

FERNANDO HENRIQUE GIELOW

**AGRUPAMENTO DINÂMICO DE SENSORES
BASEADO NA SIMILARIDADE DE LEITURA DE DADOS**

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informá-
tica, Setor de Ciências Exatas, Universidade
Federal do Paraná.

Orientador: Prof. Aldri Luiz dos Santos

Curitiba

2013

FERNANDO HENRIQUE GIELOW

**AGRUPAMENTO DINÂMICO DE SENSORES
BASEADO NA SIMILARIDADE DE LEITURA DE DADOS**

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informá-
tica, Setor de Ciências Exatas, Universidade
Federal do Paraná.

Orientador: Prof. Aldri Luiz dos Santos

Curitiba

2013

RESUMO

As Redes de Sensores Sem Fio (RSSFs) vêm sendo utilizadas em diversos tipos de aplicações de monitoramento urbano ou ambiental. Uma técnica comum nas RSSFs é o agrupamento lógico dos dispositivos da rede, ou nós, que gera uma hierarquia baseada em líderes e possibilita o melhor controle do tráfego da rede, melhorando questões de escalabilidade e eficiência em energia. Embora os protocolos de agrupamento tratem de várias categorias de dinamicidade na rede, desde a variação do posto de líder até a mobilidade dos nós na rede como um todo, são poucas as soluções que consideram as questões de similaridade entre as leituras dos nós. As relações de similaridade temporal e espacial das leituras possibilitam que os nós de leituras similares sejam mantidos no mesmo agrupamento lógico, trazendo diversas vantagens, como o uso mais eficiente de técnicas de compressão e até mesmo uma melhor detecção de eventos anômalos. Para a manutenção de agrupamentos que consideram a similaridade, uma alternativa é o uso da sincronização da leitura média dos agrupamentos nos seus nós. Esta alternativa possibilitaria um funcionamento distribuído e adaptativo dos agrupamentos. Assim, este trabalho propõe um protocolo de agrupamento lógico de nós para RSSFs que considera explicitamente a relação de similaridade espacial entre as leituras dos nós, visando o uso em aplicações para detecção de eventos, sem que haja consultas proativas a partir de uma estação-base. O protocolo proposto, chamado DDFC, emprega os princípios biológicos de vagalumes encontrados na natureza que, até o momento, são utilizados nas RSSFs para questões de sincronização de relógio, possuindo um grande potencial inexplorado. Uma avaliação do protocolo foi realizada através de simulações, que analisaram seu comportamento, comparando-o com uma variante sua e com o protocolo SCCS. Os resultados mostraram que o protocolo DDFC foi capaz de manter a agregação de leituras dos agrupamentos lógicos sincronizada, possibilitando que os nós de leituras similares se mantivessem agrupados. Além disso, o protocolo operou de maneira mais estável e dinâmica, selecionando líderes mais apropriados.

Palavras-chave: agrupamentos, vagalumes, protocolos, técnicas bio-inspiradas, RSSFs.

ABSTRACT

Wireless Sensor Networks (WSNs) have been used by several kinds of urban and nature monitoring applications. The logical clustering of nodes on WSNs is a greatly used technique, corresponding to the creation of a hierarchy of nodes based on leaders that enables better network traffic management, improving issues such as scalability and energy efficiency. Although clustering protocols handle several categories of network dynamicity, from the rotation of the leader up to nodes' mobility on the network as a whole, few are the clustering solutions that consider data similarity among nodes readings. The temporal and spatial similarity relations allow nodes with similar readings to be kept on the same logical cluster, bringing several advantages, such as improving both the efficiency of compression techniques and even the detection of anomalous events. For maintaining similarity aware clusters, the synchronization of the cluster's average reading would allow a distributed and adaptive operation. Thus, this work proposes a logic clustering protocol for WSNs that explicitly handles the spatial similarity between nodes' readings, aiming events detection applications, without the need of proactive queries yielded by the base station. The proposed protocol, named DDFC, is inspired on the biological principles of fireflies which, up to the moment, usually are used on WSNs for clock synchronization matters, still leaving a huge potential unexplored. An evaluation of the protocol was performed through simulations, which analyzed the protocol's behavior, comparing it with another variation of itself and with the SCCS protocol. Results show that the DDFC protocol managed to keep the clusters' readings aggregation synchronized, hence enabling nodes with similar readings to be kept clustered together. Furthermore, the protocol behaved with more stability, electing more appropriate cluster-heads.

Keywords: clustering, fireflies, protocols, bio-inspired techniques, WSNs.

Sumário

RESUMO	i
ABSTRACT	ii
LISTA DE FIGURAS	vi
LISTA DE ABREVIATURAS E SIGLAS	vii
NOTAÇÃO	viii
1 INTRODUÇÃO	1
1.1 Motivação	3
1.2 Objetivos	4
1.3 Contribuições	4
1.4 Estrutura da dissertação	5
2 FUNDAMENTOS	6
2.1 Redes de sensores sem fio	6
2.1.1 RSSFs no meio urbano	7
2.2 Agrupamentos de nós	8
2.2.1 Critérios para agrupamento lógico	9
2.2.2 Vantagens de agrupar os sensores	10
2.2.3 Similaridade de dados	13
2.3 Sistemas bio-inspirados	14
2.4 Resumo	17
3 AGRUPAMENTOS DINÂMICOS	18
3.1 Dinamicidade de líder	18
3.1.1 LEACH	19
3.1.2 HEED	19
3.2 Dinamicidade pela mobilidade	20
3.2.1 SPRP _G	20
3.2.2 KHOPCA	21
3.3 Dinamicidade por recriação	22
3.3.1 DCRR	22
3.4 Dinamicidade das leituras dos dados	23
3.4.1 SCCS	24

3.5	Discussão	25
3.6	Resumo	27
4	DDFC: UM PROTOCOLO DE AGRUPAMENTO	28
4.1	Visão geral	28
4.1.1	Unidade de Sincronização	30
4.1.2	Unidade de Indexação	31
4.1.3	Unidade Adaptativa	31
4.1.4	Analogia da inspiração em princípios biológicos	32
4.2	Funcionamento do DDFC	33
4.2.1	Armazenamento de vizinhança	33
4.2.2	Sincronização da agregação de leituras	33
4.2.3	Indexação local das rotas até os líderes	37
4.2.4	Adaptação dos intervalos de atuação	39
4.3	Resumo	41
5	AVALIAÇÃO DE DESEMPENHO	42
5.1	Cenário da simulação	42
5.1.1	Interpolação contínua de leituras discretas	43
5.1.2	Extensão do visualizador gráfico PyViz	44
5.2	Parâmetros e métricas	44
5.3	Avaliação de resultados	47
5.3.1	Agrupamentos formados	47
5.3.1.1	Distância média até os líderes e duração dos líderes	49
5.3.2	Similaridade de leituras dos nós agrupados	53
5.3.3	Inconsistência das rotas	56
5.3.4	Mensagens <i>beacon</i>	58
5.3.5	Discussão	59
5.3.6	Análise crítica	61
5.3.6.1	Análise do DDFC-K	62
5.3.6.2	Análise do SCCS	62
5.4	Resumo	63
6	CONCLUSÃO	64
6.1	Trabalhos futuros	65
6.2	Aplicações diretas do protocolo	66
	BIBLIOGRAFIA	73
	ANEXO	74

Lista de Figuras

2.1	Nós sensores integrados no meio urbano	7
2.2	Agrupamento lógico de nós	9
2.3	Roteamento de dados através dos líderes até a estação-base	10
2.4	Nós de leituras similares agrupados de maneira fragmentada	12
2.5	Exemplos de definições de similaridade de dados	13
2.6	Troca de mensagens de vagalumes e sincronização de períodos	15
2.7	Par de osciladores de vagalumes sincronizando com o pulsar	15
3.1	Árvore de agrupamentos formada pelo SPRP _G	21
3.2	Estrutura de agrupamentos formada pelo SCCS	25
4.1	Arquitetura em alto nível do protocolo de agrupamento DDFC	28
4.2	Posicionamento do protocolo DDFC entre as camadas de comunicação	29
4.3	Estrutura hierárquica estabelecida pelo DDFC	29
4.4	Estruturas de dados que representam a vizinhança dos nós	33
4.5	Funcionamento do agrupamento baseado em vagalumes	36
4.6	Aplicação das regras da Unidade de Indexação	39
4.7	Fluxograma de funcionamento da Unidade Adaptativa	41
5.1	Cenário base para a simulação [58])	43
5.2	Visualização do funcionamento do DDFC no PyViz	45
5.3	Topologia considerada na explicação do SCCS [26]	46
5.4	Relação do número de líderes, de agrupamentos formados e de nós solidários	48
5.5	Comportamento dos agrupamentos no DDFC-K	49
5.6	Comportamento dos agrupamentos no SCCS	49
5.7	Quantidade de nós vs. distância até o líder mais próximo	50
5.8	Duração dos líderes em número de turnos de 10s	51
5.9	Duração dos líderes para a variação DDFC-K	52
5.10	Duração dos líderes para o SCCS	53
5.11	Similaridade de leituras dentro do mesmo agrupamento	54
5.12	Similaridade de leituras na vertente DDFC-K	55
5.13	Similaridade de leituras no SCCS	55
5.14	Inconsistência das rotas através do tempo	56
5.15	Inconsistência acumulada das rotas	57
5.16	Inconsistência das rotas para a variante DDFC-K	57
5.17	Inconsistência das rotas para o SCCS	58
5.18	Mensagens <i>beacon</i> enviadas através do tempo	59

5.19	Resumo normalizado das melhores combinações de parâmetros	61
------	---	----

LISTA DE ABREVIATURAS E SIGLAS

ADV	ADVertise
AMP	AMPlitude
AMRP	Average Minimum Reachability Power
CE	Cluster Extension
CF	Cluster Forward
CH	Cluster Head
CPS	Cyber-Physical System
DCRR	Dynamic-Clustering Reactive Routing
DDFC	Dynamic Data-aware Firefly Clustering
DKLR	Distributed Kernel Linear Regression
DRHMM	Data Representation-based Hidden Markov Model
GW	GateWay
HEED	Hybrid, Energy-Efficient, Distributed clustering
IR	Invalid Routes
KHOPCA	K-HOP Clustering Algorithm
LEACH	Low-Energy Adaptive Clustering Hierarchy
LTC	Lightweight Temporal Compression
MIDDUR	MID DURation
QOSB	Quantity Of Sent Beacons
RFA	Reachback Firefly Algorithm
RSSI	Received Signal Strength Indicator
RSSF	Rede de Sensores Sem Fio
SCCS	Spatiotemporal Clustering and Compressing Schemes
SNR	Signal-to-Noise Ratio
SPRP	Shortest Path Routing Protocol
TTL	Time To Live

NOTAÇÃO

<i>ADDR</i>	Endereço do nó
<i>adap</i>	Parâmetro que indica funcionamento adaptativo
<i>aR</i>	Leitura agregada no nó armazenada na estrutura <i>NeighR</i>
<i>CThresh</i>	Parâmetro correspondente ao <i>threshold</i> de similaridade de dados
<i>int</i>	Parâmetro que indica a duração entre os intervalos de atuação do Controlador de <i>Beacons</i>
<i>iR</i>	Leitura individual do nó armazenada na estrutura <i>NeighR</i>
<i>MaxK</i>	Parâmetro de distância máxima até o líder mais próximo, tal que distância máxima corresponde à $MaxK + 1$
<i>NeighR</i>	Estrutura que armazena informações sobre as leituras dos nós vizinhos
<i>nR</i>	Número de leituras agregadas no <i>aR</i> correspondente
<i>pts(n)</i>	Pontuação do nó <i>n</i> para o Unidade de Indexação
<i>SNeigh</i>	Estrutura de nós vizinhos com leituras similares
<i>src</i>	Nó origem de uma mensagem

Capítulo 1

INTRODUÇÃO

O mundo no qual vivemos pode ser visto computacionalmente como um conjunto de variáveis e parâmetros valorados (dados), em cima do qual existem várias relações. Entretanto, embora seja inquestionavelmente grande o número de variáveis presentes e complexas as relações entre elas, dadas as suas naturezas muitas vezes dinâmicas, os modelos para a interpretação destes dados são cada vez mais comuns [1]. Esses modelos tentam desconsiderar os dados que são irrelevantes para o seu propósito. Existem, por exemplo, modelos que podem inferir a temperatura, a direção de vento e a umidade em previsões climáticas para dias futuros através de dados meteorológicos [2] e, até mesmo, modelos capazes de inferir a propagação do calor em determinados cenários hipotéticos [3], tomando em conta a estrutura química dos objetos que os compõem.

Atualmente, alguns tipos de coleta de dados são realizados no meio urbano. Já é comum o uso de sensores e radares para a detecção de velocidade de veículos, ou mesmo para saber se eles estão indevidamente parados sobre faixas de pedestres. Contudo, tais aplicações ainda são primitivas por usufruírem apenas da interpretação singular e individual dos dados, sem nenhum estabelecimento de relação entre eles. Os tipos mais comuns de relações entre dados que devem ser consideradas são as relações espaciais e temporais [4, 5]. Para diversas grandezas naturais, como a temperatura, a umidade e a luz, tais medições têm tendência de serem parecidas quando tomadas em regiões próximas, devido à sua **relação espacial**. Ademais, as medições sucessivas em uma localidade individual têm tendência de variar gradualmente, possuindo deste modo **relação temporal**.

Quando se passa a explorar e analisar a relação entre os dados de maneira coletiva, as possibilidades de aplicações mais robustas surgem. Em um cenário urbano, por exemplo, é possível verificar o tráfego nas ruas para se determinar rotas ótimas, ou mesmo analisar padrões espaciais de temperatura a fim de localizar ilhas de calor, tomando medidas que melhorem a qualidade de vida [6]. Através das leituras de luz durante a noite, poderia-se avaliar o perigo que determinadas ruas apresentam. Através de leituras de som, poderia-se determinar o nível de poluição sonora em dadas regiões ou mesmo determinar como é a propagação do som no meio.

Embora as Redes de Sensores Sem Fio (RSSF) sejam uma solução existente há anos, seu uso ainda não alcançou o seu potencial máximo [7]. Com a evolução tecnológica, os componentes de *hardware* utilizados por dispositivos de sensores sem fio se tornaram mais acessíveis, baratos e, até mesmo, computacionalmente capazes. A principal função de uma RSSF é realizar a interface de comunicação entre o meio físico e o meio computacional,

formado abstratamente por conjuntos de dados [8]. Essa interface é essencial em qualquer sistema ou mecanismo de automação de tarefas, consolidando a base para o funcionamento dos, cada vez mais comuns, *Cyber-Physical Systems* (CPSs) [9]. Ainda mais, o advento da Internet do Futuro [10] e da Internet das Coisas [11] demonstra que tais redes desempenham um papel importante na infraestrutura, sendo incorporadas para consolidar uma base responsável pela extração de diversos tipos de dados do meio físico [12].

Os protocolos de agrupamento baseados na similaridade possibilitam vantagens para diversos tipos de aplicação. Ambientes com leituras de dados dinâmicas em geral podem se beneficiar destes agrupamentos lógicos, apoiando tarefas em sistemas distribuídos com eleições de líderes [13], tal como observado em áreas importantes para a economia, segurança, e conservação meio ambiente, dentre outras. Podem-se destacar, por exemplo, aplicações decorrentes da análise de dados escalares, como irrigação agrícola no setor **econômico**; monitoramento sísmico para detecção de terremotos ou erupções de vulcões na área de **segurança**; monitoramento de poluição no ar e da água na área de **conservação de meio ambiente**; assim como detecção de incêndios, auxiliando tanto na área de segurança quanto em conservação ambiental.

Responsáveis por estimados 70% do uso de água no mundo inteiro [14], as aplicações agrícola podem se beneficiar de aplicações de agrupamentos de dados baseados na similaridade de leituras para identificar áreas de desperdício através da umidade, ajustando assim o consumo. Com medições de movimentações sísmicas, é possível o estabelecimento de padrões de agrupamentos lógicos capazes de inferir movimentos prévios à erupções com até algumas semanas de antecedência [15]. Através de relações de similaridade nas medições de poluição, é possível a detecção precisa de ilhas de calor [6], assim como estimar a qualidade de água, inferindo possíveis áreas de contaminação e de emissão e poluentes, através das relações de similaridade. Para a detecção de incêndios, podem-se medir e estabelecer padrões de similaridade e contraste entre leituras de temperatura, umidade, e direção do vento, auxiliando tanto em medidas preventivas em áreas de risco de incêndio quanto na detecção efetiva.

Uma técnica muito utilizada para a organização lógica dos dispositivos (nós) nas RS-SFs é o agrupamento lógico destes nós [16]. Essas formações são chamadas de *clusters*, correspondendo a grupos lógicos que normalmente estabelecem uma hierarquia entre os nós. Os agrupamentos são atualmente muito utilizados por possibilitar a agregação de dados [5] e também diminuir o número de mensagens transmitidas na rede, otimizando a eficiência em energia e diminuindo o número de colisões entre as mensagens [17]. Entretanto, estes agrupamentos podem também estabelecer a organização lógica considerando a similaridade dos dados e mantendo nós de leituras similares juntos. Tal organização traria vantagens como a possibilidade de agregar os dados de maneira mais eficiente, devido à correlação entre determinados tipos de dados que resulta na similaridade entre leituras tomadas em intervalos discretos, e à consequente menor perda de informações

na agregação, além de possibilitar a detecção mais robusta de eventos anômalos, decorrente da maior facilidade de identificar e comparar variações entre patamares de leituras próximas [18, 19].

1.1 Motivação

Os protocolos de agrupamento que mantêm agrupados os nós sensores com leituras de dados similares oferecem benefícios em diversas camadas da rede. Na camada de aplicação, é possível evitar a redundância de leituras de dados reportadas à estação-base. Nas camadas de aplicação e de transporte, é possível realizar uma melhor compressão de dados devido à similaridade entre os dados. Na camada de roteamento, é possível utilizar técnicas de inundação limitadas por uma função de similaridade de dados. Além disso, agrupamentos dinâmicos possibilitam a operação mais eficiente de redes que reportam eventos anômalos, devido à maior facilidade e rapidez com que eles podem ser detectados.

Entretanto, poucas pesquisas vêm sido desenvolvidas acerca de protocolos de agrupamento que tratem simultaneamente da correlação e da dinamicidade dos dados em abordagens reativas até o momento. Os protocolos de agrupamento têm sido propostos com diversos objetivos: alguns visam **se adequar à dinamicidade da mobilidade** [20, 21], outros tentam por si só **serem dinâmicos ao rotacionar líderes** [22, 23] ou mesmo **recriar formações de agrupamentos por completo** [24], mas são poucos os que **tratam da similaridade espacial dos dados** [4, 19, 25] e, menos ainda, os que **dão suporte à natureza dinâmica dos dados** [26].

Desta forma, os atuais protocolos são inadequados para operar de maneira dinâmica quanto à similaridade dos dados lidos. Embora já sejam considerados pela literatura diversos tipos de dinamicidade na elaboração de protocolos de agrupamento, aqueles que tomam a similaridade espacial como critério de agrupamento não operam adequadamente quanto à natureza dinâmica que os dados lidos podem apresentar, devido à maneira que os agrupamentos são organizados e mantidos: nos casos que se utilizam da recriação de agrupamentos, uma estação-base normalmente desempenha o papel de coordenação, centralizando também a função de coletar os dados dos sensores e, através destes dados, determinar os agrupamentos [19, 25]; outros mecanismos utilizam técnicas de inundações constantes na rede para estabelecer assim os agrupamentos [4]. Ambas as políticas não são escaláveis e acarretam latência e custos de comunicação elevados. Assim, uma maneira distribuída simples para a atividade complexa de manutenção de agrupamentos com leituras similares se faz necessária.

São encontrados na natureza diversos exemplos de operações complexas que são realizadas de maneira dinâmica e distribuída por um grande número de entidades que podem executar apenas um pequeno conjunto de tarefas simples [27]. Dentre eles, algumas espécies de vagalumes encontrados na natureza são conhecidas pela sua capacidade de piscar

seus emissores de luz com sincronia em grandes grupos, ou seja, eles separadamente convergem sobre o momento no qual devem piscar através de um ajuste gradual e distribuído. Desta forma, eles já foram inspiração para diversos trabalhos de sincronização de relógio na área de RSSFs [28, 29]. Embora os vagalumes tenham sido inspiração para muitos mecanismos de sincronização de relógio, eles não são comumente utilizados para operações diferenciadas, apresentando portanto um potencial inexplorado em operações que sincronizam parâmetros atemporais.

1.2 Objetivos

Este trabalho tem como objetivo desenvolver uma solução de formação e manutenção de agrupamentos dinâmicos que mantém nós agrupados de acordo com a similaridade da leitura de dados. Tal operação há de ser realizada de maneira distribuída e sem coordenação por parte da estação-base ou constantes recriações de agrupamentos. Enquanto existem trabalhos tratando da similaridade de dados com foco exclusivo na questão de indexação para consultas [30, 31], o protocolo a ser desenvolvido almeja criar meios *adaptativos e dinâmicos* de indexação local, considerando sempre as relações de similaridade. Em nível de aplicação, a solução tem a finalidade de atender requisitos de similaridade por parte de aplicações que dependem da fácil identificação de leituras similares.

Assim, o protocolo de agrupamento proposto possui um mecanismo que mantém sincronizada a agregação média das leituras nos agrupamento dos nós, permitindo que eles determinem facilmente quando devem se juntar ou deixar agrupamentos. Para a indexação intra-agrupamento, um sistema de regras adaptativo converge dinamicamente para o estabelecimento de rotas válidas. Tais operações são obtidas de maneira distribuída, não necessitando de coordenação na estação-base ou de operações de reestruturação completa na rede.

1.3 Contribuições

As contribuições deste trabalho são as seguintes:

- Um estudo sobre os protocolos existentes na literatura que tratam de questões de similaridade na RSSF. Estas questões foram classificadas em grupos, compreendendo diferentes tipos de similaridade, incluindo aquelas focadas na variação dinâmica dos dados. Com tal estudo, foram levantados os requisitos desejáveis para um protocolo de agrupamento adaptativo quanto à dinamicidade das leituras dos nós;
- A especificação e implementação de um protocolo de agrupamento adaptativo quanto à dinamicidade das leituras realizadas pelos nós. Tal protocolo mantém

agrupados nós que possuam leituras similares de acordo com uma função de similaridade preestabelecida, que compreende o método para determinar se um par de leituras é similar ou não, sem a necessidade de nenhuma operação centralizada em uma estação-base e sem completas reestruturações da hierarquia global dos agrupamentos;

- A especificação do protocolo apresentado consolida um uso diferente das técnicas inspiradas em vagalumes, empregadas neste trabalho pela primeira vez no escopo da sincronização de medidas atemporais. Assim, os princípios biológicos dos vagalumes são explorados pelo protocolo para sincronizar a agregação média das leituras dos nós nos agrupamentos;
- Uma avaliação do protocolo proposto, no simulador NS3, em um cenário com leituras de dados reais, obtidas pelo laboratório Intel, de Berkeley. Foi também implementada uma extensão do visualizador gráfico do simulador para a avaliação visual do protocolo. Por fim, foi feita uma rigorosa análise dos resultados, estabelecendo padrões entre os parâmetros de funcionamento e as métricas avaliadas.

1.4 Estrutura da dissertação

Esta dissertação está organizada em cinco capítulos. O Capítulo 2 apresenta os fundamentos necessários no contexto da proposta, descrevendo as características gerais das RSSFs e a técnica de agrupamento lógico de nós. Esse capítulo também apresenta o sistema bio-inspirado em vagalumes, capaz de atender a dinamicidade necessária para o agrupamento adaptativo dos nós com leituras similares. O Capítulo 3 apresenta diversos trabalhos da literatura que tratam de questões de dinamicidade no agrupamento dos nós, com o foco na dinamicidade das leituras dos nós. O Capítulo 4 detalha o protocolo bio-inspirado proposto, que mantém agrupamentos lógicos de nós com leituras de dados similares. O Capítulo 5 apresenta a análise de desempenho realizada, avaliando um cenário composto de leituras de dados reais através de simulação, comparando a proposta da dissertação, uma variante dela, e outro protocolo de roteamento. Por fim, o Capítulo 6 conclui a dissertação, apresentando as considerações finais e trabalhos futuros.

Capítulo 2

FUNDAMENTOS

Este capítulo apresenta os fundamentos necessários para o entendimento do contexto do problema e da solução desenvolvida na dissertação. A Seção 2.1 descreve as características gerais das RSSFs, assim como sua presença no meio urbano. A Seção 2.2 introduz os conceitos e as abordagens mais comuns na formação de agrupamentos, mostrando as vantagens deste tipo de técnica. A Seção 2.3 apresenta as características dos sistemas bioinspirados, bem como o conceito e o funcionamento básico da sincronização por vagalumes.

2.1 Redes de sensores sem fio

As Redes de Sensores Sem Fio (RSSFs) compreendem uma tecnologia emergente que está em evolução contínua [7, 32, 33]. Os avanços recentes nos sistemas microeletromecânicos e nas tecnologias de comunicação sem fio permitiram a criação de dispositivos (nós) cada vez mais baratos, menores e capazes, equipados com sensores, microprocessador, memória e rádio. Uma RSSF consiste tipicamente de uma quantidade de dezenas até milhares destes dispositivos, espalhados em uma região de interesse. Estes nós tem a capacidade de coleta de leituras de dados de um ambiente, de processamento, de armazenamento e de transmissão pelo meio sem fio.

Os nós que compõem as RSSFs são *normalmente* caracterizados pelo seu tempo de vida limitado, devido à sua energia finita, provida de baterias. Ademais, estas redes são comumente utilizadas em regiões de difícil acesso, como em florestas e montanhas, ou em cenários de risco, como próximo a vulcões. Desta forma, as aplicações que operam nas RSSFs *normalmente* tem suas atividades de processamento limitadas, tentando otimizar o consumo de energia a fim de prolongar o tempo de vida da rede. É difícil, senão muitas vezes inviável, realizar a troca da bateria dos nós, uma vez eles depositados na região de monitoramento.

Pela sua capacidade de realizar leituras dinâmicas do meio físico, sendo uma interface para o meio computacional, as RSSFs são fundamentais para, por exemplo, o progresso da Internet das Coisas (*Internet of Things*) [11] e dos CPSs (*Cyber-Physical Systems*) [9]. Elas oferecem a capacidade de agir autonomamente de acordo com o estado atual do meio, representado pela leitura de diversos parâmetros do meio físico [12], como, nos casos mais comuns, temperatura, luz, umidade, tremores sísmicos, ruído sonoro e, até mesmo, áudio e vídeo. A partir das medições destes dados, inúmeros tipos de aplicações se tornam possíveis. É normal o uso destas redes para realizar tarefas desde o monitoramento

ambiental e detecção de atividades vulcânicas ou terremotos até mesmo a automação de tarefas e monitoramento médico de pacientes [32].

2.1.1 RSSFs no meio urbano

No meio urbano, os nós sensores são muito utilizados no sistema de trânsito. Eles são empregados para medir a velocidade de veículos ou mesmo verificar se eles não se encontram em lugares indevidos, como quando parados sobre faixas de pedestres e, em caso de infração, capturar as imagens. Entretanto, trata-se de um uso trivial dessas redes, visto que tais dados são analisados separadamente, sem o estabelecimento de nenhum tipo de relação.

Em um cenário urbano mais elaborado, estes sensores formam uma rede se comunicando entre si para a realização de atividades mais elaboradas. No projeto CitySense [6], existem diversos sensores espalhados pela cidade, como visto na Figura 2.1, realizando medições de temperatura, umidade, velocidade e direção do vento. Com estes dados, diversas aplicações podem ser concebidas. As medições de temperatura e de umidade podem ser analisadas no meio para detectar ilhas de calor de maneira direta [34]. Desta maneira, estes dados podem ser cruzados com as medições de vento, a fim de determinar se as ilhas de calor detectadas são devido a padrões de vento, como visto em [35] ou devido a causas não tão naturais. Assim, meios de melhorar a qualidade de vida da população podem ser obtidos.

