

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Tese

Balanceamento de Carga em Nível de Aplicação para Redes de Sensores Sem Fio Reativas: Uma Abordagem Baseada em Problemas de Otimização com Restrições

Igor Avila Pereira

Pelotas, 2021

Igor Avila Pereira

Balanceamento de Carga em Nível de Aplicação para Redes de Sensores Sem Fio Reativas: Uma Abordagem Baseada em Problemas de Otimização com Restrições

Tese apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientadora: Prof^a. Dr^a. Lisane Brisolara de Brisolara
Coorientador: Prof. Dr. Paulo Roberto Ferreira Jr.

Pelotas, 2021

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

P436b Pereira, Igor Avila

Balanceamento de carga em nível de aplicação para redes de sensores sem fio reativas : uma abordagem baseada em problemas de otimização com restrições / Igor Avila Pereira ; Lisane Brisolara de Brisolara, orientadora ; Paulo Roberto Ferreira Junior, coorientador. — Pelotas, 2021.

88 f. : il.

Tese (Doutorado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2021.

1. Rede de sensores sem fio. 2. Balanceamento de carga. 3. Otimização com restrição. 4. Nível de aplicação. 5. Redes reativas. I. Brisolara, Lisane Brisolara de, orient. II. Ferreira Junior, Paulo Roberto, coorient. III. Título.

CDD : 005

Igor Avila Pereira

Balanceamento de Carga em Nível de Aplicação para Redes de Sensores Sem Fio Reativas: Uma Abordagem Baseada em Problemas de Otimização com Restrições

Tese aprovada, como requisito parcial, para obtenção do grau de Doutor em Ciência da Computação, Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas.

Data da Defesa: 22 de março de 2021

Banca Examinadora:

Prof^a. Dr^a Lisane Brisolara de Brisolara (orientadora)
Doutora em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Marilton Sanchotene de Aguiar
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

Prof. Dr. Leonardo Bidese de Pinho
Doutor em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro.

Prof. Dr. Leandro Buss Becker
Doutor em Computação pela Universidade Federal do Rio Grande do Sul.

AGRADECIMENTOS

Agradecimento especial à minha família, minha noiva Narúsci e meus amigos pela compreensão e auxílio durante o doutorado.

Agradecimentos aos professores Lisane e Paulo pela ajuda dada no desenvolvimento do trabalho de doutorado. Obrigado pelas inúmeras revisões e reuniões.

Agradeço aos meus colegas de doutorado e aos funcionários da Universidade Federal de Pelotas (UFPEL) que também me apoiaram durante o período de doutorado.

O autor agradece ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS) pelo apoio concedido pela instituição.

Pra quem gosta de nós é um prato cheio — HUMBERTO
GESSINGER

RESUMO

PEREIRA, Igor Avila. **Balanceamento de Carga em Nível de Aplicação para Redes de Sensores Sem Fio Reativas: Uma Abordagem Baseada em Problemas de Otimização com Restrições**. Orientadora: Lisane Brisolara de Brisolara. 2021. 88 f. Tese (Doutorado em Ciência da Computação) – Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2021.

Nas redes de sensores sem fio (RSSFs), a sobrecarga de nodos pode resultar na degradação de desempenho da rede exigindo a adoção de técnicas dinâmicas de balanceamento de carga para prolongar o tempo de vida e a disponibilidade dos seus serviços. Atualmente estas redes estão cada vez mais sendo utilizadas em aplicações com alta demanda para o processamento de vídeos/imagens, o que faz com que a carga de processamento, assim como a de comunicação, devam ser consideradas no balanceamento da carga. Recentemente, técnicas baseadas em heurísticas foram propostas para equilibrar a carga atuando no nível de aplicação. Entretanto, essas não possuem qualquer garantia de qualidade e por serem distribuídas requerem comunicação o que pode afetar o tempo de vida da rede. Assim, este trabalho apresenta uma nova solução a qual baseia-se em técnicas de Otimização de Problemas com Restrições para resolver o balanceamento de carga em RSSFs em nível de aplicação. A abordagem proposta explora a redundância limitada nas áreas de cobertura e emprega uma solução semi-distribuída, evitando a complexidade de algoritmos distribuídos. A abordagem proposta garante que os eventos detectados sejam processados durante a vida útil da rede e que um número fixo de mensagens serão trocadas a cada novo evento detectado. Visando minimizar o impacto negativo no tempo de vida da rede, a abordagem COP-Feedback incorpora procedimentos de controle de propagação que impedem o envio de mensagens redundantes durante o processo de balanceamento de carga. Experimentos apontam que a abordagem proposta permite à rede ter desempenho superior ao das técnicas heurísticas consideradas estado-da-arte, tanto com nodos dispostos em grade, como com nodos dispostos aleatoriamente. Os resultados também mostraram que a COP-Feedback mantém seu comportamento tanto em redes densas como em redes com menor número de nodos, o que não ocorre com as técnicas heurísticas. Além disso, COP-Feedback apresentou resultados próximos de uma solução gulosa, usada como *upper-bound* nas análises, mas que não pode ser empregada em cenários reais por requerer a centralização do estado dos nodos da rede.

Palavras-chave: Rede de sensores Sem Fio. Balanceamento de Carga. Otimização com Restrição. Nível de Aplicação. Redes Reativas. Sistema Embarcados. Sistemas Ciber-físicos.

ABSTRACT

PEREIRA, Igor Avila. **Application-Level Load Balancing for Reactive Wireless Sensor Networks: An Abordagem Based on a Restriction Optimization Problem.** Advisor: Lisane Brisolara de Brisolara. 2021. 88 f. Thesis (Doctorate in Computer Science) – Technology Development Center, Federal University of Pelotas, Pelotas, 2021.

In wireless sensor networks (WSNs), the overload of nodes can result in the degradation of network performance, requiring the adoption of dynamic load balancing techniques to prolong the service life and availability of its services. Currently, these networks are increasingly being used in applications with high demand for video and image processing, which means that the processing load, as well as the communication load, must be considered in load balancing. Recently, techniques based on heuristics have been proposed to balance the load at the application level. However, these do not have any quality guarantee and since they are distributed, they require communication, which can affect the network's lifetime. Thus, this work presents a new solution that is based on techniques of Constraint Optimization Problems to solve the load balancing in WSNs at the application level. The proposed approach explores the limited redundancy in the coverage areas and employs a semi-distributed solution, avoiding the complexity of distributed algorithms. The proposed approach ensures that the detected events are processed during the network's lifetime and that a fixed number of messages will be exchanged for each new detected event. In order to minimize the negative impact on the lifetime, the COP-Feedback approach incorporates propagation control procedures that prevent the sending of redundant messages during the load balancing process. Experiments point out that the proposed approach allows the network to perform better than the heuristic ones considered state-of-the-art, both with nodes arranged in a grid, as well as with nodes randomly arranged. The results also showed that COP-Feedback maintains its behavior both in high-density as well as low-density networks, which is not the case with heuristic techniques. In addition, COP-Feedback presented results close to that of a greedy solution, used as upper-bound in the analyzes, but which cannot be used in real scenarios because it requires centralization of the state of the network nodes.

Keywords: Wireless sensor network. Load Balancing. Constraint Optimization. Application Level. Reactive Networks. Embedded Systems. Cyber-physical Systems.

LISTA DE FIGURAS

1	Arquitetura Básica das RSSFs.	20
2	Visão Geral das Aplicações das RSSFs adaptada de YICK; MUKHERJEE; GHOSAL (2008).	21
3	Uma simulação no EBORACUM.	23
4	Diagrama de Classes do EBORACUM - Camada de Plataforma. . .	25
5	Diagrama de Classes do EBORACUM - Camada de Aplicação. . .	28
6	Representações possíveis de um DCOP com a função $F(x_1, x_2, x_3, x_4, x_5) = f_{12}(x_1, x_2) + f_{23}(x_2, x_3) + f_{25}(x_2, x_5) + f_{34}(x_3, x_4) + f_{45}(x_4, x_5)$	34
7	Funcionamento da técnica LEACH.	36
8	Funcionamento da técnica PEGASIS.	37
9	Funcionamento da técnica HEED.	38
10	Abordagem COP: (a) disparo do evento; (b) líder do esquema de balanceamento de carga; (c) e (d) vizinhos trocando mensagens com o líder para confirmar ou não o sensoriamento do mesmo evento; (e) o líder encaminhando para os vizinhos o esquema de balanceamento escolhido; e (f) somente o nodo responsável pelo processamento permanecerá com sua capacidade de processamento ativa.	56
11	Diagrama de Classes - Abordagem COP.	58
12	Abordagem COP com controle de propagação de mensagens no líder.	60
13	Cenários avaliados de RSSF em grid.	64
14	Cenários avaliados de RSSF em modo randômico.	65
15	Resultados Preliminares: Média dos Eventos Processados - 49 nodos.	66
16	Resultados Preliminares: Média dos Eventos Processados - 64 nodos.	67
17	Resultados Preliminares: Média dos Eventos Processados - 81 nodos.	67
18	Resultados Preliminares: Média dos Eventos Processados - 100 nodos.	68
19	Resultados Preliminares: Número total de eventos processados: COP x Greedy x Ant x PS.	69
20	Resultados Preliminares: Média e dispersão dos eventos detectados por abordagem variando a densidade.	69
21	Média de Eventos Processados por Dia: COP e COP-Feedback vs Demais - RSSF de 49 nodos em grid.	71
22	Média de Eventos Processados por Dia: COP e COP-Feedback vs Demais - RSSF de 64 nodos em grid.	71

23	Média de Eventos Processados por Dia: COP e COP-Feedback vs Demais - RSSF de 81 nodos em grid.	72
24	Média de Eventos Processados por Dia: COP e COP-Feedback vs Demais - RSSF de 100 nodos em grid.	72
25	Desempenho em Número de Eventos - COP e COP-Feedback vs Demais Abordagens no cenários em grid.	73
26	Desempenho em Número de Dias - COP e COP-Feedback vs Demais Abordagens nos cenários em grid.	74
27	Cenários Randômicos: Média de Eventos Processados por Dia - 49 nodos.	75
28	Cenários Randômicos: Média de Eventos Processados por Dia - 64 nodos.	75
29	Cenários Randômicos: Média de Eventos Processados por Dia - 81 nodos.	76
30	Cenários Randômicos: Média de Eventos Processados por Dia - 100 nodos.	76
31	Desempenho em Número de Eventos - COP e COP-Feedback vs Demais Abordagens no cenário randômico.	77

LISTA DE TABELAS

1	Trabalhos Relacionados.	52
2	Parâmetros de Energia.	66
3	Resultados Preliminares - Média e Desvio padrão dos Eventos Processados.	67
4	Média e Desvio Padrão dos Eventos Processados - RSSFs em <i>grid</i>	70
5	Média e Desvio Padrão dos Eventos Processados - Cenários randômicos.	77

LISTA DE ABREVIATURAS E SIGLAS

ABC	<i>Artificial Bee Colony Algorithm</i>
AG	Algoritmo Genético
BnB	<i>Branch and Bound</i>
COP	<i>Constraint Optimization Problem</i>
CSP	<i>Constraint Satisfaction Problem</i>
DAG	<i>Directed Acyclic Graph</i>
CPU	<i>Central Processing Unit</i>
DCOP	<i>Distributed Constraint Optimization Problem</i>
DFS	<i>Depth-First Search</i>
DOOTA	<i>Distributed Optimal On-line Task Allocation</i>
DSA	<i>Distributed Stochastic Search Algorithm</i>
DSA-DMRP	<i>Data Similarity Aware for Dynamic multi-hop Routing Protocol</i>
DTAS	<i>Dynamic Task Allocation Scheduling</i>
FIFO	<i>First In, First Out</i>
HEED	<i>HEED: a Hybrid, Energy-Efficient, Distributed Clustering Approach for ad-hoc Sensor Networks</i>
I.A	Inteligência Artificial
JDK	Java SE Development Kit
KHOPCA	<i>K-hop Clustering Algorithm</i>
LEACH	<i>Low Energy Adaptive Clustering Hierarchy</i>
MBPSO	<i>Modified Version of Binary Particle Swarm Optimization</i>
PPGC	Programa de Pós Graduação em Computação - UFPEL
PSO	<i>Particle Swarm Optimization</i>
RSSF	Rede de Sensores Sem Fio
RSSFm	Rede de Sensores Sem Fio Móveis
SDF	<i>Synchronous Data Flow</i>

SMA	Sistema Multiagente
TASIM	<i>Task Allocation Algorithm Based on Score Incentive Mechanism</i>
UFPEL	Universidade Federal de Pelotas
WSN	<i>Wireless Sensor Network</i>

SUMÁRIO

1	INTRODUÇÃO	15
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Redes de Sensores sem Fio	19
2.1.1	Modelagem e Simulação com EBORACUM	22
2.1.2	Balanceamento de Carga em RSSF	27
2.2	Problemas de Otimização com Restrições	31
2.2.1	COPs	31
2.2.2	DCOPs	32
3	TRABALHOS RELACIONADOS	35
3.1	Balanceamento de Carga para o Nível de Rede	35
3.2	Balanceamento de Carga para o Nível de Aplicação	40
3.3	Análise Comparativa entre as Abordagens	48
4	ABORDAGEM PROPOSTA	53
4.1	Visão Geral e Definição do Problema	53
4.2	Implementação	55
4.3	Técnica de Feedback	59
5	EXPERIMENTOS	62
5.1	Metodologia	62
5.2	Resultados Preliminares	65
5.3	Resultados COP-Feedback	69
5.3.1	Resultados Experimentais nos Cenários em Grade	70
5.3.2	Resultados Experimentais nos Cenários Randômicos	73
6	CONCLUSÃO	78
	REFERÊNCIAS	81

1 INTRODUÇÃO

As redes de sensores sem fio (RSSFs) ou WSNs (do inglês, *Wireless Sensor Networks*) são compostas por uma ou mais estações-base (*sink*) e de nodos sensores que trabalham juntos monitorando e extraindo dados de um ambiente (YICK; MUKHERJEE; GHOSAL, 2008). As RSSFs são utilizadas em aplicações de monitoramento e rastreamento de grandes áreas como, por exemplo, áreas de plantio (MOR; SAROHA, 2013) e regiões de conflito militar (MISBAHUDDIN et al., 2018; CHACHRA; MAREFAT, 2006; HOUSSAINI et al., 2018; DERAKHSHAN; YOUSEFI, 2019). Cada nodo pertencente à rede, comumente, possui recursos limitados de *hardware* como, por exemplo, uma baixa capacidade de processamento e uma interface de comunicação sem fio com cobertura limitada (MOR; SAROHA, 2013). Um nodo pode enviar e receber mensagens dos vizinhos cobertos pelo seu raio de comunicação e sensoriar a região interna do seu raio de sensoriamento (HOUSSAINI et al., 2018). Já a estação-base funciona como um *container* das informações coletadas e processadas pelos nodos. Além disso, considera-se a estação-base um dispositivo externo às RSSFs que, geralmente, está conectada à Internet.

Usualmente, os nodos empregam baterias como fonte de energia e a carga destas baterias influenciam o tempo de vida da rede e a disponibilidade dos serviços prestados. A descarga de bateria dos nodos não ocorre de maneira homogênea, uma vez que a carga de trabalho dos nodos pode variar dependendo da sua localização ou, até mesmo, da aplicação. Assim, alguns nodos podem ficar mais rapidamente sem bateria que os demais, deixando lacunas na cobertura, ou ainda isolando outros nodos, quando estes dependem de nodos inativos para se comunicar. Esta morte precoce é ainda mais limitante quando trata-se de nodos próximos à estação-base, inviabilizando o repasse dos dados coletados pelos demais nodos da rede.

A elaboração de técnicas que tenham a capacidade de gerenciar o consumo de energia dos nodos é essencial para que as RSSFs possam estar com a maior quantidade de nodos ativos pelo maior tempo possível e capturar o maior número de eventos durante o tempo de vida da rede (ROSA et al., 2018; CALISKANELLI et al., 2013a). Isso é ainda mais relevante quando o ambiente onde a rede opera é de difícil acesso,

inviabilizando ou dificultando a manutenção ou a recarga de nodos.

As técnicas de otimização em RSSFs são consideradas um grande desafio, especificamente, no que tange o controle do gasto energético dos nodos devido a natureza dinâmica e distribuída das RSSFs. A auto-adaptação exigida pelas redes, aliada a distribuição, motiva o emprego de soluções bioinspiradas (DERAKHSHAN; YOUSEFI, 2019; SAMBO et al., 2019; FERREIRA; BRISOLARA; INDRUSIAK, 2015; CALISKANELLI et al., 2013b). Os recursos limitados de *hardware* destes dispositivos inviabilizam o uso de algoritmos computacionalmente complexos (CALISKANELLI, 2014) e motivam o uso de soluções aproximativas e/ou heurísticas. O emprego de técnicas de Inteligência Artificial (IA) para ambientes de RSSFs tem sido bastante explorado (BARRIUSO et al., 2018), inclusive para o balanceamento de carga nos seus diferentes níveis (FARINELLI; ROGERS; JENNINGS, 2014; CALISKANELLI et al., 2013a; FERREIRA; BRISOLARA; INDRUSIAK, 2015; PEREIRA et al., 2020).

As técnicas de balanceamento de carga podem contemplar tanto o nível de rede quanto o nível de aplicação. As técnicas de nível de rede estão preocupadas no roteamento das informações coletadas e permitir que estes dados possam chegar à estação-base, preferencialmente, consumindo o mínimo de energia dos nodos participantes do processo. As soluções no nível de aplicação consideram também a distribuição da carga de processamento (CALISKANELLI, 2014; PEREIRA et al., 2020).

Dentre os trabalhos científicos da área de balanceamento de carga em RSSFs, encontra-se um número maior de trabalhos ligados ao balanceamento de carga em nível de rede. Isso deve-se ao fato de por muito tempo o gasto energético das redes ser atribuído, principalmente, a comunicação (ZUHAIRY; ZAMIL, 2018; ANASTASI et al., 2004; SINHA; JOSE, 2017; YANG et al., 2014). Neste sentido, a maioria dos *surveys* já publicados na área de balanceamento de carga em RSSFs contemplam somente soluções de nível de rede (MOHAMED et al., 2018; ZADE et al., 2018; RAJKUMAR et al., 2018; XU; COLLIER; O'HARE, 2017; TYAGI; KUMAR, 2013). No entanto, o consumo energético presente no processamento não pode ser mais descartado, devido ao surgimento de RSSFs mais complexas como, por exemplo, redes responsáveis por aplicações de Internet das Coisas (IoT, do inglês, *Internet of Things*) (YU; HUANG; GARCIA-ORTIZ, 2018; MISBAHUDDIN et al., 2018) e sistemas mais sofisticados de monitoramento (JIN et al., 2013). Estes sistemas de monitoramento trabalham com câmeras sem fio *real-time*, extraíndo do ambiente monitorado uma grande quantidade de dados visuais que requerem processamento de vídeo ou imagens (JIN et al., 2013) (WANG et al., 2015). O processamento de uma grande quantidade de dados ou a execução de algoritmos complexos torna o custo de computação comparável, e até algumas vezes maior, ao custo de comunicação. Nestes casos, para prolongar de maneira eficiente o tempo de vida da rede é preciso também equilibrar a carga de trabalho no nível de aplicação considerando, simultaneamente, o processamento de

eventos/tarefas e o roteamento das informações (YU; HUANG; GARCIA-ORTIZ, 2018; JIN et al., 2013).

Técnicas bioinspiradas *Pheromone Signaling* (PS) (CALISKANELLI et al., 2013a) e *Ant-based* (FERREIRA; BRISOLARA; INDRUSIAK, 2015) foram propostas para o balanceamento de carga em nível de aplicação. Estas decidem de forma descentralizada qual nodo será o responsável pelo processamento de um evento detectado por nodos com cobertura redundante em RSSFs reativas. Apesar de apresentarem melhorias, por serem heurísticas, estas não garantem um resultado ótimo.

Em outra frente, FARINELLI et al. (2008) emprega o algoritmo MAXSUM para organizar um esquema de ativação e hibernação de nodos e assim balancear a carga na rede. O MAXSUM é um algoritmo para resolver DCOP (do inglês, *Distributed Constraint Optimization Problems*) que são abordagens empregadas em sistemas distribuídos ou multiagentes (DORRI; KANHERE; JURDAK, 2018). Entretanto, os DCOPs completos para atingir a solução ótima demandam, geralmente, um crescimento exponencial de algum aspecto do algoritmo o que impede o uso em aplicações com dispositivos embarcados com limitações de hardware (FARINELLI; ROGERS; JENNINGS, 2014). Este custo adicional pode ser evitado por soluções semi-distribuídas (GUPTA; BAYAZIT; MAHAJAN, 2005) ou centralizadas. Como a redundância é limitada mesmo em RSSFs densas, uma solução semi-distribuída pode substituir tanto uma abordagem totalmente distribuída - pois exige um número menor de mensagens trocadas entre nodos - e uma abordagem totalmente centralizada (impraticável para ambientes reais de RSSFs).

Esta tese propõe uma solução baseada nos Problemas de Otimização de Restrição - COP (do inglês, *Constraint Optimization Problems*) para resolver o balanceamento de carga em nível de aplicação para RSSFs reativas de forma semi-distribuída. O objetivo é contornar o custo das soluções baseadas em DCOPs, que exigem muita comunicação, e alcançar um desempenho superior ao obtido pelas abordagens heurísticas. A abordagem proposta define, entre os nodos que detectaram um evento, o nodo mais apto para o processamento e desativa a capacidade de processamento dos demais nodos da vizinhança, evitando redundância no processamento. A deliberação ocorre em um único nodo que é nomeado líder analisando informações recebidas dos seus vizinhos. Para a redução da sobrecarga do envio de mensagens no contexto do balanceamento, foi incorporado à abordagem um mecanismo de controle de mensagens redundantes. Esta versão melhorada foi nomeada COP-Feedback e a versão anterior foi chamada apenas de COP. Através de experimentos realizados via simulação de alto nível de abstração, a eficiência da abordagem proposta é comparada às técnicas heurísticas *Ant-based* e PS e a uma solução gulosa empregada como *upper-bound*. Os cenários empregados na análise comparativa consideram RSSFs de diferentes densidades e dois esquemas de implantação diferentes (grade e randômico).

O restante do texto está organizado da seguinte forma: o Capítulo 2 contempla os conceitos fundamentais das RSSFs, dos COPs e DCOPs. Além disso, discute os principais desafios encontrados nas soluções de balanceamento de carga e aborda o simulador EBORACUM usado para a modelagem de RSSFs; o Capítulo 3 trata os trabalhos relacionados; os Capítulos 4 e 5 detalham a abordagem proposta e os experimentos realizados; e, o Capítulo 6 apresenta as conclusões e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo revisa os principais conceitos e fundamentos teóricos para a compreensão deste trabalho. Definições importantes sobre RSSFs são apresentadas na Seção 2.1. Nesta seção além de uma revisão de conceitos, aplicações e classificações, também é revisada a modelagem e simulação de RSSFs, com ênfase especial no simulador EBORACUM empregado no trabalho (Seção 2.1.1), bem como é discutido o problema de balanceamento de carga em RSSFs (Seção 2.1.2). Por fim, o capítulo apresenta os principais fundamentos a respeito dos COPs e dos DCOPs na Seção 2.2.

2.1 Redes de Sensores sem Fio

As RSSFs são constituídas por um grande número de pequenos sensores, autônomos, eletronicamente carregados e com recursos limitados de processamento, de memória, de bateria e de rádio transmissores (CALISKANELLI, 2014). O canal sem fio de comunicação dos nodos provê um meio de transferência dos dados coletados por um nodo para um outro nodo vizinho ou, até mesmo, para o meio externo da rede (POTDAR; SHARIF; CHANG, 2009). Além disso, as RSSFs possuem uma ou mais estações-base (ou *sink*) que funcionam como um canal de comunicação da rede com o ambiente externo visto que, normalmente, estas redes estão ligadas à internet, repassando as informações extraídas da rede para os usuários finais (BRITO; PERALTA, 2008).

A Figura 1 ilustra duas arquiteturas comumente encontradas em aplicações de RSSFs, considerando a conectividade entre nodos. Na Figura 1a é ilustrada uma arquitetura com conexões em *single-hop*, onde cada nodo consegue se comunicar diretamente com a estação base, enquanto na arquitetura da Figura 1b a mensagem precisará ser transmitida em vários saltos o que denomina-se *multi-hop*. Usando o roteamento *multi-hop*, um nodo que não esteja próximo à estação base, terá de transmitir a mensagem para um vizinho, que a retransmitirá para outro vizinho, até que esta chegue à estação base.

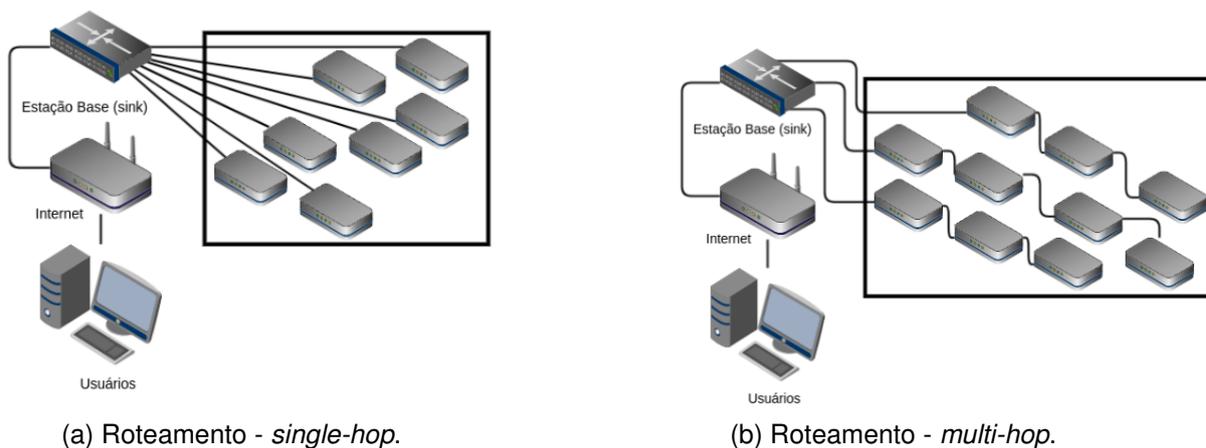
(a) Roteamento - *single-hop*.(b) Roteamento - *multi-hop*.

Figura 1 – Arquitetura Básica das RSSFs.

É possível classificar as aplicações de RSSFs em dois grupos: de **rastreamento** e **monitoramento** (YICK; MUKHERJEE; GHOSAL, 2008). As aplicações de rastreamento podem ser usadas para rastrear inimigos em ambientes de guerra, identificar algum animal e rastrear o tráfego de carros em cidades inteligentes. Já as aplicações de monitoramento, são usadas para observar as variáveis do ambiente monitorado e identificar situações de risco de insegurança podendo ser aplicadas em diferentes áreas, tais como, militar, saúde, automação industrial, meio-ambiente, entre outras. Na área de saúde, quando a rede de sensores monitora uma pessoa, um médico pode monitorar remotamente os sinais vitais do seu paciente. Na área de habitat animal estas redes monitoram, por exemplo, o comportamento de pássaros e zebras para estudos biológicos. Na área pública, utiliza-se RSSFs para monitoramento estrutural, industrial e químico, podendo monitorar a concentração de poluentes, ou ainda detectar e prevenir acidentes. Na área militar é possível monitorar aspectos de segurança como, por exemplo, a detecção de presença de veículos suspeitos em uma área monitorada (AKYILDIZ et al., 2002). A Figura 2 ilustra estes dois grupos.

