendo o dobro do raio de sensoriamento (CARDEI; WU, 2006).

A figura 2.5 ilustra essa relação. Em uma RSSF, são considerados dois nós sensores S_i e S_j . Neste caso, o círculo maior de raio R_c , representa a área de comunicação do sensor S_i . Já o círculo pontilhado menor representa a área de sensoriamento do sensor S_i , por sua vez de raio R_s . É importante ressaltar que neste trabalho a RSSF é considerada homogênea, bem como que os valores de R_c e R_s são iguais tanto para S_i , S_j , bem como para todos os sensores da rede.

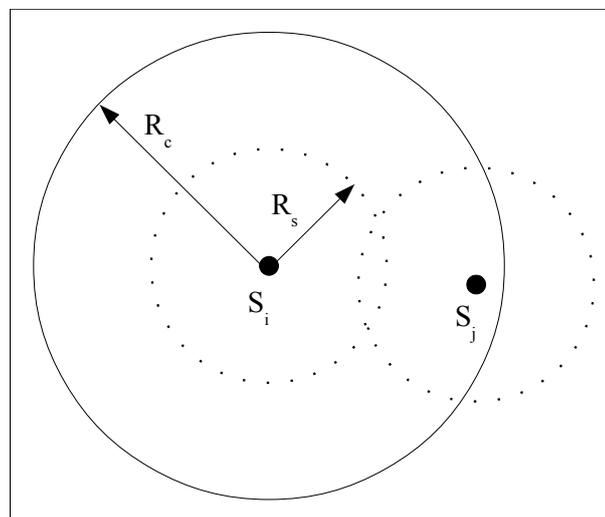


Figura 2.5: Dois sensores S_i e S_j , que sobrepõem suas áreas de sensoriamento, conseguem se comunicar entre si uma vez que $R_c \geq 2R_s$.

Dois nós são considerados vizinhos se suas áreas de sensoriamento se sobrepõem. Portanto, como mostrado na figura e a partir dos resultados obtidos pelo autores anteriormente referenciados, quando $R_c \geq 2R_s$ dois nós vizinhos estão dentro de suas áreas de comunicação.

2.3 Escalonamento de Atividade em RSSF

Uma vez que a limitação de energia de uma RSSF venha a ser um ponto vital para permitir que a rede opere por um longo período de tempo, uma abordagem interessante para conservação de energia é o estabelecimento de um esquema de escalonamento para os sensores. Os sensores no estado ativo continuam realizando as tarefas de sensoriamento e comunicação, ao passo que os nós restantes passam para o estado *sleep*, visando economizar sua energia

interna.

Dentre os trabalhos presentes na literatura, que foram levantados na revisão bibliográfica, a seguir serão feitas algumas considerações.

Em (WANG et al., 2010) é proposto um método de escalonamento distribuído *Connectivity and Coverage Maintenance Scheduling* (CCMS), que não necessita de informações de localização dos sensores. Neste trabalho, os autores definem um novo conceito de *Clique*, e a partir deste conceito os sensores são divididos em subgrupos que alternam seus estados de atividade, visando economia de energia na rede. Para os autores, um *Clique* representa um conjunto de nós sensores que são vizinhos entre si. Dessa forma, o algoritmo proposto realiza primeiro a busca por *Cliques* na rede, em seguida são atribuídos identificadores para cada um deles. Por fim, o algoritmo estabelece o esquema de escalonamento desses grupos de sensores, ativando e desativando esses conjuntos de *Cliques*.

Esse algoritmo possui algumas características semelhantes à solução proposta nesta dissertação. Características essas que possibilitaram a escolha da mesma como solução de comparação para este trabalho. Uma das características é que ambas as soluções não requerem informação de localização dos sensores. Outra característica é a inerente distribuição do algoritmo pela rede, fazendo uso da colaboração entre os sensores. Por outro lado, os autores em (WANG et al., 2010) afirmam que a maior desvantagem de sua solução é o elevado custo computacional na identificação dos *Cliques*. Diante disso, despertou-se a ideia de propor uma mecanismo que permitisse o escalonamento dos sensores, e que fosse feito de maneira menos custosa e mais robusta. Com esse objetivo, o capítulo 3 explica com maiores detalhes a solução concebida e avaliada neste trabalho.

A seção a seguir apresenta o algoritmo que inspirou a concepção da solução proposta nesta dissertação. De início, é feita uma breve introdução sobre sistemas distribuídos, situando o contexto onde o algoritmo se insere, bem como é explicado mais detalhadamente a sua execução.

2.4 Escalonamento por Inversão de Arestas

Esta seção aborda a problemática de exclusão mútua em sistemas distribuídos, apresentando o problema do jantar dos filósofos, para em seguida introduzir o algoritmo de escalonamento por inversão de arestas. Ao final da seção é mostrado um pseudocódigo desse algoritmo.

2.4.1 Sistemas Distribuídos

SDs são sistemas nos quais componentes de hardware e/ou software se comunicam e coordenam suas ações unicamente por meio da troca de mensagens, de acordo com a definição

apresentada em (COULOURIS et al., 2011). Ainda segundo os autores, uma das principais motivações para a construção e utilização de SD resulta do desejo de se compartilhar recursos. A ideia de “recurso” pode ser um tanto abstrata, uma vez que caracteriza um conjunto daquilo que pode ser compartilhado em um sistema de computadores. Nesse contexto, um sistema distribuído pode ser definido como uma coleção de processos (*processors*) interconectados por enlaces de comunicação. Em muitos casos, a comunicação entre esses processos é feita por conexões ponto-a-ponto, onde as trocas de informações são realizadas por meio do envio de mensagens através dos enlaces de comunicação.

SD abrange várias aplicações e tecnologias, sendo o domínio de aplicações de monitoramento ambiental a fonte de inspiração para o trabalho desenvolvido nesta dissertação. No contexto de monitoramento ambiental, o uso de tecnologia de sensores para monitorar e gerenciar ambientes naturais pode ser de relevante aplicabilidade, por exemplo, no diagnóstico prévio de desastres, tais como: incêndios florestais, terremotos, inundações ou *tsunami*.

Para fins de representação, neste trabalho, um sistema distribuído é representado por um grafo $G_n = (N_n, E_n)$, onde N_n é o conjunto de processos, que compartilham recurso(s). A quantidade de nós em N_n é representada por n e cada nó possui uma identificação única, onde para um nó N_i sua identificação é id_i . O conjunto de arestas do grafo E_n por sua vez, representa os enlaces de comunicação bidirecional ponto-a-ponto. Uma mensagem recebida por um processo, geralmente é formada pelo par (p, Msg) , representando que a mensagem Msg deve ser entregue ao processo p .

Quando os nós em G_n compartilham recursos uns com os outros, de maneira que um mesmo recurso não pode ser acessado por mais de um nó simultaneamente, surge a necessidade de algoritmos distribuídos para garantir a exclusão mútua (*mutual exclusion*) no acesso a esses recursos compartilhados. Na literatura esse caso é denominado de problema de exclusão mútua (*mutual exclusion problem*) (LAMPOR, 1986a) (LAMPOR, 1986b).

Dentre os problemas de exclusão mútua presentes na literatura, pode ser destacado o problema do jantar dos filósofos - *Dining Philosophers Problem* (DPP) - (KRISHNAPRASAD, 2003), que está relacionado ao compartilhamento de recursos. O DPP pode ser descrito da seguinte forma:

Cinco filósofos estão sentados ao redor de uma mesa, cinco garfos estão sobre a mesa de maneira que, para cada dois filósofos adjacentes, existe um garfo entre eles. Um filósofo necessita de ambos os garfos que estão ao seu lado para poder comer. Outra restrição é que dois filósofos vizinhos (adjacentes) não podem comer simultaneamente. Neste problema, os filósofos se encontram em um ciclo alternativo entre “pensando”, “comendo” e “com fome”.

A figura 2.6 ilustra como se configura o problema DPP, definido anteriormente. Nela é possível visualizar como estão dispostos os cinco filósofos, bem como os garfos que representam os recursos compartilhados pelo filósofos adjacentes.

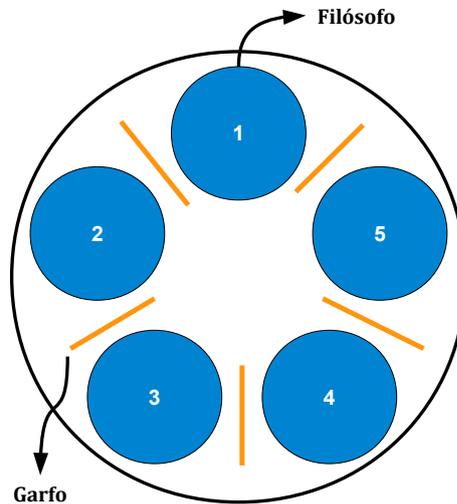


Figura 2.6: Ilustração para o *Dining Philosophers Problem*.

Uma solução para o problema consiste em um algoritmo que garanta a exclusão mútua, uma vez que um garfo pode vir a ser usado por um filósofo de cada vez. A solução deve ainda evitar a condição de *starvation*, pois cada filósofo deve conseguir comer em um tempo finito após entrar no estado “com fome”. Na representação definida anteriormente, para este problema G seria o conjunto de $n = 5$ filósofos, e cada aresta de E corresponderia aos garfos que são adjacentes à cada nó N_i . De uma maneira geral, para poder comer, um filósofo necessita obter os seus dois garfos adjacentes. Um garfo representa o recurso compartilhado por dois filósofos vizinhos. Isso significa que, adquirindo cada um dos garfos adjacentes é equivalente a garantir a exclusão mútua no acesso ao recurso que o nó necessita (BARBOSA, 1996).

2.4.2 Algoritmo EIA

O Escalonamento por Inversão de Arestas (EIA) - em inglês *Scheduling by Edge Reversal* - é apresentado em (BARBOSA, 1986), (BARBOSA; GAFNI, 1989) e (BARBOSA, 1993). Esse algoritmo também pode ser encontrado na literatura sob a nomenclatura de Escalonamento por Reversão de Arestas (ERA).

O algoritmo EIA é baseado na ideia de uma população de processos que realizam acesso a recursos compartilhados. Sistemas de recursos compartilhados podem ser representa-

dos por um grafo $G = (N, E)$, onde os processos são os nós de G e existe uma aresta orientada entre dois nós quaisquer sempre que estes compartilham recurso. Dessa forma, um recurso é representado por um clique, ou seja, um subgrafo de G totalmente conectado (FRANÇA; FARIA, 1995).

O algoritmo 1, a seguir, reproduz em pseudocódigo uma versão do EIA, apresentada em (BARBOSA, 1996).

Algoritmo 1: Pseudocódigo do algoritmo EIA.	
Dados:	P_i^j para todo $j \in N_{viz}^i$;
Entrada:	$msgProc \leftarrow$ mensagem oriunda de n_j para n_i
1	recebe ($msgProc \leftarrow n_j$);
2	$P_i^j \leftarrow$ verdadeiro ;
3	se $\forall_{j \in N_{viz}^i} P_i^j =$ verdadeiro ;
4	então
5	Acessa os recursos compartilhados;
6	para todo $j \in N_{viz}^i$ faça
7	$P_i^j \leftarrow$ <i>falso</i> ;
8	envia ($msgProc \leftarrow j$);
9	fim
10	fim

No algoritmo, P_i^j representa os processos P_j que compartilham recurso com processo P_i ; já N_{viz}^i representa a vizinhança em torno do nó n_i ; e por fim, $msgProc$ representa as mensagens trocadas entre os processos de cada nó.