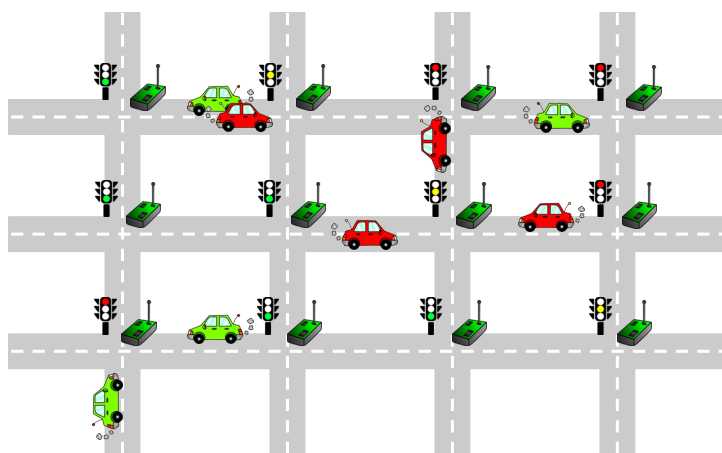


Figura 2.1: Nós sensores integrados no meio urbano

O conceito de redes veiculares vem crescendo no meio urbano. A utilização das RSSFs como base de infraestrutura auxilia estes sistemas veiculares inteligentes a cumprir suas tarefas [36]. Através da interface provida por sensores, meios de transporte público podem ser otimizados com a finalidade de reduzir a espera por veículos e aumentar o aproveitamento das linhas. Além disso, as informações coletadas podem ser utilizadas para evitar ou mitigar condições de congestionamento de veículos ao prover rotas ótimas não apenas

em questão de distância percorrida. Em situações de emergência, essa informação é de grande importância, seja para realizar um resgate rápido de uma pessoa ou para evacuar uma grande região de maneira coordenada, minimizando o afunilamento das rotas.

Em aplicações de monitoramento ambiental a distribuição dos nós pode ocorrer de maneira aleatória devido à dificuldade de posicioná-los, o que, ainda assim, é raro [37]. Um diferencial importante de RSSFs aplicadas neste meio é que os nós não são distribuídos de maneira aleatória devido à inevitável organização do meio urbano. Em aplicações urbanas reais, os nós são estrategicamente distribuídos, a fim de monitorar adequadamente o ambiente de interesse e otimizar parâmetros relevantes ao objetivo da rede, como cobertura ou qualidade da informação coletada.

Para reduzir o impacto da limitação de energia presente nas RSSFs, é comum o uso de técnicas que aproveitam o meio para obter energia solar, mecânica ou térmica [38]. Embora estas técnicas prolonguem o tempo de vida da rede e se mostrem promissoras, elas ainda não são suficientes para eliminar completamente a necessidade de troca de baterias ou garantir operação ininterrupta. Por outro lado, considerando também o posicionamento planejado, o meio urbano permite o uso da energia dos próprios postes de luz para alimentar os nós sensores [30], possibilitando assim operações de funcionamento mais complexo na rede, como o próprio CitySense, que retira energia de postes.

Além disso, é comum o uso de diversas técnicas que visam garantir *eficiência de energia*, quando finita, aumentar a *escalabilidade da rede*, ou mesmo *organizar o tráfego* originado pelos nós a fim de *evitar colisões e congestionamentos*. Uma técnica comum que trata destes problemas é o agrupamento lógico dos nós. Com ele, é estabelecida uma hierarquia que traz vantagens combinadas de abordagens distribuídas e centralizadas.

2.2 Agrupamentos de nós

Uma técnica muito utilizada para a organização dos nós nas RSSFs é o uso de agrupamentos lógicos [16, 17], ilustrados na Figura 2.2, onde as linhas contínuas indicam as ligações aos líderes dos agrupamentos. Essas formações correspondem a grupos lógicos que estabelecem uma hierarquia entre os nós e são atualmente muito utilizadas por possibilitar a agregação de dados e também diminuir o número de mensagens transmitidas na rede [5]. A Subseção 2.2.1 enumera e explica alguns parâmetros que podem ser considerados por esta operação. A Subseção 2.2.2 apresenta os benefícios gerais do uso de técnicas de agrupamento e, em seguida, a Subseção 2.2.3 apresenta conceitos gerais de similaridade de dados comumente empregados pelos protocolos que a tratam.

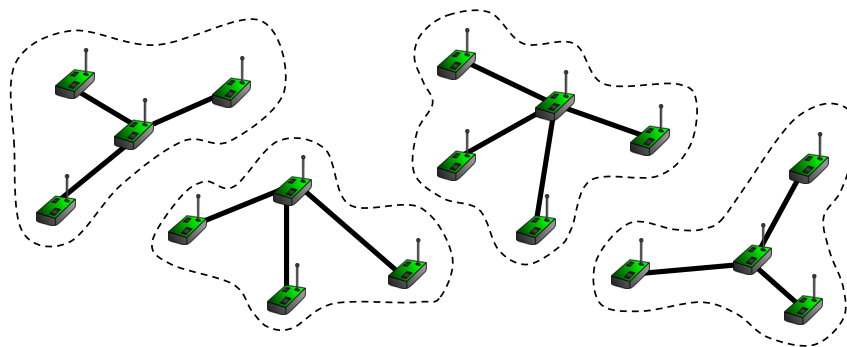


Figura 2.2: Agrupamento lógico de nós

2.2.1 Critérios para agrupamento lógico

Na fase inicial da rede, não existe diferenciação lógica entre os nós que a compõem. Desta maneira, a operação de agrupamento mais comum define na rede inicialmente aqueles nós que serão os líderes. Isto pode ser feito de maneira **centralizada em uma estação-base** que possua informação sobre a topologia e a energia dos nós, a fim de otimizar a operação da rede ou o seu tempo de vida, ou de maneira **distribuída**. Na abordagem distribuída, é realizada uma operação de seleção de líderes que normalmente se baseia em parâmetros como **quantidade ou porcentagem de líderes** desejados na rede [22] ou **energia residual** dos nós [23].

Os agrupamentos podem ser formados com base em diversos parâmetros, apresentando diferentes vantagens e desvantagens. Após a definição dos líderes, deve-se definir os agrupamentos em si. Nesta fase, na maioria dos casos, os nós comuns são responsáveis por selecionar um daqueles nós eleitos como líder para se juntar ao seu agrupamento. Existem algumas abordagens comuns para o agrupamento dos nós [39]. O critério mais comumente utilizado para determinar um líder adequado é a **distância dos nós até o líder**, que deve ser reduzida a fim de tentar prover eficiência em energia e qualidade de sinal. Entretanto, dado que a distância física até o líder não necessariamente indica boa qualidade de comunicação, alguns mecanismos usam para este fim o **RSSI** (*Received Signal Strength Indicator*) [22] ou o **SNR** (*Signal-to-Noise Ratio*) [40], que indicam de maneira direta a qualidade da comunicação entre os nós.

Existem também abordagens que utilizam um campo de **TTL** (*Time To Live*) [41], nas quais os líderes enviam mensagens de anúncio que podem se propagar por um número limitado de saltos. Esta abordagem também considera a distância ao líder, entretanto, em número discreto de saltos. Em caso de empate, outros critérios podem ser aplicados, como a **energia dos líderes**, tal que líderes com maior energia poderão suportar mais nós em seu agrupamento. Alguns mecanismos até mesmo utilizam apenas este atributo para a formação de agrupamentos, de maneira independente. Entretanto, por ser apenas um indicador inicial e que varia muito, ele é utilizado mais eficientemente em conjunto

com outros.

Embora estes parâmetros apresentados explorem questões de eficiência em custos de energia e qualidade de comunicação, os agrupamentos baseados neles nem sempre garantem o melhor funcionamento quanto ao processamento e a questões de similaridade na leitura de dados. Desta forma, existem também abordagens que utilizam a **leitura dos nós** como critério de agrupamento, tentando manter aqueles nós com leituras próximas no mesmo agrupamento. Existem tanto abordagens que exploram a similaridade em conjunto com a proximidade dos nós [26] como também abordagens que criam agrupamentos virtuais de nós não necessariamente vizinhos [42]. Enquanto a segunda abordagem garante maior facilidade de consultas globais, a primeira abordagem garante menor *overhead* e latência, além do proveito explícito das características de similaridade espacial de dados.

2.2.2 Vantagens de agrupar os sensores

Ao agrupar conjuntos de nós, um ou mais nós líderes representam cada agrupamento. Desta forma, os dados que devem ser enviados pelos nós comuns dentro deste agrupamento são inicialmente repassados para um líder que, então, realiza a retransmissão da maneira adequada. No roteamento até o nó destino, apenas líderes realizam o encaminhamento dos dados, como visto na Figura 2.3, onde as retas pontilhadas indicam as ligações dos nós comuns ao líder do agrupamento e as flechas indicam o fluxo de roteamento entre os líderes. Esta hierarquia de agrupamento pode possuir diversos níveis de líderes, possibilitando uma maior escalabilidade da rede, dado que as informações de roteamento na rede como um todo serão mantidas apenas nos líderes, que também manterão as informações locais de seu agrupamento.

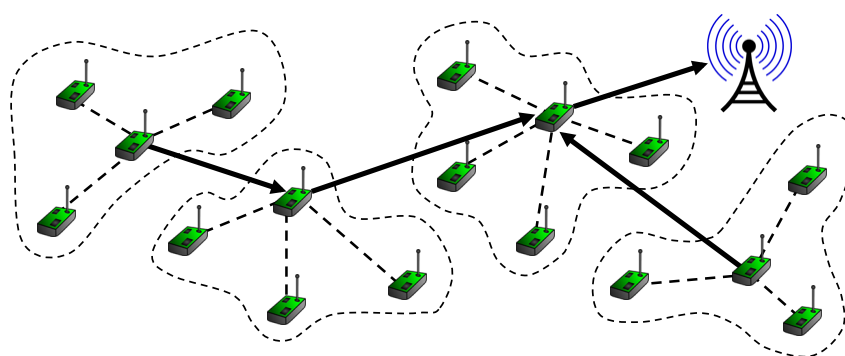


Figura 2.3: Roteamento de dados através dos líderes até a estação-base

O uso de agrupamentos introduz uma região de centralização parcial - se trata de uma rede de entidades distribuídas, na qual os líderes tem mais autoridade e responsabilidade do que os demais nós do grupo. Embora a centralização em uma RSSF se mostre inadequada em diversos quesitos [43], uma abordagem híbrida com centralização parcial possui granularidade em um nível que combina as vantagens do processamento centrali-

zado à natureza idealmente descentralizada da rede. Os diversos agrupamentos operando distribuídos entre si, mas possuindo pontos centralizados internamente nos líderes, possibilitam as seguintes vantagens:

- **Eliminação de redundância:** com alguns pontos centralizados responsáveis por receber e retransmitir os dados, eles podem verificar proativamente se os dados a serem transmitidos não possuem redundâncias. Um caso típico é a eliminação da sobreposição de imagens obtidas por diferentes câmeras em uma mesma região em aplicações multimídia [44];
- **Compressão dos dados:** analogamente à eliminação de redundâncias, um ponto centralizado pode aplicar algoritmos de agrupamento ou compressão de dados que não dependam de redundâncias, embora se aproveitem da similaridade dos dados [45]. Todas as aplicações típicas que realizam monitoramento de dados escalares podem se beneficiar;
- **Organização do tráfego da rede:** considerando as vantagens anteriores, menos fluxos de dados existem no nível mais alto da hierarquia, dado que os líderes sempre serão os responsáveis pelas transmissões. Desta forma, existirão menos colisões de mensagens e interferência no meio, possibilitando também um melhor proveito dos canais disponíveis;
- **Economia de energia:** através da redução do volume do tráfego e a consequente redução no número de colisões e retransmissões de mensagens, o uso da energia é otimizado, garantindo maior tempo de vida à rede, quando a energia é restrita;
- **Facilidade do estudo de histórico:** com menos nós redirecionando o tráfego da rede no roteamento inter-agrupamentos, que ocorre apenas entre os nós líderes, é mais fácil realizar análises do histórico de transmissões passadas.

Além disso, protocolos que mantêm os nós com leituras de dados similares agrupados proporcionariam benefícios em diversas camadas. Quando os nós não são agrupados desta maneira, mesmo que dados redundantes sejam originados em uma região, é possível em determinados cenários, como no ilustrado na Figura 2.4, que as leituras similares não tenham suas redundâncias eliminadas eficientemente e, assim, a qualidade dos dados reportados à estação base é inferior. Desta forma, a camada de aplicação depende do agrupamento de nós com leituras de dados similares para tratar eficientemente das redundâncias e garantir ainda assim dados precisos [46].

Nas camadas de aplicação e de transporte, a compressão de dados seria mais eficiente devido à similaridade entre eles. Quando são considerados dados similares, várias técnicas de compressão [47], como LTC (*Lightweight Temporal Compression*), DRHMM

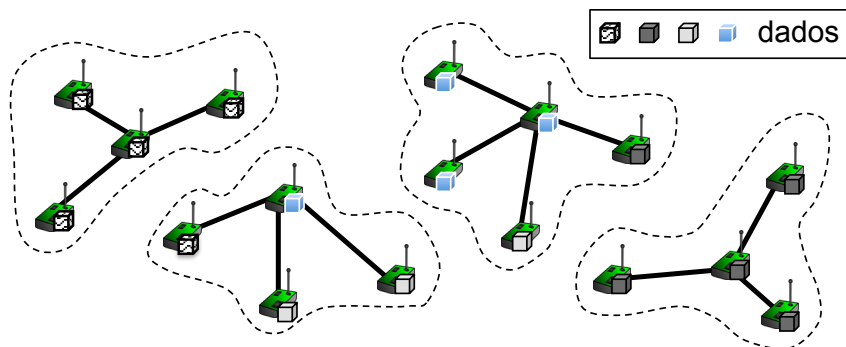


Figura 2.4: Nós de leituras similares agrupados de maneira fragmentada

(*Data Representation-based Hidden Markov Model*), DKLR (*Distributed Kernel Linear Regression*), apresentam resultados melhores em questão de tamanho dos dados comprimidos e de qualidade de compressão, no caso de compressões com perda. Assim, a camada de transporte pode também utilizar mecanismos de compressão adaptativa para o controle de congestionamento, balanceando a fidelidade do evento reportado e a quantidade de dados transmitidos.

Uma técnica muito conhecida na camada de roteamento é a disseminação de dados através de inundações [33, 48]. Contudo, esta técnica por si só é normalmente evitada, pois sobrecarrega a rede com a troca de muitas mensagens, causando colisões, latência, e consumindo muita energia. As técnicas de inundação foram aperfeiçoadas ao tentar controlar o escopo das retransmissões, seja por coordenadas geográficas ou por número máximo de retransmissões [48]. Tomando a similaridade de dados em conta, a camada de roteamento poderia utilizar também inundações controladas de acordo com a função de similaridade de dados.

Por fim, existem diversas características do ambiente de monitoramento e das próprias redes sem fio que são dinâmicas, tais como questões de topologia [33] ou mesmo de leituras de dados [26]. Levando a dinamicidade na leitura dos dados em conta, os agrupamentos dinâmicos possibilitam a operação mais eficiente de redes que devem reportar eventos anômalos. Por possuírem nós de leituras similares agrupados, é fácil detectar quando um nó tem suas leituras variando além de patamares aceitáveis e é possível comparar facilmente o contraste das leituras de agrupamentos vizinhos. Assim, a implementação de possíveis políticas de detecção na camada de aplicação, acima da tarefa de agrupamento, se torna mais robusta e mais responsiva, visto que os nós de leituras similares já se encontram organizados em uma estrutura lógica.

Todas estas vantagens provém diretamente da maneira que os agrupamentos se adaptam para que a *similaridade* dos dados dinâmicos internos a eles seja respeitada. Contudo, o conceito de similaridade é amplo e pode variar bastante de acordo com a aplicação empregada da rede. Desta forma, é necessária a definição formal de alguns tipos de similaridade de dados que podem ser levados em consideração pelos protocolos de agrupamento.

2.2.3 Similaridade de dados

A literatura na área de RSSFs tem diversas abordagens para definir a *similaridade de dados*, que depende não só do escopo tratado, mas, principalmente, do tipo dos dados lidos e da aplicação. No caso de aplicações multimídia que envolvam fluxos de vídeo [49], a similaridade comumente é associada à (a) áreas sobrepostas em quadros de imagem diferentes. Após a identificação de sobreposições de quadros, estas áreas podem ser facilmente comprimidas ou mesmo parcialmente eliminadas.

No caso de dados escalares, funções de similaridade simples podem ser utilizadas, por se tratarem de manipulações numéricas explícitas. Considerando uma similaridade atemporal, isto é, que se dá em um instante discreto e não em um período de tempo contínuo, funções comuns normalmente envolvem (b) limites L de diferença absoluta entre leituras a e b , de tal forma que $|a - b| < L$ é satisfeito; (c) limites Q de diferença porcentual entre leituras, tal que uma leitura b é dita similar à a se $|a - b| < Q * a$; (d) limites customizados de intervalos de leituras predefinidos.

A Figura 2.5 ilustra estes conceitos de similaridade. A similaridade de um quadro multimídia (a) expressa a região que é comum à quadros obtidos por diferentes câmeras. As funções de diferença absoluta entre leituras (b) são adequadas para uso quando não existe necessidade de agrupamento de intervalos específicos de leitura e quando as leituras em si podem possuir valores numéricos pequenos. No caso de valores numéricos muito pequenos, funções de similaridade porcentual (c) agrupam as leituras de maneira desigual quanto ao tamanho dos conjuntos. As leituras maiores possuirão mais leituras similares, pela tolerância porcentual ser maior, gerando um maior intervalo, como visto na figura. Finalmente, intervalos customizados (d) são normalmente utilizados quando já se tem determinados grupos de interesse distintos preestabelecidos.

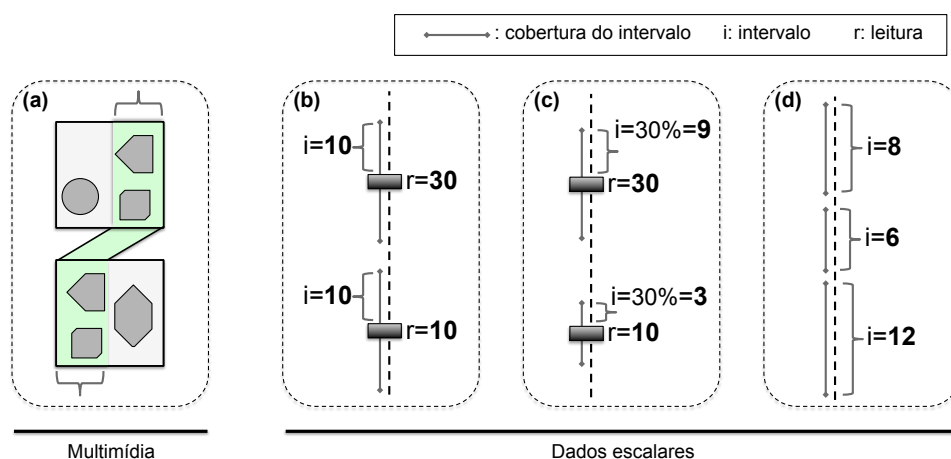


Figura 2.5: Exemplos de definições de similaridade de dados

Embora existam estas alternativas comuns, a função a ser utilizada para determinar a similaridade dos dados é muito dependente do propósito da aplicação e dos dados

lidos. Desta forma, para o tratamento adequado dos dados, os protocolos que consideram dados similares devem idealmente também ser capazes de adaptar seu funcionamento de acordo com a função de similaridade necessária, permitindo a fácil alteração da função de similaridade de dados sem impacto no funcionamento do protocolo como um todo, que deve ser capaz de atender aos requisitos dinâmicos da rede.

Correlação de dados

É importante notar que a correlação pode ser vista como a tendência de determinados dados ou tipos de dados serem similares quando comparados [18]. No caso espacial e temporal, quando dados são tomados em regiões próximas ou em tempos próximos, respectivamente. Por outro lado, embora a correlação influencie na probabilidade de similaridade, no escopo desta dissertação o conceito de similaridade compreende uma comparação atômica, independente de correlações, como nas funções apresentadas acima. Desta forma, tipos de dados sem correlação podem apresentar similaridade em dados instantes, assim como tipos de dados com correlação podem não apresentar similaridade - o que pode, inclusive, indicar um evento anômalo ou de interesse.

2.3 Sistemas bio-inspirados

As RSSFs demandam aspectos como auto-configuração, resiliência a falhas, escalabilidade, heterogeneidade, funcionamento distribuído, de modo que seu funcionamento seja garantido. Todas essas características são encontradas de maneira elegante e eficiente em sistemas da natureza. A grande dinamicidade que alguns sistemas biológicos apresentam é baseada em conjuntos pequenos de leis e regras, determinando comportamentos colaborativos que resultam em gerenciamento de recursos, alocação de tarefas, diferenciação social, sincronização, sem a necessidade de entidades externas realizando o controle [27].

Os vagalumes se enquadram no princípio biológico que faz parte da classe de *sistemas bio-inspirados* [27], sendo que algumas espécies são capazes de sincronizar o modo que piscam em conjunto de maneira distribuída. Em [50], Mirollo e Strogatz realizaram um estudo do piscar de vagalumes, modelando este comportamento através de um sistema de osciladores de pulso-acoplado. Isto é, eles assumem que cada vagalume possui osciladores que se ajustam, incrementando ou decrementando os seus valores em reação ao pulsar dos osciladores de demais vagalumes. Com uma abordagem simples de convergência, o relógio do piscar dos vagalumes é gradualmente sincronizado, como ilustrado na Figura 2.6, que mostra a troca de mensagens e a conseqüente configuração sincronizada dos vagalumes, após um dado tempo.

A Figura 2.7 ilustra de maneira genérica o processo de sincronização entre os osciladores de dois vagalumes, denominados $V1$ e $V2$. Começando a pulsar depois, o vagalume

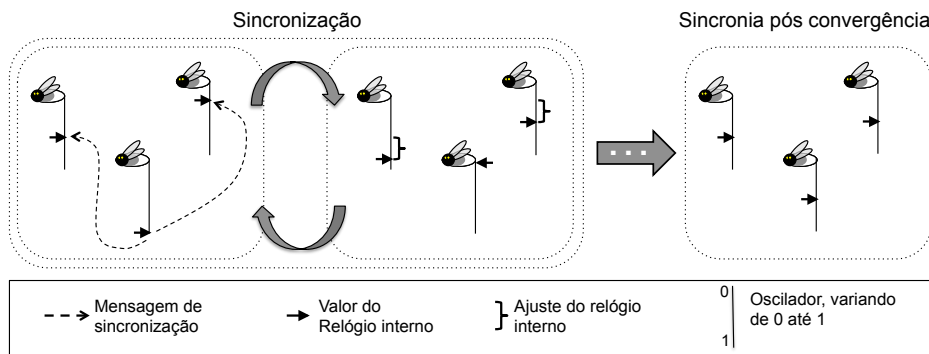


Figura 2.6: Troca de mensagens de vagalumes e sincronização de períodos

$V2$ tem o seu relógio atrasado em relação à $V1$. Desta forma, no instante $t1$, o pulsar de $V1$ leva $V2$ a adiantar o seu relógio. Análogamente, com o pulsar de $V2$ em $t2$, $V1$ atrasará o seu relógio. A mesma situação se repete nos instantes $t3$ e $t4$ até que em $t5$, os relógios são finalmente sincronizados.

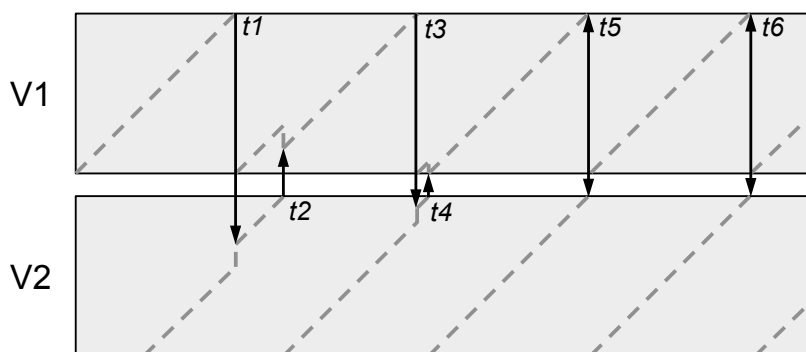


Figura 2.7: Par de osciladores de vagalumes sincronizando com o pulsar

Embora a sincronização dos relógios através de pulsos de osciladores opere de maneira aparentemente simples, esse funcionamento é resultante da modelagem do comportamento exclusivamente de vagalumes. Contudo, para seu uso em RSSFs, ainda existem características intrínsecas das RSSFs que não são tratadas de maneira direta. Tyrrell et al. [28] estudaram como os osciladores de vagalumes podem ser aplicados em redes adhoc sem fio. Eles mostraram que diversos tempos de latência devem ser considerados ou incorporados em sistemas sem fio para que os relógios sejam sincronizados de maneira eficiente. Entre esses tempos estão a latência de propagação, de transmissão, de decodificação e de refração:

- **latência de propagação** ($T_0^{i,j}$): tempo despendido para que uma mensagem seja enviada de uma dada origem i até um destino j , proporcional à distância entre tais nós;
- **latência de transmissão** (T_{T_x}): duração do envio da *palavra* de sincronização.

Embora no caso de vagalumes a mensagem enviada seja sempre igual, independente de origem, nas redes sem fio há a necessidade de diferenciação da mensagem para identificar a origem. Desta maneira, uma *palavra* de sincronização precisa ser estipulada, seja ela um padrão de pulsos ou um preâmbulo na mensagem, ambos demandando tempo para a transmissão;

- **latência de decodificação** (T_{dec}): após a mensagem recebida, há um dado tempo despendido para que ela tenha os cabeçalhos de cada camada decodificados, assim como o conteúdo da mensagem em si.
- **latência de refração** (T_{refr}): para uma maior estabilidade, um período de refração é adicionado após a transmissão de um pulso, durante o qual nenhuma alteração pode ser realizada no relógio do nó local.

Descartando o tempo de propagação e tomando o período do pulsar T do vagalume, um tempo de espera T_{wait} é proposto, de acordo com a Equação 2.1, para a transmissão da *palavra* de sincronização. Tal tempo de espera é necessário no meio sem fio para a melhor precisão da sincronia obtida entre os relógios dos nós. Após a transmissão de um pulso, nenhuma alteração de relógio pode ser realizada pelo período T_{refr} que considera a possibilidade das mensagens serem trocadas de maneira imprevisível devido ao tempo de propagação $T_0^{i,j}$, causando instabilidade nos relógios dos nós.

$$T_{wait} = T - (T_{Tx} + T_{dec}) \quad (2.1)$$

O mecanismo de sincronização RFA (*Reachback Firefly Algorithm*), proposto em [51], também considera as latências do meio sem fio. Além disso, ele utiliza uma abordagem na qual o vagalume espera o seu tempo de pulsar para realizar o ajuste do seu relógio, ao invés de realizar o ajuste assim que é detectado o pulso de um outro vagalume. Desta maneira, é possível a realização do incremento do relógio apenas uma vez com o valor acumulado e, assim, manter o comportamento do mecanismo mais estável. Essa operação é especialmente útil quando os vagalumes possuem muitos vizinhos e os pulsos recebidos ajustam o relógio alternadamente entre atrasos e adiantamentos, causando certa instabilidade. Ademais, tal mecanismo aplica latências aleatórias no envio de cada pulsar para evitar colisões no meio sem fio.

Assim como o RFA, diversos outros trabalhos tratam da sincronização sob aspecto exclusivamente temporal, seja ele considerando explicitamente o relógio interno de cada nó [28, 29] ou então focando na sincronização de operações baseadas em turnos [52, 53]. Entretanto, embora os trabalhos inspirados em vagalumes para RSSFs foquem na questão de sincronização temporal entre os nós e a questão da sincronia sob aspecto temporal seja mais evidente, é possível utilizar operação similar para manter os parâmetros de funcionamento do protocolo de agrupamento sincronizados. Em [54] foi mostrado que apesar

das sincronizações realizadas entre vizinhos espaciais serem em escopo regional, ocorre também a convergência global na sincronização. Desta forma, a técnica de sincronização inspirada em vagalumes se mostra apropriada para tratar a dinamicidade das leituras de dados dos nós na manutenção de agrupamentos como um todo.

2.4 Resumo

Este capítulo apresentou os fundamentos sobre as Redes de Sensores Sem Fio, mostrando também a sua aplicação no meio urbano. Foram também apresentados genericamente alguns critérios de similaridade utilizados por protocolos que consideram dados similares entre os nós. Por fim, os conceitos de princípios biológicos e dos vagalumes foram apresentados, indicando que os vagalumes mostram um bom potencial inexplorado na sincronização de parâmetros atemporais em RSSFs. Baseando-se nestes conceitos, o Capítulo 3 apresenta uma seleção de protocolos de agrupamento dinâmicos para RSSFs.

Capítulo 3

AGRUPAMENTOS DINÂMICOS

A natureza de uma RSSF é dinâmica de diversas maneiras. Por exemplo, aspectos como topologia, rotas e posicionamento de nós podem variar através do tempo. Após a inicialização da rede, os nós estão sujeitos a falhas, ataques e, comumente, falta de energia, podendo deixar de fazer parte da topologia da rede a qualquer instante. Além disso, o meio de comunicação sem fio está sujeito a colisões e interferências, significando que os nós podem às vezes não receber efetivamente as mensagens destinadas a eles devido às condições temporárias. A própria topologia da rede pode ser dinâmica em cenários com nós móveis e rotas dinâmicas de acordo com as condições e as posições dos nós. Além desses parâmetros convencionais, as leituras quantificadas das grandezas monitoradas pelos nós são dinâmicas, variando espacial e temporalmente.