A conectividade entre os nodos serve de garantia para que os dados coletados pelos nodos da rede possam chegar até a estação-base. Além disso, a conectividade assegura que qualquer análise feita com os dados estará utilizando os dados globais da rede - e não somente dados locais de uma região específica. Desta maneira, é essencial evitar que nodos fiquem desconectados ou isolados da rede. Muitas vezes um nodo depende de sua conexão com outros nodos para repassar dados e fazê-los chegar à estação-base (quando utiliza roteamento multi-hop). Esta comunicação entre nodos é estabelecida quando os nodos encontram-se dentro do raio de comunicação. Obstruções no ambiente podem limitar ou restringir a conectividade que, por sua vez, afeta a conectividade global da rede (YICK; MUKHERJEE; GHOSAL, 2008). Assim, não é recomendável deixar nenhum nodo isolado pois a informação deste nodo poderá não ser encaminhada para a estação-base e não ser utilizada de fato. Nodos que

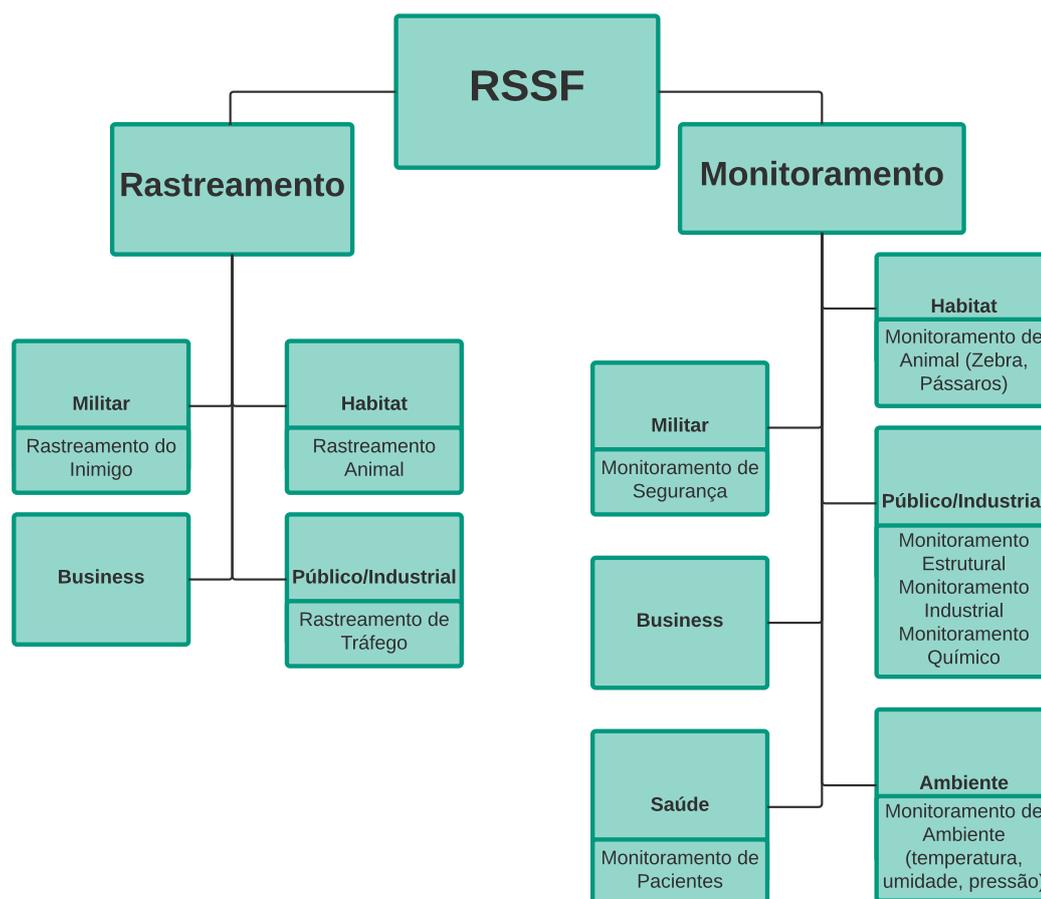


Figura 2 – Visão Geral das Aplicações das RSSFs adaptada de YICK; MUKHERJEE; GHOSAL (2008).

estão mais próximos da estação-base sofrem uma maior degradação em sua bateria pois uma maior quantidade de mensagens são retransmitidas por estes. A presença de sensores móveis é uma alternativa viável utilizada para manter a cobertura da rede por mais tempo a medida em que nodos vão se tornando inativos por esgotamento de bateria. Entretanto, o movimento dos nodos provoca um consumo adicional de bateria, se o motor empregar a mesma bateria. Desta forma, é preciso avaliar o custo-benefício de cada alternativa, definindo o melhor tipo de rede para cada aplicação.

A cobertura é um outro ponto que merece atenção nas RSSFs pois uma má disposição dos nodos no ambiente prejudica a leitura dos dados e, conseqüentemente, da mesma forma que a conectividade, faz com que a rede extraia somente informações pontuais que podem prejudicar o desejável monitoramento. Uma distribuição não uniforme pode provocar furos de cobertura na rede quando diferentes pontos de monitoramento são exigidos (SAHA; DAS, 2016). Além disso, uma distribuição não uniforme ou randômica pode resultar em sobreposição de áreas de cobertura, ou seja, muitos nodos cobrindo uma mesma área, enquanto outras áreas ficam descobertas ou com poucos nodos.

Uma RSSF de detecção de vida selvagem através do som é um bom exemplo para ilustrar tais problemas (CALISKANELLI, 2014). Como redes deste tipo são densas, um mesmo som é detectado, simultaneamente, por mais de um nodo. Nestes sistemas, é possível que vários nodos processem o mesmo som em um mesmo instante, encurtando o tempo de vida destes nodos por esgotamento de bateria e diminuindo, especificamente, o tempo de vida da rede. Se nodos próximos ficarem inativos, ou seja, com a bateria descarregada, em RSSFs com roteamento *multi-hop*, torna-se impossível encaminhar as informações dos nodos mais extremos até à estação-base.

Desde a definição do conceito de RSSFs e a introdução do nodo como elemento base que pode interagir com os demais nodos e formar sistemas complexos, torna-se cada mais valiosa a existência de ferramentas de simulação para implementação e testes de softwares/algoritmos/protocolos de uma forma mais eficiente, minimizando tempo e dinheiro (MENICHELLI; OLIVIERI, 2010). Com o intuito de suportar a modelagem e simulação de redes de sensores sem fio, vários simuladores foram construídos. Uma revisão ampla de ferramentas de simulação de RSSFs pode ser encontrada no trabalho intitulado *Survey of wireless sensor networks simulation tools for demanding applications* (KORKALAINEN et al., 2009). Neste trabalho, a ferramenta EBORACUM foi empregada como infra-estrutura de modelagem e simulação, a qual será apresentada na Seção 2.1.1.

2.1.1 Modelagem e Simulação com EBORACUM

O EBORACUM¹ é um simulador extensível utilizado para avaliação de RSSFs reativas e adaptativas, o qual provê primitivas de modelagem de alto nível de abstração. O EBORACUM pode ser facilmente estendido a fim de suportar diferentes tipos de nodos, eventos, bem como, estratégias adaptativas (BRISOLARA et al., 2016). Dentro do EBORACUM é possível desenvolver ambientes de RSSF para simulação de novas abordagens de balanceamento de carga, permitindo mensurar a eficiência destas abordagens por meio de métricas de avaliação.

O EBORACUM foi desenvolvido em JAVA e utiliza os *frameworks* Ptolemy e VirtualSense como ponto de partida para criação, visualização e simulação dos modelos propostos. Primitivas suportadas por estes *frameworks* foram reusadas e, por vezes estendidas, na construção do EBORACUM. O pré-requisito para instalação é JDK 8 (*Java Development Kit*). O EBORACUM pode ser instalado no Windows, Linux e MacOS (BRISOLARA; FERREIRA; INDRUSIAK, 2015).

Os elementos que constituem uma RSSF podem ser adicionados de maneira *drag-and-drop*. As simulações podem ser tanto construídas graficamente como também criadas através de códigos JAVA. Estes códigos, criados pelos usuários, devem instanciar um conjunto de primitivas. As primitivas são utilizadas para a criação de um

¹<https://sourceforge.net/projects/eboracum/>

arquivo *XML*². O EBORACUM utiliza o arquivo *XML* como parâmetro de entrada para a simulação. Além disso, o EBORACUM apresenta uma hierarquia de classes que são utilizadas como base e que podem ser estendidas a fim de simular outras redes ainda não contempladas. Há trabalhos que já estenderam o EBORACUM visando contemplar o maior número de configurações possíveis para os nodos e para RS-SFs (FERREIRA; BRISOLARA; INDRUSIAK, 2015; GONZALEZ; FERREIRA; BRISOLARA, 2017). O trabalho proposto por GONZALEZ; FERREIRA; BRISOLARA (2017) estendeu o simulador e avaliou a adoção de nodos móveis para aperfeiçoar a cobertura e o tempo de vida da rede. Além disso, combinou a técnica proposta de nodos móveis com outras duas técnicas bioinspiradas de balanceamento de carga e analisou o impacto da mobilidade no desempenho atingido por estas técnicas.

A Figura 3 ilustra um cenário de simulação para uma rede sem fio com nodos dispostos no formato de *grid*. Os nodos são representados por pontos azuis. Os raios de cobertura de cada nodo são representados por círculos azuis esmaecidos. Na Figura, a cor cinza expõe nodos totalmente descarregados.

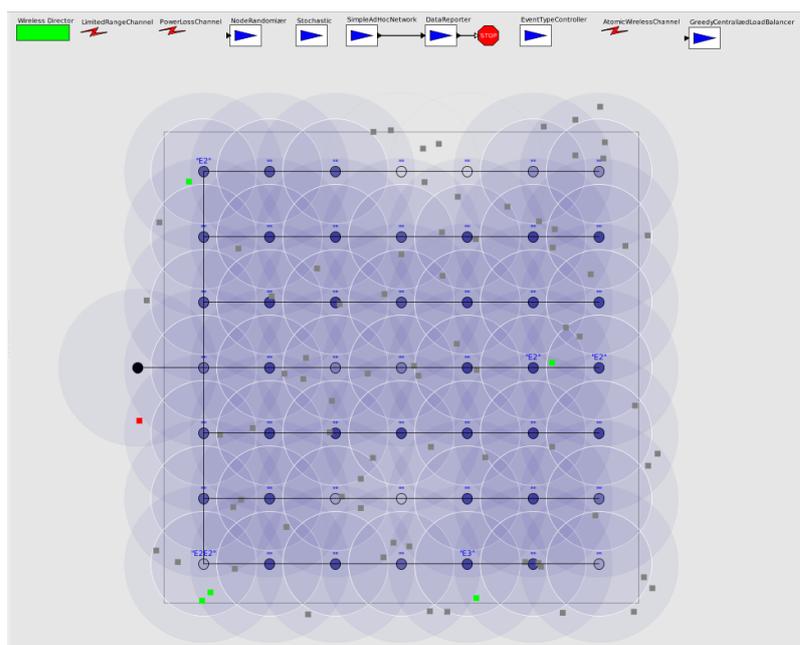


Figura 3 – Uma simulação no EBORACUM.

O EBORACUM é dividido em duas partes: a camada de **plataforma** e a camada de **aplicação**. No camada de **aplicação**, primitivas baseadas em modelos probabilísticos são empregadas para suportar avaliações de performance em redes enquanto executa cenários reais. A camada de **plataforma** reutiliza o *VirtualSense* para detalhar a comunicação entre os nodos. Toda infraestrutur reutiliza os conceitos e as facilidades do *Ptolemy II*, principalmente, para controle da simulação dos modelos (MARQUES et al., 2016).

²<https://www.w3.org/XML/>

2.1.1.1 Camada de Plataforma

Nodos no EBORACUM podem ser do tipo **nodos-sensores**, **nodos-intermediários** ou **nodos-estação-base**. **Nodos-sensores** têm uma limitada área de sensibilidade, a qual define se podem captar ou não um evento específico no ambiente. De acordo com a arquitetura baseada em redes de estações-base, um **nodo intermediário** é um nodo que está localizado entre alguma extremidade da rede e os demais nodos. Os **nodos estação-base** estão conectados através de um mesmo canal de comunicação, com os outros nodos da rede, respeitando os raios de cobertura deste canal (MARQUES et al., 2016).

Os nodos são distribuídos pela área de interesse e suas conexões são construídas, respeitando a amplitude do canal de comunicação e considerando uma forma para que toda informação salva localmente por cada nodo atinja a estação-base (ou estações-bases). Se um nodo estiver inativo ou morto (falha ou falta de bateria), as conexões são dinamicamente reconstruídas, a fim de encontrar um outro caminho para reconectar os demais nodos e evitar caminhos dependentes do nodo inativo. As conexões da rede são montadas atualmente considerando o menor caminho que um nodo tem para alcançar à estação-base. O EBORACUM é flexível o bastante para modelar vários algoritmos de roteamento através da especialização das classes fornecidas pelo *framework* (GONZALEZ; FERREIRA; BRISOLARA, 2017).

Quando um evento é capturado por um nodo, o processamento correspondente às tarefas do evento é executado e uma mensagem é encaminhada para a estação-base. Para a simulação deste processamento, os nodos possuem uma Unidade Central de Processamento - CPU (do inglês, *Central Processing Unit*) que simula a execução e o processamento das tarefas. Normalmente, assume-se que os eventos são processados usando uma estratégia do tipo FIFO (*First-In, First-Out*). Entretanto, no EBORACUM o suporte a *round-robin*, preempção prioritária ou não preemptiva podem ser incluídos no *framework*. Um evento pode ser associado a um grafo de tarefas e cada tarefa pode ter diferentes custos. Cada custo determina por quanto tempo a CPU estará ocupada e quanta energia será dissipada para processar a tarefa. Estes custos serão considerados para simular a descarga dos nodos.

Para simular o descarregamento das baterias, durante a simulação, os nodos e as estações-base irão ter suas baterias recalculadas de acordo com suas atividades. O processo de descarregamento é baseado nos ciclos de CPU e nos custos definidos para cada modo de operação (processamento, comunicação e modo *idle*). O custo energético de processamento é determinado pela soma de custos das tarefas processadas por este nodo. O custo de comunicação está relacionado somente pela operação de envio de mensagem, enquanto que o custo *idle* considera o custo de manter o sensor capaz de capturar eventos e receber mensagens. O simulador abstrai aspectos de comunicação e não trata colisão e/ou interferências de mensagens

entre os dispositivos das RSSFs (MARQUES et al., 2016).

A Figura 4 ilustra um diagrama de classes que representa uma visão de alto nível de abstração da hierarquia de primitivas propostas pelo EBORACUM para modelar RSSFs. A principal classe do digrama é a *WirelessNode*. Como nodos podem ser estações-base ou nodos-sensores, a *WirelessNode* apresenta duas subclasses que são utilizadas para representar, respectivamente, os nodos-sensores e as estações-base. Geralmente, um nodo-sensor deve detectar eventos e processar tarefas associadas ao evento. Este comportamento é representado pela classe *SimpleRSSFNode*, a qual estende a classe *BasicWirelessSensorCode*. *BasicWirelessSensorCode* é uma classe abstrata que representa um nodo-sensor que tem uma CPU interna usada para processar tarefas associadas aos eventos detectados. A CPU é representada pela interface *SensorCPU* que define alguns métodos que estabelecem o comportamento genérico de uma CPU. Essa interface tem uma implementação denominada *SimpleFIFOBasedCPU* que determina um comportamento FIFO para os eventos da CPU.

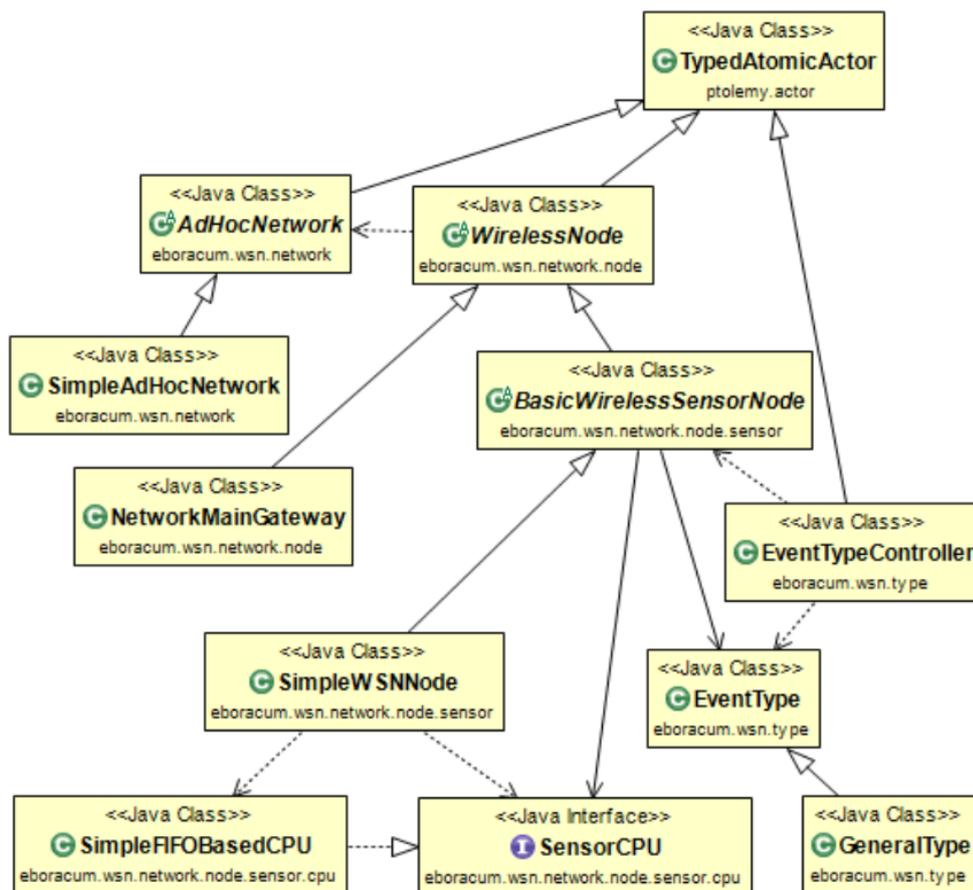


Figura 4 – Diagrama de Classes do EBORACUM - Camada de Plataforma.

2.1.1.2 Camada de Aplicação

A camada de aplicação do framework foca em aspectos não funcionais, logo a computação envolvida nas tarefas são abstraídas e a camada somente representa os custos (tempo, energia, recursos) associados nestas tarefas. Assim, para melhorar caracterizar a carga de trabalho em aplicações de redes sem fio reativas, aspectos temporais e espaciais do ambiente monitorado são incorporados no nível da camada de aplicação (MARQUES et al., 2016).

Aplicações de RSSFs reativas são comumente utilizadas para monitorar eventos. Estas aplicações dependem dos fenômenos naturais que os usuários finais estão interessados em observar como, por exemplo, mudanças de temperatura, umidade, movimentação de objetos (como veículos ou animais), entre outros (MARQUES et al., 2016).

Quando ocorre um evento dentro do ambiente em um dado local, em um determinado tempo, este evento é relacionado a um fenômeno natural. Estando na área de atuação de um nodo-sensor, um evento é somente capturado por este se estiver ativo (com bateria) quando o evento ocorrer. Assim, para representar eventos, três aspectos são relevantes: espaço, tempo e funcionalidade. Espaço refere-se a localização do evento, o tempo refere-se ao momento em que o evento ocorre, enquanto a funcionalidade indica o tipo de sensor habilitado para capturar tal fenômeno (MARQUES et al., 2016).

Na modelagem de um fenômeno natural, a incerteza e variabilidade deve ser considerada, motivando o uso de modelos probabilísticos para descrever este fenômeno. Na camada de aplicação do EBORACUM, os eventos podem ser representados por meio de modelos estocásticos, que são construídos através de observações de campo de especialistas da área (ecologistas, agrônomos, etc) e representarão aspectos do fenômeno correspondente. Estes modelos estocásticos determinam a probabilidade de um dado evento ocorrer em um determinado tempo em uma posição específica a fim de simular as variações de espaço e de tempo de eventos que ocorrem em um cenário real de uma dada aplicação. Estes modelos podem basear-se em qualquer distribuição probabilística e ser representados por um histograma 2D ou por um espectrograma 3D (MARQUES et al., 2016).

A Figura 5 ilustra um diagrama de classe hierárquico. Este diagrama de classes foca nos aspectos temporais, espaciais e funcionais dos eventos das RSSF. No topo do diagrama tem-se a classe abstrata *BasicEvent*, a qual generaliza o conceito de evento, definindo uma gama de atributos pertencentes a um determinado evento (posição, tempo de disparo). Em relação aos aspectos funcionais, qualquer evento deve ser considerado uma instância da classe *EventType*.

A classe *BasicEvent* é especializada pelas classes *AtomicEvent* e *PeriodicEvent* que representam eventos atômicos e periódicos, respectivamente. Um evento atômico

é considerado estático por natureza, mas é possível especializar-lo a fim de representar eventos estáticos que podem ter suas posições e tempos de disparo determinados por modelos estocásticos (*StochasticStaticEvent*). Um evento periódico é aquele que ocorre em intervalos e é representado pela classe *PeriodicEvent*. A classe *PeriodicEvent* tem um atributo extra denominado período que representa a duração de um ciclo de um evento repetitivo.

A classe *PeriodicEvent* pode ser usada para representar eventos estáticos e móveis. A subclasse *StochasticPeriodicStaticEvent* representa um evento estático recorrente. Quando a posição de um evento muda, isto pode ser modelado pela classe *SimpleMobileEvent* ou por alguma outra subclasse. A classe *StochasticPeriodicJumpEvent* pode ser utilizada para modelar um grande conjunto de eventos estáticos não-simultâneos com uma distribuição espacial que pode ser definida por meio de modelos estocásticos, contribuindo para a aceleração da simulação.

Um evento móvel é representado pela classe pai *SimpleMobileEvent* que define atributos que descrevem os parâmetros de movimentação como, por exemplo, a direção, a velocidade e o tempo entre mudanças de estado. Os atributos são utilizados para determinar uma nova posição do evento móvel e podem ser também descritos por distribuições de probabilidade usando *RandomMobileEvent*, *StochasticMobileEvent* ou *FullyStochasticMobileEvent*. A classe *RandomMobileEvent* é uma versão simplificada de *SimpleMobileEvent*, onde a nova posição de um evento é dada através do incremento ou decremento da menor porção aleatória (tendo como base a posição atual). A classe *StochasticMobileEvent* representa a posição inicial, a direção, a velocidade e o tempo entre mudanças de direção, mediante modelos estocásticos, mas com tempos de disparo fixados em zero. Na classe *FullyStochasticMobileEvent* todos os parâmetros, incluindo o tempo de disparo, são baseados em modelos estocásticos. A variabilidade destes parâmetros de movimentação são determinados por distribuições probabilísticas.

2.1.2 Balanceamento de Carga em RSSF

Como mencionando anteriormente, devido à dinâmica das RSSFs e os recursos limitados dos nodos, técnicas de otimização devem ser usadas durante o funcionamento da rede para estender o tempo de vida, garantir a cobertura da área de interesse e a disponibilidade dos serviços prestados. Nesta subseção, o problema de balanceamento de carga será discutido bem como quais aspectos destas RSSFs podem influenciar as abordagens de balanceamento de carga. Dentre estes aspectos encontram-se: a heterogeneidade e a homogeneidade de nodos; o esquema de implantação e de roteamento adotados; o tipo de rede; e, sua aplicação.

A ideia básica do balanceamento de carga é distribuir a carga de trabalho por todos os nodos da rede para evitar a sobrecarga de algum nodo em específico. É preciso

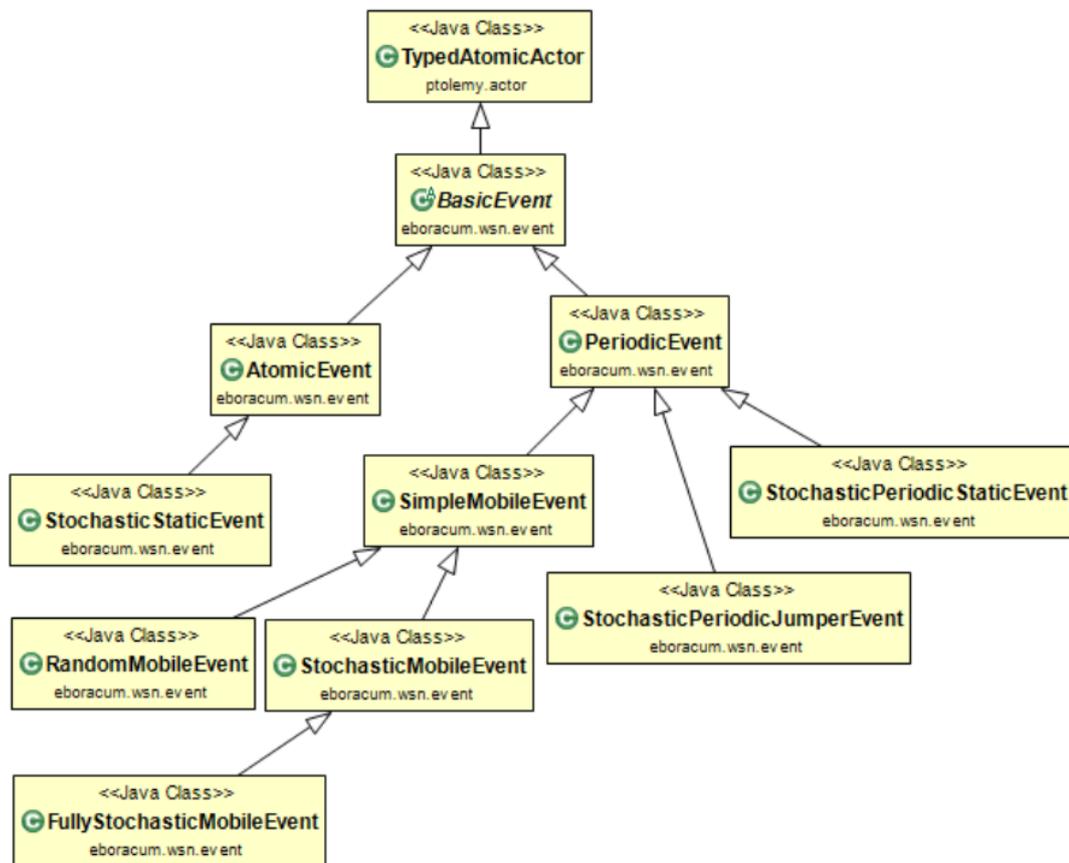


Figura 5 – Diagrama de Classes do EBORACUM - Camada de Aplicação.

gerenciar ao máximo o uso de energia para prolongar o tempo de vida da rede e, simultaneamente, coletar a maior quantidade de informação possível (CALISKANELLI; BROECKER; TUYLS, 2014; FARINELLI; ROGERS; JENNINGS, 2014). A sobrecarga de um nodo ou de alguns nodos pode resultar na degradação de desempenho da rede. A carga de trabalho a qual os nodos de uma RSSF estão submetidos pode ser considerada um somatório de tarefas de processamento, de sensoriamento e de comunicação.

Os algoritmos de balanceamento podem ser classificados em três categorias (MOR; SAROHA, 2013): estático, dinâmico e adaptativo. No balanceamento estático todo processo baseia-se no conhecimento inicial *off-line* do sistema e é realizado antes da RSSF começar a disponibilizar seus serviços. Técnicas estáticas e abordagens que implementam mecanismos centralizados de balanceamento de carga não lidam bem com a natureza dinâmica da RSSF. Algoritmos dinâmicos utilizam informações de estado do próprio sistema para a realização do processo de balanceamento, explorando sua flutuação para aperfeiçoar o desempenho e, assim, superar as limitações das abordagens estáticas. As informações podem incluir, por exemplo, a carga atual de bateria, a quantidade de eventos processados e etc. Entretanto, algoritmos

deste tipo incluem *overhead* tanto no armazenamento como na análise do estado atual da rede, em virtude da maior complexidade destes algoritmos e devido às constantes mudanças de estado que caracterizam as RSSFs. Os algoritmos adaptativos são um subtipo dos algoritmos dinâmicos que alteram, dinamicamente, seus parâmetros de forma a ajustarem seus comportamentos ao estado corrente da rede (MOR; SAROHA, 2013).

O balanceamento de carga em RSSF homogêneas é simplificado pois todos os nodos tem os mesmos raios de sensoriamento, de cobertura dos comunicadores, mesma capacidade de processamento e consomem da mesma maneira suas baterias o que simplifica a definição de técnicas para divisão da carga. Porém, quando os nodos são heterogêneos, eles podem se diferenciar em muitos aspectos, dificultando a distribuição da carga. Técnicas que se baseiam apenas no número de eventos tratados (FERREIRA; BRISOLARA; INDRUSIAK, 2015) precisarão ser revisadas, visto que uma tarefa terá um custo diferente dependendo do nodo que a realizou.

As RSSFs podem ter seus nodos distribuídos em uma região de interesse de forma aleatória (rede não-estruturada ou não-uniforme) ou seguir um esquema de implantação pré-definido ou estruturado (em malha ou em grade, por exemplo). A cobertura é um aspecto importante das RSSFs pois uma disposição inadequada dos nodos no ambiente prejudica a leitura dos dados, fazendo com que a rede extraia somente informações pontuais e não globais de todo cenário, não permitindo o desejado monitoramento. Uma distribuição não uniforme pode provocar lacunas de cobertura na rede, pois haverá áreas com maior concentração de nodos e regiões com poucos nodos e conseqüentemente lacunas de cobertura (SAHA; DAS, 2016). O esquema de implantação adotado é um fator relevante ao discutir o problema de balanceamento de carga, visto que a carga de trabalho dos nodos da rede pode ser dependente da localização de cada nodo, sobretudo a carga de comunicação em redes com roteamento multi-hop.