Para ilustrar a execução do algoritmo, a figura 2.7 apresenta o funcionamento do EIA. O grafo é composto por cinco nós ($|N| = 5$), as setas indicam o sentido das arestas do grafo. É pertinente lembrar que os nós que possuem arestas entre si compartilham de um mesmo recurso. Por exemplo, os nós 1 e 3 compartilham recurso, ao passo que os nós 2 e 4 não compartilham, por exemplo. Os nós em negrito representam aqueles nós que foram selecionado pelo escalonamento para acessarem recurso.

Como é feita essa seleção?

Serão selecionados para acessar recurso os nós que possuírem todas as arestas apontando para si mesmos e nenhuma aresta apontando para outro nó. Essa condição de seleção é importante, pois será também utilizada na solução proposta nesta dissertação.

Inicialmente, os nós 3 e 5 estão selecionados, após estes nós acessarem o recurso, eles inverterão as suas arestas. Neste caso, as arestas do nó 3 passarão a apontar para os nós 1, 2 e 4 respectivamente. Já as duas arestas do nó 5 passarão a apontar para os nós 2 e 4 respectivamente. É possível perceber que, agora, os nós 2 e 4 possuem todas as suas arestas apontando para si próprios, logo estes nós serão os selecionados para acessar recurso. O processo de inversão de arestas é novamente repetido, lembrando que só invertem arestas aqueles nós que acabaram de acessar recurso. Portanto, têm-se os nós 1 e 5 selecionados para acessar recurso e, após esse processo, tem-se o nó 3 novamente sendo selecionado.

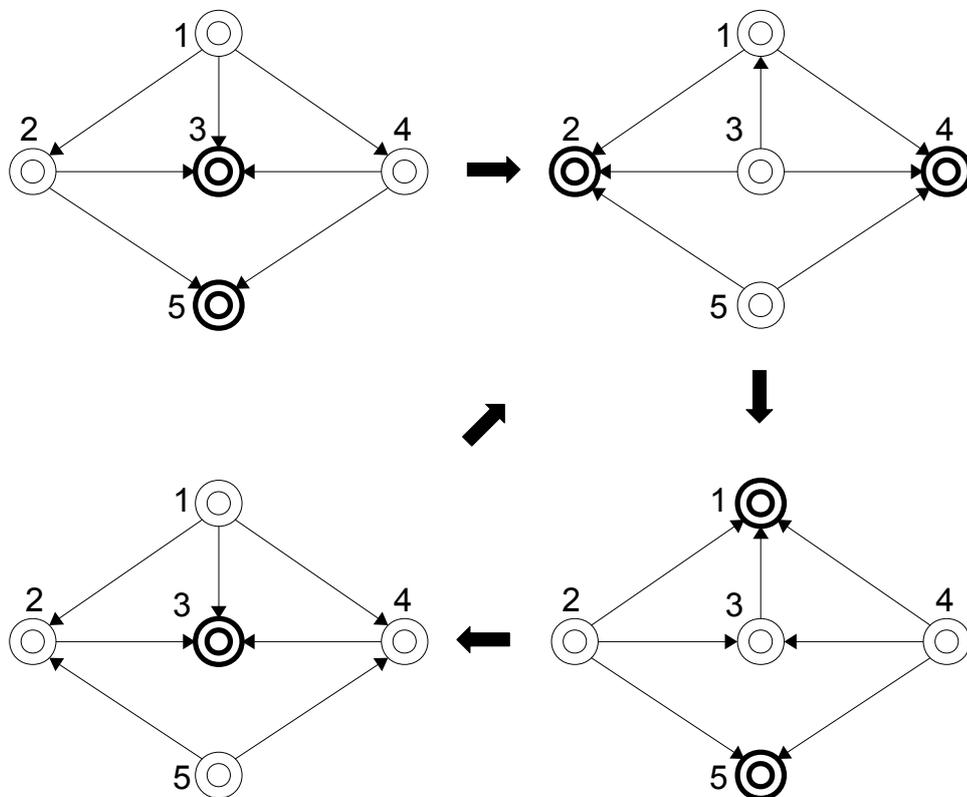


Figura 2.7: Funcionamento do Algoritmo EIA.

É pertinente destacar uma propriedade importante do EIA, um nó que foi escalonado em um dado instante só voltará a ser escalonado outra vez após todos os seus vizinhos (leia-se: nós que compartilham o mesmo recurso que ele) também terem sido escalonados. De fato, tomando como exemplo o nó 3, que é inicialmente escalonado, somente é escalonado outra vez após seus vizinhos, que são os nós 1, 2 e 4.

O capítulo 3 a seguir apresenta a proposta desta dissertação, após o levantamento bibliográfico que foi apresentado neste capítulo.

3 ESQUEMA DE ESCALONAMENTO DISTRIBUÍDO: DCS²

O trabalho proposto nesta dissertação aborda a problemática de cobertura em RSSF, desenvolvendo um mecanismo de escalonamento distribuído e colaborativo, tendo como fonte de inspiração o algoritmo EIA, apresentado no capítulo 2. Dessa forma, o algoritmo proposto neste trabalho é intitulado *Distributed and Collaborative Scheduling Scheme* (DCS²), ou Esquema de Escalonamento Distribuído e Colaborativo.

As seções a seguir definem o problema abordado pelo algoritmo DCS² (seção 3.1); como se deu o processo de adaptação do EIA para RSSF; quais abstrações foram necessárias (seção 3.2); por fim apresenta o pseudocódigo descrevendo o algoritmo proposto (seção 3.3).

3.1 Definição do problema

O problema a ser abordado está definido da seguinte maneira:

Em uma área a ser monitorada, de dimensões L e W , é depositado de maneira aleatória um conjunto $S = \{s_1, s_2, \dots, s_n\}$ de n sensores. O problema associado consiste em estabelecer o esquema de escalonamento desses nós, alternando o estado de atividade de sensoriamento de cada um, de forma a prolongar o tempo de vida dessa RSSF.

Ao longo de todo este trabalho, cada dispositivo sensor é representado por um nó e o conjunto desses nós é denotado por $N = \{n_1, n_2, \dots, n_n\}$. Para simplificar a notação, neste trabalho são utilizadas as letras i e j para identificar os respectivos nós n_i e n_j . Os valores de i e j estão no intervalo $[1, n]$. O número total de nós da rede é denotado por $n = |N|$.

Uma vez definido o problema, é necessário descrever o processo de concepção da solução proposta. As soluções que propuseram mecanismos de garantia de exclusão mútua para o domínio de sistemas distribuídos, apresentadas na seção 2.4.1 serviram como fonte de inspiração para este trabalho. A solução escolhida foi o algoritmo EIA, descrito na seção 2.4.2. Esse algoritmo foi escolhido por ser uma solução robusta, adequada para o escalonamento de processos e que, por ter sua descrição nos trabalhos levantados na literatura, possibilitaria o processo de adaptação da ideia principal do algoritmo para o contexto de RSSF.

A seção a seguir explica com maiores detalhes como foi realizado esse processo de adaptação do EIA para RSSF.

3.2 Adaptação do EIA para RSSF

Esta seção apresenta as definições do algoritmo proposto. Inicialmente, para adaptar o EIA foi necessário visualizar o contexto de exclusão mútua de SD, em seguida trazer a ideia principal para o contexto das RSSFs.

Revisitando o que foi apresentado sobre o algoritmo EIA na seção 2.4.2, para uma melhor compreensão será utilizada a mesma representação aqui neste capítulo. Dessa forma, a RSSF é representada sob a forma de um grafo $G = (N, E)$, onde os nós da rede correspondem ao vértices do grafo, que fazem parte do conjunto G . Já o conjunto E representa as arestas direcionadas do grafo. Na solução proposta, existe uma aresta ligando dois nós sensores sempre que estes compartilham recurso.

Contudo, qual seria o recurso compartilhado pelos sensores?

É exatamente nessa situação onde há o primeiro diferencial deste trabalho. A condição de concorrência, que define o recurso compartilhado entre dois sensores quaisquer, ocorre quando as áreas de sensoriamento desses sensores se sobrepõem. Todos os sensores da rede possuem o mesmo raio de sensoriamento r . Logo, por meio do valor de r é possível calcular a distância relativa entre os sensores e identificar se as suas áreas se sobrepõem. Além de identificar o quanto elas se sobrepõem. O trabalho de (MISRA et al., 2011) define as fórmulas matemáticas de como realizar esse cálculo, e é uma leitura recomendada para maiores detalhes.

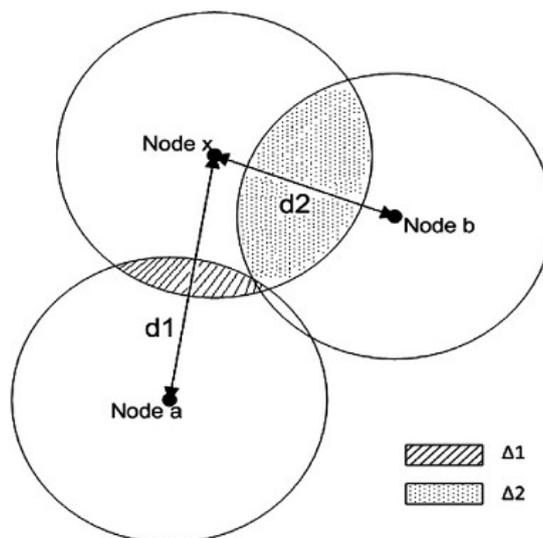


Figura 3.1: Exemplo - sobreposição de áreas de sensoriamento. Fonte: (MISRA et al., 2011).

A figura 3.1 ilustra um exemplo onde três nós possuem partes de suas áreas de sensoriamento em sobreposição com sensores vizinhos. É possível perceber que a interseção das áreas entre os nós b e x é maior do que a interseção entre x e a . No contexto de cobertura adotado neste trabalho, quando há interseções entre sensores implica dizer que há um certo nível de redundâncias, que esses sensores monitoram uma mesma área. Com base nisso, é inferido que um desses sensores pode ser colocado para dormir, ao passo que o outro sensor cobrindo a mesma área permanece ativo, realizando suas tarefas de sensoriamento.

O segundo diferencial do algoritmo está no processo de inversão de arestas. Neste trabalho, a inversão de uma aresta é feita pelo envio de uma mensagem do nó que está com a aresta apontando para si, para aquele nó onde a aresta passará a apontar.

Outro diferencial nessa adaptação é a inserção da energia residual do sensor como variável de decisão no processo de inversão de arestas. No EIA o processo que foi escalonado e acessou o recurso em seguida inverte suas arestas. Contudo, no algoritmo proposto essas inversões estão condicionadas à energia residual do sensor escalonado em relação aos seus vizinhos.

Conforme definido anteriormente, a rede é representada por um grafo, onde as arestas representam a relação de concorrência entre nós. Para estabelecer essa relação de concorrência foi necessário definir o grau de sobreposição dos nós para estes serem considerados concorrentes. Esse grau de sobreposição reflete o nível de redundância das áreas de sensoriamento dos nós.