Dada a dinamicidade da rede como um todo, a formação dos agrupamentos deve ser adaptativa e reconfigurável. É comum observar a dinamicidade de agrupamentos quanto aos seus *líderes* [22, 55, 23], quanto à *mobilidade dos nós* [20, 21], quanto à *recriação de formações* lógicas [24] e quanto às *leituras dos nós* [26]. Todos estes aspectos apresentam diferentes características e vantagens, que podem muitas vezes ser incorporadas ou tratadas simultaneamente por uma mesma solução. A seguir, são discutidos alguns dos trabalhos encontrados na literatura que consideram tais aspectos ao lidar com leituras de dados escalares.

3.1 Dinamicidade de líder

As RSSFs tipicamente têm restrições de energia devido ao uso de pilhas e baterias. Nestes casos específicos, os nós atuando como líderes de seus agrupamentos são responsáveis por boa parte do roteamento e muitas vezes executam tarefas de processamento adicionais, como a agregação ou a compressão dos dados, consumindo mais energia do que os nós comuns de seu agrupamento. Desta maneira, é normal empregar a operação de **rotação de líderes** a fim de balancear o consumo de energia interno aos agrupamentos. Essa operação alterna o papel dos nós no agrupamento para que o cargo centralizado de líder varie e, assim, o consumo de energia seja distribuído internamente entre os nós do mesmo agrupamento, evitando a sobrecarga de apenas um nó. A seguir são apresentados alguns trabalhos que alternam dinamicamente a posição de líder nos agrupamentos.

3.1.1 LEACH

O LEACH (*Low-Energy Adaptive Clustering Hierarchy*) foi uma das primeiras grandes melhorias em protocolos baseados em agrupamentos [22]. Após a formação dos agrupamentos, eles são mantidos através de rotações randômicas de líderes. A decisão de rotação neste caso parte não do nó líder atual, mas dos nós comuns de seu agrupamento. Isto evita o *overhead* com trocas de mensagens decorrentes de uma operação de eleição explícita.

Em instantes de tempo periódicos, cada nó comum n dos agrupamentos gera um número aleatório $X \in [0, 1]$. Considerando r como sendo o número da rodada discreta atual, G como o conjunto dos nós que não foram eleitos líderes nas últimas $\frac{1}{P}$ rodadas e P a porcentagem desejada de líderes na rede, a Equação 3.1 define a probabilidade $T(n)$ de um nó n se tornar líder. Caso $X < T(n)$, o nó n em questão se elege líder, anunciando em *broadcast* para que os demais nós mudem suas configurações de roteamento.

$$T(n) = \begin{cases} \frac{P}{1 - P(r \bmod \frac{1}{P})}, & \text{se } n \in G, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.1)$$

Esta probabilidade garante que após $\frac{1}{P} - 1$ rodadas, $T = 1$ para aqueles nós que não se tornaram líderes nas últimas $\frac{1}{P}$ rodadas. Desta forma, a rotação inevitavelmente acontecerá e o encargo de líder de cada agrupamento será devidamente rotacionado, balanceando o consumo de energia. Este trabalho foi estendido em [55], onde os sensores que possuem mais energia tem preferência para serem líderes. Esta medida auxilia no balanceamento inicial da energia, mas não garante que a rede tenha um tempo de vida muito superior, pois as rotações ainda são a principal medida de balanceamento no decorrer do tempo.

3.1.2 HEED

A potência de transmissão necessária para fazer os sensores se comunicarem é levada em conta pelo HEED (*Hybrid, Energy-Efficient, Distributed clustering*), e a probabilidade inicial de um sensor se candidatar a líder depende de sua energia residual [23]. Os líderes são eleitos também com base no custo de energia para comunicação dentro do agrupamento formado, sendo que a operação de agrupamento é repetida periodicamente para eleger novos líderes, rotacionando-os, desta maneira.

O HEED define uma função de probabilidade CH_{prob} para a eleição de líderes. Tomando C_{prob} como a porcentagem desejada de líderes na rede, $E_{residual}$ como a energia atual do nó e E_{max} como a energia máxima da bateria deste nó, CH_{prob} é definido na Equação 3.2. Em uma eleição baseada em rodadas, cada nó ainda não candidato gera um novo número aleatório $X \in [0, 1]$ por rodada e, caso $X < CH_{prob}$, ele será dito um candidato a líder. A cada rodada sucessiva, o valor de CH_{prob} dos candidatos é dobrado e, assim que $CH_{prob} \geq 1$, tal nó é considerado um líder definitivo. Ao final das rodadas,

caso algum nó comum não esteja coberto por um líder definitivo, ele irá também se eleger a líder definitivo.

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \quad (3.2)$$

Quando um nó é coberto por mais de um líder definitivo, ele selecionará seu líder tentando minimizar o custo médio de comunicação consequente da potência de transmissão utilizada. Sendo M o número de nós no agrupamento do líder e $MinPwr$ a função que denota a potência mínima para o líder atingir um nó i de seu agrupamento, um nó comum selecionará seu líder tentando minimizar o valor de $AMRP$, definido na equação 3.3. Tal abordagem é mais eficiente em termos de consumo de energia do que escolher sempre o líder mais próximo, dado que diferentes potências de transmissão podem ser empregadas na comunicação interna aos agrupamentos e, quanto menor ela for, menos energia será consumida.

$$AMRP = \frac{\sum_{i=1}^M MinPwr_i}{M} \quad (3.3)$$

3.2 Dinamicidade pela mobilidade

A presença da mobilidade na rede adiciona um requisito de dinamicidade associado à movimentação dos nós. Os agrupamentos podem não possuir posições fixas na topologia e, assim, a comunicação entre nós comuns e líderes ocasionalmente é impossibilitada pela distância e posições atuais, comprometendo o agrupamento lógico inicialmente formado. Desta forma, os agrupamentos necessitam ser constantemente reparados e modificados, alternando os líderes e estabelecendo hierarquias diferentes e mais heterogêneas, se necessário. A seguir são apresentados alguns trabalhos que consideram o fator dinâmico imposto pela mobilidade dos nós.

3.2.1 SPRP_G

Em [20], Islam et al. apresentam um mecanismo de criação de agrupamentos de arquitetura dinâmica chamado SPRP_G (*Shortest Path Routing Protocol*), sendo reconfigurável de acordo com a topologia atual da rede. Para isto, o mecanismo define três tipos lógicos de nós: (i) líder, (ii) *gateway* e (iii) puramente membro. Na arquitetura definida por estes nós, a comunicação não é realizada de maneira direta entre líderes, mas existe um nó *gateway* entre cada par de líderes de agrupamentos adjacentes, podendo o *gateway* pertencer a qualquer um dos dois agrupamentos em questão.

A topologia da rede é inicialmente organizada como uma árvore geradora a partir de um nó raiz r , estabelecido por definição como líder. Após o estabelecimento da raiz,

um procedimento de classificação de nós é chamado recursivamente a fim de determinar todos os líderes, *gateways* e nós puramente membros. São selecionados um a um aqueles nós que ainda não fazem parte da árvore, mas que podem alcançar um nó da árvore em um salto. Se determinado nó n selecionado possuir um vizinho líder, então n será um nó puramente membro. Caso n não possua um vizinho líder, mas possua um vizinho *gateway*, n será classificado como líder, ligando-se a tal *gateway*. Caso n possua apenas vizinhos puramente membros, ele será definido como um líder e um de seus vizinhos puramente membros se tornará um *gateway* de n . Tal procedimento é repetido recursivamente até todos nós serem classificados na árvore geradora para, ao fim, estabelecer uma arquitetura como a da Figura 3.1.

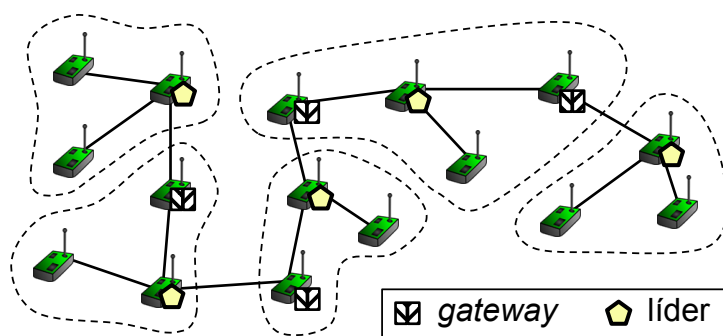


Figura 3.1: Árvore de agrupamentos formada pelo SPRP_G

Após a organização inicial, tal árvore de agrupamentos é mantida de acordo com operações que indicam a entrada ou saída de nós no grafo de conectividade da árvore. Uma vez que um nó entra no grafo, ele se comunicará com seus vizinhos a fim de determinar o seu estado segundo o mesmo algoritmo que inicialmente formou a árvore recursivamente. Na saída de um nó, é assumido que o grafo da árvore continua conexo e então se verifica a diferença entre o grafo da árvore atual e o grafo da árvore cuja raiz é o nó que saiu e, desta maneira, determina-se qual aresta conectaria ambos os grafos, inserindo-a e assim reparando as rotas.

3.2.2 KHOPCA

Baseado no algoritmo do "jogo da vida" [56], o mecanismo KHOPCA (*K-HOP Clustering Algorithm*) opera através de um conjunto de regras simples executado em cada nó [21]. Tais regras dependem apenas de informações dos vizinhos locais de cada nó da rede, garantindo convergência em um número finito de rodadas e realizando a manutenção dinâmica e contínua da estrutura de agrupamentos. Este mecanismo visa estabelecer agrupamentos de k -saltos, isto é, agrupamentos dentro dos quais cada nó comum pode atingir um líder em até no máximo k saltos.

O conjunto de regras definido pelo KHOPCA utiliza um esquema de pontuação individual dos nós, manipulando estes pontos para que eles indiquem se determinado nó

opera como um líder ou como um nó comum. São estabelecidas quatro regras que, informalmente cuidam dos casos: (i) o nó copia o peso de seu maior vizinho subtraído de um; (ii) o nó se torna um líder quando a pontuação dos vizinhos é mínima; (iii) se o nó tem peso maior do que seus vizinhos e não é líder, diminuir sua pontuação; (iv) se dois líderes se aproximarem, um deles deve deixar de ser líder. Considerando $w(n)$ a função de pontuação de um nó n , MAX a constante que indica que tal nó é um líder, MIN a menor pontuação possível para um nó e $N(n)$ o conjunto de vizinhos do nó n , a Equação 3.4 apresenta como o peso $w(n)$ do nó é atualizado de maneira assíncrona e independente com o passar das rodadas a fim de determinar os líderes, cujo $w(n) = MAX$.

$$w(n) \leftarrow \begin{cases} \max(w(N(n))) - 1, & \text{se } w(m) > w(n), \forall m \in N(n), \\ MAX, & \text{se } w(m) = MIN, \forall m \in N(n), \\ w(n) - 1, & \text{se } w(n) \neq MAX \text{ e } w(n) > w(m), \forall m \in N(n), \\ w(n) - 1, & \text{se } w(n) = MAX \text{ e } \exists m \in N(n) \text{ tal que } w(m) = MAX. \end{cases} \quad (3.4)$$

Os autores provam analiticamente que este conjunto de regras converge e que, após algumas rodadas convergindo, a pontuação de cada nó indica a distância $dist$ dele até o líder mais próximo, tal que $dist = MAX - w(n)$. Desta forma, o agrupamento de nós a k -saltos é viabilizado, tal que os parâmetros MAX e MIN refletem na distância máxima k permitida entre um líder e os nós que pertencem à sua borda. O mecanismo KHOPCA é dinâmico ao prover a adaptação contínua e dinâmica da estrutura de pontuação, de modo que essa pontuação permite a cada nó rotear seus dados até o líder mais próximo, ao selecionar como próximo salto sempre um nó de pontuação maior do que a sua.

3.3 Dinamicidade por recriação

Existem muitos protocolos capazes de criar agrupamentos adequados considerando determinadas características e necessidades da rede. O reagrupamento periódico dos nós pode ser utilizado para se adequar à dinamicidade da RSSF. Após o mecanismo de agrupamento ser aplicado, ele normalmente mantém coerência por um tempo, até que a dinamicidade do meio atue de modo que o agrupamento formado anteriormente não satisfaça mais o seu propósito. Com esta idéia, alguns mecanismos propõem que os agrupamentos devem ser recriados periodicamente, como o apresentado a seguir.

3.3.1 DCRR

Em uma abordagem reativa a eventos e baseada em redes de neurônios, o mecanismo de agrupamento do DCRR (*Dynamic-Clustering Reactive Routing*) estabelece agrupamen-

tos dinâmicos quando dados relevantes devem ser enviados à estação-base [24]. Através do conceito de estímulos de redes neurais, a detecção de um evento relevante dispara o processo de eleição de líder na região de interesse. Após a eleição do líder, ele agrega as leituras recebidas da vizinhança e envia estes dados através de um *backbone* mantido dinamicamente por um processo de criação de árvore iniciado na estação-base.

Para a detecção de eventos, o mecanismo estabelece patamares de normalidade na variação das leituras de dado realizadas. Existem dois valores definidos, um limite de estímulo T_e e um limite T_a de ação. Uma vez que a leitura l de um nó satisfaça $l > T_e$, ele envia em *broadcast* uma mensagem de estímulo MSG_E com potência de transmissão proporcional a $\frac{l}{T_e}$, informando o valor de estímulo E e a energia atual do nó. A cada mensagem MSG_E recebida, o nó agregará o valor de E em E_{sum} e, assim que $E_{sum} > T_a$, é determinado que o evento de interesse foi detectado.

Após a detecção reativa do evento, é disparado um processo de seleção de líder. Os nós que detectaram $E_{sum} > T_a$ comparam a sua energia com as informadas na quantidade $n(MSG_E)$ de mensagens MSG_E recebidas. Caso sua energia seja superior ao valor informado, uma variável d^- é incrementada, caso contrário, d^+ é incrementada, ao final, é calculado o valor de energia relativa R_d definido na Equação 3.5. Através disso, o nó com maior R_d é eleito líder da região e agregará os dados, encaminhando-os para a estação-base.

$$R_d = \frac{d^- - d^+}{n(MSG_E)} \quad (3.5)$$

3.4 Dinamicidade das leituras dos dados

Por fim, existem os mecanismos de agrupamento dinâmico que consideram as leituras dos nós que formam a RSSF. Esta categoria visa manter dinamicamente agrupamentos formados apenas por nós que tenham dados similares, de acordo com uma função de similaridade que categoriza os dados, a fim de prover as vantagens vistas no final da Seção 2.2.2. Embora existam diversos trabalhos de agrupamento de nós baseados na similaridade dos dados, eles dependem de inundações constantes [4] ou dependem da atuação constante e centralizada de uma estação-base [19, 25], ambos os casos resultando em latência e sobrecarga. Entretanto, uma característica essencial é a adaptação automática rápida e autônoma dos agrupamentos às novas leituras realizadas pelos nós. A seguir é apresentado um trabalho que cria e mantém agrupamentos dinâmicos de acordo com a similaridade dos dados lidos pelos nós.

3.4.1 SCCS

Pham et al. [26] propõem o protocolo SCCS (*Spatiotemporal Clustering and Compressing Schemes*), tratando tanto a similaridade temporal quanto espacial entre os dados. Para aproveitar da similaridade temporal, cada nó mantém um histórico de dados aplicando uma técnica que mantém apenas aqueles dados necessários para que segmentos de retas entre os pontos (*dado, tempo_de_leitura*) representem uma função temporal que indique a leitura nos determinados instantes, dado um determinado erro máximo. Para se aproveitar da similaridade espacial, o protocolo aplica um mecanismo dinâmico de agrupamento de nós.

O mecanismo de agrupamento utilizado define quatro **estados principais** de nós: (i) *ch*, líder; (ii) *gw*, *gateway*; (iii) *ext*, extensão de líder; (iv) *mem*, membro comum. Além destes, existem três **estados temporários**: (i) *ini*, estado inicial; (ii) *gwr*, candidato a *gateway*; (iii) *chc*, candidato a líder. No início, todos os nós se encontram no estado *ini* e a estação-base inicia o processo de agrupamento através de uma mensagem *CF* (*Cluster Forward*). Tal mensagem será propagada salto-a-salto pelos nós que a receberem, alterando adequadamente seus estados. Junto na mensagem *CF*, os nós sempre informarão a média das leituras realizadas no período anterior.

Ao receber a mensagem *CF*, nós no estado *ini* mudarão seu estado para *gwr*, enviando uma mensagem *SEARCH* para indicar a procura por um candidato a líder na vizinhança. Caso um nó no estado *ini* receba tal mensagem e tenha um grau de similaridade adequado, esse nó mudará seu estado para *chc*. Entre estes candidatos, será eleito a *ch* aquele com maior energia, anunciando seu estado definitivo com uma mensagem *ADV*. Nós de **estado temporário** que recebam tal mensagem e satisfaçam a similaridade necessária se tornarão membros *mem* de tal agrupamento; caso contrário, alterarão seu estado para *gwr*. Um nó temporário *gwr* que receba uma mensagem *ADV* resultante de sua própria mensagem *SEARCH* alterará o seu estado para *gw*, *gateway* definitivo.

Nós do tipo *gw* e *mem* continuarão o processo propagando mensagens *CE* (*Cluster Extension*), que informam a média das leituras do líder de seu agrupamento. Cada nó de **estado temporário** que receba uma mensagem *CE* e possua a similaridade com a leitura daquele líder, alterará o seu estado para *ext*, estendendo a cobertura deste agrupamento. Caso contrário, este nó será classificado como um *gwr*. Este processo continua, realizando o envio alternado de mensagens *CF* e *CE*, de acordo com os critérios especificados, até que a rede inteira seja coberta. Ao fim do processo, uma formação similar à apresentada na Figura 3.2 é estabelecida.

Para manter estes grupos considerando a similaridade dos dados, os líderes *ch* monitoram as leituras reportadas no agrupamento. Caso detecte-se que as leituras estão fora de um intervalo de similaridade aceitável, o líder utilizará o mecanismo de agrupamento de maneira local, restrita ao escopo interior ao seu agrupamento. Desta forma, os nós poderão

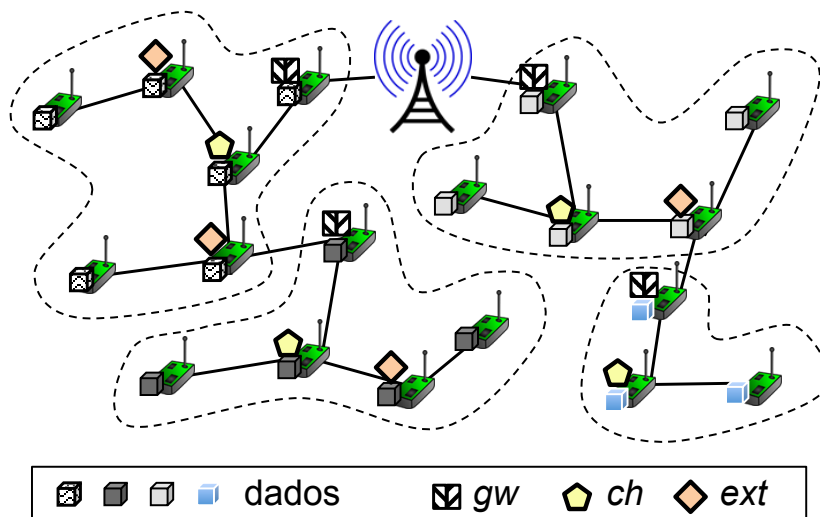


Figura 3.2: Estrutura de agrupamentos formada pelo SCCS

se dividir novamente, criando novos grupos para refletir melhor a similaridade espacial. Entretanto, nota-se que este mecanismo possibilita apenas a quebra de agrupamentos, descartando a operação de fusão deles. Assim, a estação-base realiza o monitoramento para que, caso o número destes agrupamentos ultrapasse um limite máximo, ela inicie um novo processo de agrupamento global da rede.

3.5 Discussão

As RSSFs são dinâmicas devido a diversas questões, tais como topologia, rotas e posicionamento dos nós. Desta forma, os mecanismos de agrupamento devem se adequar, sendo adaptativos e reconfiguráveis. A **dinamicidade de líder** é obtida de maneira interior aos agrupamentos, balanceando o consumo de energia através da rotação dos nós líderes. O protocolo LEACH [22] foi um dos primeiros destaques ao propor um mecanismo de rotação sem mensagens explícitas de eleição. O HEED [23] é capaz de, além de balancear, otimizar o consumo de energia interna aos agrupamentos ao minimizar a potência de transmissão necessária para a comunicação interna ao agrupamento. Entretanto, tais políticas de dinamicidade são apenas locais, não afetando diretamente a rede de maneira global.

Enquanto os protocolos LEACH e HEED operam em redes estáticas, é cada vez mais comum a **dinamicidade pela mobilidade**. Desta maneira, manter agrupamentos enquanto nós transitam arbitrariamente pela rede é complicado, havendo a necessidade de mecanismos específicos para estes cenários. O protocolo SPRP_G [20] estabelece uma árvore geradora, criando nós líderes e *gateway* a fim de conectar os seus agrupamentos dentro dela. Entretanto, tal mecanismo assume que mesmo que um nó saia do escopo de um agrupamento, a árvore sempre continuará conexa, o que simplifica muito a manutenção

com asserções irreais. O mecanismo KHOPCA [21] é inspirado no jogo da vida, operando proativamente através de um conjunto simples de regras que define agrupamentos de k -saltos até os líderes.

Levando em conta a complexidade e o custo de manter uma estrutura hierárquica de maneira dinâmica e proativa, algumas abordagens optam por atender aos requisitos de **dinamicidade por recriação** completa dos agrupamentos, seja ela periódica ou reativa. O mecanismo DCRR [24] assume que agrupamentos em uma rede dinâmica são de relevância apenas quando há a detecção de algum evento, sendo que manter a estrutura de agrupamentos de maneira contínua na rede é dispendioso. Adotando uma abordagem baseada em estímulos das redes neurais, o DCRR realiza o agrupamento de nós para a agregação de dados de modo reativo. Entretanto, ele considera a soma de estímulos e isso pode tornar a detecção de um evento mais sensível em regiões com maior densidade de nós. Ademais, abordagens que dependam do reagrupamento completo da rede como um todo introduzem custos em termos de latência e energia.

De maneira geral, as características desejáveis em um protocolo de agrupamento lógico de nós que mantenha os nós com leituras similares agrupados são: (i) *tratamento explícito das relações de similaridade espacial*, (ii) *formação dinâmica do agrupamento quanto aos nós que fazem parte dele*, (iii) *não depender da estação-base para coordenação centralizada*, (iv) *não depender da recriação constante dos agrupamentos* e, por fim, (v) *oferecer manutenção robusta*, isto é, capaz de manter a formação dos agrupamentos dinâmica e adaptativa às mudanças. Assim, a Tabela 3.1 apresenta uma avaliação dos protocolos descritos, de acordo com estas características desejáveis.

Característica	Líderes		Mobilidade		Recriação	Similaridade
	LEACH	HEED	SPRP _G	KHOPCA	DCRR	SCCS
Similaridade espacial						✓
Formação dinâmica			✓	✓	✓	✓
Base não coordena	✓	✓	✓	✓	✓	✓
Não depende de recriação	✓	✓	✓	✓		
Manutenção robusta				✓		

Tabela 3.1: Tabela comparativa dos protocolos de agrupamento

Destas abordagens, nenhuma oferece suporte apropriado à similaridade de dados, isto é, que seja robusto o bastante para suportar a dinamicidade dos dados lidos sem a necessidade de operações de grande *overhead* ou centralização. Dentre as categorias de dinamicidade apresentadas, a rotação de líderes pode ser incorporada para o balanceamento de energia e o suporte a mobilidade é necessário em casos de aplicações específicas, porém pouco comuns [57]. A dinamicidade por recriação pode aproveitar um pouco da correlação espacial entre os dados, mas de maneira indireta e ineficiente. Desta forma, é importante que protocolos tratem de maneira explícita a **dinamicidade considerando**

os dados. Na literatura, o protocolo SCCS [26] se destaca ao estabelecer agrupamentos dinâmicos e reconfiguráveis de acordo com a similaridade de dados. Entretanto, a sua manutenção não é adequada. Para se adequar à variação dos dados, é permitida apenas a operação de quebra de agrupamento. Sem que existam fusões, a estação-base deve disparar um novo processo de reagrupamento global.

Para as fusões dos agrupamentos de leituras similares, seria necessária a troca de informações quanto às leituras dos agrupamentos. Como foi visto no Capítulo anterior, os vagalumes apresentam um potencial inexplorado quanto à sincronização de dados atemporais. Desta forma, eles podem ser utilizados para sincronizar uma agregação média de leituras que represente todas as leituras contidas no agrupamento, de forma que agrupamentos adjacentes sejam capazes de determinar suas similaridades, sendo devidamente unidos.

3.6 Resumo

Este capítulo apresentou um estudo dos protocolos de agrupamento quanto a diversas formas de dinamicidade, desde a comum dinamicidade de líder para balanceamento de energia até a dinamicidade de agrupamento quanto a similaridade de dados, que além otimizar o uso de técnicas de agregação de dados, beneficia aplicações que devem observar eventos de interesse através de políticas de detecção baseada nas leituras dos nós individuais. Mostrou-se a necessidade de um protocolo que seja capaz de operar adaptativamente quanto à dinamicidade das leituras realizadas pelos nós da rede.

Capítulo 4

DDFC: UM PROTOCOLO DE AGRUPAMENTO

Este capítulo apresenta um protocolo bio-inspirado para a criação e manutenção dinâmica de agrupamentos que considera a similaridade das leituras realizadas pelos nós sem fio. O protocolo, denominado DDFC, mantém os nós com leituras similares agrupados logicamente em uma hierarquia baseada em líderes. O seu funcionamento é inspirado na operação de sincronização utilizada pelos vagalumes para piscarem simultaneamente. A Seção 4.1 apresenta uma visão geral da arquitetura do protocolo DDFC. A Seção 4.2 detalha o DDFC, identificando as estruturas locais e o funcionamento dos seus agentes.

4.1 Visão geral

O protocolo DDFC (*Dynamic Data-aware Firefly-based Clustering*) tem como objetivo criar e manter agrupamentos lógicos dos nós que possuam leituras de dados similares na RSSF. Para isso, cada nó mantém a agregação média de leituras do agrupamento localmente sincronizada, a fim de possibilitar a determinação precisa de quando um agrupamento de nós deve ser fragmentado ou quando diferentes agrupamentos devem ser unidos nos casos de, respectivamente, leituras que não satisfaçam ou satisfaçam a similaridade de dados desejada. Uma vez estabelecidos os agrupamentos lógicos, o protocolo DDFC define índices para roteamento interno aos agrupamentos, possibilitando que as mensagens dos nós comuns atinjam um líder do agrupamento, para que ele possa retomar o roteamento dos dados até uma estação-base. A arquitetura geral para o protocolo é dividida em três componentes, definidos como agentes que rodam todos simultaneamente em cada nó, como observado na Figura 4.1.

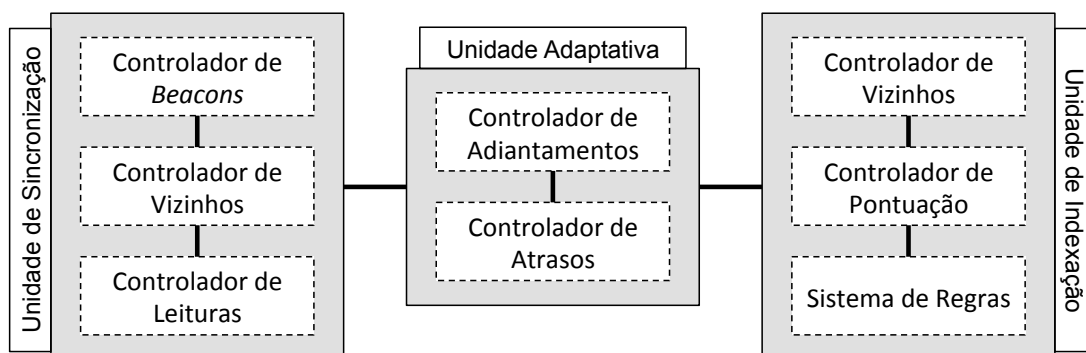


Figura 4.1: Arquitetura em alto nível do protocolo de agrupamento DDFC

O protocolo DDFC opera entre as camadas de enlace e de rede. Desta maneira, ele utiliza das comunicações em nível de enlace para realizar *broadcasts* de mensagens de

controle, criando e mantendo os agrupamentos de nós, assim como realizando o roteamento interno aos agrupamentos. Como o DDFC não provém o roteamento inter-agrupamentos, a camada de rede, acima do DDFC, é responsável por rotear as mensagens entre os líderes estabelecidos pelo protocolo DDFC. Tal organização pode ser observada na Figura 4.2.