Tradicionalmente, as RSSFs são estáticas, ou seja, seus nodos permanecem fixos na mesma posição durante todo o tempo de vida da rede. As RSSFs móveis (RSSFMs) são um caso especial onde os nodos e, algumas vezes até mesmo a estação-base, possuem a capacidade de deslocamento (YOUNIS et al., 2004) (YICK; MUKHERJEE; GHOSAL, 2008). As RSSFMs diferem-se das demais RSSFs, pois podem se auto-reorganizar reposicionando seus nodos para corrigir possíveis falhas de cobertura (YOUNIS et al., 2004). A presença de nodos móveis é uma alternativa viável para manter a cobertura das RSSFs a medida em que vão perdendo nodos por término de bateria. Entretanto, o movimento destes nodos pela rede também pode consumir bateria. Logo, a mobilidade dos nodos interfere no balanceamento da carga e deve ser levada em conta durante a execução dos processos de otimização em RSSFs.

O esquema de roteamento e a topologia da rede também influenciam na carga da RSSF. No roteamento do tipo *multi-hop*, a informação coletada e processada por um nodo passa através de nodos intermediários até chegar à estação-base (GULERIA; VERMA, 2018). Em RSSFs *single-hop*, nodos específicos encaminham de forma direta mensagens à estação-base (sem participação de nodos intermediários). Uma consequência negativa do multi-hop é que nodos localizados mais próximos à estação-base tendem a morrer mais cedo pois participam mais vezes de processos de roteamento. Nos dois modos de transmissão, o consumo desbalanceado de energia acarreta a morte prematura de alguns nodos da rede (KHAN et al., 2018).

Os nodos das RSSFs podem trabalhar de duas formas diferentes: proativa e reativa (MARQUES et al., 2016; GONZALEZ; FERREIRA; BRISOLARA, 2017, 2016). Os sensores das redes proativas realizam o sensoriamento periódico do ambiente e submetem um novo valor até a estação-base. Esse tipo de comportamento periódico provoca um número significativo de pacotes e uma grande quantidade de retransmissões em redes em *multi-hop*. Quando a rede é reativa, esta opera capturando dados somente quando um determinado evento ocorre, reduzindo o consumo energético dos sensores e o tráfego na rede. Nas redes reativas, o processamento complexo ocorre somente quando um evento acontece como, por exemplo, em RSSFs multimídia com captura de imagens ou gravação de áudio disparadas por um evento no ambiente. Quando mais de um nodo pode detectar um mesmo evento, técnicas de balanceamento de carga de nível de aplicação devem gerenciar a disponibilidade dos recursos da rede, impedindo que um mesmo evento seja processado por mais de um nodo - fato que reduziria o tempo de vida da rede (FERREIRA; BRISOLARA; INDRUSIAK, 2015).

Soluções de balanceamento de carga propostas para o nível de rede em RSSFs têm o objetivo de encontrar o melhor caminho para que a informação coletada ou processada por um nodo possa atingir à estação-base. Estas soluções, geralmente, investigam a entrega de pacotes com sucesso, a latência, os atrasos e as falhas na conectividade e a largura de banda utilizadas no contexto de roteamento. Por outro lado, as alternativas de balanceamento de carga para o nível de aplicação buscam distribuir a carga de trabalho, decidindo o nodo que deverá executar as exigências de requisição vindas da aplicação. Nas soluções de balanceamento de carga para o nível de aplicação, tarefas computacionais deverão ser distribuídas entre os nodos da rede. Neste contexto, encontram-se trabalhos que endereçam a distribuição de tarefas periódicas entre os nodos da rede e, por vezes, através da quebra destas tarefas em sub-tarefas. Já em outras abordagens propostas para RSSFs reativas, o foco é definir, dentre os nodos que detectaram um evento, o nodo responsável pelo processamento. No caso do emprego de redes em *multi-hop* esta decisão de processamento também acabará afetando as retransmissões na rede. No Capítulo 3 técnicas e abordagens

propostas para resolver este problema serão discutidas em mais detalhes.

2.2 Problemas de Otimização com Restrições

O balanceamento de carga em RSSFs pode ser visto como um problema de alocação de recursos mapeado em um problema de otimização. Os problemas de otimização com restrições (COP) aparecem em vários setores no âmbito científico, econômico e social (SGHIR; JAAFAR; GHÉDIRA, 2018). Na área de Sistemas Multiagente - SMAs (do inglês, *Multi-agent systems*), devido a descentralização, estes problemas são resolvidos por DCOPs - *Distributed Constraint Optimization Problems*. Esta seção apresentará fundamentos dos COPs e dos DCOPs.

2.2.1 COPs

Formalmente, um Problema de Otimização com Restrição - COP (do inglês, *Constraint Optimization Problem*) é definido como sendo uma tupla $\{V, D, F\}$, onde V é o conjunto das variáveis $\{v_1, v_2, \dots, v_m\}$, m o número total de agentes, D o conjunto de domínios $\{D_1, D_2, \dots, D_m\}$ e F o conjunto das restrições. Cada restrição envolve algumas das variáveis do conjunto V e define os possíveis valores de cada variável participante da restrição de acordo com cada domínio do conjunto D (MEISELS, 2008). Em um COP, cada restrição tem sua própria função de custo que é associada a uma função global que agrega os valores de custo de todas as restrições. Um COP é resolvido otimizando esta função de global de custo (FLUECKIGER; VERMAN; BERNSTEIN, 2016), maximizando ou minimizando a função de custo, dependendo da aplicação onde ele está sendo aplicado. No âmbito da abordagem proposta nesta tese, considere que o objetivo é de minimizar a função global de custo.

Nos Problemas de Satisfação por Restrição - CSPs (do inglês, *Constraints Satisfaction Problems*), uma variante dos COPs, as funções de custo são chamadas de funções de restrição pois são responsáveis por representar funções booleanas (ou lógicas) onde cada restrição resulta em um valor verdadeiro (*true*) ou falso (*false*). Nos COPs, altera-se esta nomenclatura, pois as funções de custo não representam somente restrições, mas sim, custos em relação aos valores assumidos pelas variáveis do sistema. Por este motivo, os COPs tendem a ter um processo mais complexo de resolução do que um CSP, uma vez que devem encontrar uma associação de valores que otimize o valor das funções de custo, e não somente encontrar uma associação que satisfaça todas as restrições. Desta forma, nos COPs o objetivo não é apenas encontrar uma solução mas sim encontrar a melhor solução - algo que implica uma maior exploração no espaço de estados das variáveis (GULARTE, 2013; SANTOS, 2005).

O *Branch and Bound* (BnB) é o algoritmo de COP mais básico, que se assemelha

ao algoritmo de *Back-Tracking*, sendo estendido para algoritmos mais sofisticados. O algoritmo BnB utiliza uma árvore de pesquisa que prioriza a busca por profundidade. Depois define um valor para a variável X_i , construindo um novo subproblema. Se o subproblema for localmente consistente, a pesquisa continuará tentando resolver recursivamente novos subproblemas. Ao final, o algoritmo retorna o custo da solução global do problema. A melhor atribuição completa para o subproblema é definida como sendo o novo limite superior. O procedimento retorna *true* se o problema fornecido for consistente (deve ser explorado) ou *false* se houver uma condição de parada. O procedimento de parada verifica se o custo acumulado das atribuições parciais feitas até aquele momento excede - ou não - o limite superior (MEISELS, 2008).

2.2.2 DCOPs

Um Problema de Otimização de Restrição Distribuída (DCOP - *Distributed Constraint Optimization Problems*) pode ser visto como um COP que distribui as restrições por variáveis entre seus vários agentes (VINYALS et al., 2010). Cada agente recebe informações somente das relações que envolvem suas variáveis. Os DCOPs são desejáveis quando grandes problemas devem ser resolvidos ou é preciso evitar que um mesmo agente tenha todas as informações do sistema (FLUECKIGER; VERMAN; BERNSTEIN, 2016). Estes são geralmente, usados em problemas dinâmicos onde os agentes podem entrar ou deixar o sistema a qualquer momento (RUST, 2019) e vêm ganhando destaque na comunidade científica devido a robustez e a escalabilidade. Na literatura é possível encontrar uma vasta quantidade de aplicações para DCOPs: agendamento de encontros, RSSFs, alocação distribuída de recursos em cenários de desastre, sincronização distribuída de semáforos, gerenciamento distribuído de redes elétricas e etc.

Um DCOP consiste em múltiplos agentes distribuídos que controlam um conjunto de variáveis e funções de custo. As funções são definidas sobre as variáveis e os agentes trabalham em grupo para otimizar estas funções. Assim, a solução de um DCOP consiste em associar valores a variáveis de maneira que o somatório das funções de custo seja o menor possível (DORRI; KANHERE; JURDAK, 2018). Para coordenar um SMA, cada agente pode controlar uma variável, cujo domínio consiste de todas as ações que o agente pode realizar e as funções estabelecem custos/utilidades para estas ações que os agentes podem tomar a cada instante (FARINELLI; ROGERS; JENNINGS, 2014) (YEDIDSION; ZIVAN; FARINELLI, 2014).

A formalização clássica de um DCOP é definida por uma tupla: $\langle A, X, D, F \rangle$, onde $A = \{a_1, \dots, a_m\}$ é um conjunto de agentes e $X = \{x_1, \dots, x_s\}$ é o conjunto de variáveis e, cada variável x_i é de, exatamente, um agente a_i (mas um agente pode, potencialmente, ter mais de uma variável). $D = \{D_1, \dots, D_s\}$ é um conjunto discreto e finito de variáveis de domínio onde cada variável x_i pode representar valores do

domínio D_i . $F = \{F_1, \dots, F_n\}$ é o conjunto de funções que descreve as restrições sob as variáveis. O objetivo é encontrar variáveis que possam maximizar (ou minimizar) o somatório das funções:

$$\arg \max_x \sum_i F_i(x_i) \quad (1)$$

Os algoritmos DCOPs podem ser divididos em completos e incompletos. Os completos fornecem soluções, garantidamente, ótimas ao custo de muita computação e troca de mensagens. Por outro lado, os incompletos encontram soluções mais rapidamente, mas sem garantia de encontrar a solução ótima. Os algoritmos completos podem ser classificados em parcialmente centralizados e completamente descentralizados. Os descentralizados, por sua vez, ainda podem ser divididos em abordagens baseadas em busca ou inferência (YEOH, 2010).

Diferentemente dos parcialmente centralizados, nos descentralizados cada agente tem acesso apenas às suas funções de custo. Os algoritmos de inferência geralmente utilizam programação dinâmica para propagar as funções de custo de um agente para outro. O algoritmo mais referenciado na literatura que utiliza essa abordagem é o DPOP (*Distributed Pseudo-Tree Optimization*) (PETCU; FALTINGS, 2005). Já os algoritmos baseados em busca, como o ADOPT (MODI et al., 2005), empregam estratégias de busca distribuída para explorar o espaço de soluções até encontrar aquela que possui o custo mínimo.

Todos os algoritmos utilizados para solucionar DCOPs dependem de uma representação gráfica (DORRI; KANHERE; JURDAK, 2018). Os algoritmos mais referenciados pela literatura utilizam duas representações possíveis: grafo de coordenação e grafo-fator. Grafos de coordenação, também conhecidos como grafos/redes de restrições, são utilizados em algoritmos clássicos como DPOP e ADOPT. Nesses grafos, cada variável é representada por um nodo e as funções de custo são representadas por arestas, de tal forma que uma função, por exemplo, com os parâmetros x_1 e x_2 é representada por uma aresta conectando x_1 a x_2 . Um grafo-fator, utilizado pelo algoritmo MAXSUM e suas variações (FARINELLI et al., 2008), é um grafo bipartido composto por arestas não-direcionais e dois conjuntos de nodos. Nesses grafos, cada aresta conecta nodos de conjuntos diferentes. No grafos-fatores, um conjunto representa as variáveis das funções (nodos-variáveis), enquanto o outro conjunto representa as funções (nodos-fatores). As arestas conectam as variáveis às funções sempre que uma variável for argumento para uma função. Uma aresta existe entre um nodo-fator e um nodo-variável se, e somente se, a variável é um argumento para a função representada pelo nodo-fator. Quando comparado com um grafo-fator, os grafos de coordenação são menos flexíveis na representação das funções de custo, uma vez que permitem apenas funções binárias de custo/utilidade (FARINELLI; ROGERS;

JENNINGS, 2014). A Figura 6 ilustra um comparativo das duas representações.

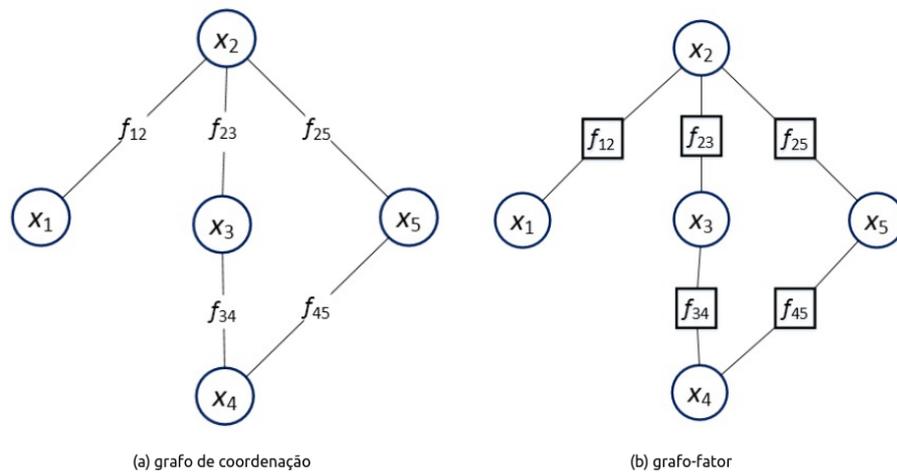


Figura 6 – Representações possíveis de um DCOP com a função $F(x_1, x_2, x_3, x_4, x_5) = f_{12}(x_1, x_2) + f_{23}(x_2, x_3) + f_{25}(x_2, x_5) + f_{34}(x_3, x_4) + f_{45}(x_4, x_5)$.

O crescimento exponencial de um algum aspecto dos DCOPs exatos ou completos são simplesmente inaceitáveis para dispositivos embarcados que possuem limitações de hardware. Mais do que isso, a maioria das abordagens ótimas exigem alguma forma de pré-processamento antes da execução do algoritmo (ex.: DPOP ou ADOPT), o que não os torna rapidamente habilitado a reagir a mudanças. No contexto de aplicação em RSSF, tanto as limitações de hardware mencionadas quanto as mudanças frequentes restringem a escolha do algoritmo a ser escolhido. Em contraste, algoritmos incompletos frequentemente convergem para soluções de baixa qualidade e, geralmente, não fornecem qualquer garantia de qualidade. A classe dos algoritmos incompletos tende a ter maior aplicação em problemas reais pois apresentam um custo computacional reduzido. Mesmo não possuindo garantia de encontrar a solução ótima, alguns algoritmos desta classe fornecem limites de distância máxima entre a solução ótima e àquela que pode ser encontrada.

3 TRABALHOS RELACIONADOS

Um dos objetivos dos projetistas de RSSFs é aumentar o tempo de vida da rede mediante a criação de soluções de balanceamento de carga que possam gerenciar a alocação de tarefas e o roteamento dos dados. Por exemplo, dentro do contexto do balanceamento em nível de rede, sensores devem coordenar suas ações para evitar que todas as mensagens passem sempre por um mesmo nodo e para o problema de balanceamento de carga em nível de aplicação, evitar que haja processamento redundante de eventos/tarefas (FARINELLI; ROGERS; JENNINGS, 2014).

Pesquisadores vem propondo alternativas capazes de decidir, por exemplo, em tempo de execução, como a carga de trabalho (processamento e comunicação) será distribuída entre os nodos da RSSF com o propósito de prolongar o tempo de vida e a disponibilidade dos serviços prestados pela rede (CALISKANELLI et al., 2013b; CALISKANELLI; BROECKER; TUYLS, 2014; FARINELLI; ROGERS; JENNINGS, 2014; FARINELLI et al., 2008; FERREIRA; BRISOLARA; INDRUSIAK, 2015). Neste sentido, este capítulo revisa trabalhos que endereçam o problema de balanceamento de carga para os níveis de rede (Seção 3.1) e de aplicação (Seção 3.2).

3.1 Balanceamento de Carga para o Nível de Rede

Técnicas de balanceamento de carga em nível de rede devem definir o melhor caminho para transmitir os dados vindos de um nodo de origem para um nodo de destino (MISBAHUDDIN et al., 2018). Além disso, as abordagens de balanceamento de carga para o nível de rede devem investigar a entrega de pacotes com sucesso, a latência, a largura de banda, os atrasos de comunicação e as falhas de conectividade entre os nodos.

Uma das mais relevantes técnicas de balanceamento de carga em nível de rede já propostas para RSSFs é o LEACH (*Low Energy Adaptive Clustering Hierarchy*) (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2002). O LEACH adota o conceito de agrupamento (*clustering*) onde cada grupo de nodos da RSSF tem um nodo líder. No LEACH, os líderes de cada grupo ou *cluster* são escolhidos aleatoriamente. Os de-

mais nodos do grupo enviam as informações coletadas para o seu respectivo líder, que compacta as informações recebidas dos nodos e as encaminha para a estação-base em um único *hop*. A Figura 7 apresenta um exemplo de aplicação da técnica LEACH em uma RSSF, ilustrando três grupos com nodos conectados entre si, e apenas um nodo por grupo conectado à estação base.

Para aperfeiçoar o desempenho do LEACH, trabalhos com variações da técnica vem propondo, por exemplo, incorporar um fator de energia para o processo de escolha dos líderes (XINGGUO; JUNFENG; LINLIN, 2016) e reduzir o tráfego de rede definindo um mesmo nodo como líder por várias iterações seguidas até esgotar totalmente sua bateria (PANDEY; KUMAR, 2019). Além destes trabalhos, a variação LEACH-C propõe centralizar a escolha dos líderes na estação-base (TRIPATHI et al., 2013).

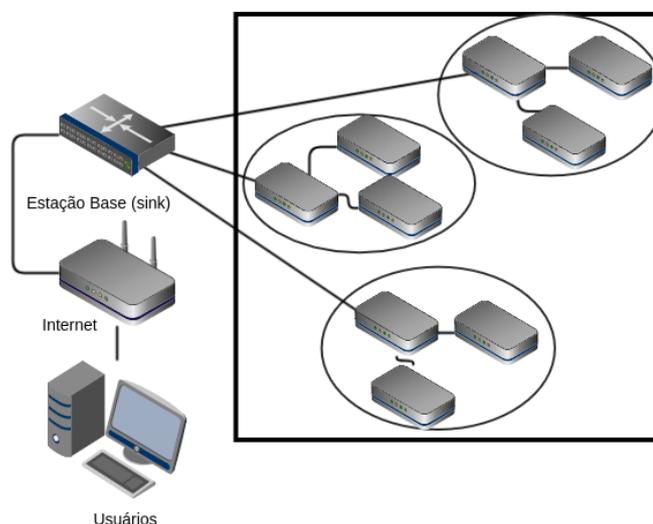


Figura 7 – Funcionamento da técnica LEACH.

O trabalho intitulado PEGASIS (*Power-Efficient GAttering in Sensor Information Systems*) (LINDSEY; RAGHAVENDRA, 2002) diferencia-se do LEACH pois não forma grupos mas, sim, uma cadeia de nodos, garantindo que cada nodo receba somente os dados de seu vizinho mais próximo e que um nodo definido, aleatoriamente, transmita estes dados à estação-base. Na abordagem, cada nodo desenvolve um pacote com seus dados e com os dados de seu vizinho e encaminha este pacote ao nodo mais próximo. Esta cadeia é obtida por uma estratégia gulosa e estabelece a sequência de nodos de menor caminho. Esta estratégia pode ser executada tanto pelos nodos ou pela própria estação-base. Caso a estação seja utilizada, após a construção da sequência, será preciso notificar os nodos da rede por *broadcast*. Se a construção da sequência estiver nos nodos, o processo começa com o nodo mais afastado da estação-base afim de assegurar que haverá nodos entre o nodo mais afastado e a estação-base.

O PEGASIS garante distâncias fixas de comunicação não havendo grandes saltos de envio e transmissão. Além disso, somente um nodo é responsável pela transmissão dos dados à estação-base. A Figura 8 ilustra o funcionamento da técnica PEGASIS que constrói uma sequência de nodos com objetivo de que os dados coletados, até mesmo de nodos mais distantes, possam chegar até a estação base. Em outro trabalho que propôs uma variação ao PEGASIS, denominada PEG-Ant (GUO; ZHANG; LU, 2010), a estratégia gulosa foi substituída por uma abordagem inspirada no comportamento de formigas.

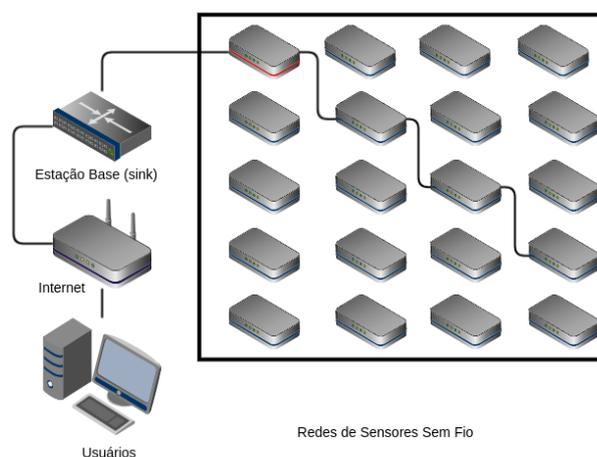


Figura 8 – Funcionamento da técnica PEGASIS.

A proposta de roteamento HEED (*Hybrid, Energy-Efficient, Distributed clustering approach for ad-hoc sensor networks*) (YOUNIS et al., 2004) aperfeiçoa o processo de definição dos líderes do LEACH, considerando a energia residual e impedindo que os líderes estejam próximos, além de controlar a sua execução estabelecendo um número máximo de iterações. O líder mais próximo da estação-base tem a responsabilidade de propagar os dados dos outros líderes para a estação-base, estabelecendo uma transmissão multi-hop inter-*cluster*, conforme ilustrado na Figura 9.

O trabalho de EVER et al. (2012) propõe uma variação do HEED, denominada UHEED, que forma grupos de tamanhos diferentes considerando a distância entre os líderes de cada grupo e a estação-base. Desta forma, quanto mais longe um líder estiver da estação-base, maior será o tamanho do seu grupo em comparação com aqueles mais próximos à estação-base. Com a criação de grupos de tamanhos diferentes, o UHEED busca reduzir a quantidade de tráfego em grupos onde o líder está mais próximo da estação-base. Além do UHEED, há uma variação do HEED denominada R-HEED (NIVEDHA; THOMAS, 2015) que impõe aos líderes um tempo pré-definido de espera para o recebimento de uma mensagem vinda da estação-base. Caso o tempo de espera termine e o líder não tenha ainda recebido nenhuma mensagem da estação, este líder deve ser substituído por outro nodo de seu grupo.

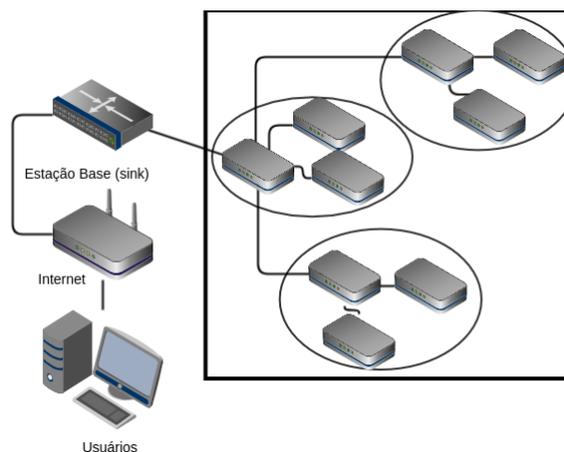


Figura 9 – Funcionamento da técnica HEED.

As inspirações derivadas do comportamento biológico de abelhas (KARABOGA; OKDEM; OZTURK, 2012; YANG; FU, 2015; PAWANDEEP; GARG; JAIN, 2016; SINHA; JOSE, 2017) e formigas (LIU; LI; WANG, 2016) foram utilizadas tanto em propostas de balanceamento de carga em nível de rede como em nível aplicação. Entretanto, nas abordagens de nível de rede a inspiração biológica baseia-se, normalmente, no caminho percorrido por estes animais em busca de comida, ou seja, das abelhas por néctar e das formigas atrás de alimento.

Em KARABOGA; OKDEM; OZTURK (2012), os autores propõem uma abordagem centralizada de balanceamento de carga em nível de rede com *clustering* onde o processo de escolha dos líderes inspira-se no comportamento das abelhas quando estão em busca de comida. A escolha dos líderes ocorre na estação-base e considera o gasto energético de comunicação entre os dispositivos como parâmetro chave para a definição dos líderes. Para justificar a decisão de centralizar a escolha na estação-base, os autores explicam que, geralmente, estes são os dispositivos com maior sofisticação de *hardware* nas RSSFs (algo que não ocorre em RSSFs homogêneas). Dentro da estação-base, este comportamento biológico inspirado nas abelhas é representado pelo algoritmo artificial de colmeias ABC (do inglês, *Artificial Bee Colony*) (KARABOGA et al., 2014). Comparada ao LEACH, que utiliza a seleção aleatória dos líderes, é esperado que esta abordagem tenha um *overhead* maior devido a realização de cálculos mais complexos.

No trabalho de SINHA; JOSE (2017), os autores propõem uma outra técnica descentralizada para balanceamento de carga em nível de rede inspirada no comportamento de abelhas. Inicialmente, a abordagem assume que todos os nodos estejam com os mesmos níveis de energia e define uma porcentagem aleatória de líderes. Em seguida, estes líderes enviam para seus vizinhos (não líderes) uma mensagem. Os vizinhos comparam a intensidade do sinal, a distância relativa e a energia residual dos líderes remetentes com propósito de definir qual deles será o seu líder dentre os inú-

meros candidatos. Os líderes devem armazenar a informação dos nodos de seu grupo e são responsáveis pelo envio de toda a informação do grupo para uma estação-base móvel. Considera-se, neste trabalho, o uso de estações-base móveis recarregáveis e com grande alcance de cobertura. Somente depois da compactação realizada por cada *sink* móvel, há o envio à estação-base principal da rede.

No trabalho desenvolvido por LIU; LI; WANG (2016), os autores propuseram uma abordagem de balanceamento de carga em nível de rede baseada no algoritmo de colônia de formigas que, anteriormente, foi proposto como solução para o problema do caixeiro viajante. Neste trabalho, os autores assumem uma RSSF com nodos dispostos de maneira não-uniforme em um ambiente retangular. Nesta abordagem, o caminho de roteamento baseia-se na concentração de feromônio produzido pela colônia de formigas durante o processo de busca por alimento. Para o roteamento multi-hop, a técnica utiliza como entrada para a função responsável pela definição da rota três parâmetros fundamentais: o valor de concentração de feromônio, o valor inverso da distância entre os nodos e seus valores residuais de energia. Além disso, para impedir a presença de muitos nodos no caminho escolhido (que prejudicaria o tempo de vida da rede), a abordagem estabeleceu três melhorias: adição de um fator de *delay* para evitar congestionamentos de pacotes; adição de um parâmetro aleatório q para explorar a topologia variável das RSSFs e considerar caminhos alternativos (em vez de permanecer no caminho com maior concentração de feromônio); e aprimorar a tabela de adjacências dos nodos próximos da estação-base com a finalidade de evitar *loops* demasiados.

O trabalho MISBAHUDDIN et al. (2018) propõe um protocolo de roteamento dinâmico multi-hop projetado para grupos de tamanho diferentes denominado de DSA-DMRP (*Data Similarity Aware for Dynamic multi-hop Routing Protocol*). Este protocolo é dividido em duas etapas: a primeira etapa realiza o agrupamento dos nodos através da similaridade dos dados usando uma técnica *fuzzy* dinâmica capaz de formar grupos de tamanhos diferentes; e a segunda etapa, essencialmente, o processo de roteamento através do algoritmo de *clustering* K-HOP (KHOPCA). Neste algoritmo de *clustering*, adiciona-se um fator de prioridade que é usado durante a escolha dos líderes e, conseqüentemente, para a definição do caminho de roteamento. Neste tipo de roteamento multi-hop com *clustering*, os líderes devem receber as informações dos nodos de seu grupo, compactá-las e encaminhá-las para um outro líder mais próximo da estação ou, envia-las diretamente à estação-base percorrendo, assim, todo o caminho de roteamento.

O fator de prioridade é obtido através de uma abordagem híbrida composta de um sistema *fuzzy* e um algoritmo de busca. Durante o cálculo do fator de prioridade, esta abordagem considera valores residuais de energia e a distância entre os líderes e a estação-base. O DSA-DMRP otimiza as regras adotadas pelo método *fuzzy*,

construindo gerações aleatórias de n vaga-lumes que são usadas pelo algoritmo de busca baseado no comportamento de vaga-lumes. Neste algoritmo, cada vaga-lume é responsável pela geração de uma solução candidata para o problema de busca. Cada novo movimento de um vaga-lume gera uma nova solução candidata mais otimizada que a anterior. Estes movimentos, denominados de atratividade, baseiam-se na distância euclidiana existente entre dois vaga-lumes adjacentes e expressam a intensidade de luz emitida por estes vizinhos.