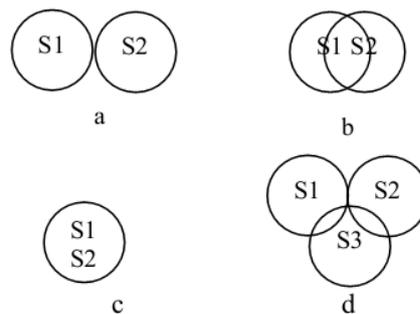


Figura 3.2: Exemplos de grau de cobertura. Fonte: (ANCUSA; BOGDAN, 2011).

A figura 3.2, apresenta diferentes casos de grau de sobreposição. Na figura a) não há sobreposição, logo o grau de sobreposição entre S1 e S2 é 0%. Em b) o grau de sobreposição é de 50%. No caso c) S1 e S2 estão completamente sobrepostos, o que implica em um grau de sobreposição de 100%. Já no caso d) os graus de sobreposição são 0% entre S1 e S2, 15% entre S1 e S3 e 15% entre S2 e S3, respectivamente.

Para os experimentos realizados (seção 4), foi definido o limiar de sobreposição a ser considerado na definição da condição de concorrência com valor de 30%. Este valor foi escolhido a partir do trabalho apresentado em (ANCUSA; BOGDAN, 2011).

Revisitando a figura que ilustrava o funcionamento do algoritmo EIA, na seção 2.4.2, a seguir é apresentada a figura 3.3. A figura ilustra a inclusão da energia residual de cada nó como variável de decisão antes de ser realizada a inversão de aresta. Para essa inclusão, foi desenvolvida uma heurística, a qual estabelece que ao inverter suas arestas, o sensor avalia se sua energia residual é maior ou igual à energia residual de cada um de seus vizinhos/concorrentes. Dessa forma, um sensor que está com todas as arestas apontadas para si vai então decidir se inverte suas arestas. Neste caso, ele avalia se sua energia residual é maior ou igual à energia de seus vizinhos. Em caso afirmativo, o sensor não inverte suas arestas e permanece ativo. Em caso negativo, o sensor inverte suas arestas e passa para o estado *sleep*, durante um período determinado.

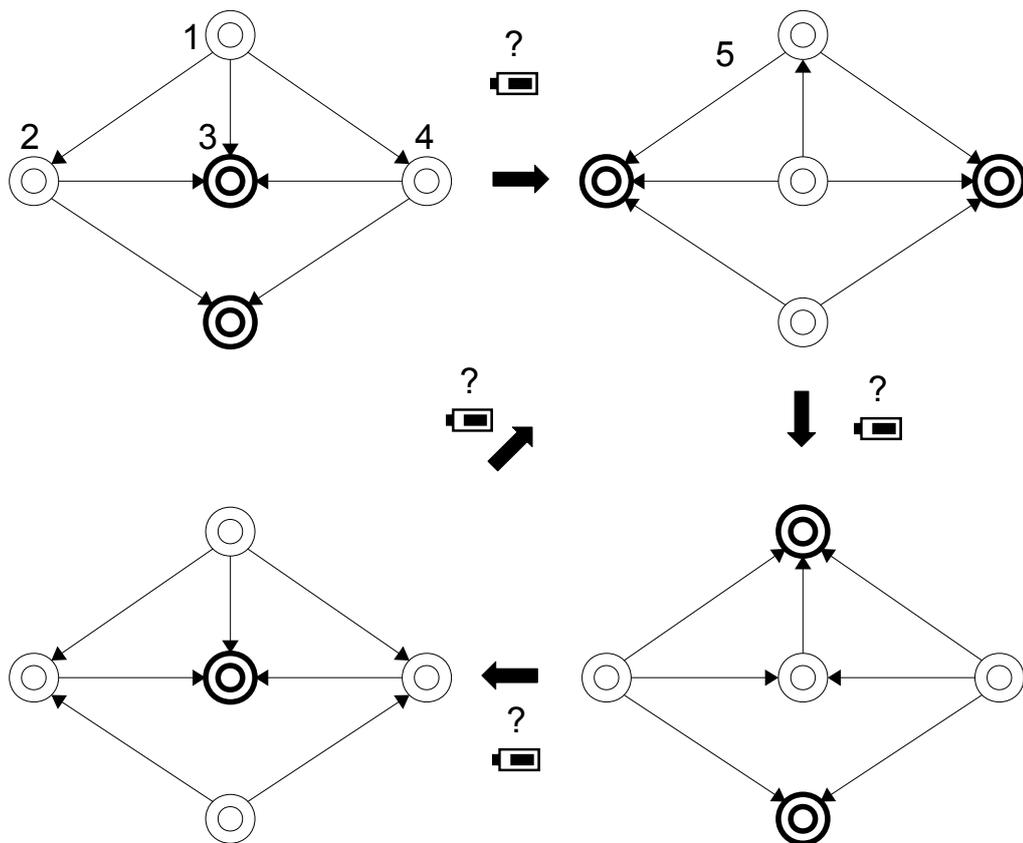


Figura 3.3: Exemplo - adaptação EIA para RSSF.

O período em que o nó permanece no modo *sleep*, com sua interface de sensoriamento desligada, é proporcional a sua energia residual. Conforme (OLIVEIRA, 2011), foi necessário empregar a fórmula apresentada nesse trabalho para determinar o período T em que um nó n_i ficará no estado *sleep*. Logo a fórmula que calcula T é definida a seguir:

$$T = \frac{\text{media}(E_{viz}^i)}{E_i} \times \text{sleeptime} \quad (3.1)$$

Na fórmula, T representa o tempo em que o nó ficará no estado *sleep*; $\text{media}(E_{viz}^i)$ representa a média de energia residual da vizinhança de n_i ; E_i representa a energia residual interna de n_i . Assim, como sugerido também no trabalho referenciado, foram feitos vários testes para se ajustar um valor adequado para a constante *sleeptime*. Durante os teste foram testados valores no intervalo entre 10 e 50 segundos, neste intervalo os melhores resultados foram obtidos com valor de 30 segundos. Após esses testes, foi definido o valor de 30 segundos para o tempo médio que cada nó deve permanecer no estado *sleep*.

O objetivo do DCS² é prolongar o tempo de vida da rede. Com esse objetivo em mente, o período de tempo que um nó deve permanecer no estado *sleep* deve ser definido de maneira que quanto menor a sua energia residual, em relação a seus concorrentes, maior será o período que ele passará no estado *sleep*. Apesar de ter sido definido esse valor como constante, será o valor de T , proporcional à energia residual, que irá realmente definir o tempo.

Na seção a seguir, o algoritmo DCS² é apresentado como pseudocódigo, bem como é explicado o seu funcionamento.

3.3 Algoritmo Proposto: DCS²

Inicialmente, é relevante explicar uma necessidade que surgiu durante o desenvolvimento dessa solução. Identificou-se que era necessário delimitar o número de sensores vizinhos/concorrentes que realizam as inversões de arestas entre si. Esta necessidade está relacionada à topologia dinâmica da rede. Dessa forma, o algoritmo realiza um agrupamento dos sensores. Contudo é importante ressaltar que não há o papel do nó central (*cluster head*) nesses agrupamentos, a ideia é unicamente de delimitar para quais sensores um determinado sensor deve enviar suas mensagens de inversão de arestas. Dessa forma, é executado o algoritmo 2, descrito em pseudocódigo a seguir. Este algoritmo realiza apenas o agrupamento de sensores vizinhos, que conseguem se comunicar diretamente, marcando os mesmos em seus respectivos grupos. O algoritmo foi retirado do trabalho apresentado por (BARI et al., 2008).

Algoritmo 2: Agrupamento inicial no DCS². Fonte: (BARI et al., 2008).

```

Dados:  $C^i \leftarrow$  para todo  $i \in N$ 
1 para todo  $n_i \in N$  faça
2   para todo  $n_j \in N$  faça
3     se  $n_j$  pode se comunicar com  $n_i$  então
4        $C^i \leftarrow C^i \cup \{n_j\}$ 
5        $C \leftarrow C - \{n_j\}$ 
6     fim
7   fim
8 fim
9 retorna  $C^i, \forall i \in N$ 

```

Inicialmente, o algoritmo verifica para cada nó n_i se este consegue se comunicar com um nó n_j qualquer. Caso seja possível essa comunicação, o nó n_j é incluído ao grupo C^i . O conjunto C representa os nós que ainda não fazem parte de nenhum agrupamento. Assim, o nó agrupado n_j é retirado de C e o algoritmo volta a ser executado pelo nó n_i .

Uma vez delimitados os grupos, cada sensor executa o DCS² para os nós que estão em seu respectivo grupo. O algoritmo 3 a seguir descreve, em pseudocódigo, a execução do DCS².

Quando um nó que está ativo vai realizar o processo de inversão de suas arestas, ele envia uma mensagem de inversão para os nós vizinhos, dentro de seu grupo. O algoritmo abaixo descreve a execução de cada nó que está com sua interface de sensoriamento desativada, ao receber uma mensagem de inversão, para decidir se passará ou não para o estado *sleep*. O sensor n_i recebe a mensagem de inversão *msgInv* do sensor n_j (linha 1), em seguida marca como recebida essa mensagem em uma tabela que contém os seus vizinhos (linha 2). Se n_i recebeu mensagens de inversão de todos os seus vizinhos, ou seja, todas as suas arestas estão apontando para ele próprio, n_i compara então se sua energia residual E_i é maior ou igual às energias residuais de seus vizinhos (linha 3). Caso seja verdadeira essa verificação, o nó ativa sua interface de sensoriamento (linha 4). Após o período que o nó permanece ativo, é realizado novamente o processo de inversão de arestas, onde o nó n_i passará agora a enviar mensagens de inversão para cada um de seus vizinhos n_j (linhas 4 a 8).

Algoritmo 3: Algoritmo DCS².

Dados: N_i^j para todo $j \in N_{viz}^i$
Entrada: $msgInv \leftarrow$ mensagem do sensor n_j para o sensor n_i

- 1 **recebe**($msgInv \leftarrow n_j$);
- 2 $N_i^j \leftarrow$ *verdadeiro*;
- 3 **se** $\forall_{j \in N_{viz}^i} N_i^j = \textit{verdadeiro} \ \& \ E_j \leq E_i$ **então**
- 4 Ativa interface de sensoriamento.;
- 5 **para todo** $j \in N_{viz}^i$ **faça**
- 6 $N_i^j \leftarrow$ *falso*;
- 7 **envia**($msgInv \rightarrow n_j$);
- 8 **fim**
- 9 **fim**

Essa foi a adaptação realizada sobre o algoritmo original de inversão de arestas. Conforme mencionado anteriormente, a energia residual dos sensores passa a ser levada em consideração no processo de decisão se o nó irá ou não ativar sua interface de sensoriamento.

O capítulo 4 apresenta como foi realizada a implementação do DCS², como foram configurados os experimentos, justificando a escolha dos parâmetros. O capítulo apresentará também a análise dos resultados obtidos.