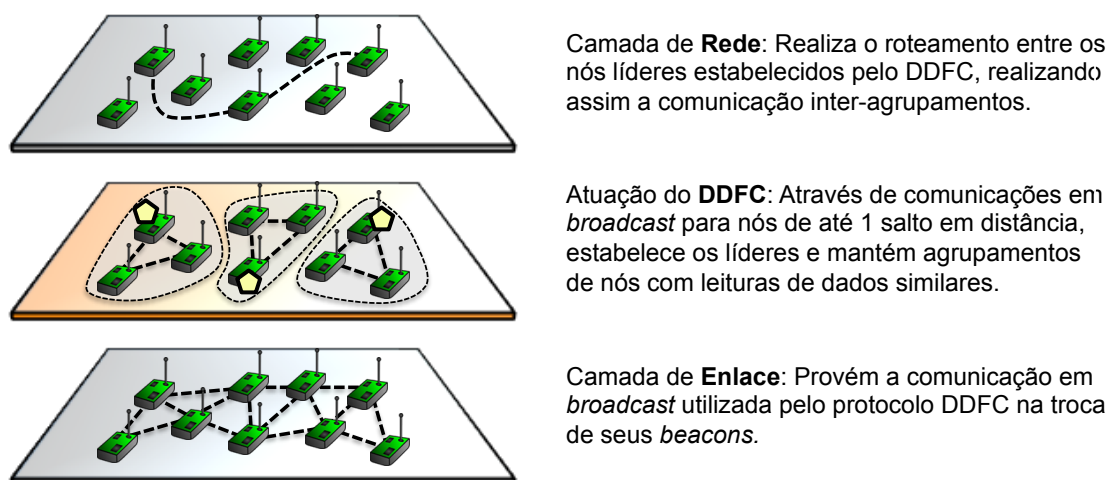


Figura 4.2: Posicionamento do protocolo DDFC entre as camadas de comunicação

Com a operação do DDFC, uma estrutura hierárquica similar à da Figura 4.3 será estabelecida, na qual as arestas sólidas representam relações de similaridade de dados de nós adjacentes. Os agrupamentos são formados mantendo apenas aqueles nós com leituras similares agrupados, de acordo com a função de similaridade definida pela aplicação da rede, como visto na Subseção 2.2.3. Como as leituras similares podem atingir grandes extensões espaciais, os agrupamentos podem por sua vez possuir mais de um líder, tal que eles são atingidos com um ou mais saltos no encaminhamento das mensagens. A hierarquia apresentada na figura é formada pelas operações dos Agentes descritos em sequência.

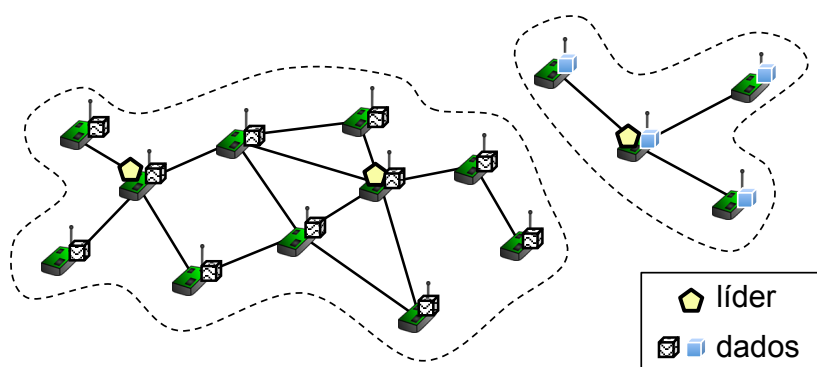


Figura 4.3: Estrutura hierárquica estabelecida pelo DDFC

4.1.1 Unidade de Sincronização

Inspirada na sincronização do piscar alcançada por algumas espécies de vagalumes biológicos, a **Unidade de Sincronização** utiliza uma operação similar para manter a agregação média das leituras de cada agrupamento sincronizada entre os nós do mesmo agrupamento. Tal operação trata apenas da similaridade espacial, isto é, a similaridade em um dado instante, tal que a similaridade temporal é tratada como consequência da sincronização contínua que ocorre no decorrer do tempo. Esse agente é dividido nos controladores de **Beacons**, de **Vizinhos** e de **Leituras**. O Controlador de *Beacons* atua através do envio periódico de mensagens *beacons*¹, análogas ao piscar dos vagalumes, e é responsável também pela interpretação de mensagens *beacon* geradas pelos demais nós. Esse controlador se comunica diretamente com o Controlador de Vizinhos. Com a interpretação dos *beacons* recebidos, o Controlador de Vizinhos é responsável por manter atualizada uma estrutura que indica os nós vizinhos que satisfazem o *threshold* de agrupamento por similaridade de dados, informação provida pelo Controlador de Leituras, e determinar quando tais relações de similaridade são estabelecidas ou destruídas.

Muitos trabalhos são bio-inspirados em vagalumes devido à sincronia que eles são capazes de atingir através de operações simples e, desta forma, a sincronização de relógios é a mais evidente no seu uso em RSSFs [28, 29]. Entretanto, a troca de mensagens para a sincronização de relógio é afetada pelos diversos tempos de latência que são decorrentes da própria natureza do meio sem fio, por depender do instante no qual a mensagem enviada chega ao seu destino. Por outro lado, a abordagem utilizada pela Unidade de Sincronização é diferente ao visar manter a sincronia não de um relógio, mas sim de um parâmetro de funcionamento atemporal. Por isso, o DDFC não sofre de imprecisões geradas pelas latências, decorrentes das condições imprevisíveis no meio sem fio. A cada intervalo de tempo, os nós atuam de acordo com as agregações de leituras sincronizadas já obtidas no instante de tempo anterior, independente da sincronia com a qual cada nó efetua as operações que determinam os agrupamentos lógicos e até mesmo da perda de mensagens, que só refletiria em um tempo de convergência maior, correspondente ao intervalo definido entre cada transmissão de *beacon*.

A operação de agrupamento lógico atua de maneira local em cada nó, utilizando-se da leitura agregada sincronizada de cada agrupamento estabelecido para identificar a similaridade entre os nós adjacentes na rede. Essa relação de similaridade é definida por uma função, que pode mudar de acordo com as necessidades da aplicação em questão. Devido à ausência do uso e da manutenção de informações globais, o DDFC é escalável, atendendo aos quesitos de dinamicidade, difíceis de serem tratados em abordagens que mantêm informações globais. A relação de similaridade de dados é expressa em cada par de nós vizinhos e, a partir do conjunto de pares similares, é possível definir de maneira

¹Um "*beacon*" pode ser definido como um alerta, usualmente sinalizado por emissões de luz.

global os agrupamentos lógicos aos quais os nós pertencem.

4.1.2 Unidade de Indexação

Como as RSSFs são muito dinâmicas em quesitos como leituras, topologia e estado de enlace, seria custoso manter a visão global dos agrupamentos lógicos atualizada em cada um dos nós. Desta forma, a Unidade de Sincronização estabelece uma estrutura global de agrupamentos lógicos que não é perceptível na visão local de cada nó. A Unidade de Indexação se utiliza das vizinhanças locais de nós com leituras similares, definidas pela Unidade de Sincronização, para estabelecer a hierarquia baseada em líderes e consequente roteamento dos nós comuns até os líderes.

Dado que os agrupamentos são estabelecidos por *thresholds* de variação de leituras, é possível que eles cubram uma extensão grande, impossibilitando que um único líder administre toda ela em um único salto. Assim, é necessário eleger mais de um líder por agrupamento, mantendo nos nós comuns, também, as informações para que eles sejam capazes de atingir estes líderes. O **Controlador de Vizinhos** da Unidade de Indexação é responsável por manter uma lista de vizinhos do nó, análogos aos vizinhos da Unidade de Sincronização - aqui são ditos vizinhos apenas aqueles sensores no raio de comunicação que possuam leituras satisfazendo o *threshold* de similaridade de dados estabelecido pelo controlador de leituras da Unidade de Sincronização. O **Controlador de Pontuação** é responsável por pontuar o nó atual através de um **Sistema de Regras** que considera a pontuação dos nós vizinhos. Essas pontuações são adaptativas ao estado atual da rede e definem dinamicamente os postos de líder e a rota que os nós comuns devem utilizar para atingir o líder mais próximo dentro de seu agrupamento.

4.1.3 Unidade Adaptativa

As mensagens enviadas pelo Controlador de Beacons são necessárias para manter a formação dinâmica coerente com as leituras dos nós. Entretanto, eventos que podem alterar os agrupamentos não são necessariamente periódicos e, desta forma, o uso de um intervalo adaptativo entre os *beacons* pode proporcionar uma dinamicidade mais eficiente em questão de similaridade de dados e até mesmo diminuir a quantidade de *beacons* enviados, quando a rede se encontra em uma configuração estável. Desta forma, o **Controlador Adaptativo** pode tanto adiantar quanto atrasar o intervalo da operação do Controlador de Beacons. Com a detecção de uma leitura que indique uma mudança no agrupamento, a operação do Controlador de Beacons pode ser **adiantada**. Ademais, esta operação pode também ser adiantada visando garantir menor tempo de convergência em casos de mudanças na estrutura de indexação pela Unidade de Indexação. Por fim, em casos de estabilidade, o período entre os *beacons* pode ser **umentado**, dado que as mudanças não

seriam significantes para alterar a formação dos agrupamentos.

As operações dos Agentes Vagalume e de Indexação são fortemente correlacionadas. A Unidade de Indexação depende das relações de similaridade de dados entre nós adjacentes estabelecidas pela Unidade de Sincronização para que ele possa determinar as pontuações dos nós, considerando como vizinhos apenas aqueles nós que fazem parte do mesmo agrupamento lógico. Além disso, o período de atuação da Unidade de Indexação é o mesmo da Unidade de Sincronização, uma vez que os nós informam suas pontuações atuais através do mesmo *beacon*. Ambos os agentes dependem de um mesmo período de atuação para se adaptar autonomamente às características dinâmicas do meio sem fio. Desta maneira, este período exerce grande influência na dinamicidade do protocolo DDFC, justificando a necessidade do seu controle adaptativo.

4.1.4 Analogia da inspiração em princípios biológicos

A Unidade de Sincronização no DDFC é inspirada nos princípios biológicos de vagalumes. Como visto na Seção 2.3, algumas espécies de vagalumes são capazes de sincronizar de maneira distribuída a maneira com a qual eles piscam. Na modelagem do comportamento de tais entidades, esta organização é decorrente de um ajuste gradual de um relógio interno, tal que com o piscar de vagalumes vizinhos, cada vagalume ajusta o seu próprio relógio de acordo com uma função de ajuste.

Analogamente às entidades biológicas, a Unidade de Sincronização envia *beacons* periódicos que são correspondentes ao piscar de cada vagalume. Estes *beacons* carregam informações sobre as leituras dos nós e é através deles que os nós vizinhos são capazes de aplicar uma função de ajuste na atual agregação média das leituras de seu agrupamento. Desta forma, enquanto com o passar do tempo os vagalumes biológicos atingem a sincronização entre as suas piscas, no DDFC com o passar do tempo os nós atingem a sincronia entre as agregações das leituras médias de cada agrupamento. Tais relações de correspondência são ilustradas na tabela 4.1.

	Entidade biológica do vagalume	Unidade de Sincronização do DDFC
Base	Piscar periódico	Envio periódico de <i>beacons</i>
Operação	Função de ajuste de relógio interno	Função de ajuste da agregação de leituras
Resultado	Sincronia entre piscadas	Sincronia entre as agregações das leituras médias nos nós de cada agrupamento

Tabela 4.1: Relação entre a entidade biológica e a unidade de sincronização

4.2 Funcionamento do DDFC

Esta seção detalha a base de funcionamento e operações realizadas pelo protocolo DDFC. Inicialmente são apresentadas as estruturas de dados² locais nos nós que dão suporte à vizinhança. Em seguida, a Unidade de Sincronização é detalhado na Subseção 4.2.2, a Unidade de Indexação na Subseção 4.2.3 e, por fim, a Unidade Adaptativa em 4.2.4.

4.2.1 Armazenamento de vizinhança

Como base do protocolo, existem duas estruturas locais simples de grande importância em cada nó sensor, contendo (i) o conjunto de vizinhos espaciais que satisfazem os *thresholds* de similaridade de dados e (ii) informações sobre as leituras dos vizinhos espaciais. A Figura 4.4 ilustra um pedaço de uma topologia, à esquerda, e as estruturas de dados armazenadas de cada nó. O nó selecionado em questão possui no total sete vizinhos, dos quais quatro possuem dados similares. Desta maneira, tal nó possui 4 vizinhos que satisfazem os *thresholds* de similaridade de dados - este conjunto de nós é mantido em uma estrutura (i) *SNeigh*, conjunto indicado por linhas pontilhadas na parte superior central da figura. Além disso, para cada um dos sete vizinhos espaciais, o nó em questão mantém uma estrutura (ii) *NeighR*, conjunto indicado por linhas pontilhadas na parte inferior central da figura, que possui informações sobre as leituras de todos os vizinhos espaciais. Tais informações envolvem a leitura individual de cada vizinho e a leitura agregada da vizinhança daquele vizinho, assim como a quantidade de nós cujas leituras foram agregadas, para se determinar aproximadamente quanto à agregação em questão representa.

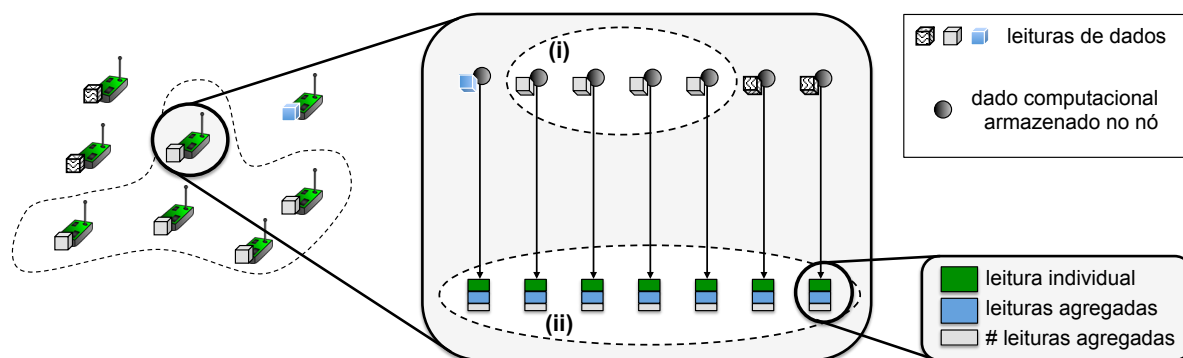


Figura 4.4: Estruturas de dados que representam a vizinhança dos nós

4.2.2 Sincronização da agregação de leituras

A operação da Unidade de Sincronização visa manter sincronizado localmente um valor que indica a agregação das leituras do agrupamento do nó em questão. Esta agregação

²Referentes aos algoritmos, não às leituras realizadas pelos nós sensores

representa uma leitura comum ao agrupamento como um todo. É importante destacar que ela não corresponde a uma operação de agregação de dados no sentido de compressão de informações para envio à estação-base, mas sim a um valor que resume a leitura do agrupamento e suporta as operações de manutenção³.

Com este valor agregado, é possível que os nós saibam quando eles devem sair de seu agrupamento, em caso de leituras muito diferentes, e quando agrupamentos vizinhos devem ser unidos devido a leituras muito próximas, satisfazendo a função de similaridade de dados aplicada. Como função padrão, o protocolo DDFC utiliza a função de *threshold* absoluto de diferença entre leituras, devido à sua maior versatilidade. Entretanto, tal função pode ser alterada como um parâmetro ajustável, dependendo das necessidades da aplicação da rede.

No início, é assumido que cada nó da rede faz parte de um agrupamento diferente. Desta maneira, os diversos agrupamentos gradualmente serão unidos, de acordo com o *threshold* de similaridade de dados. Após a convergência e formação estável inicial de agrupamentos, estes serão mantidos dinamicamente através da união e fragmentação de agrupamentos. A união e a fragmentação são resultados de operações locais de adição ou remoção de arestas, que representam a semelhança de dados entre pares de sensores, referentes à vizinhança de cada nó do agrupamento.

O Algoritmo 1 apresenta o funcionamento da Unidade de Sincronização do DDFC. Periodicamente, considerando um intervalo predefinido como parâmetro do protocolo, o Controlador de *Beacons* de cada nó envia em *broadcast* uma mensagem *beacon* informando seu identificador, denominado *ADDR*, sua leitura atual, obtida através da função *getReading()*, a leitura média agregada dos nós com leituras similares em sua vizinhança, obtida através da função *getAverageReading()* e a quantidade de vizinhos com leituras similares na vizinhança (l.1-5). Deve-se notar que o envio periódico destas mensagens insere sempre um tempo aleatório na ordem de milissegundos para evitar transmissões simultâneas, minimizando o número de possíveis colisões.

A função *getAverageReading* (l.17), provida pelo Controlador de Leituras, é utilizada para obter a média ponderada da agregação de leituras na vizinhança local que satisfaz as relações de similaridade de acordo com o Controlador de Vizinhos, ou seja, aqueles vizinhos que pertencem ao mesmo agrupamento do nó em questão. Com a leitura do nó atual (l.18-19), a média das leituras informadas pelos nós que fazem parte do mesmo agrupamento é computada (l.20-24), considerando a leitura agregada de cada um destes nós, denominada aR , e o número, denominado nR , de leituras agregadas nele como um peso (l.21). Ao fim, obtém-se a média agregada de leituras similares na região do nó (l.25), sincronizado devido à troca de *beacons* que atualiza as informações da vizinhança.

Ao receber um *beacon* (l.7), o nó saberá qual foi a sua origem, identificada como

³Operações de agregação no sentido de compressão de informação para envio à estação-base ainda podem ser executadas pela camada de aplicação, não tendo relação com o DDFC.

Algoritmo 1 Unidade de Sincronização

```

1: procedimento BEACONTIMEREXPIRE
2:    $Send(ADDR, getReading(), getAverageReading(), |SNeigh|)$ 
3:    $Wait(interval + rnd())$ 
4:    $BeaconTimerExpire()$ 
5: fim procedimento
6:
7: procedimento RECEIVEBEACON( $src, iR, aR, nR$ )
8:    $NeighR[src] \leftarrow \{iR, aR, nR\}$ 
9:    $localAvg \leftarrow getAverageReading()$ 
10:  se  $(|iR - localAvg| < CThresh)$  e  $(|getReading() - aR| < CThresh)$  então
11:     $SNeigh \leftarrow SNeigh \cup src$ 
12:  senão
13:     $SNeigh \leftarrow SNeigh - src$ 
14:  fim se
15: fim procedimento
16:
17: procedimento GETAVERAGEREADING
18:    $accumulatedReading \leftarrow getReading()$ 
19:    $numberOfReadings \leftarrow 1$ 
20:   para cada  $v \in SNeigh$  faça
21:      $temp \leftarrow NeighR[v].aR * NeighR[v].nR$ 
22:      $accumulatedReading \leftarrow accumulatedReading + temp$ 
23:      $numberOfReadings \leftarrow numberOfReadings + NeighR[v].nR$ 
24:   fim para cada
25:   retorna  $(accumulatedReading / numberOfReadings)$ 
26: fim procedimento

```

src , a leitura individual do nó origem, denominada iR , a leitura agregada aR média de sua vizinhança, e a quantidade nR de nós que foram considerados nesta agregação. Inicialmente, a estrutura $NeighR$ é atualizada (l.8) com esta informação, independente da origem e de relações de similaridade. A agregação média de leituras na região do nó atual (l.9) é usada para verificar se as leituras do nó atual e do nó origem src satisfazem a relação de similaridade de dados de acordo com o *threshold* de diferença de leituras $CThresh$ (l.10), neste caso em uma função de diferença absoluta. Caso este *threshold* seja satisfeito, a estrutura $SNeigh$ será atualizada incluindo tal origem src , caso contrário, a estrutura será atualizada removendo o seu nó src .

A função de similaridade utilizada no Algoritmo 1 pode ser decomposta em duas partes: **(i)** $|iR - localAvg| < CThresh$ e **(ii)** $|getReading() - aR| < CThresh$, que correspondem basicamente à mesma verificação de similaridade, porém com referências diferentes. A parte **(i)** verifica se a leitura iR recebida do nó vizinho, de identificador src , satisfaz o *threshold* de diferença $CThresh$ com relação ao agrupamento do nó atual. A parte **(ii)** verifica se a leitura atual $getReading()$ do nó local satisfaz o *threshold* de diferença $CThresh$ com relação ao agrupamento do nó src . Estas relações são expressas pela Equação 4.1, na qual X e Y representam respectivamente a leitura do nó atual e a leitura com a qual ela está sendo comparada, a fim de satisfazer a diferença absoluta $CThresh$.

$$\left| Y - \frac{X + \sum_{v \in S_{Neigh}} (NeighR[v].aR * NeighR[v].nR)}{1 + \sum_{v \in S_{Neigh}} (NeighR[v].nR)} \right| < CThresh \quad (4.1)$$

A Figura 4.5 ilustra um exemplo de funcionamento da Unidade de Sincronização, mostrando a sincronização da agregação de leituras de cada agrupamento e consequentes relações de similaridade de leituras. Na figura, as arestas pontilhadas indicam vizinhos puramente espaciais, enquanto as arestas sólidas indicam aqueles vizinhos que satisfazem a relação de similaridade de dados. As caixas ao lado de cada nó correspondem à estrutura vista na Figura 4.4, indicando, de cima para baixo, a leitura individual daquele nó, a leitura agregada sincronizada naquele nó, e a quantidade de leituras agregadas. Cada instante T é separado por um envio de *beacon* de cada nó. No instante inicial $T1$, as leituras agregadas de cada nó correspondem à sua leitura inicial, pois ainda não houve nenhuma troca de mensagens.

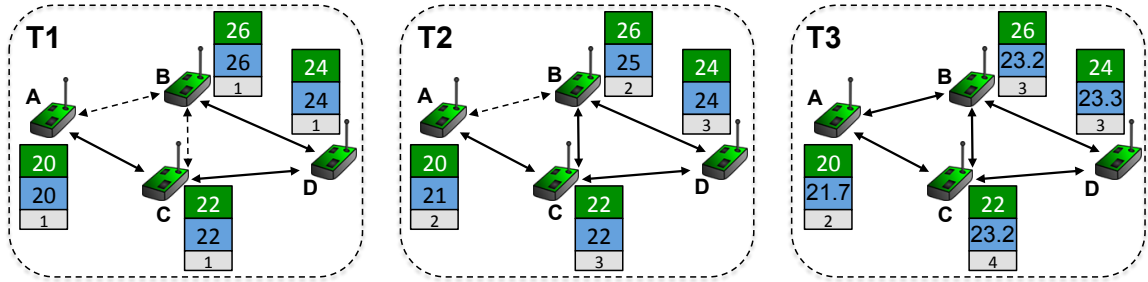


Figura 4.5: Funcionamento do agrupamento baseado em vagalumes

Assim, tomando um $CThresh = 3.0$, as arestas $((B, D), (D, C), (C, A))$ satisfazem a Equação 4.1 e estabelecem uma relação de similaridade no estado inicial $T1$. Após o estado inicial, os nós atualizam suas leituras agregadas aR_{T_n} de acordo com as leituras obtidas no instante $aR_{T_{n-1}}$ anterior, como elaborado no Algoritmo 1. No instante $T2$, $aR_{T2}(A) = \frac{20+1*22}{1+1}$, $aR_{T2}(B) = \frac{26+1*24}{1+1}$, $aR_{T2}(C) = \frac{22+1*20+1*24}{1+1+1}$, $aR_{T2}(D) = \frac{24+1*22+1*26}{1+1+1}$. Neste instante, a aresta de similaridade (B, C) passa a existir. No instante $T3$, as leituras agregadas são atualizadas novamente, $aR_{T3}(A) = \frac{21+3*22}{1+3}$, $aR_{T3}(B) = \frac{25+3*24+3*22}{1+3+3}$, $aR_{T3}(C) = \frac{22+2*21+2*25+3*24}{1+2+2+3}$, $aR_{T3}(D) = \frac{24+2*25+3*22}{1+2+3}$. Neste instante, a aresta de similaridade (A, B) passa a existir.

Com a Unidade de Sincronização operando desta maneira, cada nó possuirá sua estrutura S_{Neigh} constantemente atualizada com a troca de *beacons*. Tal estrutura indica quais nós na vizinhança do nó atual são vistos como membros do mesmo agrupamento. Assim, como cada nó saberá quais vizinhos fazem parte do mesmo agrupamento, os agrupamentos lógicos podem ser definidos de acordo com o Algoritmo 2. O agrupamento C de um nó n ($l.1$) corresponde ao conjunto formado pela união daquele nó ($l.2$) com a operação recursiva de agrupamento de cada um dos nós em seu S_{Neigh} ($l.3-5$).

Contudo, esta visão global dos agrupamentos completos dos nós não é mantida local-

Algoritmo 2 Definição dos agrupamentos lógicos

```

1: procedimento CLUSTER( $n$ )
2:    $C \leftarrow \{n\}$ 
3:   para cada  $m \in (SNeigh(n) - C)$  faça
4:      $C \leftarrow C \cup Cluster(m)$ 
5:   fim para cada
6:   retorna  $C$ 
7: fim procedimento

```

mente em cada um dos nós pois a coerência do estado reportado por estas mensagens com a formação lógica real não poderia ser garantia de maneira eficiente. Tal operação causaria grande sobrecarga devido à quantidade de mensagens de atualização e manutenção que a dinamicidade do meio e das formações lógicas demandaria. Desta forma, embora os agrupamentos lógicos existam de maneira global, eles não são visíveis por completo em nível de nó. Entretanto, conhecendo as vizinhanças locais que pertencem aos seus agrupamentos, os nós podem organizar uma hierarquia distribuída baseada em líderes para representá-los. Desta forma, com estas estruturas definidas pela Unidade de Sincronização, a Unidade de Indexação pode operar respeitando a similaridade de dados, garantindo que os nós consigam enviar mensagens para os líderes de seus agrupamentos lógicos.

4.2.3 Indexação local das rotas até os líderes

A Unidade de Indexação de um nó opera considerando os agrupamentos lógicos de acordo com a relação de similaridade de leituras que a Unidade de Sincronização estabeleceu. A seleção dos líderes e o estabelecimento das rotas até os líderes mais próximos são realizados através de um esquema de pontuação, baseado nas regras propostas pelo KHOPCA [21]. Por considerar redes móveis, o KHOPCA naturalmente é bastante adaptativo e dinâmico e, desta maneira, seu funcionamento pode ser modificado a fim de suportar a também dinâmica natureza dos dados, mesmo que operando em redes estáticas.

Como base de funcionamento, cada nó atualiza sua pontuação auto atribuída de acordo com a pontuação de seus vizinhos que pertençam ao mesmo agrupamento - informação enviada em *piggyback* na mesma mensagem *beacon* utilizada pela Unidade de Sincronização. É definido como parâmetro uma pontuação máxima que determina também, indiretamente, a distância máxima até um líder. Assim, aqueles nós que possuem pontuação equivalente a esta pontuação máxima são determinados líderes, enquanto os demais nós utilizam sua pontuação como meio de determinar o próximo salto no roteamento até o líder mais perto.

No início, todos os nós possuem pontuação individual $pts = 0$. Tomando $MaxK$ como sendo o parâmetro da pontuação máxima e a lista SN^4 de vizinhos pertencentes ao mesmo agrupamento, provida do Controlador de Vizinhos, as regras utilizadas para

⁴Mesma lista $SNeigh$, porém abreviada por questões de espaço.

a atualização dinâmica dos pontos são definidas de acordo com a Equação 4.2, baseada nas regras propostas pelo KHOPCA. A primeira condição das equações visa manter uma diferença máxima de 1 entre a pontuação dos nós adjacentes. A segunda regra define um nó como líder, maximizando seu pts para $MaxK$ caso seus vizinhos tenham pontuação mínima. A terceira regra diminui a pontuação de um nó caso ele tenha pontuação maior que seus vizinhos e não seja líder, a fim de manter a diferença máxima de 1 entre as pontuações adjacentes. Por fim, a quarta regra visa eliminar a existência de líderes, nós com $pts = MaxK$, adjacentes. Tais regras compõem o Sistema de Regras da Unidade de Indexação.

$$pts(n) \leftarrow \begin{cases} \max(pts(SN(n))) - 1, & \text{se } pts(m) > pts(n), \forall m \in SN(n), \\ MaxK, & \text{se } pts(m) = 0, \forall m \in SN(n), \\ pts(n) - 1, & \text{se } pts(n) \neq MaxK \text{ e } pts(n) > pts(m), \\ & \forall m \in SN(n), \\ pts(n) - 1, & \text{se } pts(n) = MaxK \text{ e } \exists m \in SN(n) \text{ tal que} \\ & pts(m) = MaxK \text{ e } ((|SN(m)| > |SN(n)|) \\ & \text{ou } (|SN(m)| = |SN(n)| \text{ e } m > n)). \end{cases} \quad (4.2)$$

Este sistema de regras, embora baseado nas regras propostas pelo protocolo KHOPCA [21], foi estendido para se adaptar melhor aos quesitos de dinamicidade do meio, possuindo melhor estabilidade. A Unidade de Indexação dá mais prioridade no quesito pontuação àqueles nós que possuem mais vizinhos com leituras similares - prioridade expressa na quarta regra do Sistema de Regras. Desta forma, a estabilidade dos líderes é maior. Após definidas estas regras, o Controlador de Pontuação as aplica periodicamente em cada nó.