3.2 Balanceamento de Carga para o Nível de Aplicação

As RSSFs estão cada vez mais sendo utilizadas em aplicações com alta demanda para o processamento de vídeos e imagens, de fusão de dados e de compressão de dados multimídia que exigem a execução de processos complexos (WANG et al., 2015). Esta complexidade fez com que a carga de processamento não pudesse mais ser desprezada ou desconsiderada em relação a carga de comunicação. Por este motivo, as abordagens de balanceamento de carga para o nível de aplicação vem recebendo, atualmente, a mesma importância que outrora era dada somente para as soluções de balanceamento de carga em RSSFs de nível de rede.

O objetivo do balanceamento de carga de nível de aplicação é evitar que um único nodo seja, constantemente, designado para tarefas de processamento complexas. Este processamento significativo pode ser originário de tarefas periódicas de processamento multimídia em redes proativas ou ainda processamento de captura e tratamento de eventos em redes reativas. Assim, é importante definir um esquema eficiente de alocação de tarefas que permita a colaboração entre os nodos da rede.

A alocação de tarefas em RSSFs é um problema de otimização multi-objetivo *NP-hard* (YANG et al., 2014) que impede, por exemplo, o uso de técnicas estáticas de alocação (JIN et al., 2013). Estas soluções convencionais, normalmente, não funcionam bem em ambientes dinâmicos com recursos limitados de *hardware*, como no caso das RSSFs (YANG et al., 2014; CALISKANELLI et al., 2013a). Por este motivo, algoritmos baseados em heurísticas apresentam-se como alternativas viáveis para o surgimento de novas soluções de balanceamento de carga para o nível de aplicação.

Trabalhos bioinspirados em interação animal, já haviam sido propostos para o balanceamento de carga no nível de rede baseando-se principalmente no comportamento dos insetos em busca de alimento. No entanto, nos trabalhos de balanceamento de carga em nível de aplicação, geralmente, a inspiração é na divisão da carga de trabalho que ocorre dentro das colônias e colmeias (CALISKANELLI, 2014). A divisão de trabalho emerge das relações entre os indivíduos e também entre os indivíduos e o ambiente, mediante simples padrões de comunicação, não havendo nenhum ponto central de coordenação. A alocação de tarefas deve consumir a menor quantidade

de energia possível e ser capaz de absorver as variações que acontecem constantemente no ambiente da rede.

O problema de balanceamento de carga para o nível de aplicação em RSSFs reativas foi, recentemente, endereçado pelo uso de técnicas heurísticas bioinspiradas em CALISKANELLI et al. (2013a) e FERREIRA; BRISOLARA; INDRUSIAK (2015). Ambos os trabalhos decidem de forma descentralizada qual nodo processará um evento considerando apenas os nodos que detectaram este evento, ou seja possuem coberturas redundantes. Estas técnicas são baseadas em heurísticas e portanto não encontram uma solução ótima para o problema de balanceamento de carga. Estas técnicas evitam que eventos detectados por mais de um nodo sejam processados de forma redundante. Apesar de considerarem a carga de processamento de eventos, ambas também reduzem o tráfego na rede uma vez que o processamento redundante causaria mais mensagens pela rede.

A abordagem *Pheromone Signaling* (PS) (CALISKANELLI et al., 2013a) é baseada no sistema hormonal das abelhas, a qual garante a presença de rainhas na colmeia. Durante a vida, uma abelha rainha estimula a criação de um feromônio que faz com que as abelhas operárias estejam conscientes de sua presença. Este mecanismo hormonal trabalha da seguinte forma: as abelhas operárias lambem a abelha rainha e passam o feromônio para as demais. Se o feromônio da abelha rainha não for passado para as abelhas operárias, a abelha rainha será dada como morta. Neste caso, as operárias irão selecionar uma nova larva, que será alimentada com uma grande quantidade de proteínas. Essa proteína trabalha a fim de diferenciar larvas de abelhas operárias de larvas de abelhas-rainha. Se as operárias continuarem recebendo o feromônio, elas estarão cientes de que há uma abelha rainha para orquestrar a colônia e não tomarão nenhuma providência para nomear uma nova rainha (CALISKANELLI et al., 2013a).

Inspirado no sistema hormonal das abelhas, quando a técnica PS é empregada, os nodos são, periodicamente, diferenciados dos outros nodos com a finalidade de definir seus papéis. Alguns nodos, denominados de rainha, são capazes de processar eventos capturados, enquanto os nodos remanescentes, denominados de nodos operários, permanecem em repouso. Tanto os nodos-rainha como os nodos-operários são capazes de detectar o ambiente, enfileirar e executar tarefas e se comunicar com outros nodos de sua vizinhança. Entretanto, somente os nodos-rainha irão, voluntariamente, executar as tarefas (isto é, reagir para detectar e processar o evento). Nodos-operários somente estarão responsáveis pelo repasse das mensagens. A distribuição de nodos-rainha determina a cobertura da rede e deve ser configurada para evitar redundâncias, mediante ajustes nos parâmetros do algoritmo (CALISKANELLI et al., 2013a).

A estratégia do algoritmo baseia-se na transmissão periódica do feromônio através

dos nodos-rainha que, assim, retransmitem para a sua própria vizinhança. O nível de feromônio de cada nodo decai com o tempo e com a distância em relação à fonte. Todos os nodos acumulam feromônio recebidos dos nodos-rainha. Se em um dado instante, o nível de feromônio de um nodo estiver abaixo de um *threshold*, este será convertido para um nodo-rainha. Isto, tipicamente, acontece quando o nodo está muito distante dos demais nodos-rainha ou quando o nodo-operário fica muito tempo sem receber o feromônio. Desta forma, na abordagem PS, os nodos fornecem ou não um serviço dependendo de seu estado (rainha ou operário). Esta decisão é distribuída baseada nas informações locais do nodo. As mudanças nos níveis de feromônio são usadas para orquestrar e definir a responsabilidade de cada indivíduo da rede (CALISKANELLI et al., 2013a). A otimização proposta pelo PS gerencia o *trade-off* entre a eficiência energética e a detecção de eventos em tempo de execução, maximizando a disponibilidade, enquanto reduz o consumo de energia (CALISKANELLI, 2014).

FERREIRA; BRISOLARA; INDRUSIAK (2015) propuseram uma abordagem de balanceamento de carga bioinspirada nas colônias de formigas, a qual foi denominada *Ant-based*. Assim como o PS (CALISKANELLI et al., 2013a), esta nova abordagem também foi aplicada em redes reativas com sobreposição nas áreas de cobertura. Nas colônias, as formigas são capazes de alterar seu estado atual mediante mudanças no ambiente, ou seja, dependendo do estado atual do ambiente onde vivem, as formigas desenvolvem mecanismos de divisão de trabalho. Empregando a *Ant-Based*, os nodos decidem, probabilisticamente, quem irá processar um evento detectado de uma forma totalmente distribuída. O foco é utilizar técnicas de comunicação e de decisão a fim de permitir que os nodos possam melhor distribuir, dinamicamente, a carga de trabalho do sistema. Os eventos produzem estímulos para os nodos que, baseados em seus *thresholds* internos, determinam diferentes probabilidades para o processamento destes eventos. Quando os nodos detectam um evento, cada um deles irá decidir se processa ou não o evento considerando a sua probabilidade calculada. Em síntese, a abordagem *Ant-based* trabalha modificando a probabilidade de um nodo processar um evento de acordo com o número de nodos que detectaram o mesmo evento ao mesmo tempo e o número de vezes que este mesmo nodo foi, previamente, engajado para o processamento de eventos anteriores (FERREIRA; BRISOLARA; INDRUSIAK, 2015). A abordagem *Ant-based* visa reduzir a redundância no tratamento de eventos e priorizar que o processamento seja tratado pelo nodo que trabalhou menos em processamento e que conseqüentemente tende a ter um nível de bateria mais elevado. Embora a abordagem considere a carga de processamento, seu emprego também afetará o tráfego na rede, pois quando o processamento redundante é evitado, reduz-se as mensagens trafegadas.

Um estudo comparativo publicado em FERREIRA; BRISOLARA; INDRUSIAK (2015), avaliou as abordagens *Ant-Based* e PS em diferentes cenários, variando a

densidade das redes e a distribuição dos eventos. Durante as simulações, os eventos eram distribuídos pelas RSSFs e a eficiência era avaliada através do número médio de eventos detectados por dia pela rede. Este estudo apontou que a abordagem *Ant-Based* atinge resultados bem próximos aos de PS quando é bem configurada. Além disso, a técnica *Ant-based* continua com resultados satisfatórios mesmo com a diminuição de densidade, o que não acontece com o PS que perde performance exponencialmente com a redução no número de nodos. Desta forma, *Ant-based* é, de acordo com os experimentos, menos dependente dos parâmetros iniciais, tendo uma configuração mais fácil e sendo mais estável que a abordagem PS. No estudo, os autores ainda compararam estas duas soluções com uma abordagem gulosa (não-prática) que permite a um nodo acessar o estado de bateria dos demais nodos e uma técnica aleatória, no qual sorteia-se o nodo que irá tratar o evento. Esta abordagem gulosa serviu de *upper-bound* para as demais abordagens testadas.

Em FARINELLI; ROGERS; JENNINGS (2014), a ideia de coordenar os ciclos de sensoriamento e de hibernação dos nodos e assim balancear a carga na RSSF é proposta. No trabalho, o problema de balanceamento de carga foi mapeado para um DCOP e o algoritmo MAXSUM foi empregado na resolução desta coordenação. A coordenação de ciclos foi utilizada para maximizar a probabilidade de detecção de eventos e garantir que a área monitorada estivesse ativamente monitorada por pelo menos um sensor durante todo o tempo de vida útil da rede. O algoritmo MAXSUM emprega uma representação por grafo-fator, que permite expressar explicitamente interações mais complexas entre os agentes/nodos da RSSF - algo que não seria possível com uma representação binária convencional usada por outros DCOPs. O foco era que cada agente trocasse o valor de sua variável de forma assíncrona e detectasse a possibilidade de que outra solução pudesse ser melhor do que a que está sob investigação. A abordagem permite que a computação seja assíncrona pois um agente não necessita da informação global para efetuar suas decisões locais.

A técnica foi dividida em três fases: calibração, coordenação e operação.

1. **Fase de calibração:** a fase de calibração é um procedimento relativamente longo onde os agentes precisam sincronizar seus ciclos de sensoriamento/hibernação através de troca de informação. Além disso, durante intervalos regulares, os sensores guardam, individualmente, o número de eventos observados e o número de eventos detectados por eles mesmos e por seus vizinhos;
2. **Fase de coordenação:** os números de eventos observados durante a fase de calibração atuam como um *proxy* para áreas desconhecidas de nodos-vizinhos com alguma área de cobertura sobreposta. Além disso, esta fase também captura o fato de que eventos não irão ocorrer eventualmente sob uma área inteira, mas somente em áreas específicas - ex: em estradas. Este fato modifica a fun-

ção de utilidade de cada variável - suavizando-a - mediante o acréscimo de uma equação matemática que expressa a probabilidade de um evento aparecer em uma determinada área monitorada de pelo menos um nodo;

3. **Fase de operação:** sensores seguem o esquema de ativação/hibernação determinado pela fase anterior de coordenação. Se durante esta fase um sensor falhar no algoritmo de coordenação - ele pode ser simplesmente re-executado. Além disso, se a posição dos sensores mudar ou se novos sensores forem adicionados, as fases de calibração e de coordenação podem ser executadas novamente.

A proposta do FARINELLI; ROGERS; JENNINGS (2014) foi então comparada com uma abordagem aleatória, uma abordagem onde o MAXSUM foi substituído pelo algoritmo DSA (do inglês, *Distributed Stochastic Algorithm*) e uma outra abordagem baseada no algoritmo *Simulated Annealing*. Na prática, o *Simulated Annealing* não pode ser usado para coordenar os ciclos de ativação/hibernação de nodos reais pois é uma técnica centralizada que assume ser possível ter conhecimento total da topologia da rede. Entretanto, esta foi usada como *benchmarking* de desempenho para as abordagens descentralizadas de coordenação, similar a análise comparativa realizada por FERREIRA; BRISOLARA; INDRUSIAK (2015) que empregava a técnica *Greedy* como *upper-bound*.

Da mesma forma que as abordagens PS e *Ant-based*, a abordagem baseada em MAXSUM também não garante encontrar uma solução ótima quando é utilizada em aplicações que apresentam grafos com ciclos de restrição. Ciclos de restrição acontecem quando há uma dependência mútua entre as variáveis, ou seja, quando o estado do *agente1* afeta o estado do *agente2* e, por sua vez, o estado do *agente2* afeta o *agente1* (FARINELLI et al., 2008). Vários estudos vem surgindo para garantir a convergência do MAXSUM em problemas com grafos de restrição cíclicos. Uma das alternativas mais conhecidas é excluir as restrições mais custosas do grafo de restrição com o objetivo de eliminar os ciclos (Bounded-MAXSUM). Além disso, a versão MAXSUM_AD, propõe a construção de um grafo acíclico direcionado que tende a convergir depois de um número de iterações sem remover qualquer nodo do grafo de restrição (ZIVAN et al., 2020). Por outro lado, uma alternativa seria empregar um algoritmo completo de DCOPs como, por exemplo, o ADOPT, que é capaz de encontrar soluções ótimas utilizando somente a comunicação local (MODI, 2003) (MODI et al., 2005). Entretanto, o ADOPT *standard* exige uma etapa de pré-processamento para a construção do grafo de restrições que impede sua utilização em cenários com características dinâmicas, como é o caso de ambientes de RSSFs (FARINELLI; ROGERS; JENNINGS, 2014).

No contexto de RSSF proativas, o balanceamento de carga pode ser mapeado

para um problema de alocação de tarefas em um sistema distribuído. Neste contexto, usualmente, estas técnicas dividem os nodos em grupos e as tarefas são distribuídas entre estes grupos (YANG et al., 2014) (YU; HUANG; GARCIA-ORTIZ, 2018) (WANG et al., 2015). Diferentemente, das técnicas *Ant-based* e PS, estes trabalhos não foram propostos para redes reativas e nem exploram a sobreposição das coberturas dos nodos. Nestas técnicas, todos os recursos da rede podem ser considerados na distribuição da carga, sendo seu emprego mais apropriado para redes proativas cujas tarefas são periódicas e não associadas a detecção de um evento.

No trabalho de YANG et al. (2014), autores propõem um algoritmo para alocação de tarefas aplicado a RSSFs heterogêneas baseado na otimização de enxame de partículas (PSO, do inglês *Particle Swarm Optimization*), denominado MBPSO (do inglês, *Modified Version of Binary Particle Swarm Optimization*). As tarefas e suas relações são expressas por um grafo acíclico dirigido - DAG (do inglês, *Directed Acyclic Graph*). Neste grafo, o conjunto M enumera as tarefas, o conjunto P determina os ciclos de CPU necessários para processar cada tarefa e o conjunto L representa a quantidade de dados trocada pelos nodos selecionados. A direção das arestas expressa a prioridade das tarefas dentro da rede. A rede de sensores sem fio é também expressa por um grafo, mas este é não-dirigido. Os vértices são representados pelo conjunto N que determina os nodos. O conjunto R , define a conectividade entre os nodos e e_{init} , e_j , d_{ij} e v_j representam, respectivamente, os parâmetros de energia inicial, de consumo médio, a distância entre os nodos i e j e a velocidade de processamento.

A otimização PSO tem funcionamento semelhante a um algoritmo genético (AG): inicia com uma população aleatória de soluções e vai atualizando gerações em busca de soluções ótimas utilizando uma função de *fitness*. No entanto, ao contrário dos AGs, a otimização PSO não possui operadores de evolução, como *crossover* e mutação. No PSO, as soluções potenciais, denominadas de partículas, deslocam-se pelo espaço do problema seguindo as melhores partículas atuais. Além disso, no PSO todas as partículas tendem a convergir rapidamente para a melhor solução - algo que não ocorre com todos os indivíduos de um AG (EBERHART; SHI, 2001). O trabalho de YANG et al. (2014) incorporou ao PSO a capacidade de armazenamento dos caminhos anteriormente percorridos com altos valores de qualidade e a construção de um novo procedimento de atualização para as partículas. Essas alterações trouxeram uma maior velocidade de convergência e uma maior diversidade de partículas ao algoritmo, contornando a convergência prematura para a solução ótima local, existente em outras propostas baseadas no PSO (YU; HUANG; GARCIA-ORTIZ, 2018).

A primeira linha das partículas representa a tarefa de prioridade mais alta e a última linha a de menor prioridade. Tarefas com a mesma prioridade serão dispostas em uma ordem aleatória na partícula. Para uma tarefa M qualquer, um conjunto de sensores deve ser selecionado. Cada sensor do grupo será responsável por uma fa-

tia (sub-tarefa) de M . O trabalho garante tanto a conectividade dos nodos do grupo, como também, garante que este mesmo grupo possui recursos para processar completamente a tarefa M e que cada nodo do grupo possa enviar e receber mensagens dos demais nodos deste mesmo grupo. A função de *fitness* global incorpora sub-funções de custo que, respectivamente, consideram (1) o menor tempo de execução, (2) o menor consumo de energia e; (3) o prolongamento do tempo de vida da rede (distribuição de energia). Cada partícula é codificada em um matriz $m \times n$ binária onde: m é o número total de tarefas dentro de um grafo DAG e n é número de nodos da RSSF.

O trabalho de YU; HUANG; GARCIA-ORTIZ (2018) propõe o algoritmo DOOTA (*Distributed Optimal on-line Task Allocation*) para a definição dos *clusters* de nodos, o trabalho representa este problema de balanceamento como um problema de partições de um grafo SDF (do inglês *Synchronous Data Flow*) que, sistematicamente, formula o gasto de energia e as restrições de tempo para os nodos da rede. A proposta é dividida em duas fases: (1) preparação *off-line*: focada na definição das partições (grupos); e (2) algoritmo DOOTA *on-line* de alocação de tarefas: realizada por cada grupo de maneira independente. Esta independência promove a flexibilidade na técnica, permitindo adaptar-se às mudanças dinâmicas que ocorrem, constantemente, nas RSSFs durante o tempo de vida (o que não ocorre com técnicas *off-line* e técnicas complexas centralizadas).

Um algoritmo baseado em escores foi proposto em WANG et al. (2015) para emprego em RSSFs heterogêneas. O algoritmo denominado TASIM: *Task Allocation Algorithm Based on Score Incentive Mechanism* foi proposto, primeiramente, para execução de tarefas complexas em aplicações de tempo real. O algoritmo, além de considerar o tempo limite de execução das tarefas, também realiza a migração de tarefas. Em sua dinâmica, o TASIM atribui escores de forma a recompensar nodos que finalizaram suas tarefas e penaliza nodos que não concluem a execução de suas tarefas. O foco é dividir uma tarefa complexa em sub-tarefas e alocar estas sub-tarefas para os diferentes nodos da rede afim de viabilizar uma execução colaborativa. Cada sub-tarefa pode ser designada somente para um único nodo, mas cada nodo é capaz de processar diversas sub-tarefas durante a sua vida. A abordagem assemelha-se ao (YANG et al., 2014) no que tange a execução colaborativa de tarefas complexas.

No TASIM, os nodos são divididos em três tipos: (1) estáveis, (2) instáveis e (3) mortos. Os nodos que estão com recursos suficientes e que não foram alvo de nenhum ataque malicioso e que não sofreram alguma falha são considerados estáveis. Caso contrário, são instáveis. Nodos instáveis podem migrar suas tarefas incompletas para um outro nodo, garantindo um tempo de execução aceitável. Já os nodos mortos devem ter suas tarefas realocadas pela própria abordagem. Dentro do TASIM, o processo de balanceamento de carga é dividido em três etapas: (1) alocação de tarefas no líder; (2) migração de tarefas dos nodos instáveis; e (3) realocação de

tarefas incompletas de nodos mortos. Os grupos são formados pelos nodos vizinhos e a alocação de tarefas ocorre dentro dos líderes. O nodo líder pode se comunicar diretamente com a estação-base. Além disso, como acontece em outras abordagens de balanceamento em nível de rede, pode enviar e receber mensagens dos outros líderes. Os demais nodos se comunicam com os seus vizinhos e com o seu líder. Diferente dos demais membros, o líder deve ter um alto nível de energia e uma alta velocidade de processamento. Dentro do processo de alocação, os nodos são selecionados por sua confiabilidade, histórico de participação, consumo de energia, nível de energia residual e etc. Os nodos são divididos em domínios de acordo com a quantidade de recurso e habilidade. Nodos de diferentes domínios podem se comunicar com nodos de outro domínio para completar sua tarefa de forma colaborativa. Se não houver nodos suficientes classificados como estáveis, os nodos do segundo tipo serão selecionados e, assim por diante. Os líderes, dentro da proposta, são responsáveis pela alocação de tarefas de seu grupo.

Nesta abordagem, um DAG ($G = (T, E, w, l)$) foi novamente utilizado para a modelagem das tarefas. Os vértices T expressam as tarefas que precisam ser executadas e as arestas E representam as dependências entre as tarefas. Cada tarefa tem um *deadline* de execução t_{dl} . Se há uma aresta direcionada e_{ij} vinda de um vértice t_i para um outro vértice t_j , é preciso interpreta-lá da seguinte forma: a execução da sub-tarefa t_j precisa do *output* da sub-tarefa t_i . Além disso, a sub-tarefa t_j deve ser processada somente depois de t_i ser totalmente finalizada. O peso w_i do vértice representa a quantidade de computações necessárias para a tarefa. O peso l_i da aresta direcionada expressa o valor de comunicação existente entre tarefas. Se uma tarefa t_j é agendada para um nodo, enquanto a sua tarefa predecessora t_i é agendada para um outro nodo, a comunicação entre os dois nodos é necessária. Portanto, t_j não pode começar sua execução até a comunicação esteja completa e o resultado de t_i seja totalmente recebido. Se ambas t_i e t_j sejam definidas para um mesmo nodo, a latência de comunicação é zero. Desta forma, a tarefa t_j pode começar depois que a execução de t_i esteja finalizada.

Os algoritmos genéticos também já foram usados como alternativa para o problema de balanceamento de carga em nível de aplicação. O trabalho de JIN et al. (2013) propõe um algoritmo genético evolutivo DTAS (do inglês, *Dynamic task allocation scheduling*) para redes multi-hop heterogêneas de sensores sem fio multimídia. Entretanto, algoritmos genéticos não podem ser diretamente aplicados em redes de sensores sem fio pois estes algoritmos não estão adaptados a trabalhar com redes que estão em constante mudança. Neste caso, o principal desafio do DTAS é equilibrar o tempo de execução do algoritmo genético proposto e, ao mesmo tempo, estar atento às mudanças que ocorrem, constantemente, nas RSSFs durante o tempo de vida.

Para que seja possível adotar um algoritmo genético dentro das RSSFs, a abordagem, periodicamente, executa este mesmo algoritmo genético e efetua, paralelamente, um processo de autoaprendizagem. Ao contrário dos algoritmos genéticos convencionais que, geralmente, produzem soluções somente depois de inúmeras iterações, o processo de autoaprendizagem contido no DTAS fornece soluções parciais sub-ótimas capazes de lidar com as mudanças da rede. Este espaço de soluções é atualizado por meta-heurísticas, disparadas a cada nova mudança na rede de sensores sem fio. A função de *fitness* do algoritmo genético considera a energia dos nodos e a velocidade de processamento exigida por cada tarefa. Além disso, o DTAS possui um processo de recuperação responsável por falhas nos nodos, problemas de comunicação e pelo envio dos relatórios periódicos para análise. Para modelar a carga, esta abordagem também utiliza um grafo DAG (grafo direcionado acíclico). Os vértices representam as tarefas e as arestas a quantidade de dados necessários para a transmissão. Modelando a carga de trabalho por este DAG, a abordagem acaba também considerando o custo de comunicação envolvido para solucionar o problema de alocação.

3.3 Análise Comparativa entre as Abordagens

O balanceamento de carga tem como ideia básica distribuir a carga de trabalho por todos os nodos da rede, evitando a sobrecarga de algum nodo em específico. Desta forma, técnicas de balanceamento de carga gerenciam o uso da energia e servem para prolongar o tempo de vida sem sacrificar o desempenho da rede em termos de disponibilidade e da qualidade dos serviços prestados (CALISKANELLI; BROECKER; TUYLS, 2014). O balanceamento de carga em RSSFs é um problema complexo onde, geralmente, as soluções propostas não conseguem contemplar as inúmeras configurações existentes de RSSFs, tendo seu emprego voltado, normalmente, para uma configuração específica.

Esta seção analisa e compara as abordagens de balanceamento de carga revisadas. A Tabela 1 resume as principais características de cada abordagem, nível do balanceamento, técnica empregada na otimização, além de procurar destacar uma característica relevante de cada abordagem.

As abordagens de nível de rede revisadas em sua maioria empregam técnicas de agrupamento, porém com estratégias diferentes para a definição de grupos e líderes, como indicado na Tabela 1. Dentre estas destaca-se o LEACH, que serviu de base para diversos outros trabalhos. No entanto, o LEACH seleciona, aleatoriamente, o líder de cada grupo sem considerar nenhum parâmetro - o que pode provocar líderes inadequados/ineficientes ou bem próximos (e não distribuídos uniformemente pela rede). Já no HEED, o líder é escolhido pelos valores residuais de energia e etc. Em

SINHA; JOSE (2017), os vizinhos utilizam a intensidade do sinal, a distância relativa e a energia residual dos nodos remetentes para definir qual deles será o seu líder dentre os candidatos. Estratégias bioinspiradas para escolha do líder foram também empregadas no DSA-DMRP (MISBAHUDDIN et al., 2018) e no KARABOGA; OKDEM; OZTURK (2012). O primeiro usa uma combinação de lógica *fuzzy* com um algoritmo bioinspirado nos vaga-lumes e o segundo emprega um algoritmo inspirado no comportamento das formigas na busca por comida. Dentre as técnicas revisadas para o balanceamento de carga em nível de rede, o PEGASIS se diferencia por não utilizar agrupamento. Um ponto negativo do PEGASIS é que, durante o processo de construção da sequência, pode-se gerar uma cadeia com muitos nodos, fazendo com que vários nodos gastem energia devido a um número maior de trocas de mensagens. Esta estratégia gulosa de criação da sequência no PEGASIS pode ser demorada (NEHRA; SHARMA, 2013) e obter somente soluções locais (GUO; ZHANG; LU, 2010).

No caso das técnicas de balanceamento de carga em nível de aplicação, as técnicas revisadas podem ser classificadas em dois grandes grupos: um deles foca na alocação de tarefas de processamento considerando todos recursos da rede (YU; HUANG; GARCIA-ORTIZ, 2018; JIN et al., 2013; WANG et al., 2015; YANG et al., 2014) e no outro encontram-se técnicas que alocam tarefas de processamento de eventos apenas entre os nodos que detectaram estes eventos (FERREIRA; BRISOLARA; INDRUSIAK, 2015; CALISKANELLI, 2014), como destacado na Tabela 1. No primeiro grupo, as soluções, usualmente, dividem tarefas complexas em sub-tarefas e as dividem entre os recursos da rede, atribuindo a um conjunto específico de nodos a responsabilidade pelo processamento de uma fatia desta tarefa. Já no segundo grupo, a divisão de trabalho é apenas entre os nodos que sensorearam um mesmo evento, ou seja consideram apenas aplicações em RSSF reativas. Em sua maioria, as técnicas no nível de aplicação empregam algoritmos bioinspirados aproximativos, como sintetizado na Tabela 1, visando reduzir a complexidade da divisão de trabalho.

As inspirações biológicas podem reduzir a complexidade de técnicas de balanceamento de carga fundamentadas em algoritmos exaustivos mas, em alguns casos, podem não atingir a solução ótima como, por exemplo, em alternativas de balanceamento que adotam variações de algoritmo genético (AG). Para que estas técnicas com AG tenham alta probabilidade de encontrar a solução ótima do problema, é preciso, geralmente, um longo tempo para a criação de muitas gerações que, mesmo assim, podem encontrar somente soluções locais (YU; HUANG; GARCIA-ORTIZ, 2018). Além disso, estes algoritmos devem ser ajustados pois, de forma direta, não funcionam adequadamente com as características dinâmicas das RSSFs (JIN et al., 2013). Na técnica DTAS (JIN et al., 2013), o algoritmo genético proposto pela abordagem tem seu espaço de soluções atualizado a cada nova mudança na rede. A proposta DOOTA propõe um processo de alocação de tarefas independente capaz de, rapida-

mente, atender às mudanças dinâmicas que ocorrem nas RSSFs. O MBPSO lançou mão da técnica de enxame de partículas para propor uma alternativa para alocação de tarefas, enquanto o TASIM (WANG et al., 2015) propõe um processo de alocação de tarefas baseadas em score ou recompensa, que considera o histórico de participações dos nodos, além do nível residual de energia.