4 EXPERIMENTOS

Após definir o modelo proposto nesta dissertação, no capítulo anterior, a etapa seguinte consiste na implementação da proposta e análise dos resultados obtidos. Dessa forma, ao longo deste capítulo será descrita a metodologia empregada para avaliar o desempenho do DCS², sendo feita a análise dos resultados, comparando-os com a abordagem CCMS, referida anteriormente no capítulo 2.

A seção 4.1 apresenta uma descrição do plano de testes elaborado para a realização dos experimentos, especificando também os parâmetros utilizados nas simulações. A seção 4.2 estabelece as métricas a serem consideradas para analisar o desempenho das soluções implementadas, em cada um dos experimentos do plano de testes. Essa seção explica ainda o que cada uma das métricas reflete sobre o comportamento da RSSF. Na seção 4.3, são apresentados os gráficos com os resultados obtidos, bem como é feita uma análise sobre esses resultados. Finalmente, na seção 4.4, são feitas algumas considerações sobre o capítulo.

4.1 Descrição do Plano de Testes

Esta seção descreve o plano de testes que foi seguido, bem como apresenta as configurações da rede e os parâmetros utilizados.

A solução proposta nesta dissertação - o DCS² - foi implementada, bem como os experimentos foram realizados, no simulador de redes Network Simulator - versão 2 (NS-2) (NS-2, 2012).

Uma vez que o objetivo deste trabalho é prolongar o tempo de vida da rede, por meio do gerenciamento da cobertura, os experimentos foram realizados de maneira que permitisse identificar esse comportamento desejado.

De maneira a melhor avaliar o comportamento do DCS², três redes foram implementadas. Estas redes correspondem aos três casos, que compõem o plano de testes estabelecido para esta avaliação, a saber:

Caso (i) – Todos os nós da RSSF ativados

Corresponde ao caso onde não há mecanismo de escalonamento, todos os nós da rede estão permanentemente no estado ativo.

Este caso foi definido para se verificar a cobertura e o tempo de vida quando a rede não emprega nenhum mecanismo de escalonamento de atividades sobre nós. O objetivo consiste em, ao inserir o mecanismo de escalonamento, seja possível obter ganho no tempo de vida da rede.

Caso (ii) – Implementação do CCMS

Corresponde à implementação da solução de comparação, o algoritmo CCMS, de acordo com as descrições presentes no respectivo artigo.

Caso (iii) – Implementação do DCS²

Corresponde à implementação da solução proposta neste trabalho, de acordo com as especificações apresentadas no capítulo 3. O objetivo é comparar o comportamento do DCS² com os casos de teste (i) e (ii).

É importante ressaltar que os casos do plano de testes foram configurados nas mesmas condições, ou pelo menos o mais próximo possível daquelas encontradas no artigo que descreve a solução escolhida para comparação. Isso foi feito para que a avaliação entre ambas soluções, DCS² e solução de comparação, pudesse ser feita de maneira equivalente.

Outro ponto está relacionado à fase inicial de implementação do plano de testes. Durante a fase inicial de experimentação, foram realizadas simulações com a finalidade de enriquecer a solução proposta. Essas simulações foram feitas com o objetivo de buscar identificar, nos estágios iniciais da implementação, possíveis inconsistências. Dentre as inconsistências procuradas estaria o possível ajuste equivocado de algum parâmetro, ou outros possíveis erros de implementação. Esse processo foi realizado no início, visando maturar o algoritmo proposto. Uma vez com o modelo mais amadurecido, por meio dessas simulações iniciais, foram então definidos de fato os valores que configuram a implementação e os resultados apresentados neste trabalho, os quais serão descritos adiante.

Após essa etapa de definição dos parâmetros, de como realmente se configuraram os experimentos, são listados abaixo os valores de configuração utilizados:

- ▶ Área a ser monitorada (dimensões $L \times W$): 250 x 250.
- ▶ Número de nós implantados na área (N): variando de 200 a 500 nós.
- ▶ Tempo de simulação: 500s.
- ▶ Número de cenários gerados (aleatoriamente): 30.

As dimensões L e W da área a ser monitorada, bem como o número N de nós a serem implantados foram parâmetros estabelecidos a partir daqueles referentes à solução de comparação. Foram escolhidos os mesmos valores para se aproximar das mesmas condições.

Com relação ao tempo estabelecido para cada execução da simulação, o valor listado anteriormente foi obtido na etapa inicial de simulações. Quando definido um valor menor do que 500s, os resultados obtidos não refletiam tão claramente o ganho obtido pelo DCS². Vale ressaltar também que se definido um valor muito maior do que 500s, existiria a possibilidade de resultar em algum comportamento diferente. Contudo, isso não foi verificado por enquanto neste trabalho. Sobre o número de cenários e simulações executadas, foram gerados de maneira aleatória 30 cenários(amostras) para cada um dos casos de teste. Em cada um dos cenários gerados é relevante considerar que a disposição dos pode ser diferente, uma vez que a geração foi realizada de maneira aleatória. Uma vez gerados, para cada um dos cenários, em cada um dos casos de teste, foram executadas 30 simulações. Em seguida, foram calculadas os valores médios das amostras, que serão apresentados e discutidos com mais detalhes na seção 4.3. A escolha do valor 30 para a quantidade de amostras tem como base o apresentado em (JAIN, 1991), que afirma ser 30 o número mínimo de amostras requeridas para se obter um intervalo de confiança para a média de uma população de dados. Para obter uma boa estimativa do valor médio calculado pelas simulações, foi estabelecido o intervalo de confiança com nível de confiança α de 95% (JAIN, 1991).

Tabela 4.1: Densidades empregadas nas simulações.

Quantidade total de nós (n)	Área total (área)	Densidade aproximada (n/área)
200	250x250	0,0032
250		0,0040
300		0,0048
350		0,0056
400		0,0064
450		0,0072
500		0,0080

A tabela 4.1 apresenta os valores de densidade empregados nas simulações. Conforme definido na seção 2.2, a densidade é uma função do número total de sensores por unidade de área. Logo, para cada ponto unitário de área existe esses valores de densidade. Tais valores foram obtidos a partir de experimentos, definindo o tamanho da área e a quantidade de nós de maneira a minimizar a geração de cenários onde a rede estivesse desconexa. Haja vista que os cenários são gerados aleatoriamente na ferramenta de simulação, existe a possibilidade de serem gerados cenários desconexos. Isso justifica a necessidade e a importância das simulações iniciais, descritas anteriormente, que buscam minimizar inconsistências, entre elas reduzir a probabilidade de serem gerados cenários desconexos. Como explicado na definição do algo-

ritmo, na seção 3.3, o algoritmo de inicialização realiza, além da atribuição de identificadores, essa verificação de conectividade. É importante ressaltar que todos esses processos foram realizados tendo em mente também a busca por reproduzir condições semelhantes ao cenário da solução de comparação.

Com relação à identificação dos sensores, cada nó apresenta um identificador único estabelecido de 0 a $n - 1$, onde n é o número total de nós da rede. Em todos os cenários gerados, o tráfego é gerado a partir de uma única fonte fixa, em direção a uma única EB existente. Semelhante ao trabalho desenvolvido por (OLIVEIRA, 2011), a fonte e a RSSF são identificadas como $n - 2$ e $n - 1$, respectivamente. Outra semelhança é que esses nós estão localizados em pontos extremos da área, de maneira a estimular a utilização da maior parte possível de sensores.

Tabela 4.2: Configurações padrão utilizadas no simulador.

Parâmetro	Valor
Canal de transmissão	Sem fio
Modelo de propagação de rádio	<i>TwoRayGround</i>
Camada Física	IEEE 802.15.4
Tipo de fila de interface	<i>PriQueue</i>
Tipo de camada de enlace	<i>LL</i>
Limite máximo para a fila de interface	50
Parâmetro energético de envio	0,660
Parâmetro energético de recepção	0,395
Parâmetro energético de estado ligado	0
Antena	OmniAntenna
Alcance de antena	15m
Modelo de energia	<i>EnergyModel</i>
Energia inicial dos nós	1
Energia inicial da EB	2

Outros valores específicos da ferramenta de simulação também precisaram ser configurados. A tabela 4.2 sumariza os demais parâmetros, que foram mantidos nos seus valores padrão do NS-2. Os valores referentes à parte de energia, no final da tabela, também foram deixados tal qual os valores padrão do NS-2. Muitos desses parâmetros são semelhantes aos utilizados por (OLIVEIRA, 2011).

Ainda com relação aos parâmetros de energia, a energia inicial de cada nó sensor foi definida como 1 para facilitar a visualização nos gráficos da seção 4.3.2. Portanto, a energia residual de cada nó varia de maneira decrescente de 1 a 0. A energia da EB, por sua vez, foi definida como sendo o dobro da energia dos nós para representar que a mesma, em geral, possui energia inesgotável ou pelo menos bem maior que a energia dos sensores. Dessa forma, o objetivo é reduzir a possibilidade de que a energia da RSSF se esgote enquanto a rede como um todo ainda esteja ativa.

Definidos os parâmetros envolvidos, bem como descritos os experimentos, a seção a seguir apresenta as métricas avaliadas sobre os resultados.

4.2 Métricas de Avaliação

Esta seção estabelece as métricas que foram consideradas para a análise dos resultados obtidos nas simulações dos casos de teste. O objetivo desta seção é também justificar a escolha dessas métricas, bem como explicar o que cada uma delas reflete em relação ao comportamento da rede.

Cobertura

Porcentagem da área total que é coberta pelos sensores ativos. Reflete diretamente o quão bem os sensores estão monitorando a área em questão. Mostra se os sensores que estão ativos realizando tarefas de sensoriamento, em conjunto, conseguem cobrir totalmente ou grande parte da área. Sendo a solução com ou sem mecanismo de escalonamento. Conforme discutido na seção 2.2.1, o conceito de cobertura é uma medida de QoS que estabelece a função de utilidade da rede, sobre a realização de sensoriamento, sendo este conceito estreitamente relacionado com o consumo de energia.

Energia residual

Calculada como sendo a média da energia residual de todos os nós da rede, ativos ou não. Reflete o comportamento da solução aplicada no que concerne ao consumo de energia dos nós, o quanto de energia estes nós gastam, internamente e/ou na comunicação com outros nós. Sobre isso, se o nós interagem com seus vizinhos, esta métrica está relacionada também com a energia consumida nessas interações. Essa métrica também

sofre influência com a densidade da rede, uma vez que o aumento da densidade resulta em uma maior interação entre sensores, ocasionando aumento no consumo de energia por parte deles.

Tempo de vida da rede

Conforme definido anteriormente, na seção 2.2, o tempo de vida reflete quanto a rede se mantém operante até que se esgote a energia de um sensor e este pare de funcionar, podendo ocasionar "buracos" na rede. Neste trabalho, um "buraco" é um espaço dentro da área a ser monitorada onde, por não existir(em) sensor(es) funcionando por perto, não é possível monitorar por nenhum sensor. Neste trabalho, o tempo de vida da rede foi medido e calculado em função da quantidade de total de nós implantados na área.

Tendo sido definidas as métricas a serem consideradas, na seção a seguir, são realizadas a apresentação e a análise dos resultados obtidos com os experimentos realizados.

4.3 Análise dos Resultados Obtidos

Conforme definidas as métricas na seção anterior, a seguir serão apresentados os gráficos com os resultados obtidos após a implementação dos três casos de teste, que foram detalhados seção 4.1.