Este controlador verifica as regras na mesma ordem que elas foram apresentadas, em uma sequência ordenada da primeira até a quarta. Entretanto, a cada período no qual elas são verificadas, apenas uma delas pode ser aplicada. Ou seja, se a primeira regra for verificada e aplicada, as demais não são sequer verificadas. Da mesma maneira, se uma regra não satisfizer as condições para que ela seja aplicada, são verificadas em sequência cada uma das sucessoras, parando a verificação das demais assim que a primeira delas for aplicada com sucesso. Tal operação é diferente do KHOPCA, que pode aplicar mais de uma regra em ordem e de maneira indeterminada em cada intervalo de tempo, produzindo resultados mais instáveis.

A Figura 4.6 apresenta um exemplo de como tais regras podem ser aplicadas a partir de uma topologia inicial, supondo um valor de $MaxK = 3$. As arestas sólidas entre cada par de nós indicam uma relação de similaridade de leituras entre eles e, desta maneira,

eles se consideram vizinhos, de acordo com a estrutura *SNeigh*. No instante $T1$, todos os nós possuem a pontuação mínima $pts = 0$. Dada que a verificação das regras não possui requisitos de sincronia, neste caso os nós B , C e D aplicam as regras primeiro, maximizando seus pontos através da regra 2. Como os nós A e E realizam a verificação depois, seus vizinhos já possuem $pts = MaxK$ e, desta forma, eles aplicam a regra 1. No instante $T3$, existem 3 nós adjacentes com $pts = MaxK$. Desta maneira, os nós B e D aplicam a regra 4, pois o nó C possui mais vizinhos similares. Novamente, o nó A aplica a regra 1, mantendo a diferença máxima de pontos entre nós adjacentes como 1.

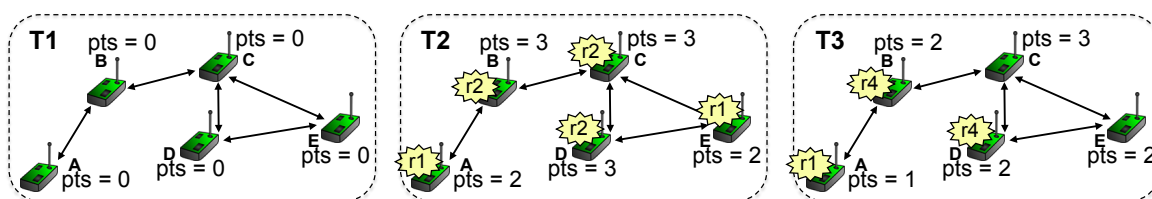


Figura 4.6: Aplicação das regras da Unidade de Indexação

A estrutura apresentada na figura é mantida dinamicamente, de acordo com a variação das leituras e da topologia. Com tal estrutura, aqueles nós que satisfaçam $pts = MaxK$ são estabelecidos líderes. Os nós comuns podem rotear os seus dados para o líder mais próximo de seu agrupamento selecionando como próximo salto um nó pertencente à estrutura *SNeigh*, cuja pontuação seja maior do que a sua atual. Desta maneira, como os líderes são aqueles com mais pontos e como as regras apresentadas estabelecem uma progressão de pontos na direção aos líderes, garante-se que ele seja atingido no final do roteamento da mensagem de um nó comum, como visto na Figura 4.6.

As pontuações resultantes do Sistema de Regras aplicado distribuidamente em cada nó dependem diretamente da relação de similaridade de leituras de um nó com seus vizinhos. Desta maneira, o reflexo da configuração atual da rede nos agrupamentos lógicos dos nós é diretamente dependente da troca periódica de *beacons* da Unidade de Sincronização. Para que os agrupamentos lógicos se adaptem com maior dinamicidade às mudanças de leituras e para que as pontuações dos nós tenham convergência mais rápida para uma configuração estável, é possível que o intervalo entre o envio das mensagens *beacon* seja ajustado de maneira adaptativa, adiantando ou, até mesmo, atrasando envios de *beacons*, no caso de situações estáveis. Para isso, a Unidade Adaptativa determina como tais intervalos podem ser atrasados ou adiantados autonomamente.

4.2.4 Adaptação dos intervalos de atuação

A Unidade Adaptativa verifica se o Controlador de *Beacons* da Unidade de Sincronização deve adiantar ou atrasar a sua atuação e consequente transmissão de *beacon*. Deve-se ressaltar que a Unidade de Indexação atua no mesmo intervalo que a Unidade de

Sincronização, pois as informações de pontuação da vizinhança dos nós são enviadas em *piggyback* no mesmo *beacon*. O Agente adaptativo mantém atualizadas as informações de similaridade de leituras acerca das vizinhanças dos nós, de maneira que a Unidade de Sincronização estabeleça agrupamentos lógicos globais satisfazendo as restrições de similaridade de dados, para que a Unidade de Indexação possa atualizar sua estrutura de pontuação de maneira dinâmica, mantendo rotas dos nós comuns até os nós líderes de seus agrupamentos lógicos. Ressalta-se que todas as verificações da Unidade Adaptativa ocorrem com um intervalo equivalente a $1/3$ do intervalo do Controlador de *Beacons*, para que existam janelas de tempo suficientemente grandes entre as verificações, para que mudanças significativas sejam mais prováveis.

Através do Controlador de Adiantamentos, a Unidade Adaptativa realiza dois testes para determinar se o Controlador de *Beacons* da Unidade de Sincronização deve ser adiantado. Caso a leitura atual do nó comparada à leitura agregada de sua vizinhança não satisfaça a relação de similaridade estabelecida pela Equação 4.1 considerando um $CTresh' = 1.5 CTresh$ mais relaxado no lugar do $CTresh$ padrão, então a operação do Controlador de *Beacons* é adiantada para ocorrer imediatamente. Neste caso, a relação de similaridade do nó com o seu atual agrupamento já não corresponde à relação estabelecida no último envio de *beacons* e, desta forma, tanto o agrupamento lógico do nó estabelecido na Unidade de Sincronização quanto as pontuações estabelecidas pela Unidade de Indexação devem ser ajustadas para refletir essa mudança.

Enquanto o caso anterior realiza um adiantamento imediato, existem outras condições menos críticas que também podem acarretar em mudanças de configuração. Desta maneira, caso a Unidade de Indexação tenha atualizado a pontuação do nó atual no seu período de atuação anterior ou o nó atual tenha recebido um *beacon* de algum vizinho desde o período em que o seu Controlador de *Beacons* atuou pela última vez, o período de espera atual até a nova atuação deste controlador é reduzido pela metade. Isto é necessário pois a mudança de pontuação de um nó individual muitas vezes acarreta em novas mudanças na vizinhança até que os agrupamentos lógicos e as pontuações dos nós converjam para uma configuração estável, i.e., sem mudanças constantes nas pontuações dos nós. Assim, o recebimento de um *beacon* pode indicar que houve uma mudança na vizinhança, seja a mudança referente à pontuações ou leituras de nós vizinhos. O adiantamento destas operações aumenta a dinamicidade do protocolo como um todo pois assim as mudanças de topologia e de leituras são refletidas mais rapidamente nas estruturas utilizadas pelo DDFC.

Dados estes adiantamentos, configurações estáveis são obtidas mais rapidamente e, uma vez obtidas, os *beacons* podem ser enviados depois de um intervalo de tempo maior. Por isso, caso nenhum dos casos de adiantamento ocorra, o período de espera atual para a atuação do Controlador de *Beacons* é ajustado para o triplo do atual. Entretanto, enquanto adiantamentos de períodos podem ocorrer acumulativamente em um mesmo

período de atuação, o atraso pode ocorrer apenas uma vez, para que a dinamicidade da adaptação a mudanças não seja comprometida. O funcionamento da Unidade Adaptativa é ilustrado no fluxograma apresentado na Figura 4.7. Deve-se notar que os losangos com contorno pontilhado indicam que a condição considera mudanças desde a última execução do fluxograma, ou seja, se a pontuação do nó mudou ou se um *beacon* foi recebido desde a última execução deste fluxograma, que ocorre com intervalos de tempo T , correspondentes a $1/3$ do período original fixo do Controlador de *Beacons*.

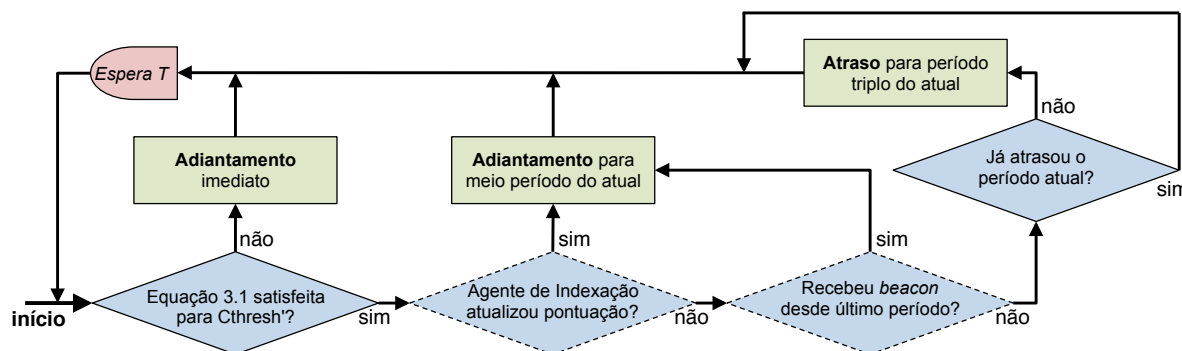


Figura 4.7: Fluxograma de funcionamento da Unidade Adaptativa

Desta maneira, a Unidade Adaptativa segue uma sequência de verificações que ocorrem em um intervalo correspondente a $1/3$ do período original fixo do Controlador de *Beacons*. Embora uma abordagem proativa pudesse ser utilizada, este funcionamento periódico é preferível pois ele garante um funcionamento mais estável e bem comportado: como a recepção de um *beacon* acarreta no adiantamento do período do Controlador de *Beacons* individual do nó, a operação proativa de adiantamento de período na recepção de cada *beacon* seria imprevisível devido ao tamanho da vizinhança e falta de sincronia nos envios. Desta maneira, a verificação da recepção em um intervalo de tempo amplo permite que o protocolo atue de maneira mais estável e, ainda assim, dinâmica.

4.3 Resumo

Este capítulo apresentou a descrição do protocolo de agrupamento bio-inspirado DDFC, que atua de maneira dinâmica quanto às leituras dos nós sensores. O funcionamento deste protocolo é baseado em três agentes principais, sendo eles a Unidade de Sincronização, para a sincronização da agregação de leituras similares em cada nó, a Unidade de Indexação, para o estabelecimento de líderes e conseqüente roteamento dos nós comuns até eles e, por fim, a Unidade Adaptativa, que regula dinamicamente o intervalo de atuação de cada um dos agentes anteriores, cujo funcionamento depende da troca de mensagens *beacon*, análogas aos piscares dos vagalumes. O próximo capítulo apresenta a avaliação de desempenho do protocolo DDFC.

Capítulo 5

AVALIAÇÃO DE DESEMPENHO

Este capítulo apresenta a avaliação de desempenho do protocolo de agrupamento DDFC para RSSFs. Esta análise leva em conta diversos parâmetros e métricas, sendo o DDFC comparado a uma variante denominada DDFC-K, que emprega as regras padrões do KHOPCA na Unidade de Indexação, e ao protocolo SCCS, apresentado na Subseção 3.4.1. A avaliação considera os agrupamentos formados, analisando tanto a similaridade das leituras, obtida pela Unidade de Sincronização, quanto a qualidade do roteamento interno aos agrupamentos, referente à Unidade de Indexação. A Seção 5.1 descreve o cenário utilizado para a simulação. A Seção 5.2 apresenta os parâmetros do protocolo DDFC e as métricas utilizadas para avaliá-lo. Por fim, a Seção 5.3 apresenta e discute os resultados obtidos.

5.1 Cenário da simulação

Para a avaliação de desempenho do protocolo de agrupamento DDFC, os protocolos e variantes analisados foram inteiramente implementados no simulador NS-3, versão 3.14.1. O cenário de avaliação representa uma situação de monitoramento de ambiente realística, sendo baseada em leituras reais de dados, a fim de se determinar a eficiência dos agrupamentos lógicos formados quanto às relações de similaridade de dados estabelecidas e à qualidade das escolhas de líderes. Assim, o cenário de simulação utilizou as leituras de dados coletadas pelo laboratório Intel, da universidade de Berkeley, disponibilizados em [58]. Tais leituras correspondem a valores de umidade, temperatura, luminosidade e voltagem da bateria dos nós em questão, por um período pouco superior a um mês. O cenário, onde as leituras foram realizadas, é ilustrado na Figura 5.1, onde os hexágonos pretos representam os nós.

Com este conjunto de leituras, o cenário empregado na simulação é composto de 54 nós. Como no cenário original, um raio de transmissão padrão seria capaz de cobrir a área toda, para qualquer nó, comprometendo os resultados obtidos, este cenário foi expandido em uma escala de 15:1, i.e., multiplicando as componentes x e y e cada coordenada por 15, assim resultando em uma região retangular de 630m por 480m, com as posições individuais de cada nó também devidamente escaladas. Foi definido assim um raio de transmissão de 100m, possibilitando uma avaliação que ainda possui relação de similaridade espacial e temporal entre as leituras realizadas.

As simulações realizadas consideram apenas as leituras de umidade e temperatura coletadas nos primeiros 1200s do período total das leituras originais. Estas leituras foram

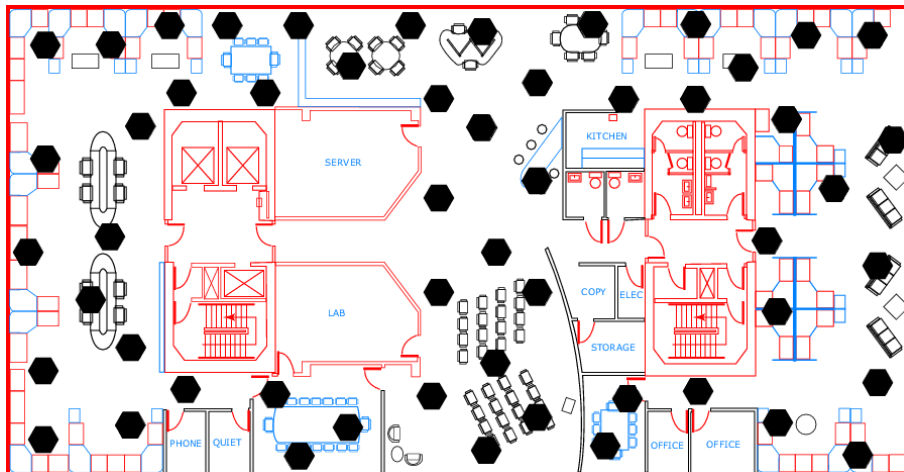


Figura 5.1: Cenário base para a simulação [58])

escolhidas pois elas demonstraram maior heterogeneidade e ainda assim grande relação espacial. Os resultados da avaliação a seguir são referentes apenas às leituras de umidade. Podem ser encontrados em anexo os resultados referentes à temperatura, que demonstraram ser similares aos resultados obtidos para a umidade.

Nota-se que não foi utilizado um protocolo de roteamento inter-agrupamentos pois o foco da avaliação se restringe ao escopo intra-agrupamento, avaliando características dos agrupamentos formados, como será visto em 5.2. Por fim, as leituras de tais nós foram coletadas em tempos discretos e diferentes para cada nó. Elas utilizam um conceito de épocas, sendo que para cada época, ou instante de tempo, apenas a leitura de um dos nós da rede foi armazenada e disponibilizada. Por isso, uma operação de normalização e interpolação das leituras foi necessária.

5.1.1 Interpolação contínua de leituras discretas

Para que os nós sejam capazes de obter uma leitura em qualquer intervalo de tempo, ainda assim mantendo as relações de similaridade espacial e temporal, uma operação de interpolação de leituras para qualquer instante de tempo é necessária. Desta maneira, tomando *currentTime()* como o tempo atual da simulação e uma variável *sR* que corresponde à lista de leituras ordenadas (*time, reading*) armazenadas no nó, o Algoritmo 3 foi criado a fim de possibilitar a obtenção de uma leitura interpolada para qualquer instante contínuo dentro do intervalo de simulação.

Neste algoritmo, o procedimento *lerp* (l.1-3) é responsável pela interpolação entre os valores *from* e *to*, sendo que o *percentageOfTo* varia no intervalo $[0, 1]$, representando a porcentagem que a ser considerada do valor *to*; analogamente, a porcentagem para o valor *from* é $1 - \text{percentageOfTo}$. No procedimento *getReading*, as leituras armazenadas em *sR* são inicialmente percorridas (l.6-8), e aquelas que correspondem a um tempo passado são descartadas. Após isso, é definida a porcentagem considerada da leitura posterior a

Algoritmo 3 Interpolação de leituras

```

1: procedimento LERP(from, to, percentageOfTo)
2:   retorna  $((1 - \text{percentageOfTo}) * \text{from} + \text{percentageOfTo} * \text{to})$ 
3: fim procedimento
4:
5: procedimento GETREADING
6:   enquanto  $(\text{currentTime}() > sR.time)$  faça
7:      $sR \leftarrow sR.next()$ 
8:   fim enquanto
9:    $pct \leftarrow (\text{currentTime}() - sR.time) / (sR.next().time - sR.time)$ 
10:  retorna  $\text{lerp}(sR.reading, sR.next().reading, pct)$ 
11: fim procedimento

```

atual (l.9), levando em conta quanto tempo faltaria até o instante no qual aquela leitura foi obtida no cenário original. Assim, a interpolação é finalmente aplicada entre a leitura atual $sR.reading$ e a posterior $sR.next().reading$ (l.10).

5.1.2 Extensão do visualizador gráfico PyViz

O PyViz é uma das ferramentas existentes para a visualização gráfica de simulações no NS3. Ele é escrito em Python e bem integrado às classes implementadas em C++, que representam os agentes de roteamento e de aplicação, por exemplo. Para a análise de resultados e a visualização gráfica dos agrupamentos lógicos formados e das agregações de leituras estabelecidas, foi implementado também um *plugin* em Python para o PyViz. Além da visualização normal da topologia que já é oferecida naturalmente por ele, este *plugin* apresenta no visualizador as arestas de similaridade de leituras entre aqueles nós que satisfazem a equação de similaridade apresentada no capítulo anterior. Ademais, tal *plugin* colore os nós em tons progressivos que refletem a leitura atual.

A Figura 5.2 apresenta um estado discreto do visualizador PyViz com o *plugin* incorporado. Cada nó possui como legenda, respectivamente: o índice do nó, (a sua leitura atual), [a leitura média agregada de sua vizinhança], {a pontuação da Unidade de Indexação}. Na instância de cenário ilustrada, foram tomados os parâmetros $MaxK = 2$ e $CThresh = 1.0$. Nota-se também que o nó de índice 4 possui leitura 0, isso se deve a uma falha em sua operação no cenário real de Berkeley, não permitindo que ele realizasse leitura alguma.

5.2 Parâmetros e métricas

O cenário, baseado nas leituras coletadas pelo laboratório Intel de Berkeley, é variado com relação a quatro parâmetros de funcionamento do protocolo DDFC: **(i)** $CThresh$, que indica a relação de similaridade entre os dados, **(ii)** $adap$, que informa o uso da Unidade Adaptativa, **(iii)** int , que indica a duração do intervalo fixo de atuação do Controlador de *Beacons*, e **(iv)** $MaxK$, que indica a distância máxima em saltos até um nó líder, tal que

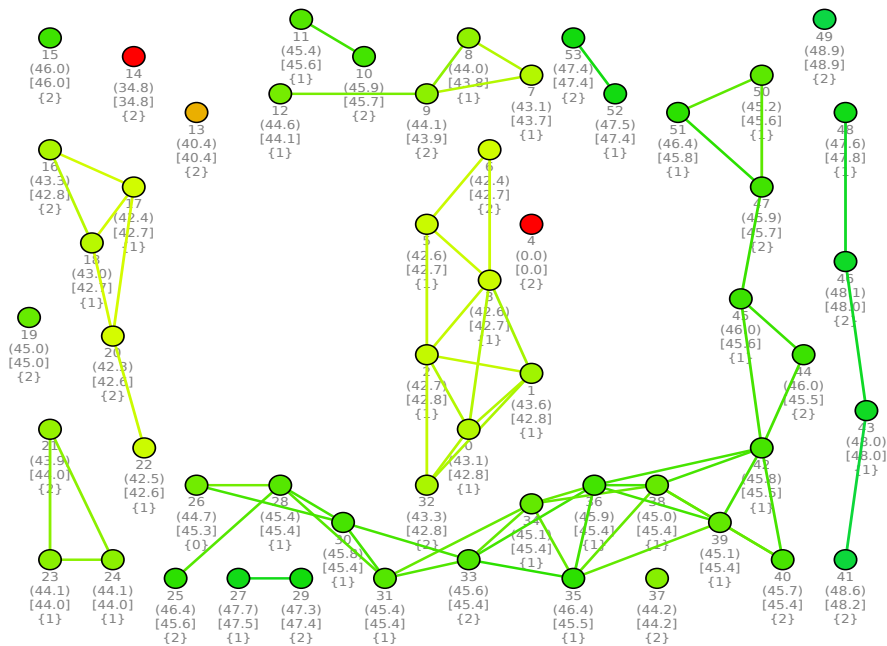


Figura 5.2: Visualização do funcionamento do DDFC no PyViz

a quantidade máxima de saltos é $MaxK + 1$. Os parâmetros das simulações, assim como a variação utilizada de cada parâmetro do protocolo, podem ser vistos na Tabela 5.1.

Parâmetros	Valores
Quantidade de nós	54
Tempo de simulação	1200s
Área de simulação	630m x 480m
Raio de transmissão	100m
$CThresh$	0.5, 1.0, 1.5, 2.0
$adap$	sim, não
int	6s, 12s, 24s, 48s, 96s
$MaxK$	1, 2, 3

Tabela 5.1: Parâmetros utilizados no cenário de simulação

Além desta variação de parâmetros para o DDFC, foi implementada uma outra variante do protocolo DDFC, denominada DDFC-K, que utiliza as regras originais do KHOPCA na Unidade de Indexação. Essa variante visa verificar se as alterações propostas foram capazes de melhorar a estabilidade e qualidade dos líderes escolhidos. O protocolo SCCS também foi implementado para a comparação de resultados. Ele foi escolhido por apresentar uma operação de manutenção de agrupamentos, que leva em conta as leituras similares como critério para agrupar os nós. O $CThresh$ é um parâmetro comum ao SCCS, compartilhando o mesmo significado e variado no mesmo intervalo apresentado na tabela anterior. O parâmetro int , no caso do SCCS, significa o intervalo

entre as verificações de similaridade de dados no agrupamento, que acarretam na sua possível fragmentação, variando também no mesmo intervalo especificado para o DDFC.

O artigo do protocolo SCCS avaliou métricas relativas principalmente quanto à duas características: (i) ao número de instruções computacionais utilizadas, número muito variável de acordo com arquiteturas de computadores e metodologia de cálculo para tal; e (ii) à características específicas da agregação temporal de dados proposta, baseada na otimização de segmentos de reta entre os pontos correspondentes às leituras leituras (mais detalhes são encontrados no artigo do SCCS [26]). Entretanto, a implementação da otimização por segmentos de reta é irrelevante no escopo desta dissertação, pois apenas a similaridade espacial é tratada pelo DDFC. Desta forma, a validação direta, i.e., analogia e comparação das mesmas métricas é impossibilitada. Contudo, como a análise dos resultados visa verificar métricas referentes à topologia formada pelos protocolos, validou-se que para o cenário apresentado na explicação do protocolo SCCS no artigo [26], a implementação realizada no NS3 foi capaz de formar a mesma configuração final de topologia, dada a inicial. O estado inicial deste cenário é ilustrado na Figura 5.3, tal que cada área contígua representa um diferente intervalo de leituras.

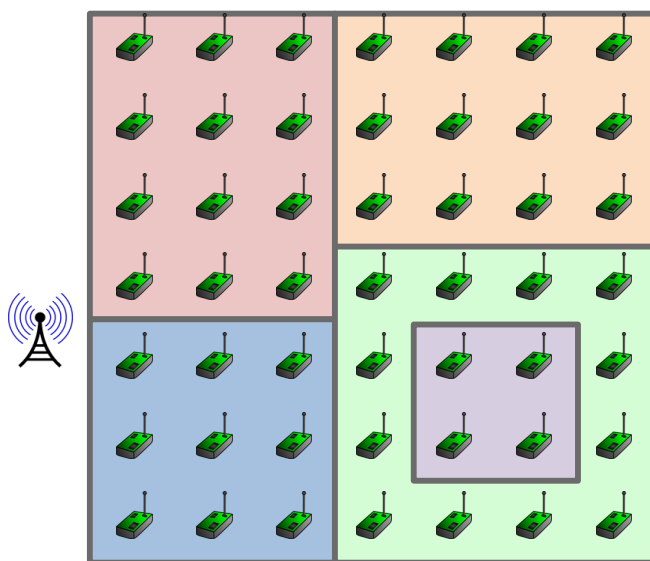


Figura 5.3: Topologia considerada na explicação do SCCS [26]

A partir de tais variações de parâmetros e protocolos, são definidas oito métricas: **número de líderes**, **número de agrupamentos**, **número de nós solitários**, **número de saltos até o líder mais próximo**, **duração dos líderes**, **amplitude média das leituras dos agrupamento**, **inconsistência de rotas** e **número de *beacons* enviados**. Estas métricas visam determinar o comportamento do protocolo com relação à variação das métricas, e o desempenho com relação à dinamicidade do protocolo para se adaptar às variações de leituras, ao *overhead* gerado e à qualidade das rotas até os líderes.

O número de líderes, de agrupamentos e de nós solitários são avaliados não só por

questões de desempenho, mas também a fim de se adequar ao propósito da aplicação - quanto à granularidade dos agrupamentos e consequente número de líderes estabelecidos, pro exemplo. O número de saltos até o líder mais próximo tem relação direta com o parâmetro $MaxK$, sendo avaliado para se determinar um valor apropriado para ele. A duração dos líderes expressa quanto tempo um líder se manteve até ter sua pontuação reduzida, dado que durações médias maiores indicam que nós apropriados foram selecionados pela Unidade de Sincronização. A amplitude das leituras dos agrupamentos expressa a diferença média entre as maiores e as menores leituras nos agrupamentos, sendo importante para se determinar a corretude do comportamento da sincronização da agregação de leituras da Unidade de Sincronização, isto é, se tal valor possui sincronia o bastante para manter apenas nós de leituras similares agrupados. A inconsistência de rotas corresponde à quantidade média de nós que não conseguem atingir seu líder dada a configuração da pontuação na rede em dado momento discreto, devendo ser minimizada. Por fim, o número de *beacons* enviados corresponde ao *overhead* gerado pelo mecanismo.

5.3 Avaliação de resultados

Esta seção apresenta a avaliação do comportamento e do desempenho do protocolo DDFC quanto aos parâmetros e métricas apresentados, sendo comparado ao desempenho do DDFC-K e do SCCS. Os resultados apresentados foram obtidos a partir de 35 simulações realizadas para cada combinação de parâmetros. Com isso, são mostrados nos gráficos intervalos de confiança de 95%. As subseções a seguir apresentam os resultados, que foram agrupados nas categorias referentes aos **agrupamentos formados**, aos **nós agrupados**, às **rotas inconsistentes**, e às **mensagens beacons**. Em cada subseção é inicialmente apresentada a análise do protocolo DDFC para posterior comparação à variante DDFC-K e ao protocolo SCCS. Por fim, são apresentadas as combinações de parâmetros com os melhores resultados obtidos para o protocolo DDFC.