As soluções *Ant-based* (FERREIRA; BRISOLARA; INDRUSIAK, 2015) e PS (CALISKANELLI, 2014) são alternativas de balanceamento de carga em nível de aplicação indicadas para RSSFs reativas com sobreposição na cobertura de nodos, ou seja, onde mais de um nodo é capaz de processar um mesmo evento (ideal para redes homogêneas). Estas técnicas minimizam o custo energético evitando, essencialmente, que mais de um nodo venha processar um mesmo evento, evitando o processamento redundante. Na técnica *Ant-based*, os nodos decidem se vão ou não processar o evento capturado mediante valores probabilísticos calculados com base no número de nodos que detectaram o mesmo evento e o trabalho anterior realizado pelo nodo. Em PS, a abordagem define quais abelhas estarão em repouso (operárias) e quais devem ser responsabilizadas pelo processamento dos eventos capturados (rainhas). Vale destacar que as técnicas propostas para o balanceamento de carga em nível de aplicação funcionam melhor em RSSFs reativas, pois a carga de trabalho nestas redes é menos uniforme que nas redes proativas, pois a carga depende dos eventos gerados pelo ambiente.

Em sistemas multiagente (SMAs), um problema pode ser resolvido de maneira distribuída onde cada agente é capaz de resolver uma fatia do problema. Os agentes devem agir de forma autônoma, raciocinar a respeito das possíveis consequências de suas ações e coordenar suas ações em conjunto afim de contemplar os objetivos gerais destes sistemas. Além disso, diversos conceitos encontrados em SMAs também podem ser vistos em RSSFs como, por exemplo, o conceito de agentes e nodos reativos e proativos, agentes/nodos heterogêneos e homogêneos, agentes/nodos móveis, agentes/nodos atuando como líderes de grupos, conectividade entre agentes/nodos e, principalmente, o problema de alocação de tarefas (DORRI; KANHERE; JURDAK, 2018). Por este motivo, devido às semelhanças entre estes sistemas, técnicas comumente usadas em problemas de SMAs também são utilizadas para a construção de novas abordagens de otimização no âmbito das RSSFs como os problemas DCOPs e, mais especificamente, o algoritmo MAXSUM (FARINELLI; ROGERS; JENNINGS, 2014). O MAXSUM foi usado em uma solução de balanceamento de carga em nível de aplicação para RSSFs que permite com que a rede possa minimizar (ou eliminar) a redundância de processamento definindo o nodo mais apto para processar o evento detectado mediante ciclos de ativação e de hibernação da capacidade de processamento dos nodos participantes do processo (FARINELLI; ROGERS; JENNINGS, 2014).

Um ponto que dificulta a comparação das soluções para balanceamento de carga é a diversidade de definições utilizadas para mensurar o tempo de vida da rede. Há soluções que consideram o número de nodos vivos; outras levam em conta a cobertura; algumas baseiam-se na conectividade; outras na comunicação intergrupos e etc. A morte do primeiro nodo é uma das definições mais populares, principalmente, dentro das RSSFs com roteamento single-hop. Ainda, como ressaltado nesta revisão, na maioria das soluções, o balanceamento de carga ocorre em um único nível, deixando um dos níveis sem o devido balanceamento. Desta forma, as abordagens desconsideram o fato de que o consumo do nodo é resultante não só da comunicação (nível de rede), mas também do sensoriamento e processamento (nível de aplicação).

Em comparação com as abordagens anteriores propostas para redes reativas, a abordagem COP/COP-Feedback proposta nesta tese tem funcionamento semelhante ao da técnica *Ant-based* em dois pontos: (1) todo processo de balanceamento de carga ocorre quando um evento disparado é detectado por nodos de uma região da RSSF e; (2) ambas utilizam o número de eventos processados previamente para a deliberação do nodo responsável pelo processamento do evento. Enquanto em FARNELLI; ROGERS; JENNINGS (2014) o nível atual de bateria dos nodos é usado para ativar ou desativar nodos, o que pode não ser uma informação facilmente acessível em nível de aplicação. Um diferencial entre *Ant-based* e a abordagem proposta é que esta última centraliza no líder o processo de deliberação, enquanto a *Ant* trabalha de forma totalmente descentralizada. O PS se aproxima mais do trabalho de FARNELLI; ROGERS; JENNINGS (2014), pois ambos trabalham com ciclos e a definição de papéis/estados independe do disparo do evento. Além disso, tanto as abordagens heurísticas, *Ant-based* e PS, como a abordagem baseada no MAXSUM, não garantem uma solução ótima para o problema. O MAXSUM tem sua qualidade comprometida quando é utilizado em aplicações com grafos de restrição com ciclos - como é o caso das RSSFs.

Tabela 1 – Trabalhos Relacionados.

Trabalho/Citação	Nível	Técnica	Característica
LEACH (HEINZELMAN; CHANDRAKASAN; BALAKRISHNAN, 2000)	rede	agrupamento	escolha aleatória dos líderes
HEED (YOUNIS et al., 2004)	rede	agrupamento	escolha dos líderes pela energia residual e etc.
KARABOGA; OKDEM; OZTURK (2012)	rede	agrupamento bioinspirado	escolha dos líderes centralizada na estação-base
SINHA; JOSE (2017)	rede	agrupamento bioinspirado	escolha descentralizada dos líderes
DSA-DMRP MISBAHUDDIN et al. (2018)	rede	agrupamento	escolha de líderes com fator de prioridade
PEGASIS (LINDSEY; RAGHAVENDRA, 2002)	rede	estratégia gulosa	distância constante de comunicação entre nodos
LIU; LI; WANG (2016)	rede	bioinspirada	caminho definido pela concentração de feromônio
PS (CALISKANELLI, 2014)	aplicação	bioinspirada	alternância entre nodos ativos/inativos
<i>Ant-based</i> (FERREIRA; BRISOLARA; INDRUSIAK, 2015)	aplicação	bioinspirada	divisão de eventos baseada na carga anterior de trabalho
MAXSUM (FARINELLI; ROGERS; JENNINGS, 2014)	aplicação	bioinspirada (DCOP)	ativação e desativação da capacidade de processamento
DOOTA (YU; HUANG; GARCIA-ORTIZ, 2018)	aplicação	agrupamento	alocação de tarefas de forma independente e <i>on-line</i>
DTAS (JIN et al., 2013)	aplicação	bioinspirada (AG)	processo em duas etapas para melhorar o desempenho
TASIM (WANG et al., 2015)	aplicação	baseada em escore	alocação de tarefas em sub-tarefas para RSSFs proativas
MBPSO (YANG et al., 2014)	aplicação	bioinspirada (PSO)	armazenamento de caminhos anteriores na partícula
COP/COP-Feedback	aplicação	COP	abordagem semi-distribuída para RSSFs reativas

4 ABORDAGEM PROPOSTA

Neste capítulo apresenta a abordagem proposta, que foca no problema de balanceamento de carga no nível de aplicação em RSSFs reativas. A solução proposta é semi-distribuída baseada em um problema COP e impede que mais de um nodo processe um mesmo evento, empregando um número otimizado de mensagens. Na Seção 4.1 será apresentada a ideia geral da abordagem proposta, exemplificando seu funcionamento e posteriormente definindo o mapeamento do problema para um COP. O algoritmo proposto, bem como a implementação da abordagem são discutidos na Seção 4.2 e a técnica de *feedback* empregada para otimização da comunicação é apresentada da Seção 4.3.

4.1 Visão Geral e Definição do Problema

Devido a característica distribuída e a dinâmica da RSSF, torna-se inviável um processo de balanceamento de carga totalmente centralizado, onde um único dispositivo precisaria conhecer o estado atual de todos os nodos da rede. Já uma solução distribuída, não exige que o conhecimento seja centralizado, mas pode requerer muita comunicação dependendo do número de nodos envolvidos no processo. Além disso, soluções DCOPs empregam algoritmos mais complexos e custosos. Como a redundância é limitada mesmo em RSSFs densas, uma solução semi-distribuída provê a comunicação necessária para considerar o estado dos nodos, mas limitando-se aos nodos envolvidos ou que compartilham uma mesma área de cobertura. Desta forma, a abordagem proposta surge como uma alternativa viável capaz de contornar os principais impeditivos encontrados em soluções totalmente distribuídas e totalmente centralizadas.

A abordagem visa definir qual o nodo que processará o evento, dentre os nodos que também perceberam o mesmo evento, mantendo apenas o nodo escolhido com a capacidade de processamento. Desta forma, o processamento redundante é evitado e assim o consumo de energia dos nodos pode ser economizado. Para isso, a abordagem emprega um esquema de líder. O líder será responsável por definir quais

os nodos participantes do processo, e dentre eles, escolher um deles considerando a carga de trabalho que os nodos tiveram no passado. A ideia é que o nodo que trabalhou menos, venha a ser o escolhido e assim evitando-se a sobrecarga de nodos e aumentando o tempo de vida da rede como todo.

A Figura 10 ilustra o funcionamento da abordagem proposta em uma RSSF com 9 nodos dispostos em grade e com uma sobreposição de coberturas de no máximo 4 nodos (Figura 10a). Neste exemplo, a deliberação ocorre em um único nodo (Nodo 1), que foi definido como líder (nodo circulado na Figura 10b). Em primeiro lugar, o líder precisa saber qual dos seus vizinhos também detectou o mesmo evento, que no exemplo, são os nodos 2, 4 e 5. Estes nodos, juntamente com o nodo responsável pela deliberação (Nodo 1), participarão do processo de balanceamento de carga (Figura 10b e Figura 10c). Cada participante, vizinho do líder, que também detectou este mesmo evento envia uma mensagem de feedback ao líder, contendo o número de eventos que processou no passado (Figura 10d). Dessa forma, a abordagem proposta tem um funcionamento semi-distribuído, pois mescla uma etapa de deliberação centralizada com duas etapas distribuídas: a primeira para definição dos participantes - por meio de mensagens - e a última responsável por encaminhar o esquema deliberado para os participantes do processo de balanceamento de carga e garantir que cada nodo respeite, individualmente, o que foi deliberado (habilitando ou desabilitando sua capacidade de processamento). No exemplo, o processo de deliberação realizado internamente no Nodo 1 (Figura 10e) definiu o Nodo 5 (circulado na Figura 10f) como responsável pelo processamento do evento detectado dentro da área de cobertura comum aos nodos 1, 2, 4 e 5. Tanto o Nodo 1 (líder), como os Nodos 2 e 4 devem se esquivar dessa responsabilidade, desativando temporariamente sua capacidade de processamento e preservando suas cargas de bateria (estes nodos estão marcados com um X na Figura 10f).

Mapeando o problema alvo para um COP conforme foi apresentado na subseção 2.2.1, cada nodo é transformado em um agente responsável por uma variável do conjunto V - que estão sob um mesmo domínio $D = \{0, 1\}$. Em D , 1 representa a ativação e 0 o estado de hibernação dos agentes/nodos. A função de custo (Equação 2) empregada na abordagem proposta é a soma-produto dos possíveis valores das variáveis (v_i) dos agentes participantes do balanceamento de carga (n), juntamente com o número de eventos processados (j_i) por cada nodo antes do evento atual ser disparado.

$$F = \sum_{i=1}^n v_i * j_i \quad (2)$$

Neste caso, a função de custo para o exemplo da Figura 10 considera apenas as variáveis v_1 , v_2 , v_4 e v_5 dos agentes a_1 , a_2 , a_4 e a_5 que representam, respec-

tivamente, os nodos 1, 2, 4 e 5. Além disso, a função seria expressa por $F = v_1 * j_1 + v_2 * j_2 + v_4 * j_4 + v_5 * j_5$. Nesta função de custo, j_1, j_2, j_4 e j_5 representam o número de eventos processados por cada nodo/agente antes de capturar o evento atual. O objetivo é encontrar o menor valor diferente de zero e, conseqüentemente, definir os melhores valores possíveis para as variáveis dos agentes/nodos envolvidos, considerando o domínio D (0 e 1). Assim, considerando que os valores j_1, j_2, j_4 e j_5 são respectivamente 8, 7, 7 e 6. Apenas para a variável v_5 , correspondente ao Nodo 5, será atribuído o valor 1 e as demais ficarão em 0, resultando no menor valor para a função de custo do exemplo.

No problema de balanceamento de carga, as funções de custo devem encontrar os menores valores e definir o status de ativação ou hibernação de cada nodo participante do processo - mas garantindo que o evento seja processado por algum nodo. Na abordagem proposta de balanceamento de carga, o nodo escolhido para o processamento tem a menor quantidade de eventos processados anteriores à captura do último evento disparado, ou seja, em comparação com os demais vizinhos que tenham também detectado este evento, o nodo escolhido se envolveu menos em processamentos de eventos no passado. Se o custo de processamento for maior que o de envio de mensagens, provavelmente, o nodo escolhido terá o maior nível de bateria, preservando assim os nodos com menores níveis de bateria.

4.2 Implementação

O funcionamento da abordagem é descrito pelo *Algoritmo 1 - Abordagem Proposta*. Quando um evento ocorre na RSSF, um ou mais nodos podem detectar este mesmo evento. Assim, um nodo é definido aleatoriamente como o líder do balanceamento de carga (linha 2) e que, internamente, será o responsável pelo processo de deliberação. A eleição do líder ocorre mediante trocas de mensagens e garante a presença de um único nodo-líder no processo de balanceamento de carga. O líder deve enviar uma mensagem para os demais nodos da vizinhança com a intenção de confirmar a participação destes vizinhos no balanceamento de carga, ou seja, elencar os vizinhos que também detectaram o evento (linhas 3 a 4). Somente os vizinhos que também detectaram este mesmo evento devem notificar o líder com uma mensagem de feedback (linhas 5 a 13). Com as informações dos vizinhos (linha 15), o líder deve definir os melhores valores possíveis das variáveis de cada agente de forma semelhante ao que acontece no algoritmo *Branch-and-Bound* (linha 16-17). O nodo escolhido será definido pela função de deliberação do líder na linha 17. Se o próprio líder for escolhido para tratar o evento (linha 18 a 27), ele se ativará e processará o evento (linhas 21 e 23), caso contrário, se desativará (linha 26). Após definir a solução, o líder notifica os seus vizinhos (linha 28) para que, individualmente, cada vizinho mude o valor de sua

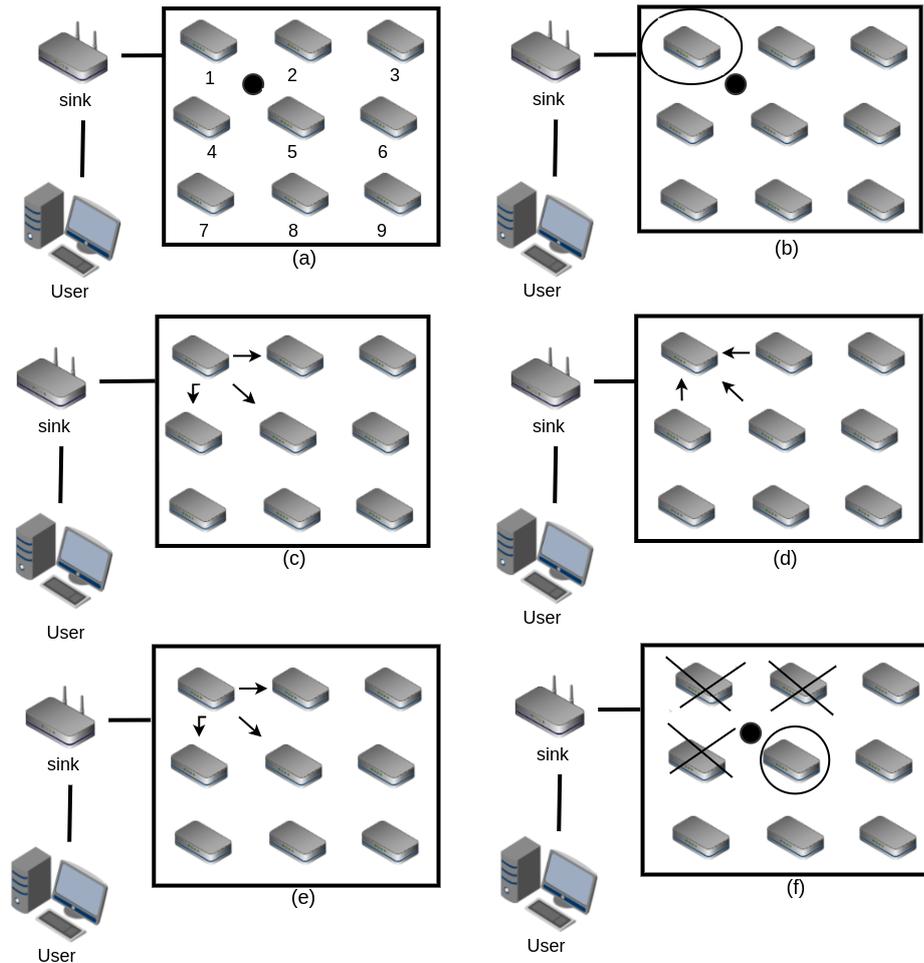


Figura 10 – Abordagem COP: (a) disparo do evento; (b) líder do esquema de balanceamento de carga; (c) e (d) vizinhos trocando mensagens com o líder para confirmar ou não o sensoriamento do mesmo evento; (e) o líder encaminhando para os vizinhos o esquema de balanceamento escolhido; e (f) somente o nodo responsável pelo processamento permanecerá com sua capacidade de processamento ativa.

variável e, conseqüentemente, mude seu estado de ativação ou hibernação (linhas 29 a 42). Quando o vizinho recebe uma mensagem que coincide com seu nome, este ativará sua capacidade de processamento e processará o evento (linhas 33 a 37), caso contrário este desativa sua capacidade de processamento (linha 40).

O algoritmo proposto para a abordagem de balanceamento de carga em nível de aplicação foi então implementado no simulador EBORACUM. O simulador possui uma hierarquia de classes que foi estendida para esta implementação. Uma nova classe nomeada *COPAgent* foi criada, a qual implementa a interface *BasicAgent* do EBORACUM. Esta nova classe possui o método *deliberate()*, responsável pela lógica de deliberação e invocado pelo líder do balanceamento de carga. Além desta nova classe que representa o nodo-agente, foi criada uma outra classe denominada *COPControlledWSNNode*, derivada da classe abstrata *ControlledWSNNode*. Devido ao relacionamento de herança e a reescrita do método *receiveMessage()* (poliformismo), a classe

Algorithm 1 - Abordagem Proposta

```

1: //leader
2: leader ← randomNodeEventDetected();
3: tempEvent ← leader.eventSensed();
4: leader.sendMessageToNeighbours(tempEvent);
5: //neighbors
6: for all n from leader.getNeighboursOfNode() do
7:   tempEvent ← n.receiveMessage();
8:   temp ← n.eventSensed();
9:   if temp = tempEvent then
10:    //neighbor's feedback
11:    n.sendMessageToNeighbours(temp);
12:   end if
13: end for
14: //leader
15: leader.receiveAllMessage();
16: //the load balancing function
17: chosenNode ← leader.deliberation( $\sum_{i=1}^n v_i * j_i$ );
18: // if the leader for chosenNode
19: if chosenNode.getName() = leader.getName() then
20:   //activation state
21:   leader.setStatus(on);
22:   //processing
23:   leader.sensorNodeAction();
24: else
25:   //hibernation state
26:   leader.setStatus(off);
27: end if
28: leader.sendMessageToNeighbours(chosenNode);
29: //neighbors
30: for all n from leader.neighboursOfNode() do
31:   chosenNode ← n.getReceiveMessage();
32:   //if neighbor is the chosen node
33:   if chosenNode.getName() = n.getName() then
34:     //activation state
35:     n.setStatus(on);
36:     //processing
37:     n.sensorNodeAction();
38:   else
39:     //hibernation state
40:     n.setStatus(off);
41:   end if
42: end for

```

ControlledWSNNode tornou-se responsável pela lógica das mensagens dos nodos-agentes. As classes *COPAgent* e *COPControlledWSNNode* representam o comportamento individual de cada nodo como, por exemplo, se detectou um evento, se rece-

beu ou enviou alguma mensagem, se decidiu ou não processar o evento detectado. Cada instância de *ControlledWSNNode* representa um nodo que é controlado por um agente (instância de *COPAgent*). Em outras palavras, toda instância de *COPControlledWSNNode* está associada a uma instância de *COPAgent*. Além destas classes, há uma classe de simulação (*COPSimulation*), derivada de outra classe abstrata (*BenchmarksGenerator*), responsável por definir os parâmetros dos cenários de teste. A Figura 11 ilustra o diagrama UML das principais classes desenvolvidas na implementação da abordagem COP no simulador EBORACUM que já havia, anteriormente, incorporado as demais técnicas Ant, PS e Greedy usadas na análise comparativa (Capítulo 5).

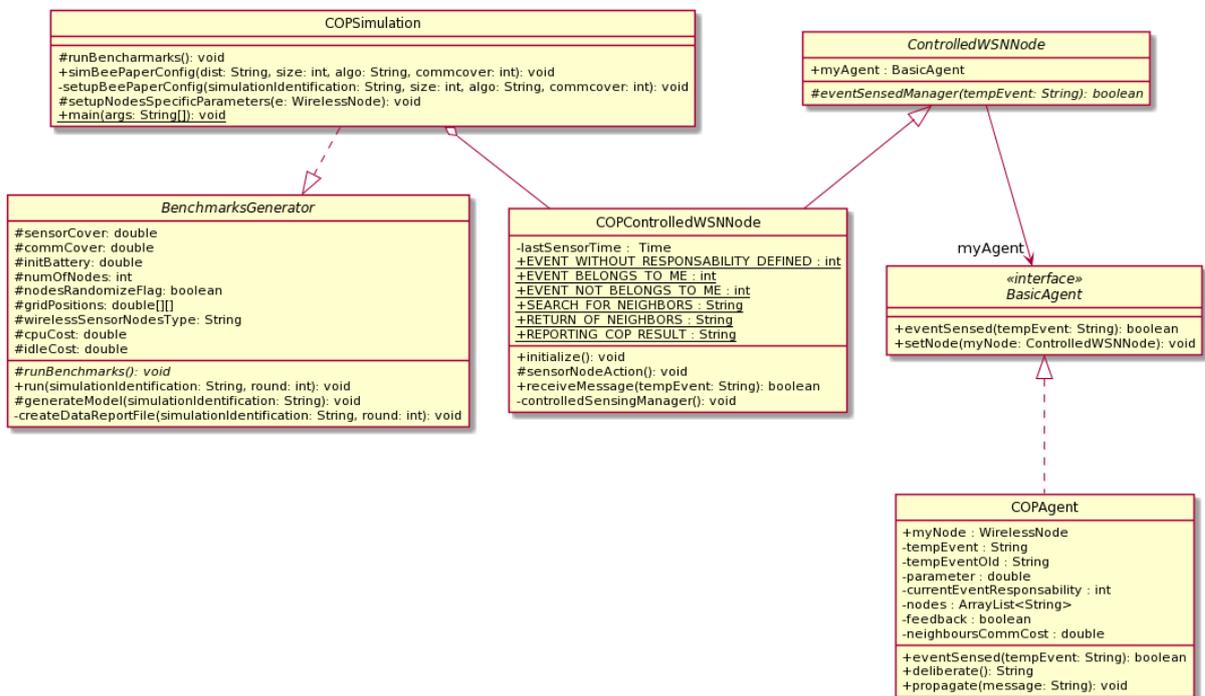


Figura 11 – Diagrama de Classes - Abordagem COP.

A abordagem começa pelo método de detecção de eventos (*eventSensed()*) dos objetos da classe *COPAgent*, passa para o método *receiveMessage()* de *COPControlledWSNNode* e retorna para o método *deliberate()* da instância de *COPAgent* que representa o líder. O líder define o esquema de balanceamento de carga e encaminha este resultado para os demais vizinhos. Em seguida, cada nodo-participante recebe o resultado da deliberação através do *receiveMessage()* e invoca, internamente, o método *sensorNodeAction()* de sua instância de *COPControlledWSNNode*. Assim, dentro do método *sensorNodeAction()*, cada nodo deve respeitar o que foi definido pela deliberação, ou seja, o nodo escolhido deve processar o evento detectado - ativamente previamente sua capacidade de processamento - e os demais vizinhos devem se esquivar desta responsabilidade.

A classe *COPSimulation* foi usada para definir os parâmetros de configuração da

rede e dos nodos adotados pelas simulações durante os experimentos como, por exemplo, a quantidade de nodos da rede, a disposição dos nodos e os valores dos raios de sensoriamento e de comunicação de cada nodo da rede. Os resultados da simulação podem ser exportados em um relatório (.CSV), que armazena os dados de cada execução. Através deste relatório é possível visualizar a quantidade de eventos processados por cada nodo por dia, o número de mensagens enviadas e recebidas e etc.

4.3 Técnica de Feedback

Visando reduzir o número de mensagens empregadas pela abordagem, uma técnica de controle de propagação de mensagens foi então incorporada à abordagem proposta. O objetivo é impedir o envio de mensagens duplicadas ou desnecessárias do líder para os participantes e vice-versa. Assim, sempre que um novo evento é detectado por algum nodo, este verifica, internamente, se a mensagem correspondente àquele evento já foi enviada. Caso contrário, envia a mensagem. Se a mensagem já foi enviada é porque o processo de balanceamento de carga para este evento já foi inicializado por uma mensagem anterior e o processo de deliberação já tem um líder. Desta forma, cada vizinho encaminha uma única mensagem de *feedback*. Apesar do envio ocorrer por *broadcast*, a mensagem será endereçada, exclusivamente, ao líder e portanto será descartada pelos demais nodos. Visando também otimizar a comunicação, o líder encaminha uma única mensagem aos vizinhos em busca de participantes para o processo de deliberação (no início do balanceamento de carga) e uma outra mensagem ao final do processo com o objetivo de propagar o resultado da deliberação para os demais vizinhos (mensagens também enviadas por *broadcast*).

Antes da incorporação desta otimização, todo nodo que detectava um evento submetia, obrigatoriamente, uma nova mensagem para os vizinhos em busca de nodos que também tivessem detectado o mesmo evento. Entretanto, basta ao líder o envio desta mensagem de propagação, pois somente o líder deve elencar quais vizinhos que, juntamente com ele, farão parte do processo de balanceamento de carga. Assim, toda vez que um novo evento é detectado por qualquer nodo este verifica, internamente, se já recebeu ou não esta mesma mensagem de notificação de outro nodo (líder). Caso já tenha recebido, a verificação impede que esta mensagem seja novamente propagada - deixando explícito que o processo de balanceamento de carga para este evento já foi previamente iniciado pelo envio da mensagem anterior e o processo de deliberação já tem um líder. Por outro lado, os vizinhos continuam retornando ao líder mensagens que confirmam ou não sua participação no processo. Deste modo, há em cada nodo outro controle interno que restringe a uma única mensagem de *feedback*. Assim, o líder receberá uma mensagem por vizinho para cada

evento detectado.

A Figura 12 ilustra as mensagens trocadas pelos nodos participantes do processo de balanceamento de carga antes e depois da incorporação do controle de propagação no líder. Na Figura 12a todo nodo que detecta um novo evento envia uma nova mensagem de notificação para os seus vizinhos. Assim, o líder além de enviar mensagens para os vizinhos também recebia dos vizinhos o mesmo tipo de mensagem. Estas mensagens duplicadas eram descartadas, porém o envio destas acarretava uma descarga de bateria dos nodos. Isto provocava um reenvio massivo de mensagens entre os nodos com o simples objetivo de notificar algo que possivelmente a maioria dos vizinhos já havia observado, ou seja, a mensagem de detecção deste novo evento para estes nodos era algo redundante. Com o controle de propagação de mensagens integrado, o foco está na definição dos papéis de cada nodo durante o processo de deliberação, pois quando o primeiro nodo envia este tipo mensagem ele, automaticamente, torna-se líder. Como ilustrado na Figura 12b somente o líder enviará as notificações de detecção de um novo evento para os vizinhos.