A análise dos resultados se faz necessária também para verificar se os objetivos previamente traçados foram alcançados, tais como: taxa de cobertura, economia de energia e melhoria no tempo de vida da rede. Dessa forma, os gráficos estão divididos nas seguintes subseções: Área coberta pelos sensores; Energia residual média e Tempo de vida.

4.3.1 Área Coberta pelos Sensores

Conforme apresentado anteriormente, na seção 4.2, a cobertura será avaliada neste trabalho como sendo a porcentagem da área total, que está sendo coberta pelos sensores ativos.

Inicialmente, foi simulada uma rede com todos os nós ativos continuamente, e sem qualquer mecanismo de escalonamento. Este compreende ao caso (i) do plano de testes. O gráfico na figura 4.1 apresenta a comparação entre este caso com todos os nós ativos, o algoritmo proposto DCS² e a solução CCMS, no que concerne à cobertura. É possível perceber que, para o caso (i), a maioria dos cenários conseguiu uma cobertura completa (seção 2.2.1). Esse resultado deve-se ao fato de que com todos os sensores ativos foi possível cobrir toda a área.

Outro ponto relevante que necessita ser explicado diz respeito às linhas do gráfico equivalentes aos casos onde não ocorreu a cobertura completa. Os sensores são depositados de

maneira aleatória na área em questão. Dentre as estratégias de distribuição (seção 2.2.1), este trabalho adota o modelo de distribuição aleatória não uniforme. Esse modelo visa aproximar a solução a um ambiente real. Isso significa dizer que os sensores estão sim distribuídos de maneira aleatória, e podem existir casos onde uma determinada região é composta por poucos ou nenhum sensor, ao passo que em outras regiões estão presentes uma quantidade maior de sensores. Dessa forma, os casos no gráfico onde a cobertura não é completa (valor 1) corresponde exatamente a esses casos onde há uma grande disparidade na distribuição dos sensores na área em questão.

Com a densidade mais baixa, número de nós entre 200 e 300, o gráfico mostra que o DCS² obteve desempenho menor que o CCMS. A porcentagem de área coberta neste intervalo estava pouco acima dos 90%, ao passo que a solução de comparação já estava em torno dos 95%.

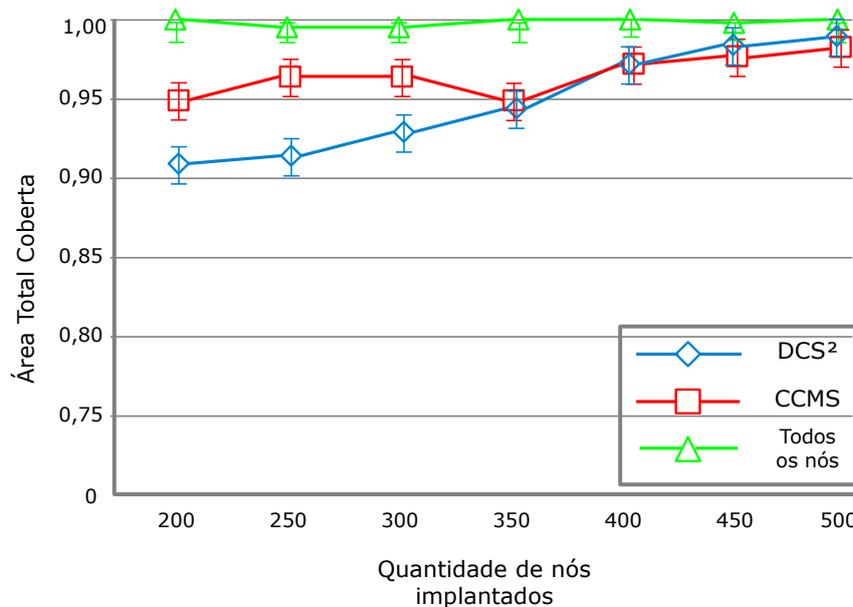


Figura 4.1: Porcentagem da área total, coberta pelos sensores - todos os nós implantados ativos, CCMS e DCS².

Com o aumento da densidade, a partir de 350 nós na rede, o mecanismo de escalonamento DCS² conseguiu se equiparar aos valores da solução de comparação, obtendo próximo à marca de 95% de área total coberta pelos sensores escalonados. A partir de 400 nós na rede, ele consegue ultrapassar a linha dos 95%. A simulação foi estendida para a quantidade de 500 nós na rede, visando verificar se as soluções ainda se equiparavam. Pelo gráfico é possível perceber que, a partir de 450 nós na rede, o DCS² consegue obter cobertura melhor do que o CCMS. Esse fato pode ser um indício de que o DCS² obtém resultados melhores do que a solução de comparação, quando em cenários de maior densidade.

Analisando o gráfico, é possível inferir que o DCS² obteve desempenho menor do que a solução de comparação, nos casos onde a rede apresentava uma densidade menor, entre 200 a 300 nós. Por outro lado, com o aumento da densidade, é possível perceber uma tendência de crescimento na porcentagem de área coberta.

Com essas observações, pode-se discutir a possibilidade de averiguar posteriormente o comportamento do algoritmo para um tempo de simulação maior, bem como para um número ainda maior de sensores. Essa análise mais aprofundada não foi realizada neste trabalho, mas consta dentre os direcionamentos de trabalhos futuros e melhorias do algoritmo.

A seção a seguir avalia o desempenho dos algoritmos sob a métrica de energia residual média da rede.

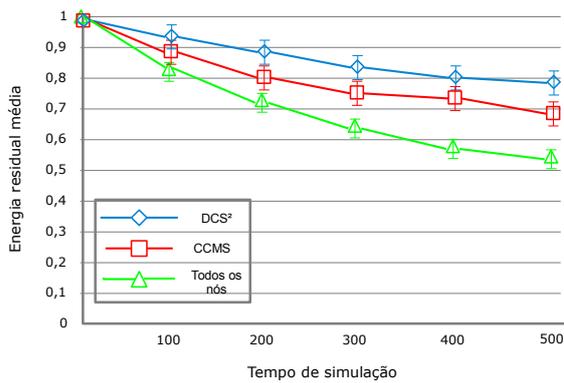
4.3.2 Energia Residual Média

Conforme definido anteriormente, na seção 4.2, a energia residual da rede foi calculada como sendo a média da energia residual de todos os nós da rede, ativos ou não, escalonados ou não. Os gráficos apresentados na figura 4.2, a seguir, apresentam a energia residual média da rede em função do tempo de simulação. Cada gráfico corresponde a uma configuração de densidade, de quantidade de nós, diferente (de 200 a 500 nós).

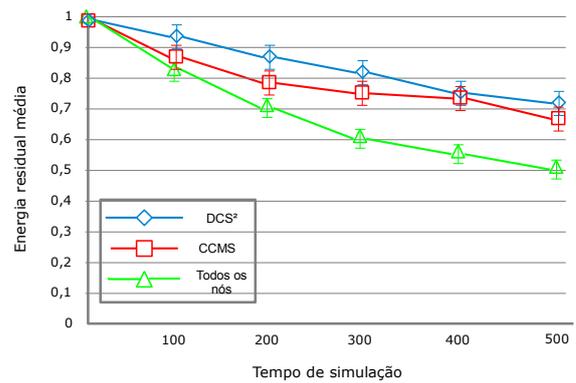
Em cada um desses gráficos, onde $200 \leq n \leq 500$ nós, já é possível perceber que tanto o DCS² quanto o CCMS conseguem conservar mais energia ao longo do tempo de simulação, em comparação à solução com todos os nós ativos (caso (i)). Isso reflete o que já havia sido discutido anteriormente, quanto aos mecanismos de escalonamento se mostrarem como sendo boas soluções para economia de energia em RSSF.

O gráfico 4.2a mostra que, para $n = 200$, ambas as soluções são superiores ao caso (i), mas ainda as linhas do gráfico estão relativamente próximas. É possível perceber também que o DCS² já apresenta um pequeno ganho de economia em relação ao CCMS. No gráfico 4.2b, com uma pequena elevação no número de nós ($n = 300$), é possível perceber que as duas soluções se distanciam mais do caso (i). Esse gráfico ainda mostra que o DCS² continua obtendo desempenho melhor, apesar de bem próximo da solução de comparação.

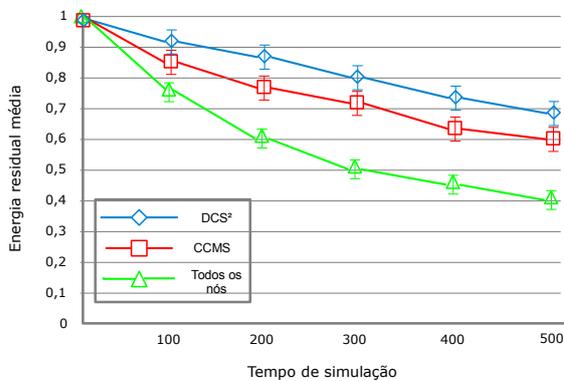
No gráfico 4.2c já é possível perceber um relativo distanciamento nos valores da energia residual do DCS² em relação ao CCMS, e destes em relação à implementação com todos os nós ativos.



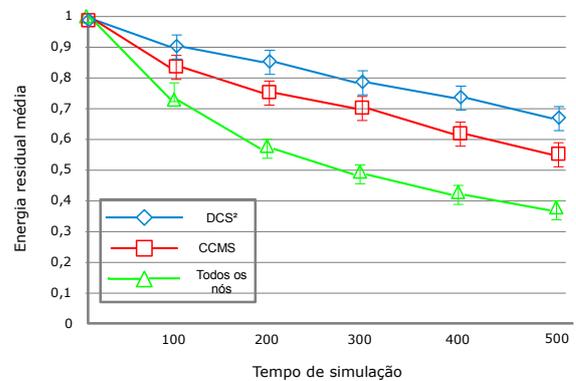
(a) 200 nós implantados.



(b) 300 nós implantados.



(c) 400 nós implantados.



(d) 500 nós implantados.

Figura 4.2: Energia residual média dos nós da rede.

No gráfico 4.2d é possível visualizar as mudanças mais consideráveis no comportamento até então das soluções. Com um aumento na densidade de 400 para 500 nós, o gráfico mostra que a solução DCS² consegue um ganho relativamente maior de economia de energia, em relação ao CCMS. A energia residual média dos sensores se configura pouco abaixo de 0,7 ao final da simulação, ao passo que na solução de comparação esse valor já se encontra abaixo de 0,6. É importante ressaltar também que já nos primeiros tempos (a partir de 200 segundos) de simulação, o DCS² já obtém uma economia maior. Por outro lado, o DCS² consegue economizar energia pois seu mecanismo de definição de vizinhança é relativamente mais simples e robusto.

A seguir, é apresentada a última métrica estabelecida para verificar o desempenho este trabalho, a avaliação do tempo de vida médio da rede.