5.3.1 Agrupamentos formados

A Figura 5.4 apresenta um conjunto de gráficos que avalia a influência do parâmetro $CThresh$ no número de líderes eleitos, de agrupamentos formados, e de nós solitários - isto é, cujo agrupamento é formado apenas por ele só. À esquerda, nota-se que quanto maior o $CThresh$, menor o número de líderes na rede. Ademais, o $MaxK$ gera a mesma influência, sendo mais notável entre os patamares de $MaxK = 1$ e $MaxK = 2$. Isso acontece porque quanto maior for o parâmetro $CThresh$, menos agrupamentos existirão, como visto no gráfico central, pois nós com leituras mais distantes serão agrupados. O $MaxK$ atua de acordo com as regras propostas na Unidade de Indexação, sendo que quanto maior for este valor, menor é o número de líderes. Entretanto, nota-se que neste

cenário a diferença é acentuada apenas para $MaxK = 2$, quando $MaxK = 3$ não há uma diferença significativa no número de líderes, indicando que mesmo que os nós comuns possam utilizar mais saltos para atingir um líder, são poucos os nós que de fato utilizam o número máximo de saltos. Por fim, o número de nós solitários tende a diminuir na medida que o parâmetro $CThresh$ aumenta, visto que com um valor maior para este parâmetro, nós com leituras mais divergentes podem ser agrupados com mais facilidade.

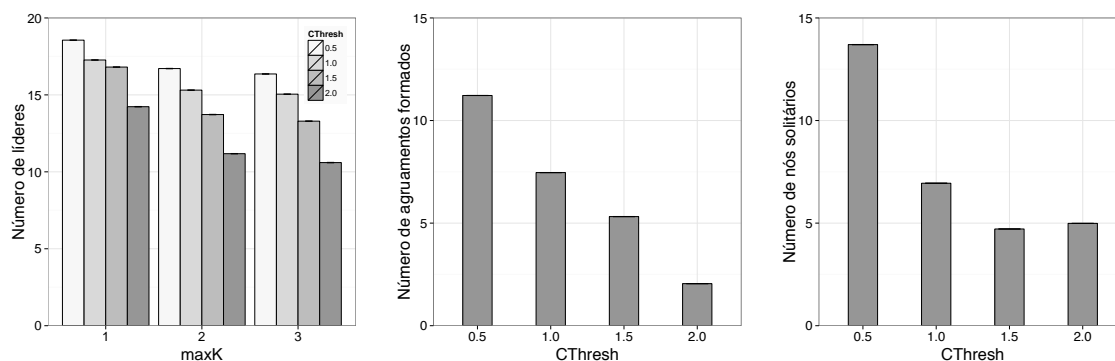


Figura 5.4: Relação do número de líderes, de agrupamentos formados e de nós solidários

Nestes gráficos, o imperceptível intervalo de confiança apresentado indica que em 95% das vezes, serão obtidos dados muito similares aos apresentados nos gráficos. Esta precisão se deve ao fato de o cenário ser estático, isto é, não possuir mudanças quanto à posição dos nós e quanto às suas leituras. Entretanto, a operação do protocolo DDFC não é determinística, ou seja, ela depende de fatores de aleatoriedade. Por isso, tal intervalo de confiança ínfimo indica que o protocolo opera de maneira controlada e estável.

Comparação com DDFC-K e SCCS

A Figura 5.5 apresenta os resultados obtidos para a variação DDFC-K. Pode-se notar que a variante DDFC-K possui resultados similares aos resultados do DDFC, isto é, com o mesmo intervalo de confiança. Isto indica que as alterações nas regras propostas não chegam a modificar a hierarquia da rede como um todo, não sendo evidentes nestas métricas.

O protocolo SCCS, por outro lado, mostrou um comportamento muito distinto, como visto na Figura 5.6. Inicialmente, o número de líderes estabelecidos é menor, fato justificado pela hierarquia mais complexa do SCCS, que estabelece além de líderes (*cluster-heads*), nós *gateway* conectando cada par adjacente de agrupamentos. Contudo, nota-se que, além disso, a variação no número de líderes e de agrupamentos formados não acompanha o crescimento no valor $CThresh$ de maneira linear. Isto ocorre porque os gráficos apresentam valores médios destas métricas no tempo total de simulação, sendo que no SCCS existe a tendência do número de agrupamentos aumentar até um dado limite, pois

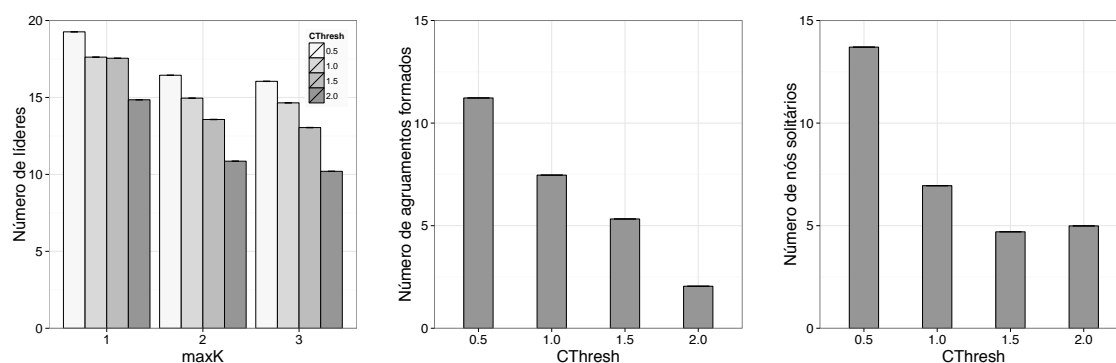


Figura 5.5: Comportamento dos agrupamentos no DDFC-K

a operação de manutenção do SCCS consiste basicamente da quebra dos agrupamentos, sem existir nenhuma união. Com isso, mesmo com um $CThresh$ grande o bastante para que existam poucos agrupamentos, eles são continuamente fragmentados em um número maior de agrupamentos, sem uniões de agrupamentos similares para normalizar tal média.

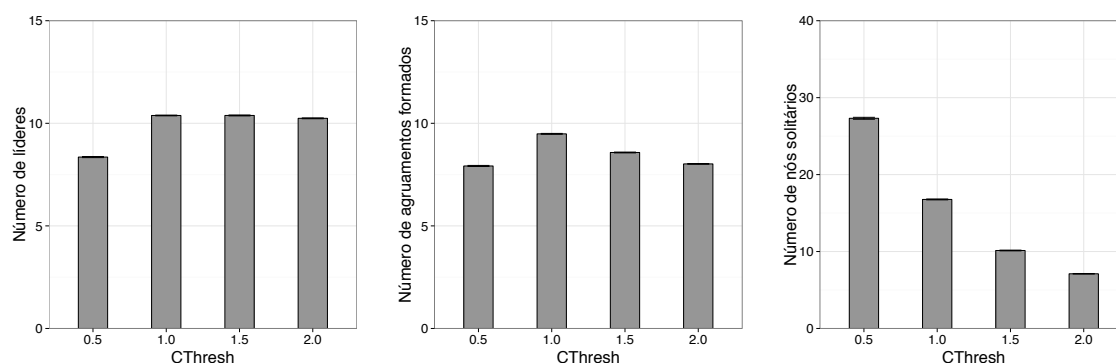


Figura 5.6: Comportamento dos agrupamentos no SCCS

A métrica de nós solitários demonstrou desempenho inadequado no SCCS, sendo que para valores muito pequenos de $CThresh$, muitos nós da rede não são capazes de se unir a um agrupamento. Isso ocorre pois, além de só ser possível a quebra de agrupamentos antes do processo de reagrupamento global ser disparado pela estação-base, o algoritmo do SCCS não garante que os nós se consolidem em um estado permanente, observou-se que muitos deles se mantiveram no estado temporário GWR (candidato a *gateway*) até que a estação-base disparasse outro processo de reagrupamento completo da rede. Ademais, será mostrado adiante que como o SCCS não emprega a média de leituras do agrupamento como um todo, o $CThresh$ acaba exercendo uma restrição maior.

5.3.1.1 Distância média até os líderes e duração dos líderes

A Figura 5.7 indica a relação de quantos nós existem para cada quantidade de saltos até que um líder seja atingido. Cada linha vertical representa uma combinação dos parâmetros

informados no eixo X, e o eixo Y informa a quantidade de nós que possuem determinada distância até o líder mais próximo. A distância em si é expressa pelo tamanho do ponto utilizado no gráfico, e a cor do ponto indica o valor do parâmetro $CThresh$. Nota-se de imediato que a maior concentração de nós, na maioria dos casos, está a 2 saltos do líder mais próximo, exceto quando é utilizado um valor de $CThresh = 0.5$. Neste caso, a maioria dos nós se encontra a apenas um salto do líder mais próximo. Isso se deve ao fato de que com um $CThresh$ muito pequeno, existirá um número muito maior de agrupamentos e, desta forma, mais líderes e mais nós próximos aos líderes.

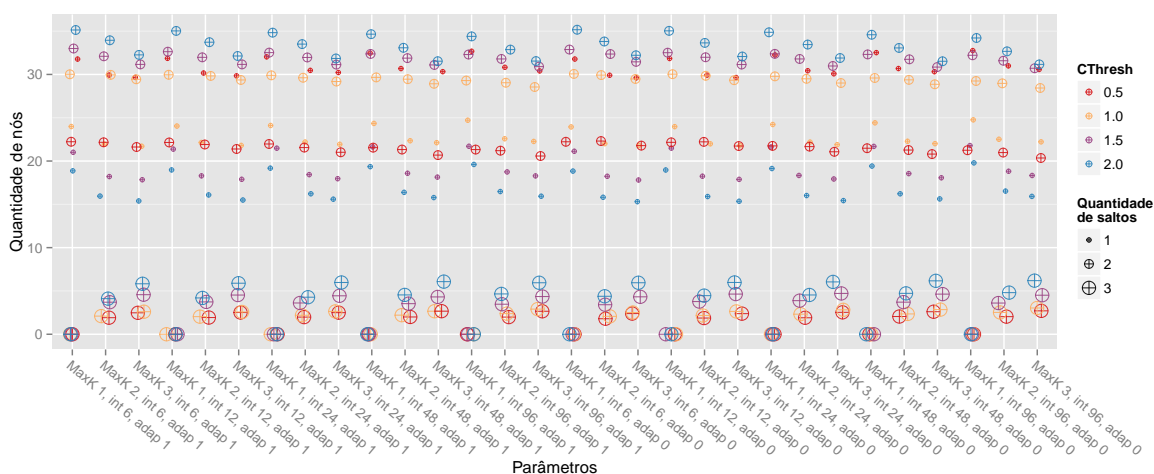


Figura 5.7: Quantidade de nós vs. distância até o líder mais próximo

Percebe-se também que embora para $MaxK$ maiores exista a tendência de ocorrerem mais saltos com maiores distâncias até o líder mais próximo, para $MaxK = 3$ a distância média já não é tão maior do que para $MaxK = 2$. Ademais, embora com um parâmetro $MaxK = 3$ seja possível que existam nós com até 4 saltos de distância até o líder mais próximo, este caso se mostrou quase inexistente, sendo removido do gráfico para evitar poluição visual. Por fim, percebe-se que os parâmetros $adap$ e int não exerceram grande influência na distância média de saltos até o líder mais próximo.

A Figura 5.8 apresenta um conjunto de histogramas que avalia a relação entre o número de líderes e a sua duração, determinada em quantidades de turnos, dado que um turno representa um intervalo de tempo de 10s. O conjunto de histogramas é apresentado em um quadro que varia horizontalmente o parâmetro int e verticalmente o parâmetro $CThresh$. De imediato, nota-se que em todos os casos a maior concentração de nós se encontra na duração de 120 turnos, que neste caso corresponde ao tempo de vida inteiro da rede. A grande duração dos nós líderes, estabelecidos de acordo com a pontuação da Unidade de Indexação, indica que as regras utilizadas permitem determinar nós capazes de se manter estáveis como líderes, mesmo com a dinamicidade das leituras. Isso se deve em especial à quarta regra utilizada pela Unidade de Indexação, que dá prioridade como líder àqueles

nós que possuem maior vizinhança com leituras comuns.

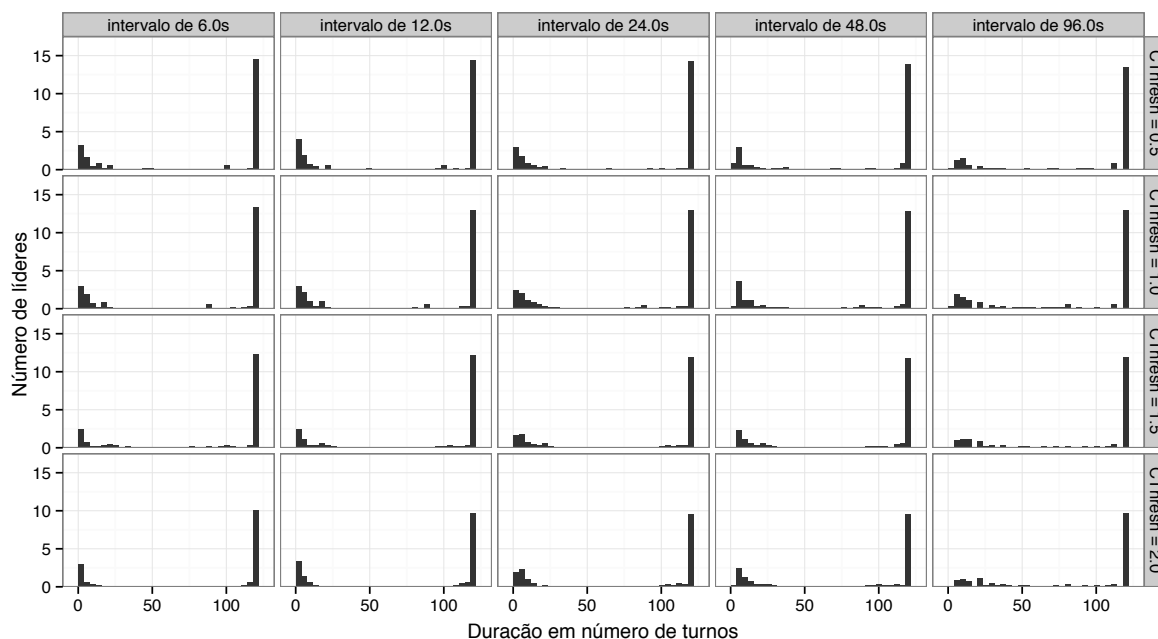


Figura 5.8: Duração dos líderes em número de turnos de 10s

Quando o $CThresh$ aumenta, o número de líderes diminui bastante. Embora seja mais aparente a diferença no número de nós de duração máxima, essa diminuição se dá em todos os casos, e ocorre pois com um $CThresh$ maior, existirão menos líderes na rede, como visto na Figura 5.4. Quando são considerados intervalos maiores para a atuação do Controlador de *Beacons*, a duração dos líderes é mais dispersa nos casos intermediários, diminuindo não só a quantidade de líderes com duração muito baixa, isto é, até 10 turnos, como também a quantidade de líderes de duração máxima.

Comparação com DDFC-K e SCCS

Para a métrica de distância dos nós comuns até o líder mais próximo, a variante DDFC-K apresentou os mesmos resultados que o DDFC e, desta forma, o gráfico foi omitido. Justifica-se, entretanto, que este comportamento é decorrente da hierarquia básica similar estabelecida pelo DDFC-K, como foi visto na Figura 5.4. O SCCS, por outro lado, não foi avaliado quanto à esta métrica pois não seria possível realizar uma comparação direta significativa entre os dois, dado que o SCCS possui diversas categorias de nós e, principalmente, não chega a consolidar uma topologia com número de líderes e agrupamentos estável devido às operações exclusivas de quebras de agrupamentos.

No caso da estabilidade dos nós líderes estabelecidos, por outro lado, tanto o SCCS quanto o DDFC-K apresentaram menor estabilidade. Na Figura 5.9 é apresentado o histograma da duração dos líderes para a variação DDFC-K, que corresponde ao protocolo

DDFC utilizando as regras originais do KHOPCA na Unidade de Indexação. Comparando este histograma ao do protocolo DDFC com as modificações de regras propostas, pode-se observar que, embora os parâmetros *int* e *CThresh* tenham exercido a mesma influência no resultado, a duração dos líderes observadas no DDFC-K é sempre inferior, indicando pior escolha dos líderes. Isso se deve ao fato do DDFC empregar regras modificadas que otimizam a estabilidade ao dar prioridade na escolha dos líderes aqueles nós que possuem maior vizinhança de leituras similares.

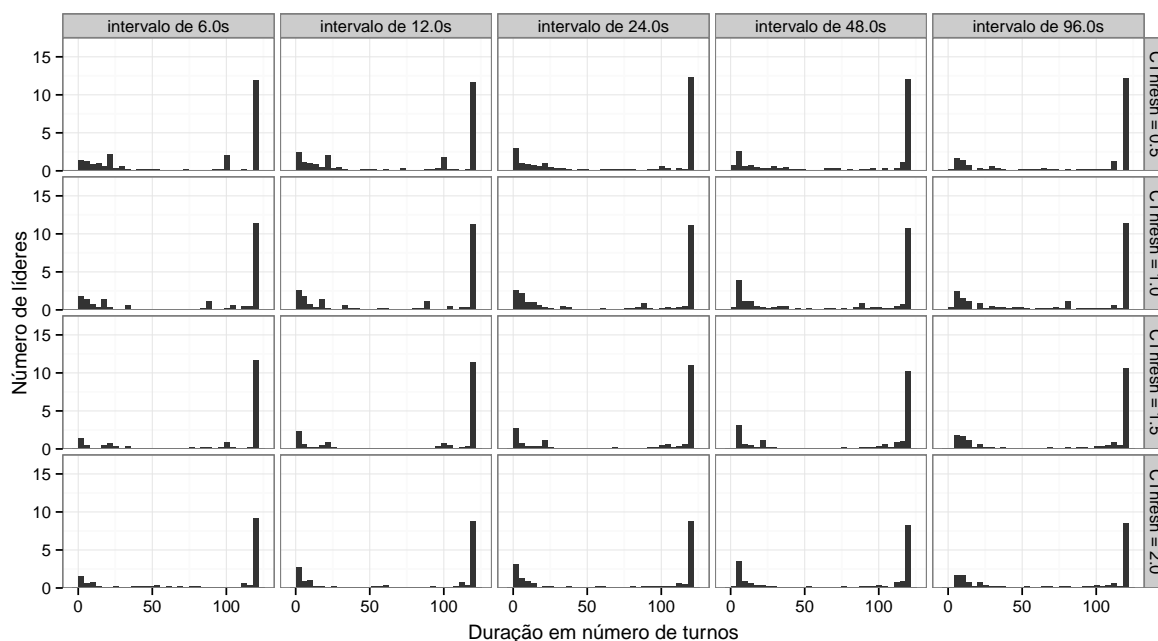


Figura 5.9: Duração dos líderes para a variação DDFC-K

A Figura 5.10 apresenta o histograma de duração dos líderes do protocolo SCCS. Nota-se de imediato a maior distribuição dos líderes em durações inferiores. O protocolo SCCS não consegue eleger nós líderes adequadamente quando o *CThresh* é mínimo pois, independente dos líderes eleitos, existe a quebra dos agrupamentos que é mandatória quando as leituras no agrupamento divergem. Além disso, ele exerce mais restrição do que no DDFC, como será visto na Subseção 5.3.2. Observa-se, contudo, um comportamento mais estável para *CThresh* = 1.0, se destacando para o caso de *int* = 6.0. Valores muito altos para *CThresh* também apresentaram instabilidade, pois quanto maior este valor, maior a extensão dos nós agrupados e, desta forma, maior também a probabilidade do agrupamento necessitar de uma quebra devido aos *thresholds* de similaridade.

De maneira geral, é difícil estabelecer padrões na duração dos nós líderes com relação aos parâmetros avaliados, o protocolo SCCS como um todo opera de modo não determinística, dependendo muito da ordem com a qual as mensagens são trocadas durante a fase de criação dos agrupamentos. Diferente do DDFC, que fragmenta e une os agrupamentos dinamicamente e de maneira simples e rápida, o SCCS pode apenas quebrá-los,

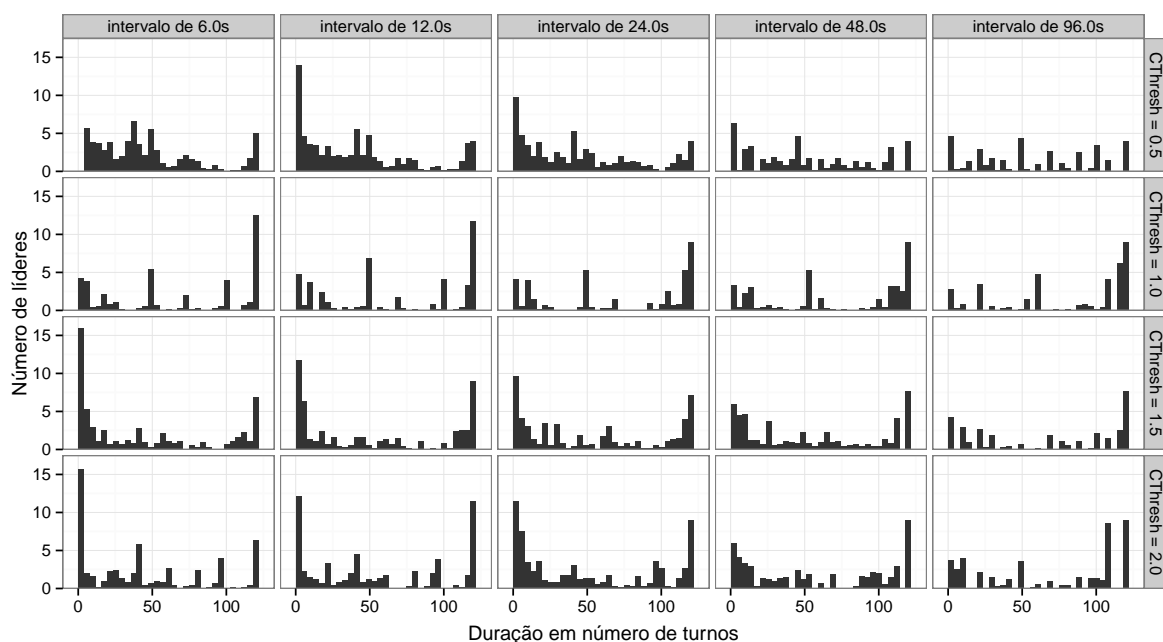


Figura 5.10: Duração dos líderes para o SCCS

dependendo de uma completa reestruturação de todos os agrupamentos para realizar algo equivalente às uniões transparentes e abstratas do DDFC.

5.3.2 Similaridade de leituras dos nós agrupados

A fim de determinar se a Unidade de Sincronização foi capaz de agrupar nós com leituras similares, é utilizada a métrica amplitude, que corresponde à diferença entre a menor e a maior leituras dentro de um agrupamento. A Figura 5.11 mostra a amplitude dos agrupamentos com relação aos parâmetros avaliados. De imediato, percebe-se que ela aumenta proporcionalmente ao $CThresh$, mas se mantém sempre inferior à $2 * CThresh$. Isso mostra que a Unidade de Sincronização foi capaz de agrupar nós com leituras similares, pois dado um valor médio m , um agrupamento aceitaria novos nós no intervalo $[m - CThresh, m + CThresh]$, cuja amplitude é exatamente $2 * CThresh$.

Nota-se que o parâmetro $MaxK$ não exerce grande influência na amplitude dos agrupamentos. Embora ele influencie no número de líderes, como visto na Figura 5.4, o número de agrupamentos se mantém o mesmo, dependendo apenas da relação de similaridade de leituras entre os nós e do $CThresh$. O uso de um intervalo adaptativo reduz a amplitude dos agrupamentos, porém quase imperceptivelmente. Isso acontece pois embora uma abordagem adaptativa permita que as mudanças de leituras e agrupamentos lógicos se tornem mais dinâmicas e rápidas, ela não exerce grande influência na amplitude em si, visto que o aumento de dinamicidade acarreta na adaptação mais rápida, apenas. Assim, tal abordagem adaptativa é benéfica quanto à detecção mais rápida de eventos anômalos.

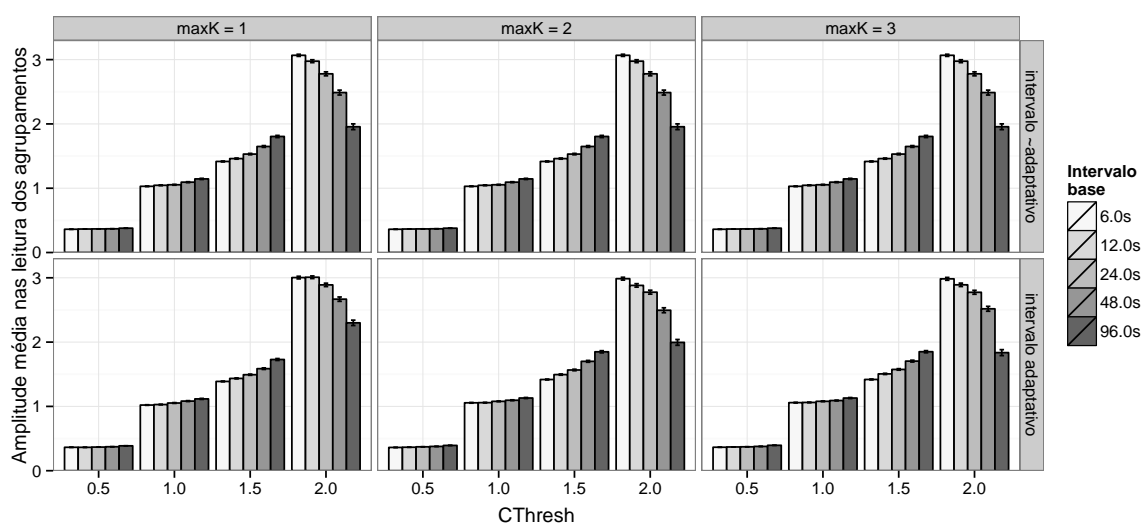


Figura 5.11: Similaridade de leituras dentro do mesmo agrupamento

Além disso, quanto maior o parâmetro *int*, intervalo base da operação do Controlador de *Beacons*, maior é a amplitude das leituras nos agrupamentos. Isso ocorre porque com intervalos de operação muito grandes, os nós demorarão mais tempo para trocarem *beacons*, independente de se a abordagem é adaptativa quanto aos intervalos ou não - a maioria dos casos de adiantamento de período ocorre de maneira relativa, diminuindo à uma fração ainda referente ao tempo total do período. Ademais, percebe-se que para $CThresh = 2.0$, o comportamento da variação de amplitude de acordo com o *int* é anômalo. Embora este comportamento não possa ser inteiramente justificado, é possível que isso ocorra pois com $CThresh = 2.0$ a rede opera com apenas 2 agrupamentos, como visto na Figura 5.4. A organização da rede em apenas dois agrupamentos lógicos é anômala devido ao tamanho imenso que esses agrupamentos atingem. Isso mostra que o parâmetro $CThresh$ deve ser ajustado de maneira sensível às leituras coletadas, sendo que $CThresh = 2.0$ é uma valoração inadequada para a operação no cenário descrito.

Comparação com DDFC-K e SCCS

Como visto na Figura 5.12, o DDFC-K apresentou amplitudes quase iguais ao DDFC. No caso do uso de intervalos não adaptativos, a amplitude encontrada é exatamente a mesma. Por outro lado, quando são utilizados intervalos de *beacon* adaptativos, existe uma pequena flutuação na amplitude encontrada, mas sem consolidar um padrão constante em todos os casos. Isso indica que isso ocorre devido à mudança nos tempos de envio dos *beacons* decorrentes da diferença na indexação realizada.