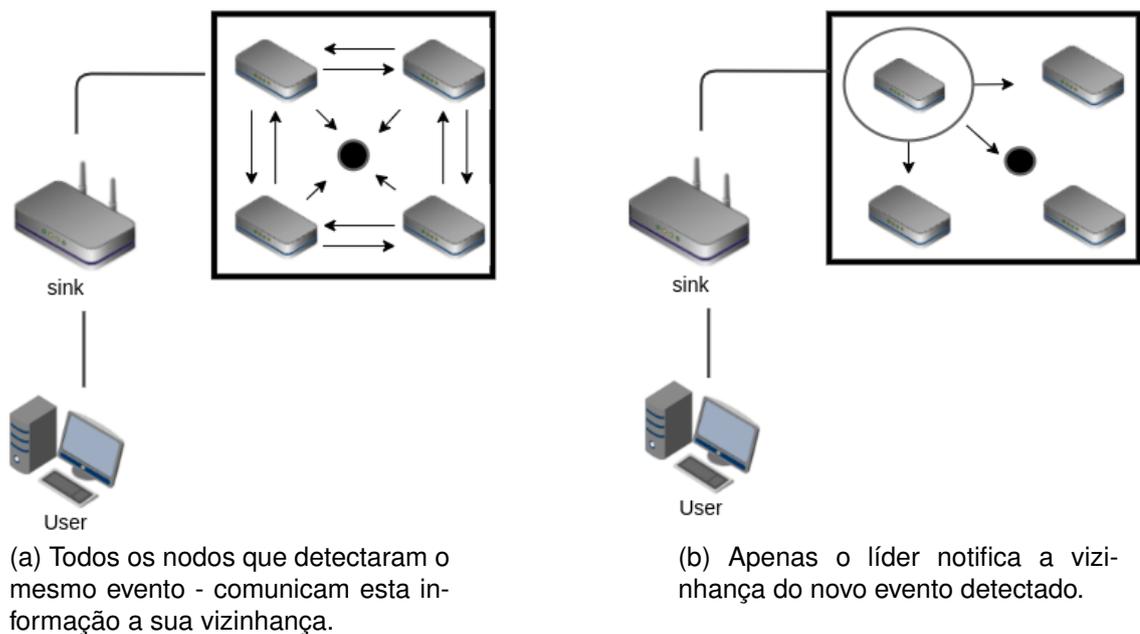


Figura 12 – Abordagem COP com controle de propagação de mensagens no líder.

Para a mensagem de confirmação dos vizinhos há também uma verificação com o objetivo de evitar que mais de uma vez um mesmo vizinho retorne ao líder sua mensagem de confirmação de detecção e, conseqüentemente, sua participação no processo. Neste sentido, o foco é evitar que vizinhos que já propagaram sua resposta venham encaminhá-la novamente. Sem este ajuste, toda nova mensagem recebida pelos vizinhos era considerada uma nova mensagem vinda do líder e era respondida novamente (o que acontecia inúmeras vezes). O que o controle faz é testar se o vizinho já enviou seu feedback ao líder em algum outro momento anterior. Caso a

mensagem já tenha sido enviada anteriormente, este controle impede um novo envio. Estes procedimentos minimizam o número de mensagens trocadas pelos nodos e incrementam o tempo de vida útil da rede.

5 EXPERIMENTOS

Este capítulo apresenta e discute os experimentos conduzidos para avaliar a proposta de balanceamento de carga para RSSFs reativas em nível de aplicação. Além disso, o capítulo discute seus resultados, sempre comparando com as abordagens concorrentes. A Seção 5.1 aborda a metodologia utilizada durante os experimentos; a Seção 5.2 contempla os resultados preliminares realizados em cenários com nodos em grade comparando a abordagem COP (ainda sem a técnica de feedback) com as demais soluções já previamente implementadas no EBORACUM e, a Seção 5.3 trata dos experimentos realizados com a abordagem COP-Feedback (com a técnica de feedback), tanto nos mesmos cenários com nodos em grade como em cenários com nodos dispostos aleatoriamente.

5.1 Metodologia

Como ambiente de simulação foi empregado o EBORACUM¹ (BRISOLARA et al., 2019; BRISOLARA; FERREIRA; INDRUSIAK, 2015; BRISOLARA et al., 2016), que é uma infraestrutura para modelagem e simulação de RSSFs. O EBORACUM foi escolhido por prover um framework orientado a objetos, e portando extensível, facilitando assim a implementação de novas abordagens de balanceamento de carga. Além disso, seu emprego possibilita a comparação dos resultados com as abordagens heurísticas de balanceamento de carga já incorporadas, anteriormente, neste simulador.

Os experimentos avaliam a abordagem baseada em COP, nas suas duas versões: preliminar (sem controle de propagação de mensagens, chamada COP) e otimizada (com controle, chamada COP-Feedback) em diferentes cenários de RSSFs. Seus resultados são comparados com as abordagens *Ant-based*, PS e com uma técnica Greedy centralizada. A Greedy é usada como *upper-bound* na análise comparativa, por apresentar resultados ótimos. Entretanto, como eventos futuros não são conhecidos durante as simulações no EBORACUM, considera-se que a solução encontrada

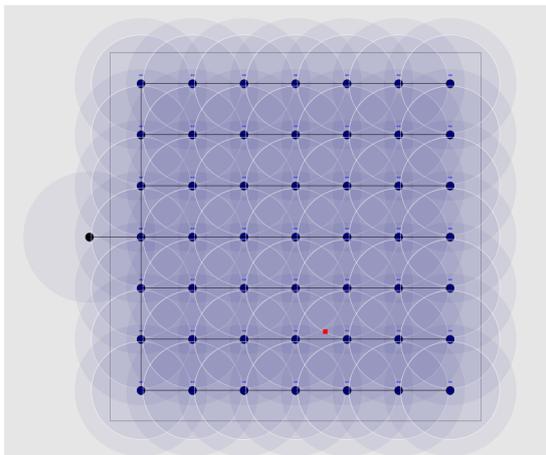
¹<http://sourceforge.net/projects/eboracum/>.

por *Greedy* é ótima local e não global. Além disso, por ser uma abordagem totalmente centralizada, *Greedy* não é viável em um ambiente real, pois considera que um nodo tenha acesso ao estado da bateria dos demais nodos, como se houvesse uma memória compartilhada entre eles, o que só é possível em simulação.

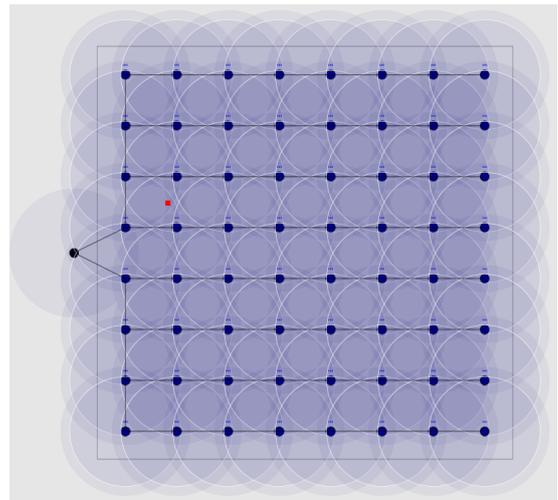
Nas simulações, foram avaliados cenários de RSSFs com nodos implantados em malha (ou grade) e RSSF com nodos dispostos aleatoriamente. Ao longo dos experimentos, variou-se também a densidade da rede considerando 49, 64, 81 e 100 nodos. Além dos nodos sensores, uma estação-base localizada ao lado foi empregada em todos os cenários experimentados. Os nodos sensores foram implantados em uma área de interesse quadrada de 810km^2 ($900\text{m} \times 900\text{m}$). As Figuras 13a, 13b, 13c e 13d ilustram os cenários em grade com 49, 64, 81 e 100, respectivamente. Para os experimentos com RSSF com nodos dispostos aleatoriamente foram empregados quatro novos cenários, variando a quantidade de nodos considerando também 49, 64, 81 e 100 nodos (Figura 14).

Nos cenários em grade, os nodos estão conectados a um de seus quatro vizinhos (o mais próximo à estação base) ou diretamente à estação base. O esquema de implantação apresenta cobertura de detecção redundante igual a 4 nodos, no máximo, na configuração de 49 nodos e a redundância aumenta nas RSSFs mais densas. Já nos cenários aleatórios, a redundância varia muito e inclusive podem haver regiões sem cobertura de nodos sensores ou ainda nodos sensores desconectados.

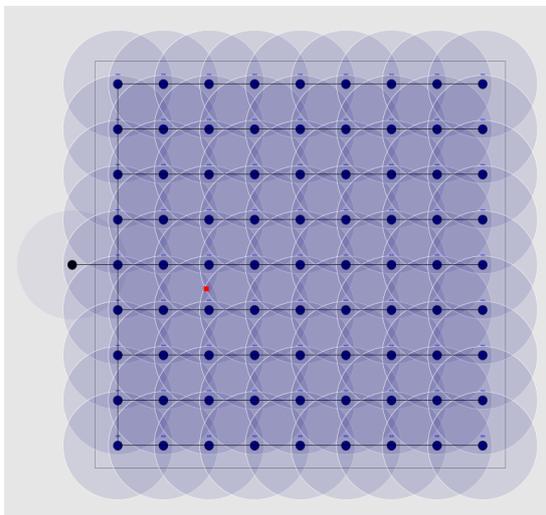
Em todos os experimentos foi considerado o raio de cobertura de 120 e o raio de comunicação de, respectivamente, 160, 140, 120 e 120. Para definição dos custos energéticos empregados na descarga e a carga total da bateria foram considerados os consumos apresentados na Tabela 2. A capacidade da bateria é de 5400000 miliampere por segundo (mAs), mas inicia-se a simulação com os nodos com 50% de bateria pois considera-se que com 50% de carga os nodos param de operar. Cada evento equivale a 14 tarefas e as mensagens geradas por estes eventos têm 3 bytes de tamanho. A descarga na transmissão é de 0,0006 mAs por byte transmitido e no ciclo ocioso 0,3 mAs. No ciclo ocioso, considera-se que a antena está ativada e portanto o nodo está recebendo mensagens. Considerou-se que as mensagens trocadas pelas abordagens de balanceamento de carga Ant-based, PS e COP equivale a 1 byte, representado uma descarga de 0,0006 mAs. A descarga do nodo nos ciclos de processamento é de 3,57 mAs. Como o evento está associado a 14 tarefas, cada uma delas associada a um ciclo de CPU, o custo de $cpu = 50$ foi considerado para a aplicação do estudo de caso. O tamanho da mensagem transmitida pós processamento do evento e o processamento requerido pelo evento, foram definidos considerando uma RSSF equipada com sensores acústicos capazes de capturar sons gerados por animais em uma determinada área e o processamento é usado para identificação do som detectado, similar ao estudo de caso empregado em (CALISKANELLI et al., 2013b).



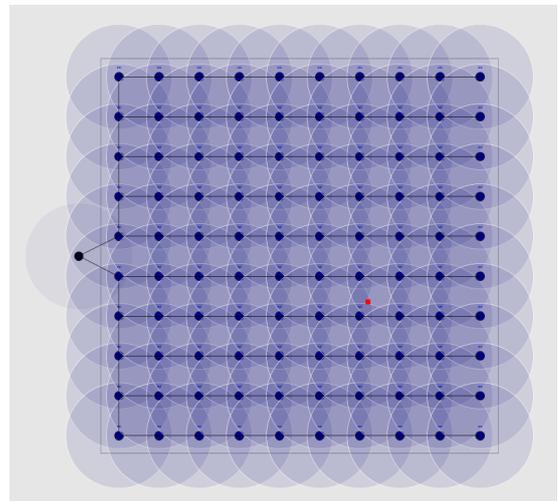
(a) RSSF - 49 nodos.



(b) RSSF - 64 nodos.



(c) RSSF - 81 nodos.



(d) RSSF - 100 nodos.

Figura 13 – Cenários avaliados de RSSF em grid.

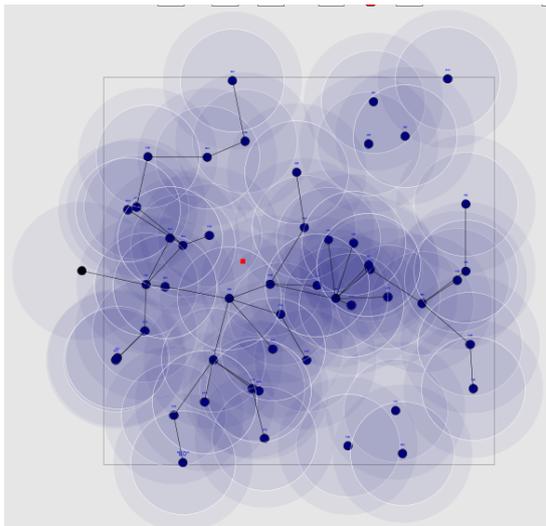
Os custos de energia e a capacidade de bateria baseiam-se em um nodo-sensor modelo, tendo como referência os valores informados no *datasheet* do equipamento IRIS Mote².

Durante os experimentos, eventos são gerados para emular a carga de trabalho da RSSF. Esta carga, combinada com os custos de energia de cada ciclo (processamento, envio de mensagem e ocioso), é usada para simular a descarga da bateria dos nodos. Os eventos aparecem no espaço seguindo uma distribuição Uniforme e a frequência de eventos é determinada por uma distribuição *Poisson* com um intervalo de função entre eventos de [1,120] segundos. Com o uso de uma distribuição uniforme, os eventos tem a mesma probabilidade de acontecer em qualquer lugar da área de interesse. Além disso, o número total de eventos gerado a cada simulação

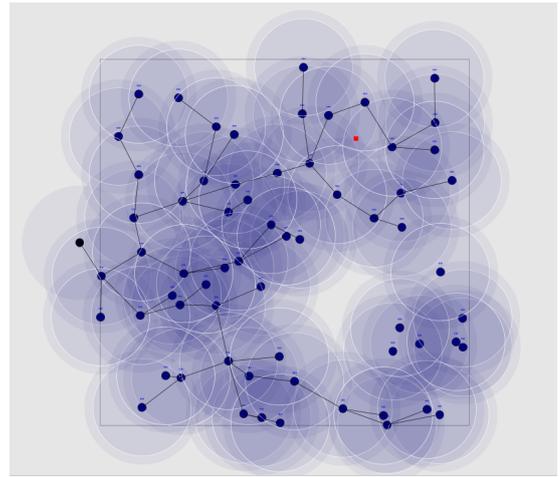
²http://www.nr2.ufpr.br/~adc/documentos/iris_datasheet.pdf

não é fixo ou conhecido a priori. Este número varia pelo emprego da distribuição de Poisson e dependerá do tempo de simulação, ou seja, do tempo de vida da rede.

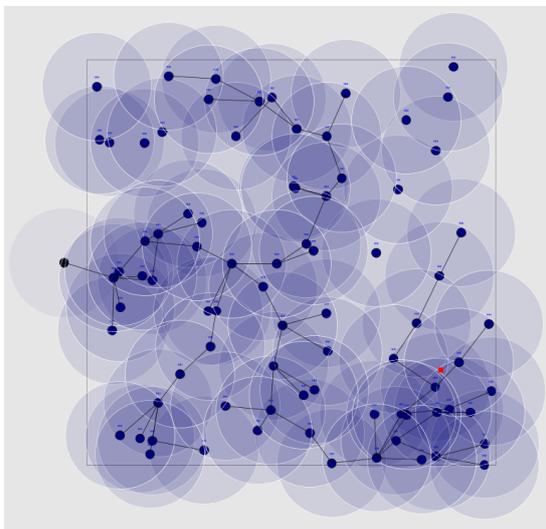
Utilizou-se a média de 30 execuções para cada cenário diferente avaliado. O *teste-t de Student* foi adotado para analisar a significância estatística entre a diferença do resultado médio obtido pela abordagem proposta e por cada uma das abordagens comparadas para cada cenário.



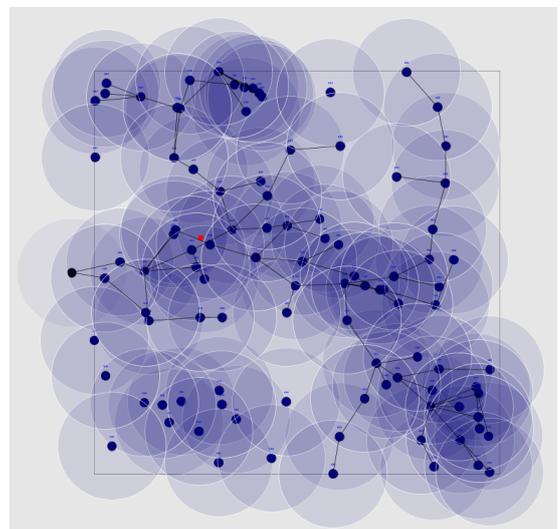
(a) RSSF - 49 nodos.



(b) RSSF - 64 nodos.



(c) RSSF - 81 nodos.



(d) RSSF - 100 nodos.

Figura 14 – Cenários avaliados de RSSF em modo randômico.

5.2 Resultados Preliminares

Primeiramente, foi implementada e avaliada a abordagem COP semi-distribuída sem controle de propagação de mensagens. Esta seção discute os resultados que esta abordagem, nomeada COP, apresentou em comparação às abordagens heurís-

Tabela 2 – Parâmetros de Energia.

Parâmetros de Energia	Valor
Capacidade da Bateria	5400000 mAs
Taxa de descarga ociosa	0,3 mAs
Taxa de descarga por tarefa	3,57 mAs
Taxa de descarga (1 byte)	0,0006 mAs

ticas (PS e *Ant-based*) e a Greedy. A comparação considerou em um primeiro momento o número médio de eventos processados por dia para cada abordagem. Os gráficos das Figuras 15, 16, 17 e 18 representam os resultados obtidos pelas diferentes abordagens nas quatro densidades experimentadas, com 49, 64, 81 e 100 nodos, respectivamente. Nos gráficos é possível observar que a abordagem COP (ainda sem controle de propagação) foi capaz de atingir o mesmo número de eventos detectados por dia que Greedy (limite superior). Em outras palavras, todos os eventos são detectados com sucesso pela rede enquanto ela está viva, o melhor resultado possível sob essa perspectiva.

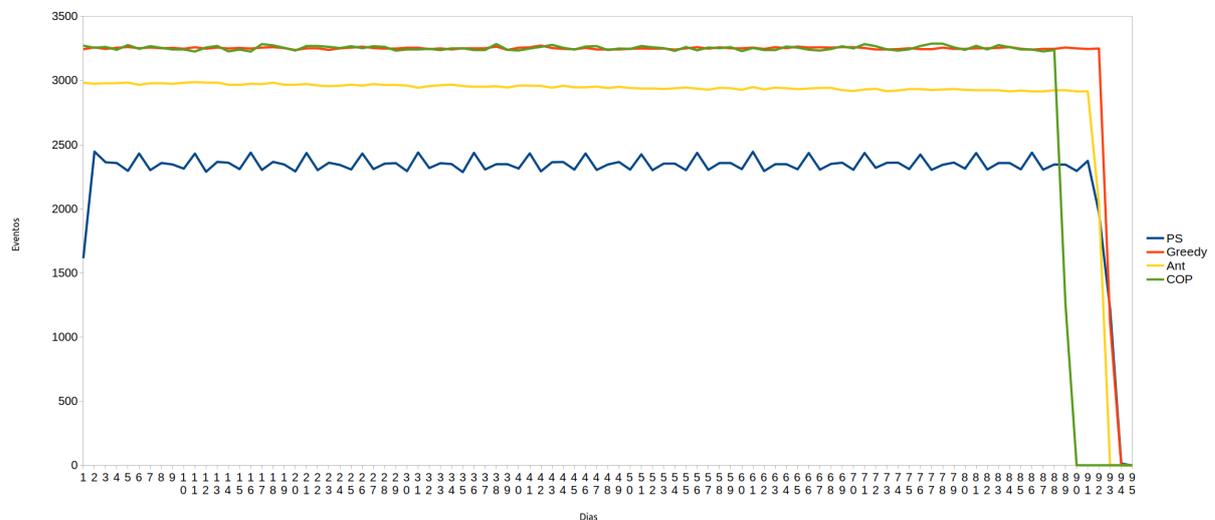


Figura 15 – Resultados Preliminares: Média dos Eventos Processados - 49 nodos.

Também é possível observar nos gráficos das Figuras 15, 16, 17 e 18 que a abordagem COP leva a rede a ter uma vida útil mais curta em todos os cenários. O COP exige mais comunicação do que os outros, por isso consome mais bateria a cada dia. No entanto, em relação ao número total de eventos detectados pela rede, conforme resultados apresentados na Tabela 3, o COP tem um desempenho melhor do que a abordagem estado-da-arte para densidades mais baixas, 49 e 64 nodos. O COP direciona as redes para detectar 6% mais eventos do que o *Ant-based* para o cenário de 49 nodos e 4% para o cenário de 64 nodos. *Ant-based* é melhor para 81 nodos,

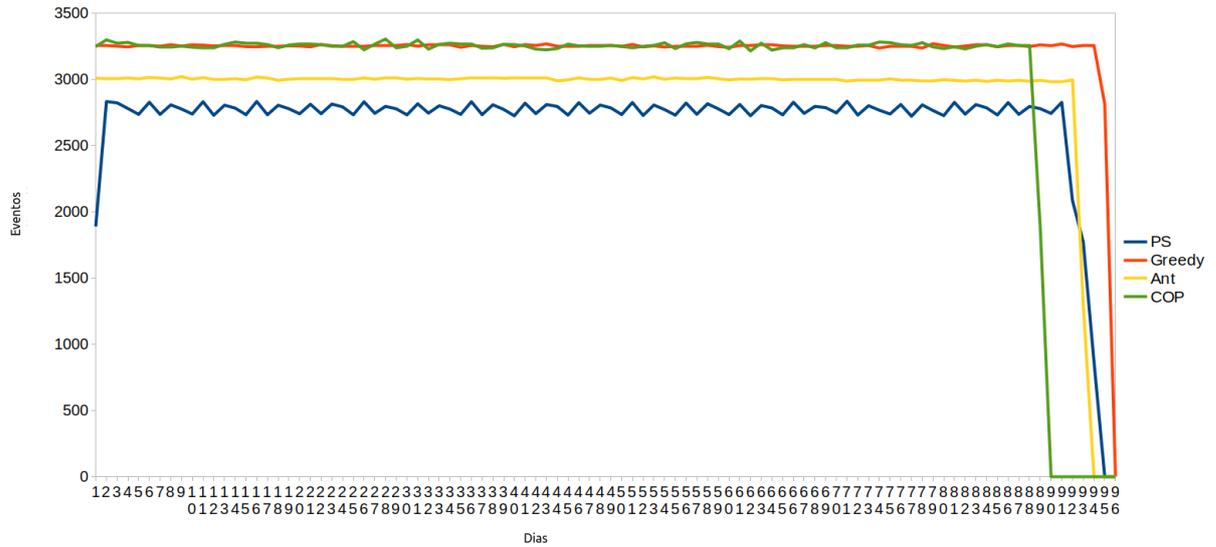


Figura 16 – Resultados Preliminares: Média dos Eventos Processados - 64 nodos.

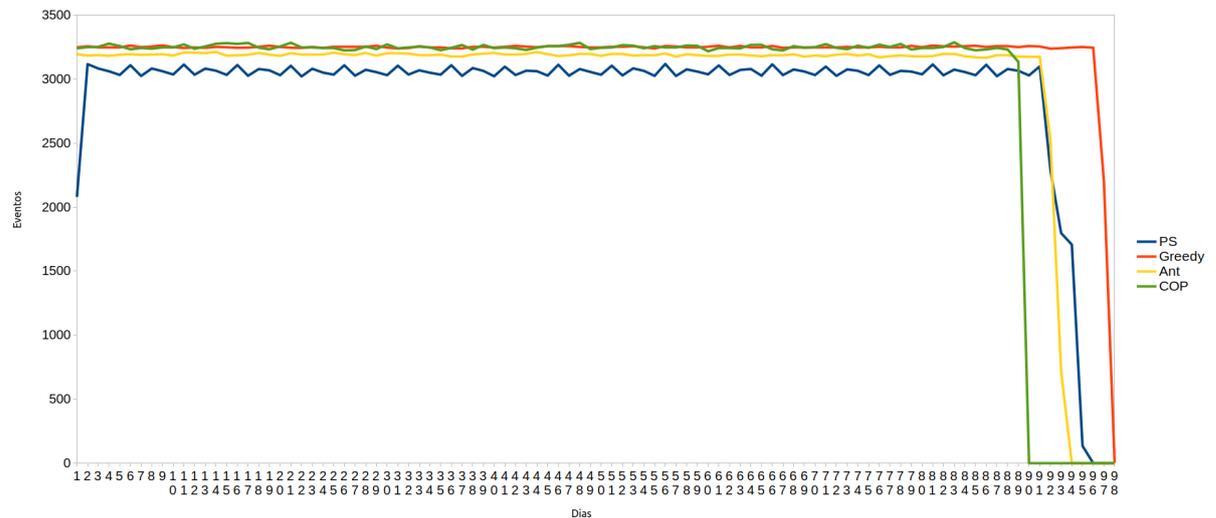


Figura 17 – Resultados Preliminares: Média dos Eventos Processados - 81 nodos.

levando a rede a detectar 1,5% mais eventos que o COP, e o PS é melhor para 100 nodos, levando a rede a detectar 1% mais eventos que o COP.

Tabela 3 – Resultados Preliminares - Média e Desvio padrão dos Eventos Processados.

	49 nodos		64 nodos		81 nodos		100 nodos	
Abordagem	Eventos	σ	Eventos	σ	Eventos	σ	Eventos	σ
Greedy	300317,4	222,2	308538,3	45,5	314375,8	106,9	318704,3	322,9
COP	287439,9	203,5	288102,6	143,7	289280,9	16,5	292289,3	53,1
Ant	270477,4	96,9	277377,9	177,1	293565,1	265,1	292815,4	330,3
PS	216476,1	282,3	256462,8	250,5	283683,1	373,5	295401,8	659,3

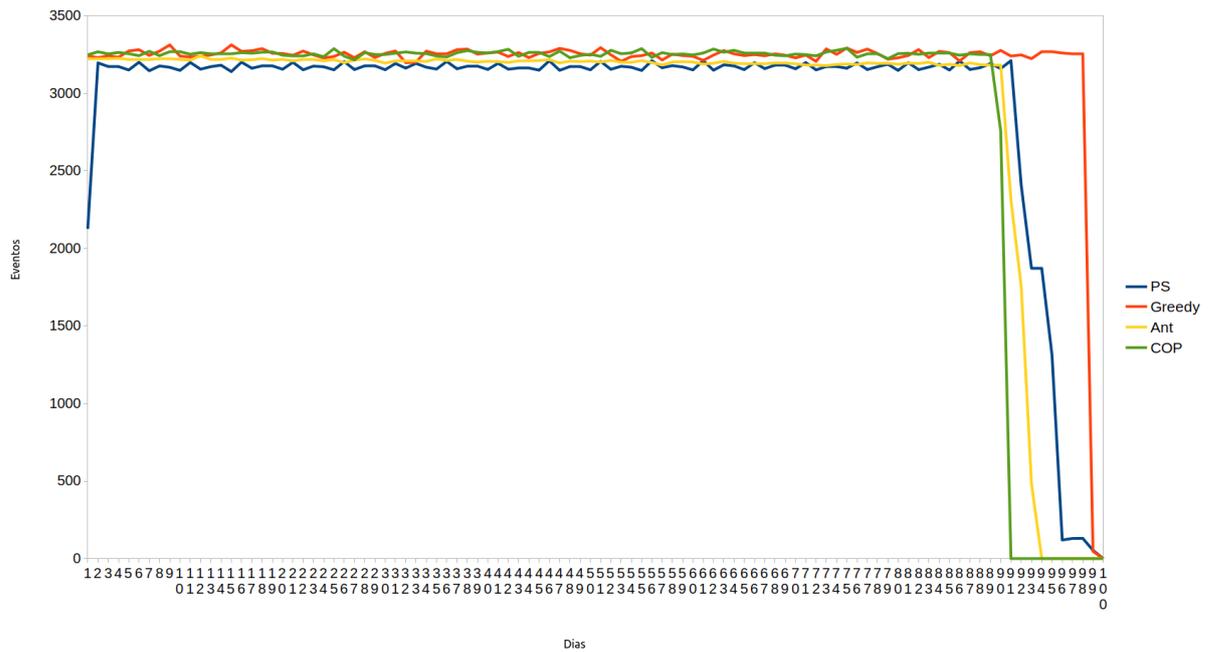


Figura 18 – Resultados Preliminares: Média dos Eventos Processados - 100 nodos.

A Figura 19 ilustra o número total de eventos detectados por cada abordagem para cada densidade. Pode-se observar que o PS e *Ant-based* têm seu desempenho dependente da densidade da rede. O PS tem o incremento mais notável no desempenho à medida que a densidade aumenta. Apesar de ser menos sensível à variação de densidades, *Ant-based* também apresenta uma degradação do desempenho em redes de baixa densidade. Em contraste, a COP é estável, mantendo um desempenho similar, independentemente da densidade da rede. Este comportamento ocorre porque a rede detecta todos os eventos que acontecem no cenário todos os dias, os nodos se comunicam sempre alterando o mesmo número de mensagens e a rede morre após um período de tempo razoavelmente previsível.