4.3.3 Tempo de Vida da Rede

O capítulo 2 apresentado anteriormente define o conceito de tempo de vida utilizado neste trabalho. Dessa forma, a proposta deste trabalho é conseguir obter ganho no tempo de vida, prolongando o tempo no qual a rede se mantém operante. Uma vez que o tempo de vida é definido como o tempo decorrido até o primeiro nó da rede esgotar sua energia residual interna, o gráfico 4.3 apresenta o tempo de vida médio da rede, para cada uma das densidades simuladas ($200 \leq n \leq 500$). Pelo gráfico é possível perceber que o DCS² obtém desempenho consideravelmente melhor do que o caso (i), mesmo nos casos onde a densidade é menor ($n = 200$). Isso é consequência do processo de escalonamento empregado, que consegue prolongar o tempo de o primeiro nó da rede esgotar sua energia.

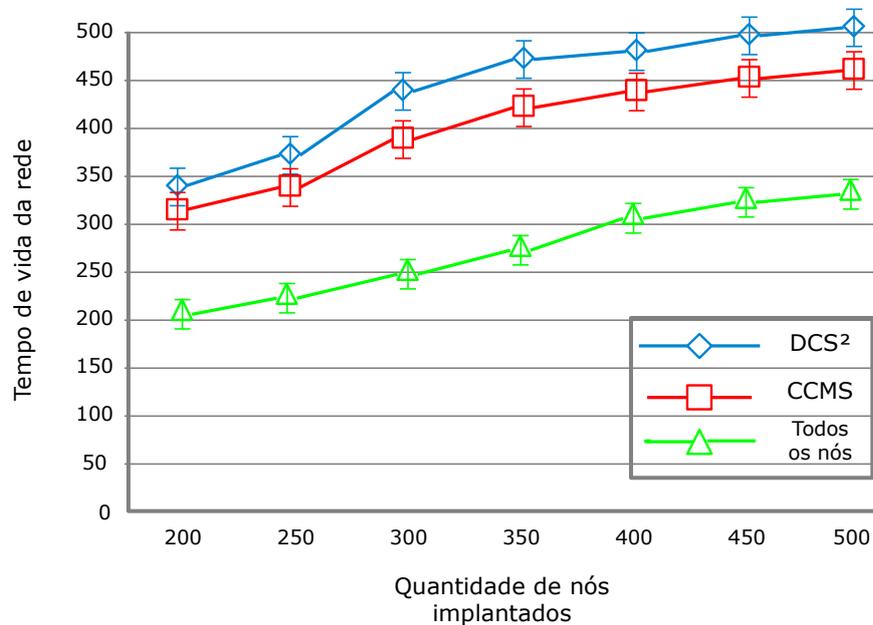


Figura 4.3: Comparação dos tempos de vida da rede com DCS², CCMS e com todos os nós ativos.

Outra análise importante está relacionada ao desempenho do DCS² em relação ao CCMS. A energia residual média foi analisada na seção 4.3.2. Foi constatado que para densidades menores, ou seja, $n \leq 350$, as duas soluções apresentavam um comportamento aproximado. Contudo, há uma vantagem do DCS² sobre o CCMS, conforme já foi discutido. Ao analisar agora o tempo de vida da rede, é possível perceber que em densidades menores ($n \leq 250$) já é possível identificar uma vantagem do DCS². Para $n = 300$ e 350 houve uma vantagem ainda maior desta solução. A partir desse valor, $n \geq 400$, a vantagem do DCS² permanece, fato que reforça a inferência de que o DCS² apresenta um comportamento melhor que o CCMS com o aumento da densidade da rede.

Considerando todas essas análises do comportamento do DCS² com relação ao tempo de vida médio da rede, é possível inferir que o algoritmo se comporta de maneira satisfatória, permitindo um prolongamento do tempo de vida da rede. Ao aumentar a densidade da rede, o algoritmo mostrou uma tendência de crescimento no gráfico maior, em relação ao CCMS.

A próxima seção conclui este capítulo, fazendo uma análise geral dos resultados aqui apresentados.

4.4 Conclusão do capítulo

Este capítulo apresentou como foram realizados os experimentos, por meio das simulações. O capítulo descreve inicialmente, o plano de testes que foi desenvolvido e que serviu para guiar a execução dos experimentos. Em seguida foram apresentadas as condições e parâmetros estabelecidos na ferramenta de simulação (NS-2). Após isso, o capítulo define quais foram as métricas consideradas para a coleta de dados, geração de gráficos e análise dos resultados. O objetivo foi avaliar o comportamento do algoritmo segundo essas métricas, bem como compará-lo com uma solução de comparação existente, que também foi implementada. Finalmente, o capítulo apresenta a análise dos resultados obtidos com os experimentos. Para cada uma das métricas, foram analisados os comportamentos dos experimentos definidos em cada um dos três casos que consistiram no plano de testes. A solução proposta neste trabalho (DCS²), bem como a solução de comparação (CCMS), ambas soluções de escalonamento, foram conjuntamente comparadas com o caso de teste onde não há mecanismo de escalonamento. Depois ambas as soluções foram comparadas entre si e o algoritmo DCS² demonstrou comportamento satisfatório dentro das métricas estabelecidas.

O capítulo a seguir conclui este trabalho como um todo, apontando as considerações relevantes acerca desses resultados apreciados no presente capítulo.

5 CONSIDERAÇÕES FINAIS

Esta dissertação propõe o DCS², um mecanismo de escalonamento distribuído e colaborativo para RSSF densas. Este capítulo versa sobre as considerações relevantes, concluindo o que foi exposto nos capítulos anteriores. Dessa forma, a seção 5.1 expõe as conclusões inferidas a partir dos resultados obtidos e discutidos no capítulo 4. As principais contribuições deste trabalho são relacionadas na seção 5.2. Finalmente, na seção 5.3 é feita uma discussão acerca dos trabalhos futuros que podem ser oriundos desta dissertação.

5.1 Conclusões

O DCS² conseguiu desempenhar de maneira satisfatória o papel que lhe foi atribuído, pois conseguiu atuar no escalonamento de atividades dos nós sensores, levando a uma extensão do tempo de vida da rede, quando comparado com uma solução sem mecanismo de escalonamento e com uma solução de comparação que utiliza mecanismo de escalonamento. Aplicações de monitoramento, que empregam uma grande quantidade de sensores em um ambiente no qual não há possibilidade de recarregar as baterias desses sensores, são consideravelmente dependentes do esquema de utilização da energia residual de seus nós sensores. Portanto, um mecanismo que consegue prolongar o tempo de vida dessas redes torna-se importante para permitir uma melhor usabilidade dessas aplicações.

Para avaliar o comportamento do DCS², foi estabelecido um plano de testes, descritos na seção 4.1. O plano de testes consiste em três casos, os quais foram implementados e simulados para compor a avaliação deste trabalho. Além do plano de testes, foram definidas as métricas listadas abaixo, para a coleta de dados oriundos das simulações. Uma vez realizados os experimentos e analisados os resultados, foi possível concluir que o algoritmo proposto se mostrou robusto e escalável, à medida que foram testados vários níveis de densidade na rede.

- ▶ Porcentagem da área total, coberta pelos sensores;
- ▶ Energia residual média dos nós da rede;
- ▶ Tempo de vida da rede.

Da mesma forma que o DCS² conseguiu obter ganho no tempo de vida de RSSFs, ele também conseguiu garantir um bom percentual de cobertura da rede. O mecanismo de escalonamento proporcionado pela colaboração incentivada entre os nós sensores, por meio das inversões de arestas entre nós concorrentes, permitiu que os sensores da rede como um todo operassem de forma alternada, economizando suas energias individualmente.

No capítulo anterior, na subseção 4.3.1, foi avaliado o desempenho do algoritmo sob a métrica de porcentagem de cobertura. Nesse ponto o DCS² não conseguiu ultrapassar a solução CCMS, conforme pode ser identificado no respectivo gráfico. Quando a densidade da rede foi aumentada, o DCS² conseguiu se igualar à solução de comparação. Dessa forma, no geral, o algoritmo conseguiu melhores resultados no que concerne ao prolongamento de energia, e resultados satisfatórios no que concerne à cobertura.

5.2 Contribuições

A principal contribuição desta dissertação foi o algoritmo DCS², que implementa um mecanismo distribuído e colaborativo para escalonamento de atividade dos nós em RSSF densas. O algoritmo foi inspirado na solução para exclusão mútua em sistemas distribuídos, denominada EIA, apresentada no capítulo 2. É importante apontar que o algoritmo faz uso apenas de informações locais de cada sensor e de sua vizinhança. As interações entre um nó e seus vizinhos são realizadas de maneira a identificar as condições de concorrência, para em seguida serem formados os subgrupos de nós que realizarão as inversões de arestas entre si.

O DCS² conseguiu estabelecer o esquema de escalonamento de atividade dos nós, incentivando a colaboração entre eles, por meio das inversões de arestas, representadas pelo envio de uma mensagem de inversão. Esse esquema conseguiu prolongar o tempo de vida da rede, ao mesmo tempo que manteve os níveis de cobertura da área em níveis acima dos 90%.

Outra contribuição são os códigos das implementações (CCMS e DCS²) em C++ e OTcl, para o NS-2, que são disponibilizados a pesquisadores interessados em evoluir este trabalho. Vale ressaltar que já existe a iniciativa de implementação desse código para a versão mais atual dessa ferramenta de simulação, tão logo haja possibilidade de tecnologia para fazê-lo. Maiores detalhes sobre essa questão serão descritos na próxima seção.

5.3 Trabalhos Futuros

Esta seção discute acerca das perspectivas de refinamentos e trabalhos futuros, a partir do estado atual deste trabalho. É relevante relacionar os possíveis direcionamentos para amadurecer ainda mais a solução proposta.

De início, é relevante explicar os motivos de ter sido escolhida a ferramenta NS-2, quando já existe uma versão mais recente desse simulador. De fato, já está disponível para a comunidade científica o Network Simulator - versão 3 (NS-3) (NS-3, 2012). Contudo, até o momento de conclusão dos experimentos dessa dissertação ainda não se encontrava disponível uma implementação estável da camada física (ADAMS, 2006) necessária para se implementar a simulação de RSSF. Por outro lado, foram descobertas iniciativas que caminham nessa linha, como por exemplo em (WPAN, 2012), que apresenta códigos iniciais para implementação do padrão IEEE 802.15.4. Outra categoria de trabalhos são os que visam utilizar o NS-3 para emular os experimentos configurados em outros simuladores. Em (RILISKIS et al., 2011) é proposto um *framework* para emulação de redes de sensores implementadas em linguagem nesC, como é o caso do simulador TOSSIM (TOSSIM, 2012) que simula aplicações do sistema operacional TinyOS, escritas nessa linguagem (TINYOS, 2012).

O DCS² também pode ser integrado em conjunto com o protocolo PROST, desenvolvido por (NETO, 2009), que é um protocolo de roteamento sensível ao tempo, onde poderia ser analisada a influência do mecanismo de escalonamento sobre o tempo de entrega das informações de contexto coletadas pela rede. Outra opção interessante a ser considerada é integrar ao protocolo de roteamento BiO4SeL de (RIBEIRO, 2009). Esta solução é baseada em colônia de formigas e utiliza princípio de computação autônoma, já discutidos aqui neste trabalho, no capítulo 2. Uma vez realizada essa integração, o trabalho resultante poderia ainda ser comparado com o trabalho desenvolvido por (OLIVEIRA, 2011). Este trabalho propõe uma ampliação ao protocolo BiO4SeL, adicionando uma heurística de escalonamento para os sensores.