O gráfico na Figura 5.13 mostra as amplitudes dos agrupamentos estabelecidos pelo protocolo SCCS. Percebe-se que elas são sempre menores, na faixa de aproximadamente 50% da amplitude gerada pelo DDFC. Isso ocorre por dois motivos. Inicialmente, o DDFC

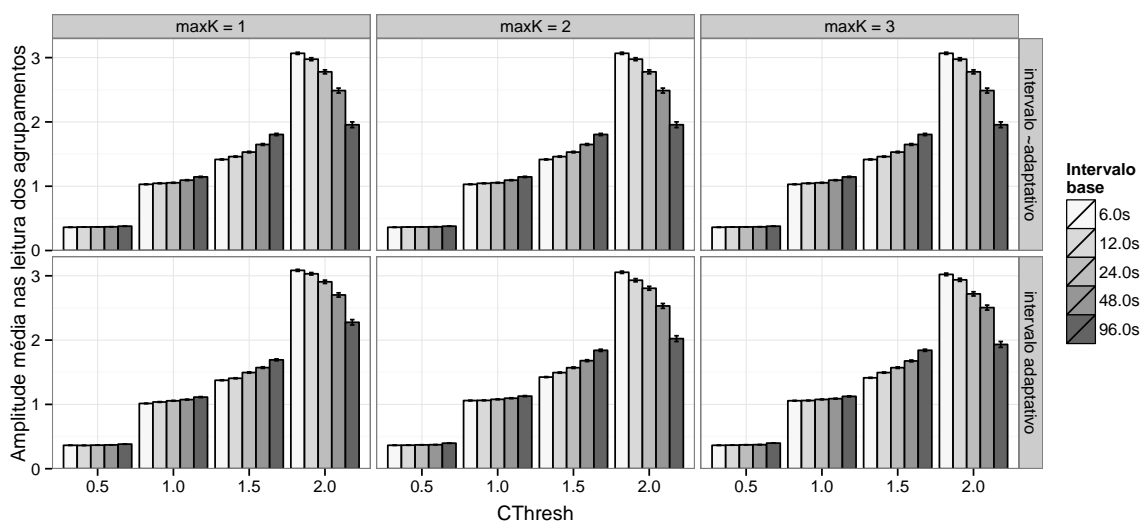


Figura 5.12: Similaridade de leituras na vertente DDFC-K

usa a média agregada das leituras no agrupamento para realizar a comparação de similaridade entre os nós sensores - o que acarreta em um comportamento mais flexível no reconhecimento de novas leituras como integrantes do agrupamento, pois o agrupamento tem uma leitura normalizada entre todos. Por outro lado, no SCCS a leitura usada para a comparação é sempre a leitura do líder que inicialmente estabeleceu o agrupamento, existindo menos flexibilidade. Além disso, como o SCCS pode apenas quebrar os agrupamentos, sem que ocorram uniões dinâmicas, os nós pertencentes ao mesmo agrupamento são em quantidade muito inferior, contribuindo para a relação de similaridade mais forte.

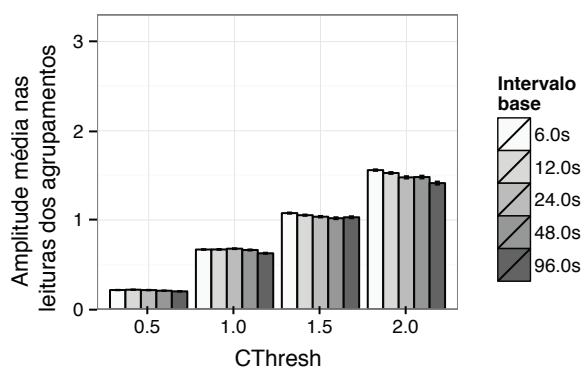


Figura 5.13: Similaridade de leituras no SCCS

Embora isto explique o mal comportamento do SCCS para o caso de $CThresh = 0.5$, que gera um intervalo muito pequeno, nos demais casos o seu desempenho não melhora de maneira significativa, mesmo se comparado ao DDFC nos demais casos. Assim, todos os protocolos e variantes respeitaram o *threshold* de similaridade imposto, mas o comportamento do SCCS é menos flexível e isso, na prática, faz com que ele considere um intervalo de cerca de 50% do que era esperado.

5.3.3 Inconsistência das rotas

Como a pontuação dos nós é dinâmica e se modifica e adapta continuamente, de acordo com a troca de *beacons*, é possível que existam estados nos quais a pontuação reflita em rotas inválidas de nós até seus líderes. Desta maneira, a Figura 5.14 mostra o número médio de rotas inválidas existentes na rede, de acordo com os parâmetros enumerados no eixo X. O tamanho das esferas indica a quantidade de rotas inválidas, enquanto a cor indica o intervalo de confiança, tal que quanto mais claro o tom de azul, maior o intervalo.

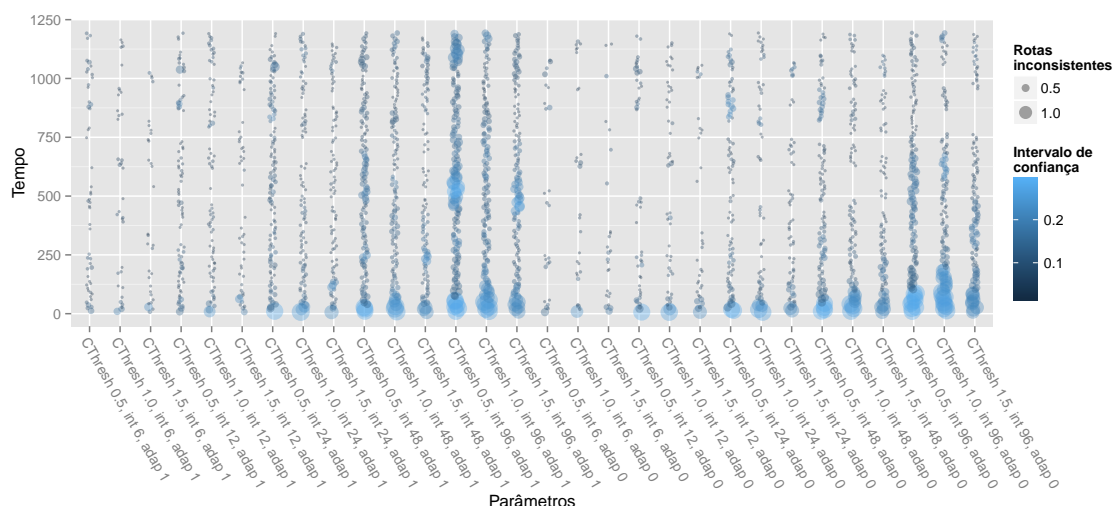


Figura 5.14: Inconsistência das rotas através do tempo

Nota-se que o período com mais rotas inválidas e maior intervalo de confiança é sempre no começo da simulação. Este é o período no qual cada nó é líder de seu próprio agrupamento, sendo que há um tempo de convergência até que os nós com leituras próximas se agrupem logicamente. Este tempo de convergência é referente ao parâmetro *int*, sendo que intervalos menores permitem que uma configuração estável seja encontrada mais rápido, da mesma forma que com um intervalo menor, menor a inconsistência das rotas no decorrer do tempo.

Além disso, conclui-se que com um valor muito pequeno para o *CThresh*, 0.5 neste caso, o número de rotas inconsistentes aumenta, devido à similaridade de dados mais estrita e conseqüente maior frequência de mudanças nos agrupamentos. Por fim, a adaptabilidade (*adap*) do intervalo de atuação do Controlador de *Beacons* diminui a inconsistência das rotas no início do tempo da rede, diminuindo o tempo de convergência para uma configuração estável. Entretanto, o número de rotas inválidas no meio do tempo de vida da rede aumenta, refletindo uma conseqüência do aumento exagerado do intervalo de atuação do Controlador de *Beacons* e conseqüente operação da Unidade de Indexação.

A Figura 5.15 mostra o valor da inconsistência média acumulada das rotas nos cenários que obedecem os parâmetros indicados - isto é, a média de rotas inválidas no tempo de

simulação inteiro. Percebe-se que quanto maior o *int*, maior a quantidade de rotas inválidas. Embora uma abordagem adaptativa melhore o tempo de convergência no começo da rede, ela acaba gerando mais rotas inválidas considerando o tempo da rede como um todo. Logo, a aplicação deve definir se a detecção adaptativa mais flexível justifica o número médio levemente maior de rotas inválidas. Por fim, este gráfico varia também o parâmetro *MaxK*. Percebe-se que, quanto maior o *MaxK*, maior a quantidade de rotas inválidas, devido ao fato das maiores distâncias possíveis entre nó comum e líder aumentarem.

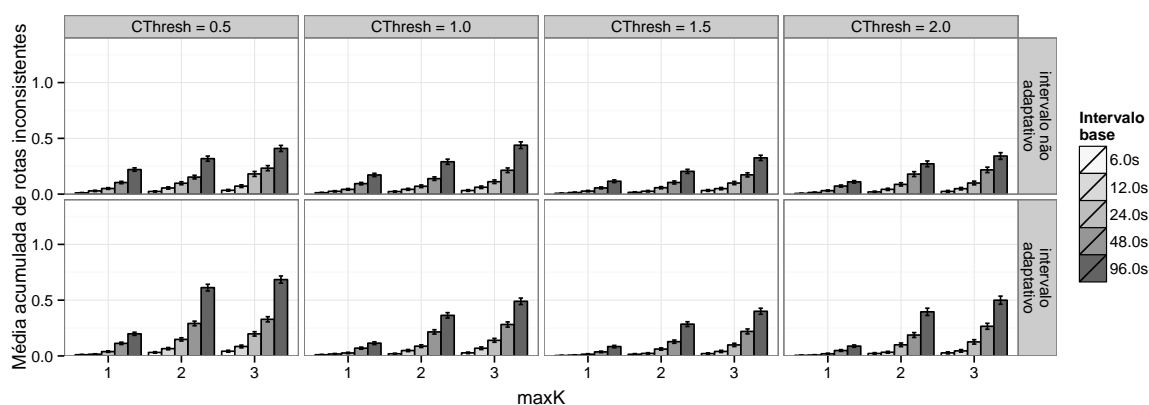


Figura 5.15: Inconsistência acumulada das rotas

Comparação com DDFC-K e SCCS

O gráfico na Figura 5.16 mostra a inconsistência acumulada das rotas para a variante DDFC-K. Em todos os casos, as modificações propostas no sistema de regras melhoraram a estabilidade dos líderes estabelecidos ao dar prioridade àqueles com maior vizinhança de leituras similares. Com isso, a inconsistência das rotas é menor no DDFC.

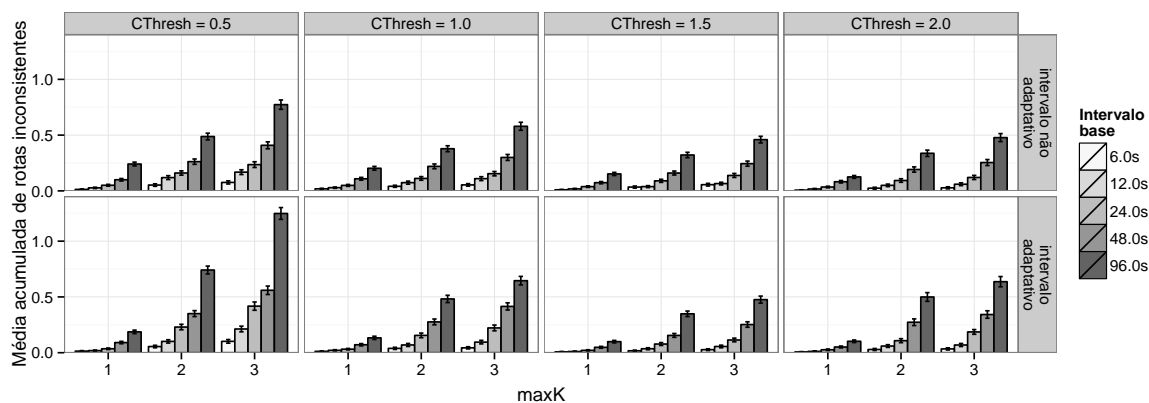


Figura 5.16: Inconsistência das rotas para a variante DDFC-K

A Figura 5.17 apresenta a inconsistência acumulada das rotas para o protocolo SCCS. Dois pontos importantes são observados. Inicialmente, a inconsistência acumulada das rotas no protocolo SCCS é muito superior, devido à sua instabilidade e constante fragmentação de agrupamentos. Além disso, não existe padrão evidente entre a inconsistência das rotas e o parâmetro *int* entre as operações de quebra de agrupamento, pois tal intervalo corresponde apenas à quebra de agrupamentos, sem que haja operações restauradoras. No DDFC, a operação de indexação empregada é dinâmica e adaptativa, mesmo que em um instante exista uma rota inválida, o estado dos nós sempre converge para uma configuração válida rapidamente, devido à abordagem adaptativa. No SCCS, não existem reparos de rotas senão a completa reestruturação dos agrupamentos.

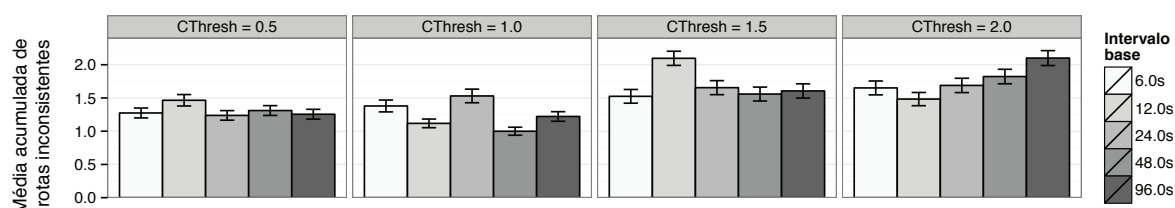


Figura 5.17: Inconsistência das rotas para o SCCS

5.3.4 Mensagens *beacon*

Na Figura 5.18 são indicadas as mensagens *beacons* enviadas através do tempo. O tamanho da esfera indica a quantidade de mensagens enviadas naquele instante e quanto mais clara a tonalidade do azul, maior o intervalo de confiança do ponto. De imediato percebe-se que para abordagens não adaptativas, o envio de *beacons* é de fato periódico, ocorrendo a cada intervalo *int*. Existem alguns pontos de tom azulado claro, apresentando certa variância, pois mesmo sendo uma abordagem periódica, um pequeno tempo aleatório na ordem de milissegundos é introduzido a cada *beacon* enviado, para evitar muitas transmissões no mesmo exato instante de tempo.

O Parâmetro *MaxK* influenciou muito na variância dos dados, sendo que quanto maior for seu valor, mais os dados variam para cada instante no tempo. Por fim, no caso de abordagens adaptativas, o número total de *beacons* enviados normalmente é inferior, porém os *beacons* são enviados com menos sincronia entre si, a depender da operação da Unidade Adaptativa. Desta maneira, o *overhead* de ambas as abordagens adaptativas e não adaptativas é similar, sendo que apenas nos casos não adaptativos o número de *beacons* enviados é sempre o mesmo.

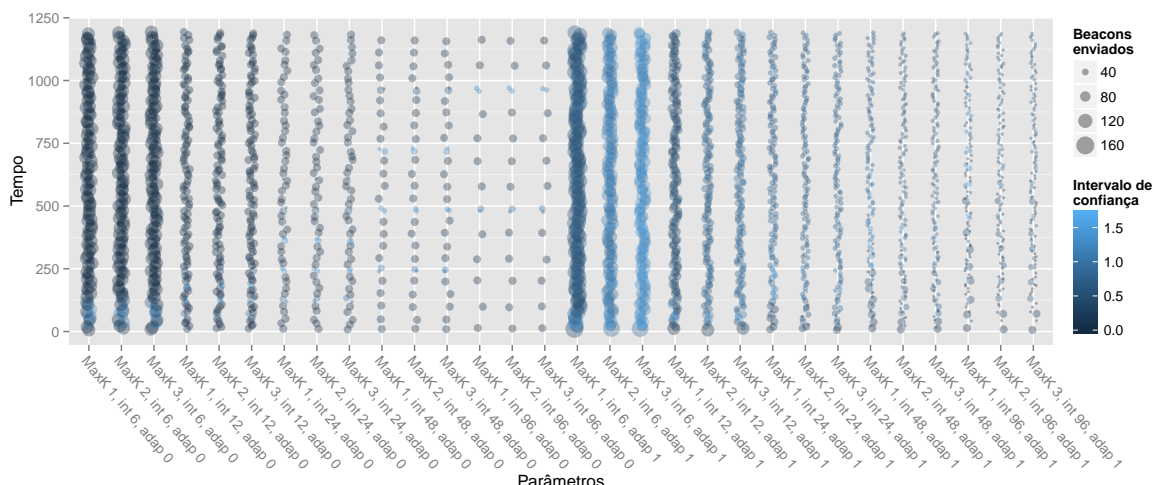


Figura 5.18: Mensagens *beacon* enviadas através do tempo

Comparação com DDFC-K e SCCS

Com relação às mensagens *beacon* enviadas, ambos os protocolos DDFC-K e SCCS operaram de maneira similar ao DDFC e, assim, seus gráficos são omitidos. Isto ocorre pois o intervalo entre as mensagens de controle enviadas é um parâmetro de mesma semântica para cada protocolo e variante e, assim, para abordagens não adaptativas, o número de mensagens de controle enviadas é o mesmo para os mesmos intervalos. No caso do DDFC-K, o uso do intervalo adaptativo alterou o tempo no qual os *beacons* foram enviados, mas ainda assim, a quantidade respeita o intervalo de confiança dos resultados para o DDFC. O protocolo SCCS possui um comportamento também similar ao DDFC sem o agente adaptativo, pois as mensagens de controle são trocadas com os mesmos intervalos descritos na variação do parâmetro *int*. Contudo, o SCCS possui mensagens de controle adicionais para a reestruturação completa dos agrupamentos - mas ainda assim em número não significativo, pois o número de tais mensagens é limitada ao número de nós da rede e a operação de recriação dos agrupamentos ocorre poucas vezes no tempo de vida da rede.

5.3.5 Discussão

Dados os quatro parâmetros estabelecidos para serem avaliados, foi feita uma análise do desempenho das 21 melhores combinações de parâmetros para o protocolo DDFC. Este estudo foi realizado quanto a quatro métricas, sendo elas, **qosb**: Quantidade de *beacons* enviados; **amp**: Amplitude média dos agrupamentos; **midDur**: Duração média dos líderes estabelecidos e **ir**: Quantidade média de rotas inválidas. Na análise, foram aplicadas operações de normalização, de maneira que cada métrica pudesse ser expressa com um valor de qualidade máximo de até 1.0.

A Equação 5.1 foi utilizada para normalizar as métricas *qosb*, *amp* e *ir*, que devem ter valores reais menores para serem melhores. Analogamente, a Equação 5.2 foi utilizada para normalizar a métrica *midDur*, que deve ter valores reais maiores para ser melhor. Tais Equações garantem que os valores normalizados de cada uma das métricas se encontram no intervalo $[\frac{1}{3}, \frac{3}{3}]$. Desta maneira, é possível selecionar as combinações dos parâmetros que apresentam os melhores resultados de acordo com estas métricas. Cabe ressaltar que tais operações de normalização atribuem a mesma importância a cada uma das métricas e consideram que cada métrica melhora linearmente de acordo com o seu valor individual, entre os valores mínimos e máximos.

$$valor_normalizado = \frac{3}{3} - \left(\frac{2}{3} * \frac{valor_individual - valor_minimo}{valor_maximo - valor_minimo} \right) \quad (5.1)$$

$$valor_normalizado = \frac{1}{3} + \left(\frac{2}{3} * \frac{valor_individual - valor_minimo}{valor_maximo - valor_minimo} \right) \quad (5.2)$$

A Figura 5.19¹ ilustra o desempenho normalizado dos melhores casos de funcionamento do DDFC, quanto à cada uma destas métricas. Cada 1/4 da esfera representa o desempenho relativo da métrica referente à esta parte do gráfico, como indicado pela legenda. O valor expresso para cada métrica é identificado pelo quão cheio está o seu quarto da esfera, sendo que esferas cheias indicam um valor máximo de 1.0 e quartos de esfera absolutamente vazios indicariam um valor mínimo 0.0.

Nota-se que embora as soluções adaptativas normalmente tenham obtido desempenho melhor nas métricas *qosb*, *amp* e *midDur*, a métrica *ir* apresenta desempenho significativamente inferior. Ademais, nenhum valor de *CThresh* superior a 1.0 apresentou um bom comportamento, indicando que quando a rede é coberta por um número muito pequeno de agrupamentos lógicos, o desempenho do protocolo é um pouco inferior. O mesmo comportamento é percebido para o parâmetro *MaxK*, que na maioria dos melhores casos assumiu o menor valor considerado pelas simulações. O parâmetro *int* atua de duas formas: enquanto valores menores de *int* melhoram a métrica *ir*, eles pioram a métrica *qosb*, havendo assim um *tradeoff* entre o número de *beacons* enviados e a consistência das rotas existentes na rede. Nota-se também que nenhum caso de *int* = 6 se enquadrou nos melhores casos, devido à quantidade grande de *beacons* enviados, que degrada a métrica *qosb*.

¹Por questões de espaço na figura, *CThresh* foi abreviado como *cTh* e *MaxK* como *mk*.

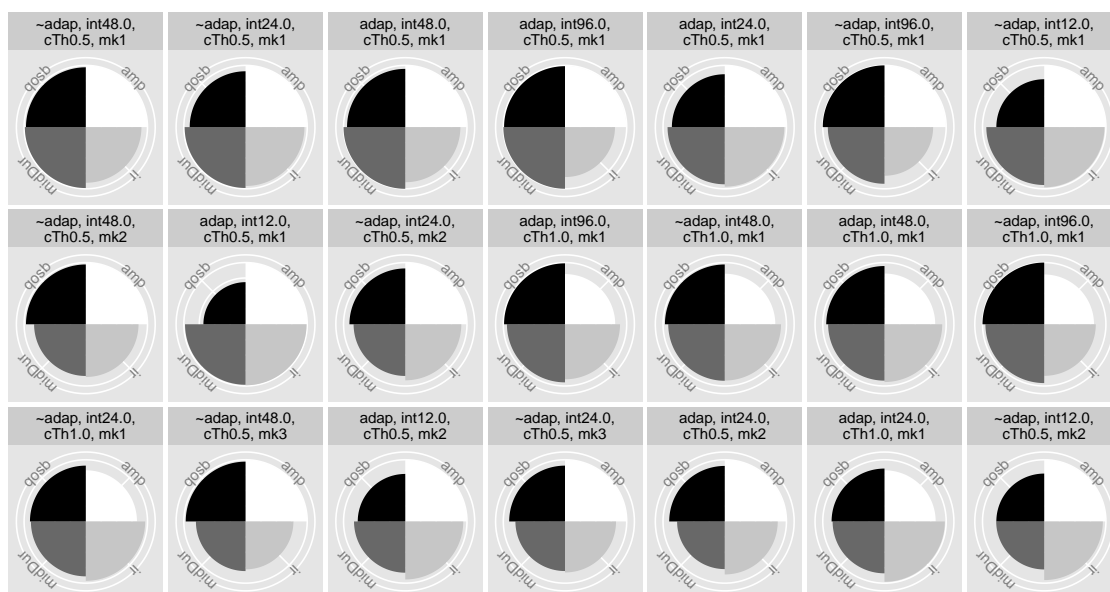


Figura 5.19: Resumo normalizado das melhores combinações de parâmetros

5.3.6 Análise crítica

Com os resultados apresentados, pôde-se estabelecer diversas relações entre os parâmetros e as métricas. No DDFC, o $CThresh$ influencia diretamente no número de agrupamentos formados, pois é ele que estabelece o critério de similaridade de leituras que deve ser satisfeito para que nós se agrupem. Da mesma forma que um valor menor para o $CThresh$ aumenta o número de agrupamentos formados, ele também aumenta o número de líderes eleitos, dado que ele tem relação com o número de agrupamentos. Assim, com o número menor de agrupamentos formados, os nós da rede normalmente ficam mais próximos dos líderes, utilizando menos saltos para atingi-los.

O Parâmetro $MaxK$, assim como o $CThresh$, influencia no número de líderes estabelecidos pelo DDFC, mas de maneira mais branda. Ademais, embora o aumento neste parâmetro possibilite a existência de nós mais distantes de líderes na rede, a maior parte dos nós ainda assim se encontra próximo de líderes, sendo que de $MaxK = 2$ para $MaxK = 3$ já não houve grande mudança. Por outro lado, notou-se um aumento considerável no número de rotas inconsistentes na medida que o $MaxK$ aumentou.

O intervalo de atuação int do Controlador de *Beacons* afeta diretamente a amplitude das leituras dos agrupamentos lógicos, sendo que intervalos menores diminuem a amplitude dos agrupamentos. Da mesma maneira, a inconsistência das rotas é inversamente proporcional ao parâmetro int . Esses comportamentos se devem ao número de *beacons* enviados, que aumenta para intervalos de atuação menores e, assim, possibilita uma maior frequência de atualização dos agrupamentos

O uso de intervalos adaptativos no DDFC foi capaz de diminuir o tempo de convergên-

cia inicial para uma configuração estável e garante maior dinamicidade na reconfiguração dos agrupamentos em caso de mudanças grandes nas leituras. Entretanto, em casos normais, o uso de um intervalo adaptativo resultou em maior inconsistência de rotas. Isto indica que a operação de atraso do período do Controlador de *Beacons* provém maior dinamicidade de resposta a eventos anômalos, isto é, que variem bastante as leituras dos nós; em contrapartida, ele aumenta levemente o número médio de rotas inválidas na rede, o que no decorrer do tempo não é crítico devido à natureza adaptativa do protocolo, que sempre converge para uma configuração estável e correta quanto às rotas.

Por fim, mostrou-se que, independente da combinação de parâmetros utilizadas, a amplitude dos agrupamentos sempre é inferior a $2 * CThresh$. Isso evidencia que a abordagem de sincronização de agregações de leituras nos nós foi bem sucedida. Através desta sincronização, os nós foram capazes de serem agrupados logicamente e dinamicamente, de acordo com a similaridade entre as suas leituras.

5.3.6.1 Análise do DDFC-K

A variante DDFC-K do protocolo tem como única diferença as regras empregadas na Unidade de Indexação, que neste correspondem às regras originais do protocolo KHOPCA. O protocolo DDFC realizou mudanças nas regras, visando gerar resultados mais estáveis e estabelecer nós mais adequados como líder. Para isso, as regras são aplicadas em uma ordem fixa e priorizam aqueles nós com mais vizinhos de leitura similar no cargo de líder.

Os resultados apresentados demonstram que, embora a variante DDFC-K se comporte de maneira similar ao DDFC quanto ao número de líderes e agrupamentos estabelecidos, o DDFC-K de fato elege líderes menos adequados. Isso é mostrado pela menor duração média dos nós eleitos nos postos de líder. Com esta pequena instabilidade, é maior a chance de que a configuração das rotas em um dado instante discreto seja inválida, como foi mostrado. Dadas estas considerações, confirma-se que o DDFC opera de maneira mais estável, elegendo líderes mais adequados.

5.3.6.2 Análise do SCCS

O protocolo SCCS se assemelha ao DDFC ao criar agrupamentos que mantêm nós com base na similaridade entre suas leituras. Contudo, a manutenção do SCCS é realizada pela quebra de agrupamentos em fragmentos menores, quando nós do agrupamento original deixam de satisfazer a relação de similaridade pre estabelecida. Sem nenhuma operação que una agrupamentos de leituras similares, o SCCS depende da completa reestruturação dos agrupamentos da rede, que é um processo disparado pela estação-base.

A manutenção baseada apenas na fragmentação de agrupamentos quando leituras internas perdem a relação de similaridade que compartilharam anteriormente não é dinâ-

mica o bastante. Mesmo o SCCS considerando também a similaridade temporal, existem flutuações no decorrer do tempo tais que em um período curto é possível que determinados nós deixem de fazer parte de um agrupamento e logo voltem a compartilhar leituras similares, sem que seja possível reingressar neste mesmo agrupamento.

Decorrente disso, a amplitude observada nos agrupamentos do SCCS é menor, indicando que os nós que permanecem agrupados possuem leituras mais próximas. Contudo, a falta de dinamicidade do protocolo implica na deterioração gradual não só dos agrupamentos, como também das rotas utilizadas, que foram muito mais frequentemente inválidas. Para corrigir o grande grau de fragmentação dos agrupamentos e as rotas que se tornam inválidas, é necessária uma completa reestruturação da rede, disparada a partir da estação-base.

Tal operação de reestruturação gera grande *overhead*, latência, e torna os agrupamentos inutilizáveis durante o breve período no qual eles são recriados. Desta forma, o protocolo SCCS não atende à questão de dinamicidade de maneira adaptativa. No caso do DDFC, as operações de manutenção são simples e dinâmicas, sendo que a informação dos agrupamentos é mantida de maneira regional em cada nó. Por isso, mesmo que em dados instantes existam rotas inválidas, a solução utilizada se adapta naturalmente, em poucos instantes, mantendo os agrupamentos dinamicamente sem que hajam rotas inválidas por grandes períodos de tempo.

5.4 Resumo

Este capítulo apresentou a avaliação do comportamento e desempenho do protocolo DDFC, comparado à uma vertente DDFK-K que utiliza as regras originais do KHOPCA, e ao SCCS. Foram utilizadas leituras reais de nós, que provaram que a operação de sincronização da agregação de leituras dos agrupamentos lógicos foi bem sucedida. Tal operação se mostrou eficiente para manter os nós agrupados dinamicamente de acordo com a sua similaridade de leituras. Também mostrou-se que, embora o uso de intervalos adaptativos aumente a dinamicidade e diminua o tempo de convergência inicial, ele aumenta o número de rotas inválidas durante a simulação. O funcionamento do DDFC foi mais estável que a variante DDFC-K, e apresentou resultados sempre superiores ao SCCS nas métricas avaliadas.