Para analisar melhor a diferença na qualidade dos resultados gerais alcançados pelas abordagens avaliadas, a Figura 20 ilustra a média de eventos processados considerando as quatro densidades. Se a densidade da rede é desconhecida anteriormente, através de COP, as redes de menor densidade terão um desempenho melhor do que as que adotam *Ant-based*, como foi discutido acima. Por outro lado, durante os experimentos, os parâmetros de *Ant-based* e PS foram mantidos idênticos. Além disso, embora o PS e o *Ant-based* tenham parâmetros dependentes da densidade da RSSF (BRISOLARA et al., 2019), a abordagem COP não possui nenhum parâmetro para ser configurado ou ajustado de acordo com a densidade - o que também representa outra vantagem contra PS e *Ant-based*.

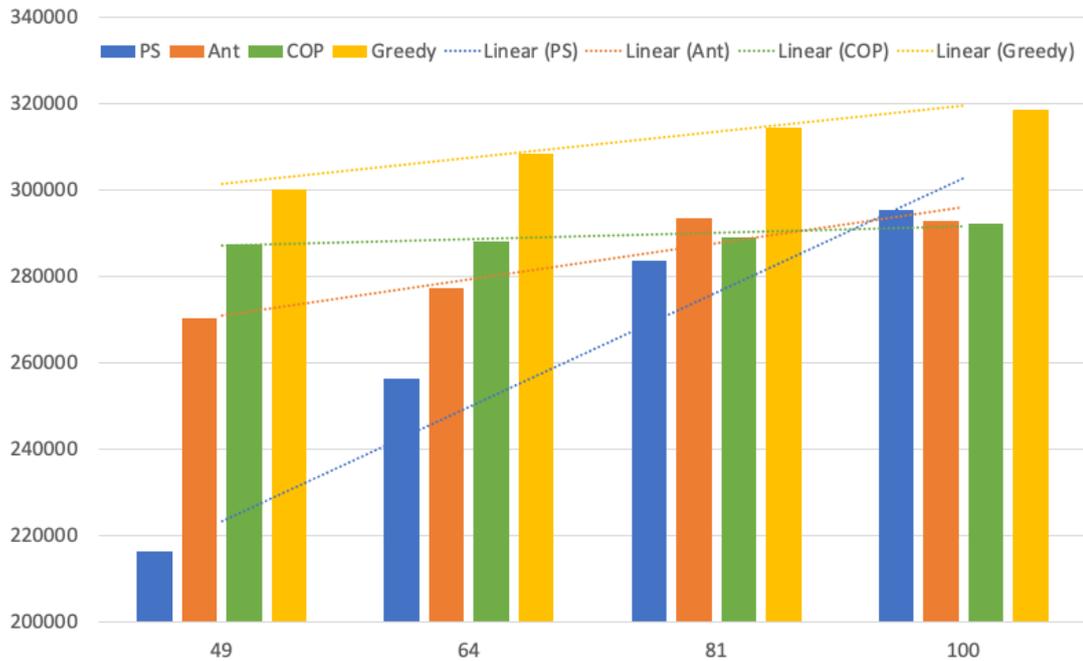


Figura 19 – Resultados Preliminares: Número total de eventos processados: COP x Greedy x Ant x PS.

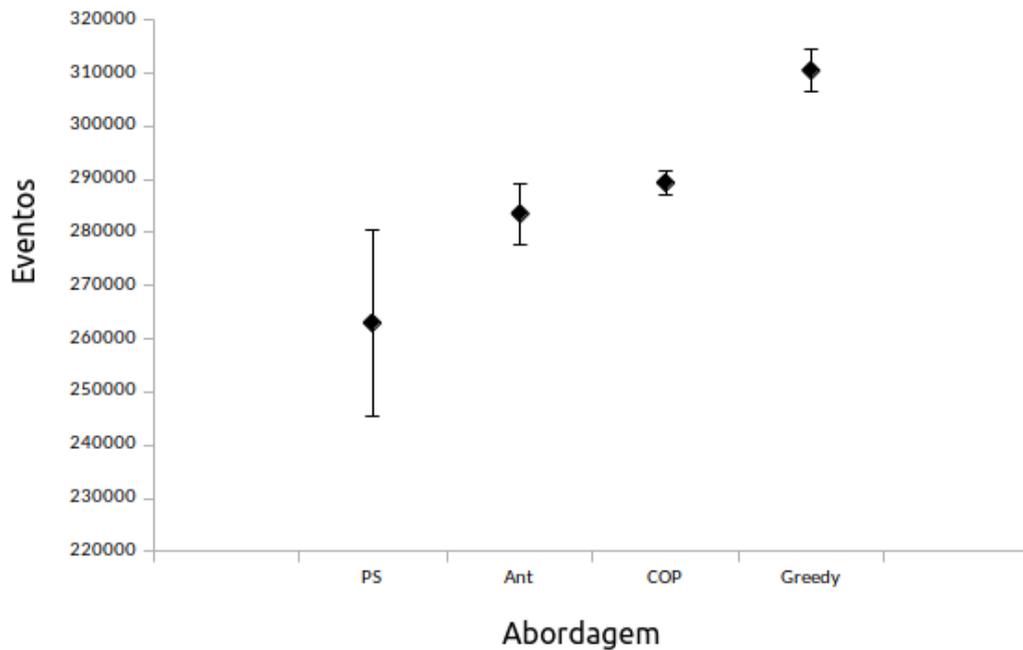


Figura 20 – Resultados Preliminares: Média e dispersão dos eventos detectados por abordagem variando a densidade.

5.3 Resultados COP-Feedback

Esta subsecção avalia o desempenho obtido com a abordagem baseada em COP após a inclusão da técnica de feedback (COP-Feedback), que otimizou a comunicação empregada pela abordagem. Os experimentos permitiram comparar o COP-Feedback

com sua versão preliminar (COP), com as abordagens heurísticas, assim como com a Greedy, considerando cenários de RSSF em grade bem como RSSFs com nodos dispostos aleatoriamente.

5.3.1 Resultados Experimentais nos Cenários em Grade

Nesta seção, a abordagem COP-Feedback é comparada às demais abordagens experimentadas em RSSF com quatro densidades diferentes (49, 64, 81 e 100 nodos). Conforme mostrado na Tabela 4, em relação ao número total médio de eventos detectados pela rede, a abordagem COP com controle de propagação (COP-Feedback) tem um desempenho melhor do que a abordagem estado-da-arte para todas densidades.

Os gráficos das Figuras 21, 22, 23 e 24 representam os resultados para as médias de eventos processados por dia para as diferentes densidades experimentadas, com 49, 64, 81 e 100 nodos, respectivamente. Os resultados apontam que a abordagem proposta, chamada COP-Feedback, foi capaz de atingir resultados bem próximos ao número de eventos detectados por dia pelo Greedy, usado como *upper-bound*. Comparado ao Greedy, o COP-Feedback obteve uma redução na média de eventos processados de 1% a 5%, piorando conforme a densidade aumenta. Em síntese, os eventos são detectados com sucesso pela rede enquanto ela está funcionando, ou seja, o melhor resultado possível sob essa perspectiva. Esse comportamento é uma melhora notável em comparação com o melhor resultado apresentado pela própria abordagem COP sem propagação de mensagens e, anteriormente, por *Ant-based*. Em comparação com COP, COP-Feedback teve uma melhora de desempenho que variou entre 3,14% (cenário de 49 nodos) e 4,97% (cenário de 81 nodos).

Tabela 4 – Média e Desvio Padrão dos Eventos Processados - RSSFs em *grid*.

Abordagem	49 nodos		64 nodos		81 nodos		100 nodos	
	Eventos	σ	Eventos	σ	Eventos	σ	Eventos	σ
Greedy	300317,4	222,2	308538,3	45,5	314375,8	106,9	318704,3	322,9
COP-Feedback	296772,8	247,9	302905,6	293,6	304395,2	149,1	302706	334,8
COP	287439,9	203,5	288102,6	143,7	289280,9	16,5	292289,3	53,1
Ant	270477,4	96,9	277377,9	177,1	293565,1	265,1	292815,4	330,3
PS	216476,1	282,3	256462,8	250,5	283683,1	373,5	295401,8	659,3

A Figura 25 ilustra o número total de eventos detectados ao empregar cada uma das abordagens nas quatro densidades. Pode-se ver como o PS e o *Ant-based* têm seu desempenho dependente da densidade da rede. O PS tem o incremento mais notável no desempenho à medida que a densidade aumenta e conseqüentemente a redundância na cobertura aumenta. Apesar de ser menos sensível à variação de densidades, o *Ant-based* também apresenta uma degradação do desempenho em redes

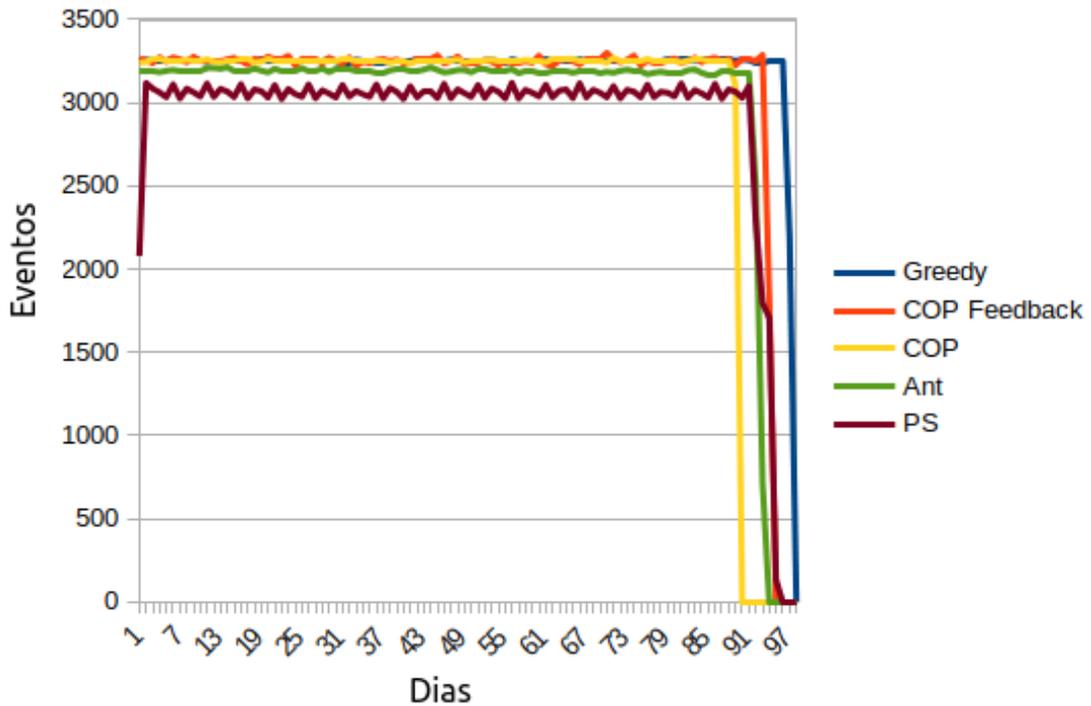


Figura 23 – Média de Eventos Processados por Dia: COP e COP-Feedback vs Demais - RSSF de 81 nodos em grid.

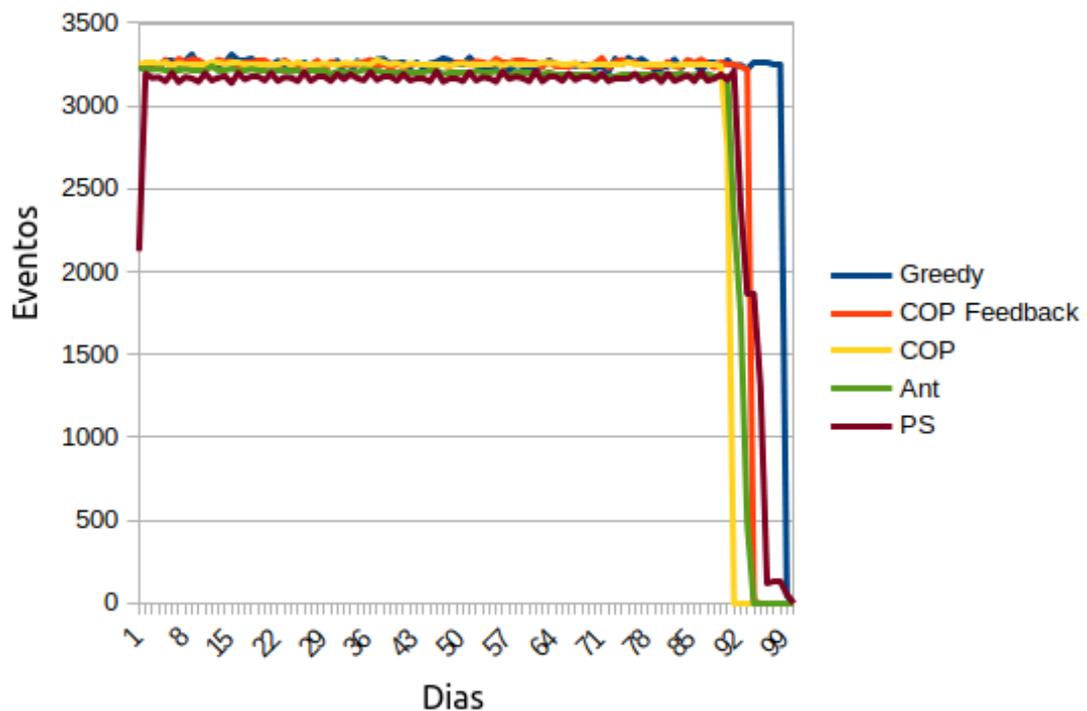


Figura 24 – Média de Eventos Processados por Dia: COP e COP-Feedback vs Demais - RSSF de 100 nodos em grid.

tamento ocorre porque nas abordagens COP e COP-Feedback a rede detecta todos os eventos que acontecem no cenário todos os dias, os nodos se comunicam sempre alterando o mesmo número de mensagens e a rede morre após um período de tempo

razoavelmente previsível.

A densidade também impacta no tempo de vida da rede obtido em cada abordagem. A Figura 26 ilustra a curva de duração da rede (em dias) em cada abordagem para as quatro densidades. Em relação ao número de dias, COP, COP-Feedback, assim como o *Ant-based*, apresentaram um crescimento mais lento no tempo de vida (em dias) em comparação com as demais abordagens quando aumenta-se a densidade da rede. Comparado ao COP, o COP-Feedback mantém a rede funcionando por mais dias graças à técnica de feedback incorporada, melhorando o tempo de vida de 3,2% para 5,5%. Em contrapartida, as abordagens Greedy e PS tendem a aumentar de forma mais significativa o tempo de vida com o aumento da densidade e redundância na cobertura. No entanto, o tempo de vida da rede não pode ser considerado em isolado, pois no caso da abordagem PS empregada na RSSF com 100 nodos, a rede funciona por mais dias mas tem sua cobertura reduzida nos últimos dias e, conseqüentemente, acaba perdendo uma quantidade significativa de eventos durante este período (Figura 24).

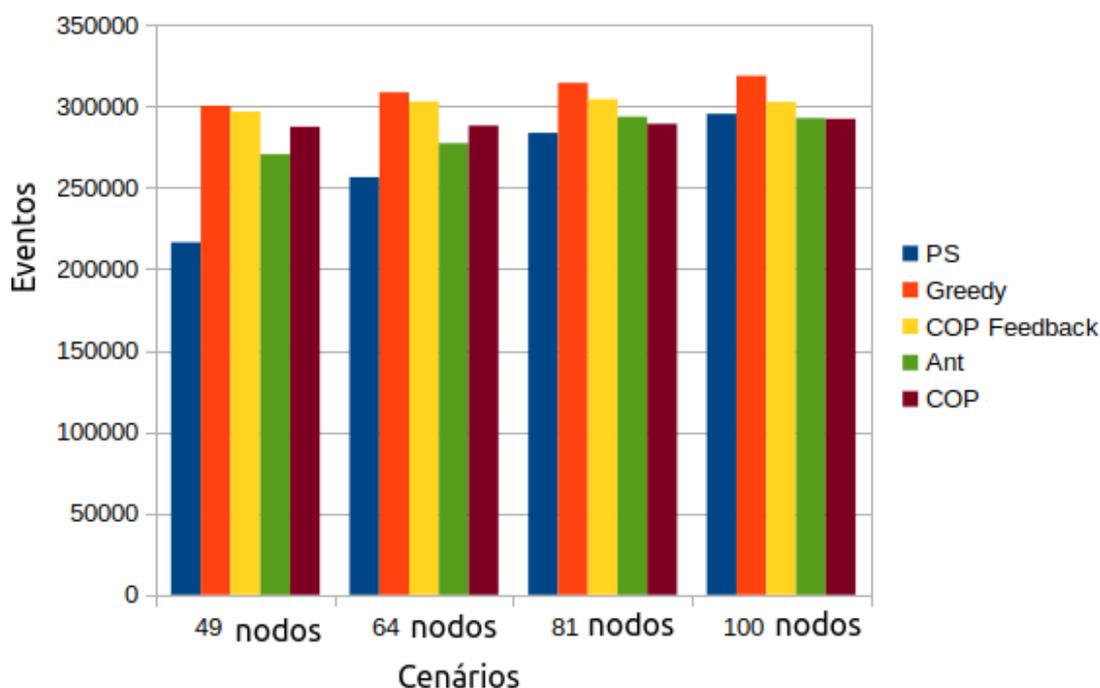


Figura 25 – Desempenho em Número de Eventos - COP e COP-Feedback vs Demais Abordagens no cenários em grid.

5.3.2 Resultados Experimentais nos Cenários Randômicos

Com o objetivo de investigar como a implantação dos nodos afetaria o desempenho das abordagens, foram criados quatro novos cenários de RSSFs, usando agora um arranjo aleatório de nodos, respeitando o mesmo número de nodos das redes em grid (49, 64, 81 e 100 nodos). Os cenários randômicos experimentados podem ser

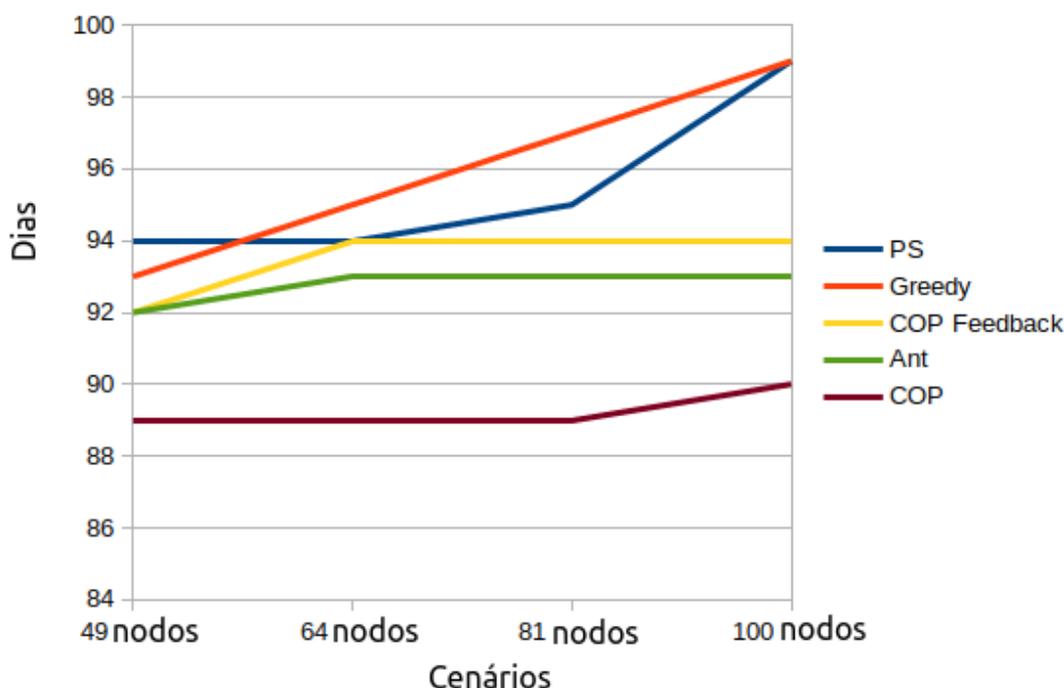


Figura 26 – Desempenho em Número de Dias - COP e COP-Feedback vs Demais Abordagens nos cenários em grid.

visualizados na Figura 14.

As Figuras 27, 28, 29 e 30 ilustram a média de eventos processados por dia pelas RSSF empregando cada uma das abordagens, nos cenários randômicos com 49, 64, 81 e 100 nodos, respectivamente. A Tabela 5 sumariza os resultados de número médio de eventos processados e o desvio padrão das médias para cada abordagem para os cenários de RSSFs com disposição aleatória de nodos. Independe da densidade e da disposição dos nodos, a abordagem COP-Feedback, a qual incorpora o controle de propagação de mensagens, processa em média mais eventos que todas as abordagens avaliadas, exceto que a abordagem Greedy. No cenário de 49 nodos, o Greedy processa 1,85% mais eventos e no cenário de 81 nodos, 5,29%, mencionando apenas a menor e a maior diferença observada. Este comportamento é similar ao que foi observado no comparativo Greedy versus COP-Feedback nos cenários de RSSF com nodos em grade.

Comparado ao COP, o COP-Feedback apresenta melhorias na média de eventos processados que variam de 0,38% a 1,77%, mas dependendo do cenário, valores inferiores aos obtidos nos cenários de grid experimentados. Este era um comportamento esperado pois na implantação randômica a RSSF, geralmente, apresenta lacunas de cobertura e eventos podem ser perdidos. Comparando os gráficos de média de eventos processados por dia dos cenários em grid com os dos cenários randômicos, observa-se que a implantação aleatória dos nodos afetou especialmente a abordagem PS, a qual apresentou os piores resultados em todos os cenários com implanta-

ção randômica (PS tinha o melhor resultado das abordagens heurísticas no cenário de 100 nodos em grid). No entanto, este impacto negativo foi menor nas abordagens baseadas em COP enquanto Ant se manteve com um comportamento estável.

A Figura 31 compara a média de eventos sensoreados pelas abordagens nas trinta execuções e para os quatro cenários experimentados, onde observa-se que o COP-Feedback perde apenas para o Greedy em todos os cenários. Os resultados obtidos por COP-Feedback e COP são superiores aos obtidos pelas abordagens heurísticas em todos os cenários randômicos.

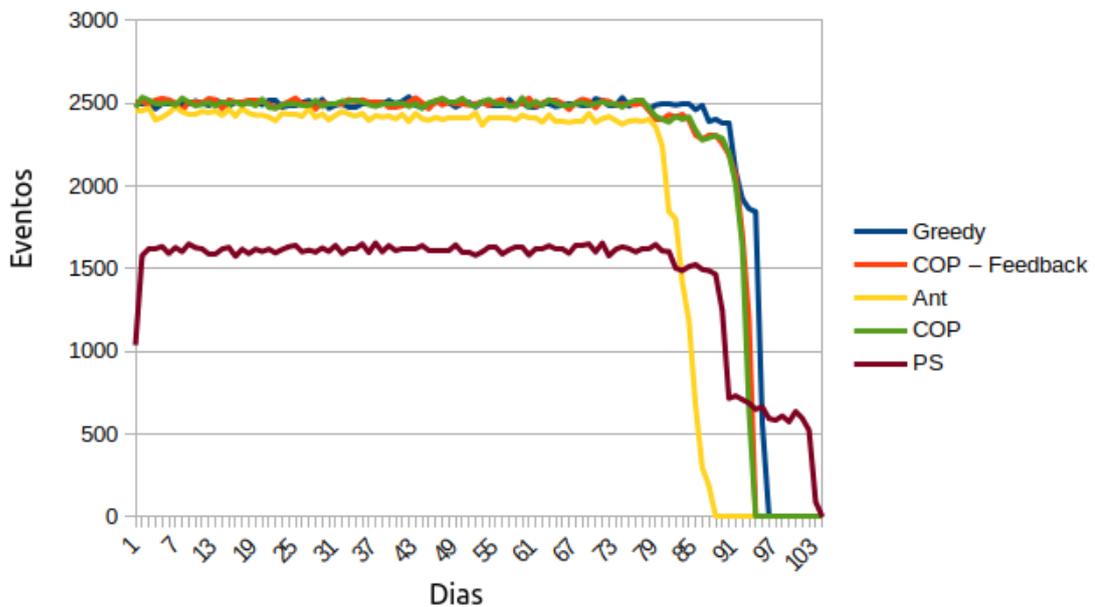


Figura 27 – Cenários Randômicos: Média de Eventos Processados por Dia - 49 nodos.

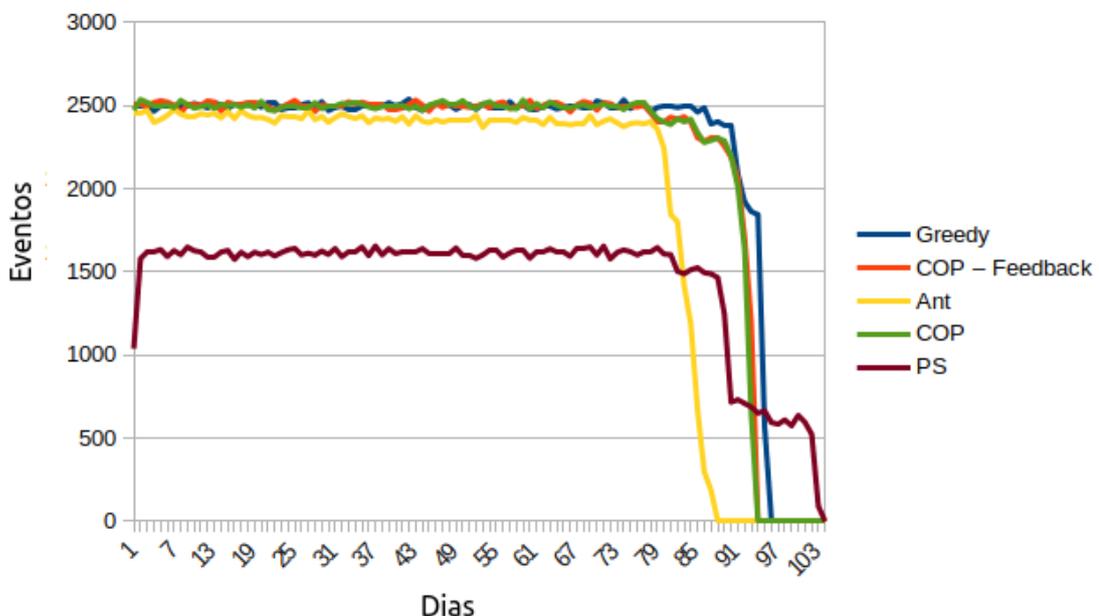


Figura 28 – Cenários Randômicos: Média de Eventos Processados por Dia - 64 nodos.

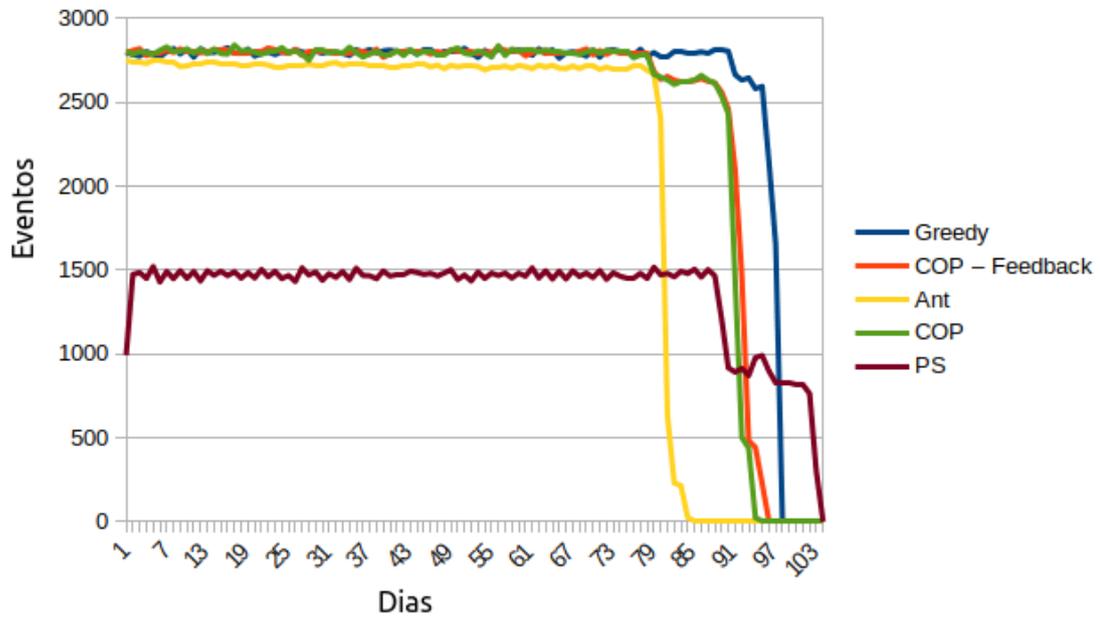


Figura 29 – Cenários Randômicos: Média de Eventos Processados por Dia - 81 nodos.