Outra abordagem interessante seria analisar o comportamento do algoritmo no cenários de RSSF móveis, verificando a influência da mobilidade dos nós tanto nos quesitos de cobertura, quanto no tempo de vida da rede.

Este trabalho foi realizado sobre RSSF homogêneas. Contudo, avaliar o desempenho do DCS² em um cenário de RSSF heterogêneas se mostra um caminho igualmente interessante de se estudar.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMS, J. An introduction to ieee std 802.15.4. In: *Aerospace Conference, 2006 IEEE*. [s.n.], 2006. p. 8 pp. Disponível em: <<http://dx.doi.org/10.1109/AERO.2006.1655947>>.

AKL, A.; GAYRAUD, T.; BERTHOU, P. A metric for evaluating density level of wireless sensor networks. In: *Wireless Days (WD), 2011 IFIP*. [s.n.], 2011. p. 1 –3. ISSN 2156-9711. Disponível em: <<http://dx.doi.org/10.1109/WD.2011.6098166>>.

AKYILDIZ, I. F. et al. Wireless sensor networks: a survey. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Elsevier North-Holland, Inc., New York, NY, USA, v. 38, p. 393–422, March 2002. ISSN 1389-1286. Disponível em: <<http://dl.acm.org/citation.cfm?id=585775.585777>>.

ANASTASI, G. et al. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 7, p. 537–568, May 2009. ISSN 1570-8705. Disponível em: <<http://dl.acm.org/citation.cfm?id=1465737.1465996>>.

ANCUSA, V.; BOGDAN, R. A method for determining ad-hoc redundant coverage area in a wireless sensor network. 2011.

BANTZ, D. F. et al. Autonomic personal computing. *IBM Systems Journal*, IBM Corp., Riverton, NJ, USA, v. 42, n. 1, p. 165–176, jan. 2003. ISSN 0018-8670. Disponível em: <<http://dx.doi.org/10.1147/sj.421.0165>>.

BARBOSA, V. C. *Concurrency in systems with neighborhood constraints (distributed systems, parallel processing, dining philosophers, simulated annealing)*. Tese (Doutorado) — University of California, Los Angeles, 1986. AAI8702613. Disponível em: <<http://www.cs.ucla.edu/tech-report/198.-reports/860035.pdf>>.

BARBOSA, V. C. *Massively parallel models of computation: distributed parallel processing in artificial intelligence and optimisation*. Upper Saddle River, NJ, USA: Ellis Horwood, 1993. ISBN 0-13-562968-3.

BARBOSA, V. C. *An introduction to distributed algorithms*. Cambridge, MA, USA: MIT Press, 1996. ISBN 0-262-02412-8, 9780262024129. Disponível em: <[http://210.212.249.121/ebook/ece/An Introduction to Distributed Algorithms - B. Valmir \(MIT, 1996\) WW.pdf](http://210.212.249.121/ebook/ece/An%20Introduction%20to%20Distributed%20Algorithms%20-%20B.%20Valmir%20(MIT,%201996)%20WW.pdf)>.

BARBOSA, V. C.; GAFNI, E. Concurrency in heavily loaded neighborhood-constrained systems. *ACM Trans. Program. Lang. Syst.*, ACM, New York, NY, USA, v. 11, n. 4, p. 562–584, out. 1989. ISSN 0164-0925. Disponível em: <<http://doi.acm.org/10.1145/69558.69560>>.

BARI, A.; JAEKEL, A.; BANDYOPADHYAY, S. Clustering strategies for improving the lifetime of two-tiered sensor networks. *Comput. Commun.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 31, n. 14, p. 3451–3459, set. 2008. ISSN 0140-3664. Disponível em: <<http://dx.doi.org/10.1016/j.comcom.2008.05.038>>.

- BERNS, A.; GHOSH, S. Dissecting self-* properties. In: *Self-Adaptive and Self-Organizing Systems, 2009. SASO '09. Third IEEE International Conference on*. [s.n.], 2009. p. 10 – 19. Disponível em: <<http://dx.doi.org/10.1109/SASO.2009.25>>.
- BING, S. et al. An ant-based on-demand energy routing protocol for ad hoc wireless networks. In: *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*. [s.n.], 2007. p. 1516 –1519. Disponível em: <<http://dx.doi.org/10.1109/WICOM.2007.382>>.
- CAMPOS, N. G. da S. *Máquina de Inferência Autônoma Distribuída para RSSF*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, Agosto 2010. Disponível em: <<http://www.dominiopublico.gov.br>>.
- CARDEI, M.; WU, J. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Computer Communications*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 4, p. 413–420, fev. 2006. ISSN 0140-3664. Disponível em: <<http://dx.doi.org/10.1016/j.comcom.2004.12.025>>.
- CARVALHO, C. G. N. de. *Correlação Espaço-Temporal Multivariada na Melhoria da Precisão de Predição para Redução de Dados em Redes de Sensores sem Fio*. Tese (Doutorado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, 2012. Disponível em: <<http://www.teses.ufc.br>>.
- CARVALHO, S. R. C. *Economia de energia em rede de sensores utilizando boas práticas da metodologia Seis Sigma*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, Setembro 2012. Disponível em: <<http://www.teses.ufc.br>>.
- CARVALHO, S. R. C. et al. Uma proposta para gerenciamento de energia em redes de sensores utilizando a metodologia seis sigma. *Workshop de Gerência e Operação de Redes e Serviços, 2012, Ouro Preto. Anais do XVII Workshop de Gerência e Operação de Redes e Serviços. Sociedade Brasileira de Computação (SBC)*, p. 149 – 162, Abril 2012. Disponível em: <<http://sbrc2012.dcc.ufmg.br/app/pdfs/p-05/wgrs/WGRS-ST4-3.pdf>>.
- CASTILLO-EFFER, M. et al. Wireless sensor networks for flash-flood alerting. In: *Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on*. [s.n.], 2004. v. 1, p. 142 – 146. Disponível em: <<http://dx.doi.org/10.1109/ICCDSCS.2004.1393370>>.
- CAVALCANTE, T. M.; GARCIA, F. P.; ANDRADE, R. M. C. Avaliação de desempenho de mecanismos de segurança para redes de sensores sem fio. *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2012, Ouro Preto. Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 277–290, 2012. ISSN 2177-496X. Disponível em: <<http://sbrc2012.dcc.ufmg.br/app/pdfs/p-03/ST6.2.pdf>>.
- CHAMBERLAND, J.-F.; VEERAVALLI, V. Decentralized detection in sensor networks. *Signal Processing, IEEE Transactions on*, v. 51, n. 2, p. 407 – 416, feb 2003. ISSN 1053-587X. Disponível em: <<http://dx.doi.org/10.1109/TSP.2002.806982>>.
- COULOURIS, G. et al. *Distributed Systems: Concepts and Design*. 5th. ed. USA: Addison-Wesley Publishing Company, 2011. ISBN 0132143011, 9780132143011.

- DIETRICH, I.; DRESSLER, F. On the lifetime of wireless sensor networks. *ACM Trans. Sen. Netw.*, ACM, New York, NY, USA, v. 5, n. 1, p. 5:1–5:39, fev. 2009. ISSN 1550-4859. Disponível em: <<http://doi.acm.org/10.1145/1464420.1464425>>.
- DOBSON, S. et al. A survey of autonomic communications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, ACM, New York, NY, USA, v. 1, n. 2, p. 223–259, dez. 2006. ISSN 1556-4665. Disponível em: <<http://doi.acm.org/10.1145/1186778.1186782>>.
- FAN, G.; JIN, S. Coverage problem in wireless sensor network: A survey. *Journal of Networks*, Academy Publisher, v. 5, n. 9, p. 1033–1040, September 2010. Disponível em: <<http://dx.doi.org/10.4304/jnw.5.9.1033-1040>>.
- FRANÇA, F. M. G.; FARIA, L. Optimal mapping of neighbourhood-constrained systems. In: *Proceedings of the Second International Workshop on Parallel Algorithms for Irregularly Structured Problems*. London, UK, UK: Springer-Verlag, 1995. (IRREGULAR '95), p. 165–170. ISBN 3-540-60321-2. Disponível em: <<http://dl.acm.org/citation.cfm?id=646009.676841>>.
- GAKKESTAD, J.; HANSEN, L. Powering wireless sensor networks nodes in northern europe using solar cell panel for energy harvesting. In: *2011 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. [s.n.], 2011. p. 1 –5. ISSN 2157-4952. Disponível em: <<http://dx.doi.org/10.1109/NTMS.2011.5720640>>.
- GAO, T. et al. Vital signs monitoring and patient tracking over a wireless network. In: *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. [s.n.], 2005. p. 102 –105. Disponível em: <<http://dx.doi.org/10.1109/IEMBS.2005.1616352>>.
- GARCIA, F. P.; SOUZA, J. N. de; ANDRADE, R. M. C. An energy-efficient passive monitoring system for wireless sensor networks. *PhD Forum In Second IFIP Conference on Sustainable Internet and ICT for Sustainability*, 2012.
- HILL, J. et al. System architecture directions for networked sensors. In: *Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM, 2000. (ASPLOS-IX), p. 93–104. ISBN 1-58113-317-0. Disponível em: <<http://doi.acm.org/10.1145/378993.379006>>.
- HORN, P. *Autonomic Computing: IBM's Perspective on the State of Information Technology*. [S.l.], 2001. Disponível em: <<http://researchweb.watson.ibm.com/autonomic>>.
- HU, X.-M. et al. Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. *Trans. Evol. Comp*, IEEE Press, Piscataway, NJ, USA, v. 14, p. 766–781, October 2010. ISSN 1089-778X. Disponível em: <<http://dx.doi.org/10.1109/TEVC.2010.2040182>>.
- JAIN, R. *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling*. [S.l.]: Wiley, 1991. I-XXVII, 1-685 p. (Wiley professional computing). ISBN 978-0-471-50336-1.
- KAR, K.; BANERJEE, S. Node Placement for Connected Coverage in Sensor Networks. mar. 2003. Disponível em: <<http://hal.inria.fr/inria-00466114>>.

- KEPHART, J. O.; CHESS, D. M. The vision of autonomic computing. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 36, n. 1, p. 41–50, jan. 2003. ISSN 0018-9162. Disponível em: <<http://dx.doi.org/10.1109/MC.2003.1160055>>.
- KO, B.-J.; ROSS, K. W.; RUBENSTEIN, D. Conserving Energy in Dense Sensor Networks via Distributed Scheduling. mar. 2003. Disponível em: <<http://hal.inria.fr/inria-00466091>>.
- KRISHNAPRASAD, S. Concurrent/distributed programming illustrated using the dining philosophers problem. *J. Comput. Sci. Coll.*, Consortium for Computing Sciences in Colleges, USA, v. 18, n. 4, p. 104–110, abr. 2003. ISSN 1937-4771. Disponível em: <<http://dl.acm.org/citation.cfm?id=767598.767617>>.
- LAMPORT, L. The mutual exclusion problem: part i — a theory of interprocess communication. *J. ACM*, ACM, New York, NY, USA, v. 33, n. 2, p. 313–326, abr. 1986. ISSN 0004-5411. Disponível em: <<http://doi.acm.org/10.1145/5383.5384>>.
- LAMPORT, L. The mutual exclusion problem: partii — statement and solutions. *J. ACM*, ACM, New York, NY, USA, v. 33, n. 2, p. 327–348, abr. 1986. ISSN 0004-5411. Disponível em: <<http://doi.acm.org/10.1145/5383.5385>>.
- LORINCZ, K. et al. Sensor networks for emergency response: challenges and opportunities. *Pervasive Computing, IEEE*, v. 3, n. 4, p. 16 – 23, oct.-dec. 2004. ISSN 1536-1268. Disponível em: <<http://dx.doi.org/10.1109/MPRV.2004.18>>.
- MARIN-PERIANU, M.; HAVINGA, P. D-fler: a distributed fuzzy logic engine for rule-based wireless sensor networks. In: *Proceedings of the 4th international conference on Ubiquitous computing systems*. Berlin, Heidelberg: Springer-Verlag, 2007. (UCS'07), p. 86–101. ISBN 3-540-76771-1, 978-3-540-76771-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=1775574.1775584>>.
- MEGERIAN, S. et al. Exposure in wireless sensor networks: theory and practical solutions. *Wirel. Netw.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 8, n. 5, p. 443–454, set. 2002. ISSN 1022-0038. Disponível em: <<http://dx.doi.org/10.1023/A:1016586011473>>.
- MEGERIAN, S.; POTKONJAK, M. Wireless sensor networks. In: PUBLISHERS, W. (Ed.). *Encyclopedia of Telecommunications*. [s.n.], 2002. p. 1 – 13. Disponível em: <http://www.cs.ucla.edu/~miodrag/papers/Megerian_Wiley_2004.pdf>.
- MEGUERDICHIAN, S. et al. Coverage problems in wireless ad-hoc sensor networks. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [s.n.], 2001. v. 3, p. 1380 –1387 vol.3. ISSN 0743-166X. Disponível em: <<http://dx.doi.org/10.1109/INFCOM.2001.916633>>.
- MISRA, S.; KUMAR, M. P.; OBAIDAT, M. S. Connectivity preserving localized coverage algorithm for area monitoring using wireless sensor networks. *Computer Communications*, v. 34, n. 12, p. 1484 – 1496, 2011. ISSN 0140-3664. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0140366410001118>>.
- NETO, J. B. B. *PROST - Um Protocolo de Roteamento Sensível ao Tempo para Redes de Sensores Sem Fio*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, 2009. Disponível em: <<http://www.dominiopublico.gov.br>>.

NETO, J. B. B.; ANDRADE, R. M. C.; NETO, P. F. R. Wireless sensor networks advances for ubiquitous computing. In: _____. *Designing Solutions-Based Ubiquitous and Pervasive Computing: New Issues and Trends*. [s.n.], 2010. cap. 09. Disponível em: <<http://dx.doi.org/10.4018/978-1-65120-843-2.ch009>>.

NS-2. *Network Simulator – versão 2*. 2012. Disponível em: <http://nslam.isi.edu/nslam/>. Acesso em: 17 de fevereiro de 2013.

NS-3. *Network Simulator – versão 3*. 2012. Disponível em: <http://www.nslam.org>. Acesso em: 17 de fevereiro de 2013.

OLIVEIRA, C. H. S. *Gerenciamento Autônomo de Energia em Redes Sensores Sem Fio Através do Escalonamento de Atividades dos Nós*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, Novembro 2011. Disponível em: <http://www.mdcc.ufc.br/teses-e-dissertacoes/doc_download/167-148-camila-helena-souza-oliveira->>.

PERLMUTTER, M.; ROBIN, L. High-performance, low cost inertial mems: A market in motion! In: *2012 IEEE/ION Position Location and Navigation Symposium (PLANS)*. [s.n.], 2012. p. 225 – 229. ISSN 2153-358X. Disponível em: <<http://dx.doi.org/10.1109/PLANS.2012.6236884>>.

QUITADAMO, R.; ZAMBONELLI, F. Autonomic communication services: a new challenge for software agents. *Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, Hingham, MA, USA, v. 17, n. 3, p. 457–475, dez. 2008. ISSN 1387-2532. Disponível em: <<http://dx.doi.org/10.1007/s10458-008-9054-9>>.

RAGHUNATHAN, V. et al. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, v. 19, n. 2, p. 40 –50, mar 2002. ISSN 1053-5888. Disponível em: <<http://dx.doi.org/10.1109/79.985679>>.

RIBEIRO, L.; CASTRO, M. de. Bio4sel: A bio-inspired routing algorithm for sensor network lifetime optimization. In: *2010 IEEE 17th International Conference on Telecommunications (ICT)*. [s.n.], 2010. p. 728 –734. Disponível em: <<http://dx.doi.org/10.1109/ICTEL.2010.5478805>>.

RIBEIRO, L. B. *BiO4SeL: Uma Abordagem Baseada em Colônia de Formigas para a Otimização do Tempo de Vida de Redes de Sensores Sem Fio*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, Agosto 2009. Disponível em: <http://www.mdcc.ufc.br/teses/doc_download/118-117-levi-bayde-ribeiro>.

RILISKIS, L.; OSIPOV, E.; MARÓTI, M. *TOS-NS3: a framework for emulating wireless sensor networks in the ns3 network simulator*. 2011. <http://www.nslam.org/workshops/wns3-2010/tinyos-ns3-workshop-v2.pdf>. Acesso em: 17 de fevereiro de 2013.

ROCHA, A. R. da. *Um Serviço de Clusterização Semântica para Redes de Sensores sem Fio Autônomas*. Tese (Doutorado) — Universidade Federal do Ceará, Fortaleza – Ceará – Brasil, 2012. Disponível em: <<http://www.teses.ufc.br>>.

SCHMECK, H. Organic computing - a new vision for distributed embedded systems. In: *Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time*

Distributed Computing. Washington, DC, USA: IEEE Computer Society, 2005. (ISORC '05), p. 201–203. ISBN 0-7695-2356-0. Disponível em: <<http://dx.doi.org/10.1109/ISORC.2005.42>>.

SCHMECK, H. et al. Adaptivity and self-organization in organic computing systems. *ACM Trans. Auton. Adapt. Syst.*, ACM, New York, NY, USA, v. 5, n. 3, p. 10:1–10:32, set. 2010. ISSN 1556-4665. Disponível em: <<http://doi.acm.org/10.1145/1837909.1837911>>.

SIMON, G. et al. Sensor network-based countersniper system. In: *Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004. (SenSys '04), p. 1–12. ISBN 1-58113-879-2. Disponível em: <<http://doi.acm.org/10.1145/1031495.1031497>>.

SOHRABY, K.; MINOLI, D.; ZNATI, T. *Wireless Sensor Networks: Technology, Protocols, and Applications*. John Wiley & Sons Inc., 2007. 1–37 p. ISBN 9780470112762. Disponível em: <<http://dx.doi.org/10.1002/9780470112762.ch1>>.

STANKOVIC, J. Wireless sensor networks. *Computer*, v. 41, n. 10, p. 92–95, october 2008. ISSN 0018-9162. Disponível em: <<http://dx.doi.org/10.1109/MC.2008.441>>.

TINYOS. *An open-source OS for the networked sensor regime*. 2012. [Http://www.tinyos.net](http://www.tinyos.net). Acesso em: 17 de fevereiro de 2013.

TOSSIM. *TinyOS mote simulator*. 2012. [Http://docs.tinyos.net/index.php/TOSSIM](http://docs.tinyos.net/index.php/TOSSIM). Acesso em: 17 de fevereiro de 2013.

VOIGT, T.; RITTER, H.; SCHILLER, J. Utilizing solar power in wireless sensor networks. In: *2003. LCN '03. Proceedings. 28th Annual IEEE International Conference on Local Computer Networks*. [s.n.], 2003. p. 416 – 422. ISSN 0742-1303. Disponível em: <<http://dx.doi.org/10.1109/LCN.2003.1243167>>.

WANG, B. Coverage problems in sensor networks: A survey. *ACM Computing Surveys*, ACM, New York, NY, USA, v. 43, n. 4, p. 1–53, October 2011. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1978802.1978811>>.

WANG, H. A review of six sigma approach: Methodology, implementation and future research. In: *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*. [s.n.], 2008. p. 1 –4. Disponível em: <<http://dx.doi.org/10.1109/WiCom.2008.1887>>.

WANG, L. et al. A clique base node scheduling method for wireless sensor networks. *Journal of Network and Computer Applications*, Academic Press Ltd., London, UK, UK, v. 33, n. 4, p. 383–396, jul. 2010. ISSN 1084-8045. Disponível em: <<http://dx.doi.org/10.1016/j.jnca.2010.03.002>>.

WANG, X. et al. Integrated coverage and connectivity configuration in wireless sensor networks. In: *Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2003. (SenSys '03), p. 28–39. ISBN 1-58113-707-9. Disponível em: <<http://doi.acm.org/10.1145/958491.958496>>.

WANT, R.; FARKAS, K. I.; NARAYANASWAMI, C. Guest editors' introduction: Energy harvesting and conservation. *IEEE Pervasive Computing*, IEEE Computer Society, Los Alamitos, CA, USA, v. 4, p. 14–17, 2005. ISSN 1536-1268. Disponível em: <<http://doi.ieeecomputersociety.org/10.1109/MPRV.2005.12>>.

WANT, R.; PERING, T.; TENNENHOUSE, D. Comparing autonomic and proactive computing. *IBM Syst. J.*, IBM Corp., Riverton, NJ, USA, v. 42, n. 1, p. 129–135, jan. 2003. ISSN 0018-8670. Disponível em: <<http://dx.doi.org/10.1147/sj.421.0129>>.

WERNER-ALLEN, G. et al. Deploying a wireless sensor network on an active volcano. *Internet Computing, IEEE*, v. 10, n. 2, p. 18 – 25, march-april 2006. ISSN 1089-7801. Disponível em: <<http://dx.doi.org/10.1109/MIC.2006.26>>.

WPAN Ir. *low-rate WPAN (802.15.4)*. 2012. Disponível em: <<http://code.nsnam.org/tomh/ns-3-lr-wpan/>>.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Analysis of a prediction-based mobility adaptive tracking algorithm. In: *Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on*. [s.n.], 2005. p. 753 – 760 Vol. 1. Disponível em: <<http://dx.doi.org/10.1109/ICBN.2005.1589681>>.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. *Computer Networks*, v. 52, n. 12, p. 2292 – 2330, 2008. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128608001254>>.

ZHANG, H.; HOU, J. C. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, v. 1, n. 1-2, p. 89–124, 2005. Disponível em: <http://theory.utdallas.edu/Feng_Zou/group3_seminar/Sensing_Coverage_and_Connectivity_in_Large_Sensor_Network.pdf>.

ZHU, C. et al. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications - Simulation and Testbeds*, v. 35, n. 2, p. 619 – 632, 2011. ISSN 1084-8045. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1084804511002323>>.

ZHU, Y. et al. Unexpected properties and optimum-distributed sensor detectors for dependent observation cases. *Automatic Control, IEEE Transactions on*, v. 45, n. 1, p. 62 –72, jan 2000. ISSN 0018-9286. Disponível em: <<http://dx.doi.org/10.1109/9.827356>>.