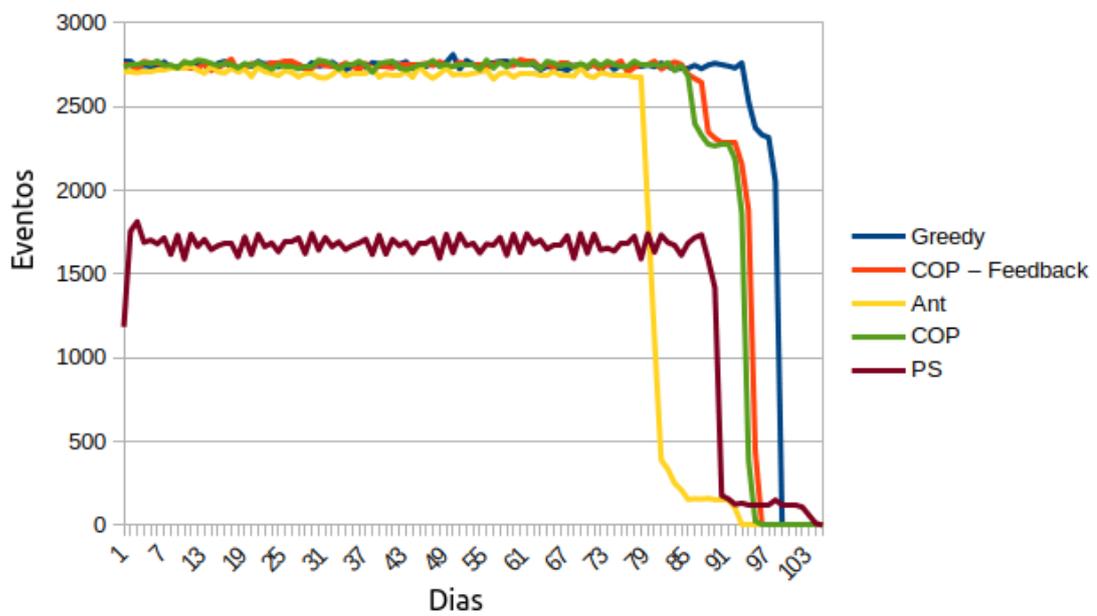


Figura 30 – Cenários Randômicos: Média de Eventos Processados por Dia - 100 nodos.

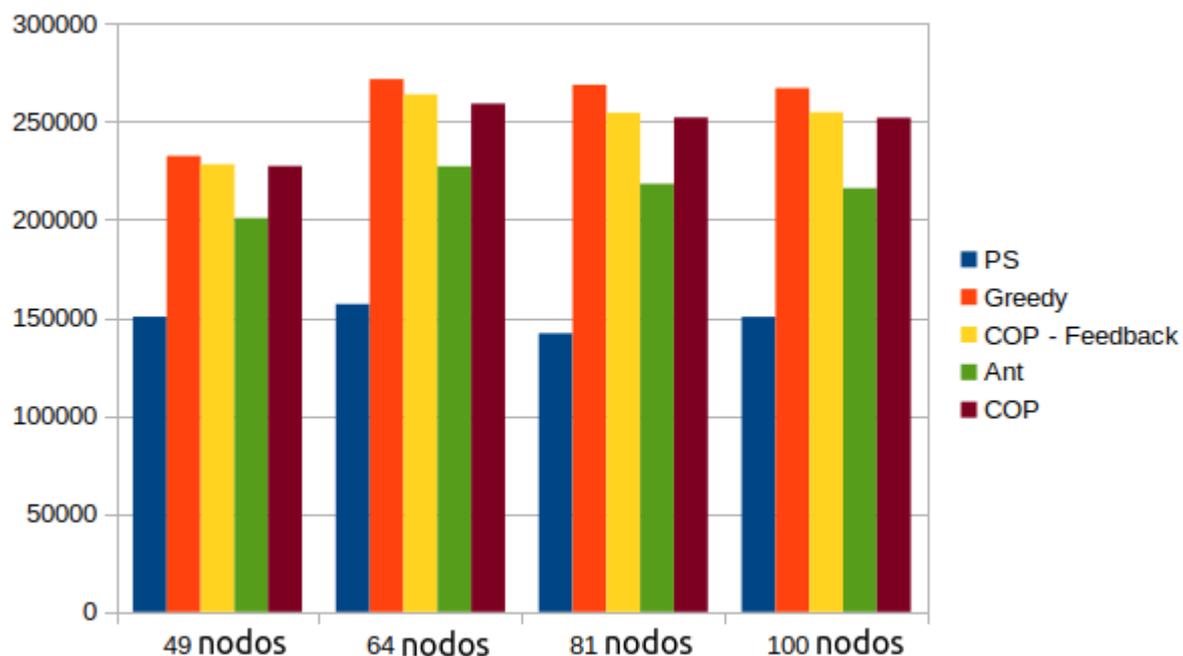


Figura 31 – Desempenho em Número de Eventos - COP e COP-Feedback vs Demais Abordagens no cenário randômico.

Tabela 5 – Média e Desvio Padrão dos Eventos Processados - Cenários randômicos.

Abordagem	49 nodos		64 nodos		81 nodos		100 nodos	
	Eventos	σ	Eventos	σ	Eventos	σ	Eventos	σ
Greedy	232468	228,1	271578,2	228,1	268708,8	138,4	267024,4	101,6
COP - Feedback	228178	170,9	263820,6	407,9	254485,8	448,1	254726,8	273,3
Ant	200733,4	414,5	227141	414,5	218263,13	526,0	215981,6	858,2
COP	227318,4	221,9	259150	498,2	252047	459,1	251895,2	391,8
PS	150425	356,4	156875,6	356,4	141993,2	232,8	150421,6	553,1

6 CONCLUSÃO

As técnicas de balanceamento de carga em RSSFs devem aumentar o tempo de vida e preservar a disponibilidade dos serviços prestados. Entretanto, as otimizações de balanceamento de carga para estas redes ainda são consideradas um grande desafio, especificamente, no que tange o balanceamento da carga de trabalho entre os nodos no nível de aplicação, considerando a carga não só de comunicação como também de processamento requeridos pela aplicação. Recentemente foram propostas técnicas bioinspiradas e distribuídas para resolver o problema em redes reativas, mas que por serem heurísticas podem não atingir uma solução ótima. Por outro lado, soluções baseadas em DCOPs também foram propostas para coordenar os ciclos de ativação e de hibernação dos nodos e assim balancear a carga de processamento em RSSFs, porém requerem algoritmos complexos que podem ter custos proibitivos.

Esta tese propôs uma abordagem semi-distribuída de balanceamento de carga para o nível de aplicação, a qual baseia-se em Problemas de Otimização de Restrição - COP (do inglês, *Constraint Optimization Problems*) e foca em redes reativas com sobreposição nas áreas de cobertura. A abordagem centraliza a escolha do nodo mais adequado para processar o evento detectado, selecionando o nodo com a menor carga de trabalho anterior entre os nodos que detectaram o mesmo evento - evitando o uso de soluções computacionais de alto custo. Comunicação entre nodos vizinhos é empregada para determinar participantes e informar o trabalho realizado. Além disso, para evitar trocas desnecessárias de mensagens, a abordagem proposta incorpora uma técnica de controle de propagação que impede o envio de mensagens redundantes encaminhadas pelo líder para os demais participantes e também destes para o líder no contexto do balanceamento da carga. Assim, sempre que um novo evento é detectado por algum nodo, este verifica, internamente, se já enviou ou não a mesma mensagem - deixando claro que o processo de balanceamento de carga para este evento já foi inicializado por uma mensagem anterior e o processo de deliberação já possui, possivelmente, um líder. Cada vizinho do líder encaminha uma única mensagem de *feedback* que será recebida apenas pelo líder - sendo desconsiderada pelos demais. Desta forma, as mensagens redundantes trocadas pela abordagem antes

da incorporação desta técnica de controle foram evitadas. Assim, o líder encaminha uma única mensagem aos vizinhos em busca de participantes para o processo de balanceamento de carga e uma outra mensagem para propagar o resultado de sua deliberação, enquanto cada vizinho passa a enviar somente uma mensagem de *feedback* - contendo o número de eventos processados e confirmando sua participação no processo de balanceamento de carga.

Nos experimentos realizados foram utilizadas configurações de RSSFs com nodos dispostos no formato de grade e aleatório, variando as densidades em 49, 64, 81 e 100 nodos e gerando eventos com base em modelos estocásticos. De acordo com os resultados, a abordagem proposta processou em média um número maior de eventos do que as abordagens heurísticas *Ant-based* e PS, em todos cenários avaliados. Além disso, tanto a abordagem COP como a COP-Feedback obteve resultados bem próximos à técnica gulosa não-prática para RSSFs reais - também usada nas análises como *upper-bound*. Em relação ao número de eventos detectados por dia, a abordagem COP/COP-Feedback mantém seu desempenho ainda em baixa densidade, não exigindo sobreposição excessiva entre os nodos (sendo mais estável que *Ant-based* e PS). Outra vantagem é o fato de que a abordagem COP/COP-Feedback não possui parâmetros dependentes da densidade a serem ajustados para uma determinada configuração de rede, conforme é exigido por *Ant-based* e PS. A abordagem COP/COP-Feedback é indicada quando os eventos não devem ser perdidos e uma pequena redução na vida útil não é um problema - enquanto as abordagens heurísticas tendem a melhorar a vida útil, estas perdem eventos ao longo de todo o tempo de vida da rede.

Experimentos foram realizados em redes homogêneas considerando apenas eventos não-simultâneos. No entanto, redes com disparo simultâneo de eventos estão cada vez mais sendo encontradas em ambientes reais (redes com milhares de nodos e uma grande área de monitoramento). Assim, como trabalho futuro, pretende-se avaliar o impacto de eventos simultâneos no desempenho da técnica proposta. Nodos heterogêneos com configurações diferenciadas de hardware também podem ser empregados, visando melhorar a cobertura ou a comunicação em algumas regiões da rede onde interferências exigem transmissores mais potentes. Estes nodos terão diferentes capacidades de comunicação, de processamento e, conseqüentemente, custos energéticos diferenciados. Assim, novos experimentos devem ser realizados para mensurar o impacto de nodos heterogêneos na abordagem proposta. Antunes; Brisolara; Ferreira (2020) explorou diferentes relações de custos de comunicação e processamento para avaliar o emprego de *sinks* móveis em RSSFs. Desta forma, esta mesma ideia pode ser também empregada para explorar a heterogeneidade de nodos e sua implicação nas técnicas de balanceamento de carga propostas neste trabalho.

Pretende-se ainda explorar variantes da função de custo empregada na aborda-

gem proposta, levando em conta outros parâmetros como, por exemplo, algo que explore as propriedades temporais e espaciais dos nodos ou, até mesmo, algo mais trivial como o nível atual de bateria de cada nodo - ao invés de somente a quantidade de eventos processados. Mais informação ou uma informação mais aprimorada pode levar a obtenção de uma solução melhor, porém há de se ter uma preocupação com uma sobrecarga imposta por uma função de custo mais complexa. Outro aprimoramento interessante à abordagem seria a garantia de que o nodo escolhido para o processamento teria um caminho viável para encaminhar seus dados até a estação base.

REFERÊNCIAS

AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. Wireless sensor networks: a survey. **Comput Networks**, [S.l.], v.38, n.4, p.393–422, 2002.

ANASTASI, G. et al. Performance Measurements of Motes Sensor Networks. In: ACM INTERNATIONAL SYMPOSIUM ON MODELING, ANALYSIS AND SIMULATION OF WIRELESS AND MOBILE SYSTEMS, 7., 2004, New York, NY, USA. ... ACM, 2004. p.174–181. (MSWiM '04).

Antunes, J. R.; Brisolara, L.; Ferreira, P. R. UAVs as Data Collectors in the WSNs: Investigating the Effects of Back-and-Forth and Spiral Coverage Paths in the Network Lifetime. In: X BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING (SBESC), 2020., 2020. ... [S.l.: s.n.], 2020. p.1–8.

BARRIUSO, A. L. et al. Combination of multi-agent systems and wireless sensor networks for the monitoring of cattle. **Sensors (Switzerland)**, [S.l.], v.18, n.1, jan 2018.

BRISOLARA, L. et al. Application modeling for performance evaluation on event-triggered wireless sensor networks. **Design Automation for Embedded Systems**, Hingham, MA, USA, v.20, n.4, p.269–287, dec 2016.

BRISOLARA, L.; FERREIRA, P. R.; INDRUSIAK, L. S. Impact of Temporal and Spatial Application Modeling on Event-Triggered Wireless Sensor Network Evaluation. In: BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING (SBESC), 2015. ... IEEE, 2015.

BRISOLARA, L.; Lorenzatto Braga, M.; BRAGA, A.; Roberto Ferreira, P. Parameter Tuning in Load Balancing Techniques for Wireless Sensor Networks through Genetic Algorithms. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS (BRACIS), 8., 2019. ... IEEE, 2019. p.42–47.

BRITO, L. M. P. L. de; PERALTA, L. M. R. An Analysis of Localization Problems and Solutions in Wireless Sensor Networks. **Polytechnical Studies Review**, [S.l.], p.146–172, 2008.

CALISKANELLI, I. **A Bio-inspired Load Balancing Technique for Wireless Sensor Networks**. 2014. 180p. Tese (Doutorado em Ciência da Computação) — The University of York. (March).

CALISKANELLI, I.; BROECKER, B.; TUYLS, K. Multi-robot coverage: A bee pheromone signalling approach. In: ARTIFICIAL LIFE AND INTELLIGENT AGENTS SYMPOSIUM, 2014. ... [S.l.: s.n.], 2014. p.124–140.

CALISKANELLI, I. et al. Bioinspired Load Balancing in Large-Scale WSNs Using Pheromone Signalling. **International Journal of Distributed Sensor Networks**, [S.l.], v.9, n.7, p.172012, 2013.

CALISKANELLI, I. et al. Bioinspired load balancing in large-scale WSNs using pheromone signalling. **International Journal of Distributed Sensor Networks**, [S.l.], v.2013, n.July, 2013.

CHACHRA, S.; MAREFAT, M. Distributed algorithms for sleep scheduling in wireless sensor networks. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2006. ICRA 2006., 2006. ... [S.l.: s.n.], 2006. v.2006, n.May, p.3101–3107.

DERAKHSHAN, F.; YOUSEFI, S. A review on the applications of multiagent systems in wireless sensor networks. **International Journal of Distributed Sensor Networks**, [S.l.], v.15, n.5, 2019.

DORRI, A.; KANHERE, S. S.; JURDAK, R. Multi-Agent Systems: A Survey. **IEEE Access**, [S.l.], v.6, p.28573–28593, 2018.

EBERHART; SHI, Y. Particle swarm optimization: developments, applications and resources. In: IEEE CONFERENCE ON EVOLUTIONARY COMPUTATION, ICEC, 2001. ... IEEE, 2001. v.1, p.81–86.

EVER, E. et al. UHEED - An Unequal Clustering Algorithm for Wireless Sensor Networks. In: INTERNATIONAL CONFERENCE ON SENSOR NETWORKS - VOLUME 1: SENSORNETS,, 1., 2012. ... SciTePress, 2012. p.185–193.

FARINELLI, A.; ROGERS, A.; JENNINGS, N. R. Agent-based decentralised coordination for sensor networks using the max-sum algorithm. **Autonomous Agents and Multi-Agent Systems**, [S.l.], v.28, n.3, p.337–380, 2014.

FARINELLI, A.; ROGERS, A.; PETCU, A.; JENNINGS, N. R. Decentralised coordination of low-power embedded devices using the max-sum algorithm. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS-VOLUME 2, 7., 2008. ... [S.l.: s.n.], 2008. v.2, n.Aamas, p.630–637.

FERREIRA, P. R.; BRISOLARA, L.; INDRUSIAK, L. S. Decentralised load balancing in event-triggered WSNs based on ant colony work division. In: EUROMICRO CONFERENCE ON SOFTWARE ENGINEERING AND ADVANCED APPLICATIONS, 41., 2015. ... IEEE, 2015.

FLUECKIGER, A.; VERMAN, M.; BERNSTEIN, A. Improving Approximate Algorithms for DCOPs Using Ranks. **Workshop on MAS**, [S.l.], 2016.

GONZALEZ, A. V.; FERREIRA, P. R.; BRISOLARA, L. B. de. Mobile Nodes as a Dynamic Management Strategy to Improve Coverage in Wireless Sensor Networks. In: {VI} BRAZILIAN SYMPOSIUM ON COMPUTING SYSTEMS ENGINEERING ({SBESC}), 2016. ... IEEE, 2016.

GONZALEZ, A. V.; FERREIRA, P. R.; BRISOLARA, L. B. de. Efficiency Evaluation of Strategies for Dynamic Management of Wireless Sensor Networks. **Wireless Communications and Mobile Computing**, [S.l.], v. Volume 201, p.13, 2017.

GULARTE, A. R. **Uma Revisão dos Algoritmos para Solução de Problemas de Otimização de Restrição**. Rio Grande: [s.n.], 2013. 62p.

GULERIA, K.; VERMA, A. K. An energy efficient load balanced cluster-based routing using ant colony optimization for WSN. **International Journal of Pervasive Computing and Communications**, [S.l.], v.14, n.3/4, p.233–246, 2018.

GUO, W.; ZHANG, W.; LU, G. PEGASIS protocol in wireless sensor network based on an improved ant colony algorithm. **2nd International Workshop on Education Technology and Computer Science, ETCS 2010**, [S.l.], v.3, p.64–67, 2010.

GUPTA, A.; BAYAZIT, A. A.; MAHAJAN, Y. **Multi-agent System Architectures for Wireless Sensor Networks**. 10–1–10–20p.

HEINZELMAN, W. B.; CHANDRAKASAN, A. P.; BALAKRISHNAN, H. An application-specific protocol architecture for wireless microsensor networks. **IEEE Transactions on wireless communications**, [S.l.], v.1, n.4, p.660–670, 2002.

HEINZELMAN, W. R.; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 33., 2000. ... [S.l.: s.n.], 2000. v.10, n.c, p.10 pp.

HOUSSAINI, D. E.; KHRIJI, S.; BESBES, K.; KANOUN, O. Wireless sensor networks in agricultural applications. **Energy Harvesting for Wireless Sensor Networks: Technology, Components and System Design**, [S.l.], n. November, p.323–342, 2018.

JIN, Y.; VURAL, S.; GLUHAK, A.; MOESSNER, K. **Dynamic task allocation in multi-hop multimedia wireless sensor networks with low mobility**. [S.l.]: Multidisciplinary Digital Publishing Institute, 2013. 13998–14028p. v.13, n.10.

KARABOGA, D.; GORKEMLI, B.; OZTURK, C.; KARABOGA, N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. **Artificial Intelligence Review**, [S.l.], v.42, n.1, p.21–57, jun 2014.

KARABOGA, D.; OKDEM, S.; OZTURK, C. Cluster Based Wireless Sensor Network Routing Using Artificial Bee Colony Algorithm. **Wireless Networks**, Secaucus, NJ, USA, v.18, n.7, p.847–860, 2012.

KHAN, Z. et al. Efficient routing for corona based underwater wireless sensor networks. **Computing**, [S.l.], 2018.

KORKALAINEN, M.; SALLINEN, M.; KÄRKKÄINEN, N.; TUKEVA, P. Survey of wireless sensor networks simulation tools for demanding applications. In: THE FIFTH INTERNATIONAL CONFERENCE ON NETWORKING AND SERVICES - ICNS, 2009. ... [S.l.: s.n.], 2009. p.102–106.

LINDSEY, S.; RAGHAVENDRA, C. S. PEGASIS: Power-efficient gathering in sensor information systems. In: AEROSPACE CONFERENCE, 2002. ... IEEE, 2002. v.3, p.3.

LIU, X.; LI, S.; WANG, M. An Ant Colony based Routing Algorithm for Wireless Sensor Network. **International Journal of Future Generation Communication and Networking**, [S.l.], v.9, n.6, p.75–86, jun 2016.

MARQUES, M. R.; BRISOLARA, L.; FERREIRA, P. R.; INDRUSIAK, L. S. Eboracum: An extensible framework for high-level modeling and evaluation of reactive and adaptable WSNs. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION, ETFA, 2016. ... [S.l.: s.n.], 2016. v.2016-Novem, p.1–8.

MEISELS, A. Constraints Optimization Problems - COPs. In: **Distributed Search by Constrained Agents: Algorithms, Performance, Communication**. London: Springer London, 2008. p.19–26.

MENICHELLI, F.; OLIVIERI, M. Tiktak: a scalable simulator of wireless sensor networks including hardware/software interaction. **Wireless Sensor Network**, [S.l.], v.2, n.11, p.815, 2010.

MISBAHUDDIN, M. et al. Dynamic multi-hop routing protocol based on fuzzy-firefly algorithm for data similarity aware node clustering in WSNs. **International Journal of Computers, Communications and Control**, [S.l.], v.13, n.1, p.99–116, 2018.

MODI, P. J. **Distributed constraint optimization for multiagent systems**. 2003. Tese (Doutorado em Ciência da Computação) — University of Southern California.

MODI, P. J.; SHEN, W.-M. M.; TAMBE, M.; YOKOO, M. Adopt: Asynchronous Distributed Constraint Optimization with Quality Guarantees. **Artif. Intell.**, Essex, UK, v.161, n.1-2, p.149–180, 2005.

MOHAMED, R.; SALEH, A.; ABDELRAZZAK, M.; S. Samra, A. Survey on Wireless Sensor Network Applications and Energy Efficient Routing Protocols. **Wireless Personal Communications**, [S.l.], v.Volume 101, 2018.

MOR, S.; SAROHA, M. V. Load balancing in wireless sensor networks. **International Journal of Software and Web Sciences (IJSWS) ISSN**, [S.l.], p.63–2279, 2013.

NEHRA, V.; SHARMA, A. K. PEGASIS-E: Power efficient gathering in sensor information system extended. **Global Journal of Computer Science and Technology**, [S.l.], 2013.

NIVEDHA, V.; THOMAS, A. K. Rotated Hybrid, Energy-Efficient and Distributed (R-HEED) clustering protocol in WSN. **International Journal of Applied Engineering Research**, [S.l.], v.10, n.20, p.16416–16420, 2015.

PANDEY, S.; KUMAR, R. Re-LEACH: An Energy-Efficient Secure Routing Protocol for Wireless Sensor Networks. In: INTERNATIONAL CONFERENCE ON COMPUTER NETWORKS AND COMMUNICATION TECHNOLOGIES, 2019, Singapore. ... Springer Singapore, 2019. p.777–787.

PAWANDEEP, P.; GARG, M.; JAIN, N. An Energy Efficient Routing Protocol using ABC to Increase Survivability of WSN. **International Journal of Computer Applications**, [S.l.], v.143, p.37–42, 2016.

PEREIRA, I. A.; BRISOLARA, L. B. D.; ROBERTO, P.; JR, F. Application-Level Load Balancing for Reactive Wireless Sensor Networks : An Approach Based on Constraint Optimization Problems. In: BRAZILIAN CONFERENCE ON INTELLIGENT SYSTEMS, 2020, Rio Grande. ... [S.l.: s.n.], 2020. p.1–15.

PETCU, A.; FALTINGS, B. **A scalable method for multiagent constraint optimization**. [S.l.]: Morgan Kaufmann Publishers Inc., 2005.

POTDAR, V.; SHARIF, A.; CHANG, E. Wireless Sensor Networks: A Survey. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS, 2009. ... IEEE, 2009.

RAJKUMAR et al. Energy Efficient Routing Clustering Algorithms For Wireless Sensor Networks. **International Journal of Advanced Networking and Applications**, [S.l.], v.10, n.3, p.3880–3887, dec 2018.

ROSA, R. L. et al. A Battery-Free Smart Sensor Powered with RF Energy. In: INTERNATIONAL FORUM ON RESEARCH AND TECHNOLOGY FOR SOCIETY AND INDUSTRY (RTSI), 4., 2018. ... [S.l.: s.n.], 2018. p.1–4.

RUST, P. **pydcop Documentation - Release 0.1.2a1**. Disponível em: <<https://pydcop.readthedocs.io>>.

SAHA, D.; DAS, N. Self-organized area coverage in wireless sensor networks by limited node mobility. **Innovations in Systems and Software Engineering**, [S.l.], v.12, n.3, p.227–238, 2016.

SAMBO, D. W. et al. Optimized clustering algorithms for large wireless sensor networks: A review. **Sensors (Switzerland)**, [S.l.], v.19, n.2, 2019.

SANTOS, F. dos. **Implementação distribuída do algoritmo Adopt**.

SGHIR, I.; JAAFAR, I. B.; GHÉDIRA, K. A multi-agent based optimization method for combinatorial optimization problems. **International Journal on Artificial Intelligence Tools**, [S.l.], v.27, n.5, 2018.

SINHA, S.; JOSE, D. V. An augmented artificial bee colony algorithm for data aggregation in wireless sensor networks. **International Journal of Civil Engineering and Technology**, [S.l.], v.8, n.12, p.651–661, 2017.

TRIPATHI, M.; GAUR, M. S.; LAXMI, V.; BATTULA, R. B. Energy efficient {LEACH}-C protocol for wireless sensor network. In: THIRD INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND INFORMATION TECHNOLOGY ({CIIT} 2013), 2013. ... Institution of Engineering and Technology, 2013.

TYAGI, S.; KUMAR, N. A Systematic Review on Clustering and Routing Techniques Based Upon LEACH Protocol for Wireless Sensor Networks. **J. Netw. Comput. Appl.**, London, UK, UK, v.36, n.2, p.623–645, 2013.

VINYALS, M.; PUJOL, M.; RODRIGUEZ-AGUILAR, J. A.; CERQUIDES, J. Divide-and-coordinate: DCOPs by agreement. **Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS**, [S.l.], v.1, n.Aamas, p.149–156, 2010.

WANG, F. et al. A task allocation algorithm based on score incentive mechanism for wireless sensor networks. **International Journal of Distributed Sensor Networks**, [S.l.], v.11, n.8, p.286589, jan 2015.

XINGGUO, L.; JUNFENG, W.; LINLIN, B. {LEACH} Protocol and Its Improved Algorithm in Wireless Sensor Network. In: INTERNATIONAL CONFERENCE ON CYBER-ENABLED DISTRIBUTED COMPUTING AND KNOWLEDGE DISCOVERY ({CYBERC}), 2016. ... IEEE, 2016.

XU, L.; COLLIER, R.; O'HARE, G. M. P. A Survey of Clustering Techniques in WSNs and Consideration of the Challenges of Applying Such to 5G IoT Scenarios. **IEEE Internet of Things Journal**, [S.l.], v.4, n.5, p.1229–1249, 2017.

YANG, J. et al. Task allocation for wireless sensor network using modified binary particle swarm optimization. **IEEE Sensors Journal**, [S.l.], v.14, n.3, p.882–892, mar 2014.

YANG, Y.; FU, G. Clustering Routing Algorithm in Wireless Sensor Networks Based on artificial Bee Colony and Assistant Cluster Heads. **{MATEC} Web of Conferences - International Conference on Engineering Technology and Application (ICETA 2015)**, [S.l.], v.22, p.4, 2015.

YEDIDSON, H.; ZIVAN, R.; FARINELLI, A. Explorative max-sum for teams of mobile sensing agents. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS, 2014. ... [S.l.: s.n.], 2014. p.549–556.

YEOH, W. **Speeding up distributed constraint optimization search algorithms**. [S.l.]: University of Southern California, 2010. 239p. v.3434536, n.December.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, [S.l.], v.52, n.12, p.2292–2330, 2008.

YOUNIS, O.; FAHMY, S.; O, Y.; S, F. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. **IEEE Transactions on mobile computing**, [S.l.], v.3, n.4, p.366–379, 2004.

YU, W.; HUANG, Y.; GARCIA-ORTIZ, A. Distributed optimal on-line task allocation algorithm for wireless sensor networks. **IEEE SENSORS JOURNAL**, [S.l.], v.18, n.1, p.446–458, 2018.

ZADE, S.; SAMBHE, N.; KAMBLE, S.; PALEKAR, V. Review on Improving Lifetime of Network Using Energy And Density Control Cluster Algorithm. In: IEEE INTERNATIONAL STUDENTS' CONFERENCE ON ELECTRICAL, ELECTRONICS AND COMPUTER SCIENCE (SCEECS), 2018. ... IEEE, 2018. p.1–5.

ZIVAN, R.; PARASH, T.; COHEN-LAVI, L.; NAVEH, Y. Applying Max-sum to asymmetric distributed constraint optimization problems. **Autonomous Agents and Multi-Agent Systems**, [S.l.], v.34, n.1, p.1–29, 2020.

ZUHAIRY, R. M.; ZAMIL, M. G. A. Energy-efficient load balancing in wireless sensor network: An application of multinomial regression analysis. **International Journal of Distributed Sensor Networks**, [S.l.], v.14, n.3, p.1550147718764641, 2018.