Capítulo 6

CONCLUSÃO

O agrupamento lógico de nós em RSSFs é uma técnica eficiente para a organização do tráfego, que possibilita maior escalabilidade, otimiza o uso de técnicas de compressão, diminui o número de colisões e melhora, assim, a eficiência da rede em termos de banda e latência. Contudo, o atual uso de agrupamentos lógicos em protocolos que consideram a similaridade das leituras não é adequado às suas relações dinâmicas. Quando estas relações são bem tratadas, aplicações mais robustas e precisas são viabilizadas, como detecções de ilhas de calor, de tremores sísmicos para predição de erupções ou terremotos, origens de poluição sonora ou na água, entre outros.

Ainda existe a necessidade de um protocolo capaz de realizar a manutenção dinâmica dos agrupamentos, de acordo com a natureza dinâmica das leituras de dados. Essa dinamicidade das leituras reflete diretamente na complexidade de se projetar um protocolo adaptativo à elas. Assim, uma abordagem intrinsecamente distribuída é necessária para manter as formações de nós de maneira auto-organizável e distribuída.

Diversos trabalhos tem buscado nos sistemas da natureza por soluções para problemas complexos, devido à elegância e à sua característica distribuída. Dentro deste escopo, enquanto trabalhos inspirados em vagalumes para RSSFs visam tratar da questão de sincronia de relógio entre os nós, eles demonstram ter um potencial até então inexplorado. Ao invés de sincronizar relógios, podem-se manter sincronizadas agregações de leituras de dados, possibilitando com isso a manutenção dinâmica e distribuída dos agrupamentos lógicos de nós, se adaptando à dinamicidade de suas leituras.

O protocolo DDFC (*D*ynamic *D*ata-aware *F*irefly-based *C*lustering) foi proposto para a criação e manutenção adaptativa de agrupamentos lógicos de nós, mantendo sempre agrupados aqueles nós de leituras similares. Inspirado em vagalumes, ele se utiliza de mensagens *beacon* análogas ao piscar dos vagalumes para manter sincronizada a agregação das leituras dos nós de cada agrupamento. Esta leitura permite definir quais nós vizinhos satisfazem a similaridade de leituras desejável para o agrupamento, possibilitando a fragmentação e a união de agrupamentos lógicos, em uma operação de manutenção distribuída. A partir das leituras dos vizinhos de leituras similares definidos pela Unidade de Sincronização, a Unidade de Indexação é utilizada para manter as rotas internas aos agrupamentos, estabelecendo índices em cada nó, que aumentam com a proximidade do nó em questão a um líder e permitem, assim, o fácil roteamento das mensagens. Devido à importância do intervalo entre as mensagens *beacon*, o DDFC emprega uma Unidade Adaptativa para gerenciar o intervalo entre as mensagens enviadas por cada nó, aumen-

tando sua frequência em condições de rede instáveis, ou reduzindo ela em condições de estabilidade.

O protocolo DDFC foi implementado no simulador NS3 e avaliado de acordo com as leituras reais obtidas do laboratório Intel, de Berkeley. Ele foi comparado à uma variante DDFC-K, que utiliza as regras originais do protocolo KHOPCA, e ao protocolo SCCS, também implementados no simulador NS3. As simulações demonstram que o DDFC é capaz de manter os nós agrupados de maneira dinâmica, respeitando os limiares de similaridade de dados. Os resultados obtidos avaliaram a relação entre os parâmetros de configuração do protocolo, indicando que o número de saltos máximos até um líder não exerce grande influência para números superiores a 2. Ademais, o uso de intervalos adaptativos apresenta ganhos e oferece maior dinamicidade ao protocolo, aumentando levemente, em contrapartida, a inconsistência das rotas até os líderes. O DDFC-K foi capaz de agrupar os nós através da agregação sincronizada das leituras dos agrupamentos, satisfazendo sempre o limiar de similaridade definido e, assim, consolida um novo uso para técnicas inspiradas em vagalumes.

Quando comparado ao DDFC-K, as modificações propostas no sistema de regras geraram uma configuração hierárquica mais estável, que seleciona líderes mais adequados. Tal estabilidade reflete diretamente no número de rotas inválidas, que é menor no DDFC. Ademais, o protocolo DDFC apresentou resultados sempre superiores ao SCCS, devido à maneira completamente adaptativa e dinâmica com a qual o DDFC opera - os agrupamentos e as rotas são mantidos e reparados continuamente e dinamicamente através de mensagens *beacon*. Em contrapartida, SCCS depende de operações de reestruturação completas, sem a possibilidade de que agrupamentos sejam unidos, mas apenas quebrados. Assim, o estado dos agrupamentos formados é deteriorado até um nível mínimo de qualidade, quando eles são completamente reestruturados em uma dispendiosa operação.

6.1 Trabalhos futuros

Como decorrência, a evolução desta dissertação possibilitaria investigar tanto aspectos do funcionamento do DDFC quanto questões de cenário e de implementação. Quanto ao seu funcionamento, os agentes utilizados podem ser alterados ou estendidos a fim de melhor operação, dada uma aplicação alvo mais específica. Quanto ao cenário, diferentes tipos de leituras de dados reais podem ser consideradas nas simulações realizadas. Quanto à implementação, o protocolo DDFC pode ser prototipado em sensores reais.

Tomando em conta os Agentes, a Unidade Adaptativa opera adiantando ou atrasando os intervalos de envio de mensagens *beacon*, dada as condições de estabilidade da rede, tal que condições mais instáveis se beneficiam de intervalos menores. Este agente pode ser melhorado para que ele possa utilizar operações de atraso e adiantamento de maneira mais expressiva, através da análise de histórico de mensagens e leituras passadas, por

exemplo. A Unidade de Indexação opera estabelecendo índices para o roteamento interno aos agrupamentos, ele pode ser estendido ao considerar também o roteamento externo aos agrupamentos, que transmitiria as mensagens entre os líderes e estação-base. A Unidade de Sincronização, por sua vez, é a entidade bio-inspirada que mantém sincronizadas as agregações das leituras dos nós em cada agrupamento, criando e mantendo estes. Este agente ainda pode ter seu funcionamento analisado com relação à diversas funções de similaridade distintas, dado um determinado cenário.

Ademais, a simulação atual empregou leituras internas à um ambiente pequeno, que foi devidamente escalado para que os resultados fossem coerentes, dado o raio de transmissão dos nós. Assim, podem ser procurados mais dados de leituras reais para que o protocolo seja avaliado em cenários diferenciados. Além disso, é possível realizar comparações com outros tipos de leituras, analisando o comportamento do protocolo para elas. Por fim, por ser baseado em princípios biológicos, o DDFC estabelece comportamentos complexos e distribuídos através de agentes simples, viabilizando uma possível prototipação rápida em sensores sem fio reais, a fim de validar o protocolo através de experimentação.

6.2 Aplicações diretas do protocolo

As RSSFs dão suporte a aplicações em diversas áreas [59], dentre as quais podem-se destacar exemplos no escopo do protocolo desenvolvido nesta dissertação. O DDFC vem a servir como um bloco de construção primário para sistemas distribuídos dependentes de informações relacionadas às extensões espaciais de dados similares em ambientes com leituras dinâmicas. Dentre estas, citam-se o monitoramento de umidade em aplicações agrícolas; monitoramento sísmico e vulcânico; monitoramento de poluição no meio urbano; monitoramento da qualidade de água; detecção prévia de indício de incêndios em florestas. Abaixo seguem alguns exemplos de aplicação do DDFC quanto a estes casos previstos.

O uso de água por aplicações agrícolas é responsável por estimados 70% do uso de água no mundo inteiro [14]. Desta forma, o monitoramento do gasto adequado de água é essencial para reduzir custos e desperdício. Assim, através de medições de umidade e determinação de extensões e distribuições espaciais do consumo, é possível estipular políticas de economia e controle nos sistemas de irrigação.

No caso de atividades sísmicas e vulcânicas, é importante determinar a maneira com a qual os tremores se comportam e quais áreas compreendem. Com medições precisas, é possível estabelecer padrões entre coordenadas geográficas, simples de serem obtidas através de GPS, e o posicionamento de placas tectônicas. Como visto em [15], através das medições de tremores e estabelecimento de padrões permitem a inferir a possibilidade de movimentação precedentes à uma erupção em até algumas semanas.

O monitoramento da poluição no meio urbano é a aplicação quase precisa do projeto CitySense [6]. Assim como nele, é possível a fácil detecção de ilhas de calor, decorrentes da

poluição, através de inferências quanto à temperatura de diferentes regiões. Além de ilhas de calor, equipamentos específicos de sensores são capazes de determinar diretamente a presença de gases nocivos no ambiente, possibilitando assim determinar também regiões com maiores concentração de poluição em ambientes urbanos, a fim de tomar devidas providências governamentais.

O monitoramento da poluição tem uma instanciação específica no caso da qualidade da água. Neste, é possível empregar sensores que medem níveis de determinados produtos químicos, determinando da extensão destes níveis e de como eles variam. Com isso, pode-se inferir fontes de poluição, identificando áreas de contaminação a fim de tomar medidas que limitem a dispersão, no caso de substâncias nocivas, por exemplo.

Para a detecção prévia de incêndios em florestas, podem-se medir dados como temperatura, umidade, e direção do vento. Com estas medições e a determinação da extensão geográficas de medidas consideradas de risco, possibilita-se estimar áreas de risco. Assim, medidas preventivas podem ser tomadas, ou o monitoramento nestas áreas consideradas de risco aumentado.

Referências Bibliográficas

- [1] Chunxiao Zhao and Lijun Wang. Computational Modeling of Complex Systems: Case Study in Real World. In *Hybrid Intelligent Systems, HIS '09*, pages 33–38, 2009.
- [2] V M Krasnopolsky, M S Fox-Rabinovitz, and A A Belochitski. Development of neural network convection parameterizations for numerical climate and weather prediction models using cloud resolving model simulations. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2010.
- [3] Arantza Mendioroz, Raquel Fuente-Dacal, Estibaliz Apinaniz, and Agustin Salazar. Thermal diffusivity measurements of thin plates and filaments using lock-in thermography. *Review of Scientific Instruments*, 80(7):074904–074904, 2009.
- [4] Sunhee Yoon and Cyrus Shahabi. The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. *ACM Transactions on Sensor Networks*, 3(1), March 2007.
- [5] Eduardo F Nakamura, Antonio A F Loureiro, and Alejandro C Frery. Information fusion for wireless sensor networks: Methods, models, and classifications. *Computing Surveys (CSUR)*, 39(3), September 2007.
- [6] Rohan Narayana Murty, Geoffrey Mainland, Ian Rose, Atanu R Chowdhury, Abhimanyu Gosain, Josh Bers, and Matt Welsh. CitySense: An Urban-Scale Wireless Sensor Network and Testbed. In *IEEE Conference on Technologies for Homeland Security*, pages 583–588, 2008.
- [7] Craig Partridge. Realizing the future of wireless data communications. *Communications of the ACM*, 54(9), September 2011.
- [8] Zhi-yong Bai and Xin-yuan Huang. Design and Implementation of a Cyber Physical System for Building Smart Living Spaces. *International Journal of Distributed Sensor Networks*, 2012:1–9, 2012.
- [9] Ragunathan Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: The next computing revolution. In *47th ACM/IEEE Design Automation Conference (DAC)*, pages 731–736. ACM Request Permissions, 2010.
- [10] Jianli Pan, S Paul, and R Jain. A survey of the research on future internet architectures. *IEEE Communications Magazine*, 49(7):26–36, 2011.

- [11] Hua-Dong Ma. Internet of Things: Objectives and Scientific Challenges. *Journal of Computer Science and Technology*, 26(6):919–924, 2011.
- [12] Tomás Sánchez López, Damith C Ranasinghe, Mark Harrison, and Duncan Mcfarlane. Adding sense to the Internet of Things. *Personal and Ubiquitous Computing*, 16(3), March 2012.
- [13] John Augustine, Gopal Pandurangan, and Peter Robinson. Fast byzantine agreement in dynamic networks. In *PODC '13: Proceedings of the 2013 ACM symposium on Principles of distributed computing*. ACM Request Permissions, July 2013.
- [14] World Business Council for Sustainable Development. *Water: Facts and Trends*. World Business Council for Sustainable Development, 2005.
- [15] Florent Brenguier, Nikolai M Shapiro, Michel Campillo, Valerie Ferrazzini, Zacharie Duputel, Olivier Coutant, and Alexandre Nercessian. Toward Forecasting Volcanic Eruptions using Seismic Noise. *Nature Geoscience*, pages 126–130, June 2007.
- [16] Joydeep Banerjee, Souvik Kumar Mitra, Pradipta Ghosh, and Mrinal K Naskar. Memory based message efficient clustering (MMEC) for enhancement of lifetime in wireless sensor networks using a node deployment protocol. In *Proceedings of the 2011 International Conference on Communication, Computing & Security*, pages 71–76. ACM Request Permissions, February 2011.
- [17] Dan J Dechene, A El Jardali, M Luccini, and A Sauer. A survey of clustering algorithms for wireless sensor networks. *Computer Communications*, pages 2826–2841, 2007.
- [18] Mehmet C Vuran, Ozgur Akan, and Ian F Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 45(3):245–259, 2004.
- [19] Ilka Afonso Reis, Gilberto Câmara, Renato Assunção, and Antônio Miguel Vieira Monteiro. Data-aware clustering for geosensor networks data collection. *Anais XIII Simpósio Brasileiro de Sensoriamento Remoto*, pages 6059–6066, 2007.
- [20] Muzahidul Islam, Shahrum Abdullah, Koichi Wada, Jiro Uchida, and Wei Chen. An efficient routing protocol on a Dynamic Cluster-based Sensor Network. In *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 161–165, 2011.

- [21] Matthias R Brust, Hannes Frey, and Steffen Rothkugel. Dynamic multi-hop clustering for mobile hybrid wireless networks. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, page 130, New York, USA, 2008. ACM Press.
- [22] Wendi B Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *System Sciences*, pages 3005–3014, 2000.
- [23] Ossama Younis and Sonia Fahmy. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 2004.
- [24] Bin Guo and Zhe Li. A dynamic-clustering reactive routing algorithm for wireless sensor networks. *Wireless Networks*, 15(4):423–430, August 2007.
- [25] Xiaochen Wu, Peng Wang, Wei Wang, and Baile Shi. *Data-aware clustering hierarchy for wireless sensor networks*. Springer-Verlag, May 2008.
- [26] Ngoc Duy Pham, Trong Duc Le, Kwangjin Park, and Hyunseung Choo. SCCS: Spatiotemporal clustering and compressing schemes for efficient data collection applications in WSNs. *International Journal of Communication Systems*, 23(11):1311–1333, November 2010.
- [27] Ozgur B Akan and Falko Dressler. A survey on bio-inspired networking. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 54(6):881–900, 2010.
- [28] Alexander Tyrrell, Gunther Auer, and Christian Bettstetter. Fireflies as role models for synchronization in ad hoc networks. In *Proceedings of the 1st international conference on Bio inspired models of network, information and computing systems*, pages 1–7. ACM, December 2006.
- [29] Alexander Tyrrell and Gunther Auer. Imposing a Reference Timing onto Firefly Synchronization in Wireless Networks. In *Proceedings of the 65th Vehicular Technology Conference*, pages 222–226, 2007.
- [30] Sergio S Furlaneto, Aldri L dos Santos, and Carmem S Hara. An Efficient Data Acquisition Model for Urban Sensor Networks. *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, pages 113–120, April 2012.
- [31] Jurandy Almeida, Eduardo Valle, Ricardo da S. Torres, and Neucimar J. Leite. DAHC-tree: An Effective Index for Approximate Search in High-Dimensional Metric Spaces. *Journal of Information and Data Management*, 1(3):375, September 2010.

- [32] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 52(12):2292–2330, August 2008.
- [33] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.
- [34] Shuo Liu, Guomo Zhou, and Lufeng Mo. The green space monitoring system based on Wireless Sensor Network. In *Education Technology and Computer (ICETC)*, pages 180–183, 2010.
- [35] C J G Morris, I Simmonds, and N Plummer. Quantification of the Influences of Wind and Cloud on the Nocturnal Urban Heat Island of a Large City. *Journal of Applied Meteorology*, 40(2), 2001.
- [36] Hussein T Mouftah. Wireless Sensor Network Architectures for Intelligent Vehicular Systems. In *Proceedings of the 3rd International Conference on New Technologies, Mobility and Security*, 2011.
- [37] Matt Welsh. Sensor networks for the sciences. *Communications of the ACM*, 53(11), November 2010.
- [38] Winston K G Seah, Zhi Ang Eu, and Hwee-Pink Tan. Wireless sensor networks powered by ambient energy harvesting (WSN-HEAP) - Survey and challenges. In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009 Wireless VITAE*, pages 1–5, 2009.
- [39] Dimitrios Amaxilatis, Ioannis Chatzigiannakis, Christos Koninis, and Apostolos Pyrgelis. Component Based Clustering in Wireless Sensor Networks. *The Computing Research Repository (CoRR)*, May 2011.
- [40] G Li and T Znati. RECA: a ring-structured energy-efficient clustering architecture for robust communication in wireless sensor networks. *International Journal of Sensor Networks*, 2007.
- [41] Jun Wang, Yong-Tao Cao, Jun-Yuan Xie, and Shi-Fu Chen. Energy Efficient Backoff Hierarchical Clustering Algorithms for Multi-Hop Wireless Sensor Networks. *Journal of Computer Science and Technology*, 26(2):283–291, March 2011.
- [42] Shuchun Yang, Zhipeng Gao, Rima Huang, and Xuesong Qiu. A Data Correlation-Based Virtual Clustering Algorithm for Wireless Sensor Network. In *Mobile Ad-hoc and Sensor Networks (MSN)*, pages 335–336, 2011.

- [43] Dujdow Buranapanichkit, Antonio Vittorioso, Giancarlo Fortino, and Yiannis Andreopoulos. Comparison Between Centralized And Distributed Coordination For TDMA Operation In Wireless Sensor Networks. *London Communications Symposium*, September 2011.
- [44] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*, 14(6):32–39, 2007.
- [45] Joa-Hyoung Lee and In-Bum Jung. Adaptive-Compression Based Congestion Control Technique for Wireless Sensor Networks. *Sensors*, January 2010.
- [46] Myung Ho Yeo, Mi Sook Lee, Seok Jae Lee, and Jae Soo Yoo. Data Correlation-Based Clustering in Sensor Networks. In *International Symposium on Computer Science and its Applications (CSA)*, pages 332–337. IEEE, 2008.
- [47] Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan, and Kiyomichi Araki. Practical data compression in wireless sensor networks: A survey. *Journal of Network and Computer Applications*, 2011.
- [48] Nicholas Chang and Mingyan Liu. Revisiting the TTL-based controlled flooding search: optimality and randomization. In *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM Request Permissions, September 2004.
- [49] Rui Dai and Ian F Akyildiz. A spatial correlation model for visual information in wireless multimedia sensor networks. *IEEE Transactions on Multimedia*, 11(6):1148–1159, 2009.
- [50] Renato E Mirolo and Steven H Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, pages 1645–1662, 1990.
- [51] Geoffrey Werner-Allen, Geetika Tewari, Ankit Patel, Matt Welsh, and Radhika Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM Request Permissions, November 2005.
- [52] Ozalp Babaoglu, Toni Binci, Mark Jelasity, and Alberto Montresor. Firefly-inspired Heartbeat Synchronization in Overlay Networks. In *Self-Adaptive and Self-Organizing Systems, 2007 SASO*, pages 77–86, 2007.
- [53] Naoki Wakamiya and Masayuki Murata. Synchronization-Based Data Gathering Scheme for Sensor Networks. *IEICE Transactions on Communications*, E88-B(3):873–881, March 2005.

- [54] Dennis Lucarelli and I-Jeng Wang. Decentralized synchronization protocols with nearest neighbor communication. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM Request Permissions, November 2004.
- [55] Wendi B Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002.
- [56] Martin Gardner. Mathematical games: The fantastic combinations of John Conway’s new solitaire game “life”. *Scientific American*, 1970.
- [57] Natalija Vljajic and D Stevanovic. Sink mobility in wireless sensor networks: a (mis) match between theory and practice. *International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 386–393, 2009.
- [58] Berkeley. <http://db.csail.mit.edu/labdata/labdata.html>, October 2012.
- [59] Shafiullah Khan, Al-Sakib Khan Pathan, and Nabil Ali Alrajeh. Wireless Sensor Networks: Current Status and Future Trends. *Wireless Sensor Networks: Current Status and Future Trends*, November 2012.

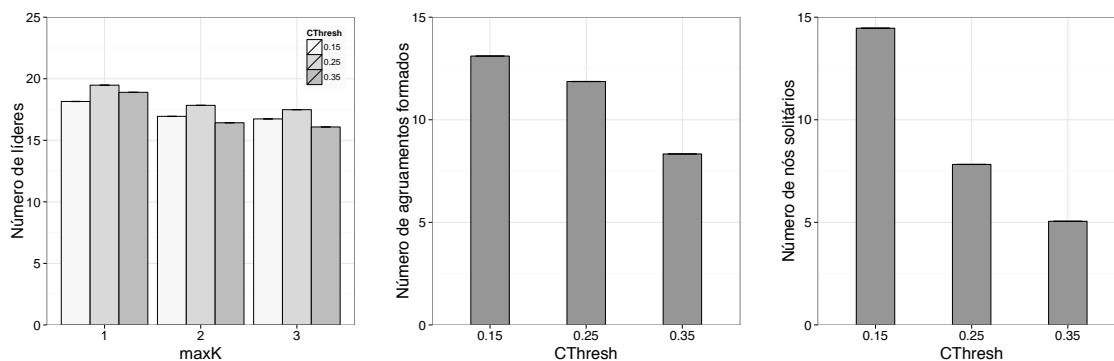
ANEXO

RESULTADOS PARA LEITURAS DE TEMPERATURA

Este anexo apresenta os resultados da avaliação de desempenho dos protocolos e variantes DDFC, DDFC-K E SCCS quanto às leituras de temperatura, realizadas pelo laboratório Intel, de Berkeley. Os resultados obtidos para este cenário são similares aos apresentados para as leituras de umidade, no Capítulo 5, correspondendo ao mesmo cenário, com as mesmas operações de interpolação de leituras e mesmos parâmetros e métricas, com exceção do parâmetro $CThresh$. Nota-se que o parâmetro $CThresh$ variou diferentemente, sendo valorado como (0.15, 0.25, 0.35). Tal variação foi necessária devido à menor variação da temperatura dentro do ambiente, sendo ela muito mais homogênea do que a umidade e necessitando de valores inferiores para este parâmetro.

Agrupamentos formados

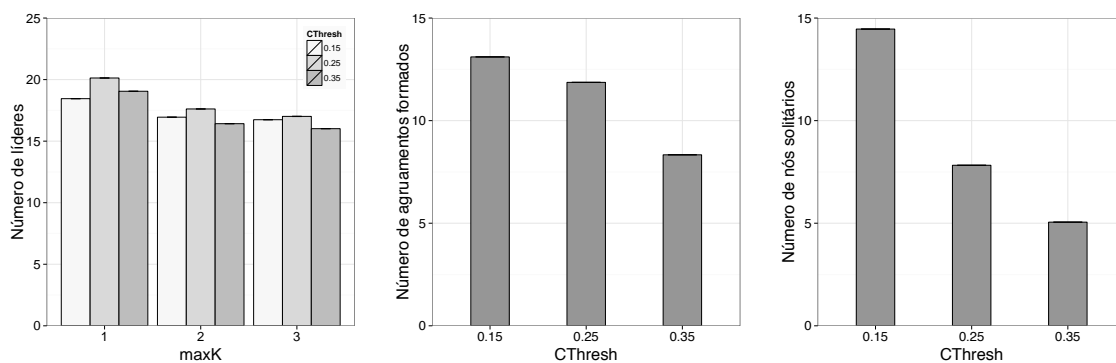
Quanto às métricas referentes à categoria de agrupamentos formados, percebe-se que o número de agrupamentos e de nós solitários seguiu a mesma tendência de diminuir de acordo com o $CThresh$. Nota-se, contudo, que para leituras de temperatura, o parâmetro $CThresh$ influencia o número de líderes de maneira um pouco diferente. A tendência observada para a umidade é que quanto maior este parâmetro, menor o número de líderes. Nos gráficos observados abaixo, o menor número de líderes corresponde ao caso de $CThresh = 0.15$, parâmetro mínimo considerado. Isso ocorre pois um valor de $CThresh = 0.15$ é muito baixo para as leituras de temperatura realizadas, demonstrando-se inadequado para este cenário e acarretando em formações menos estáveis, devido à relação de similaridade de dados muito estrita.



Relação do número de líderes, de agrupamentos formados e de nós solitários

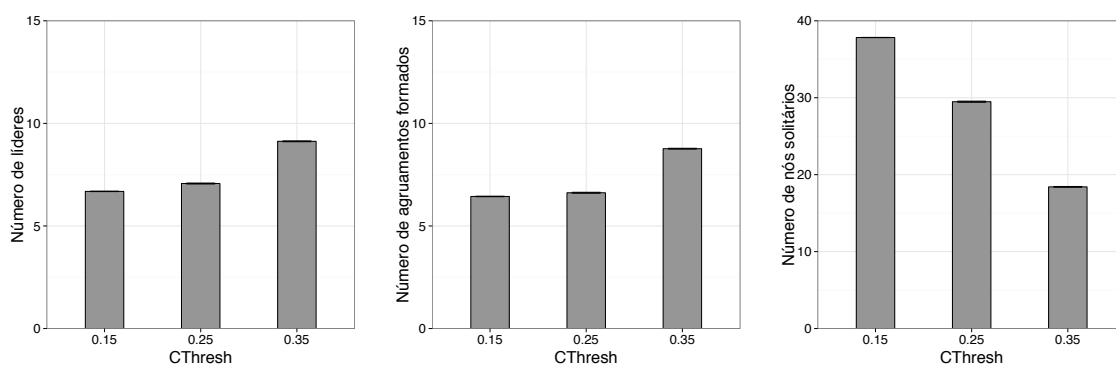
Comparação com DDFC-K e SCCS

Assim como para as leituras de umidade, as leituras de temperatura apresentaram comportamento igual para o protocolo DDFC e sua variante DDFC-K quanto às métricas de agrupamentos formados.



Comportamento dos agrupamentos no DDFC-K

Para leituras de temperatura, o SCCS apresentou um número de nós solitários muito superior às leituras de umidade. Isso ocorreu devido às leituras muito mais homogêneas e ao fato de o SCCS ser influenciado de maneira diferente pelo *CThresh*, como visto no Capítulo 5. A quantidade de agrupamentos formados em relação ao *CThresh*, se comporta também de maneira um pouco diferente, aumentando o número de agrupamentos formados para valores superiores de *CThresh*. Este aumento é decorrente do número muito grande de nós solitários. Na medida que o número de nós solitários decresce, com o aumento de *CThresh*, o número de agrupamentos aumenta.

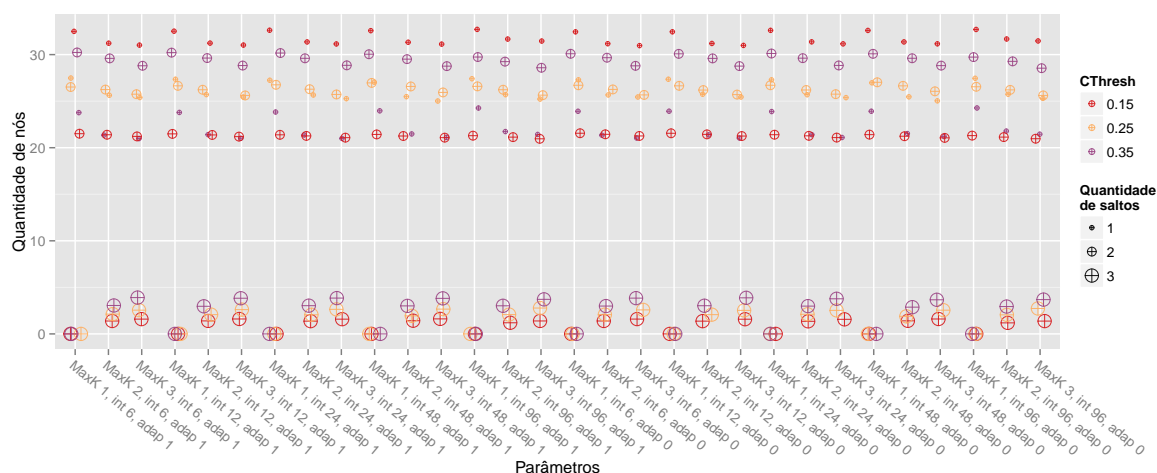


Comportamento dos agrupamentos no SCCS

Distância média até os líderes e duração dos líderes

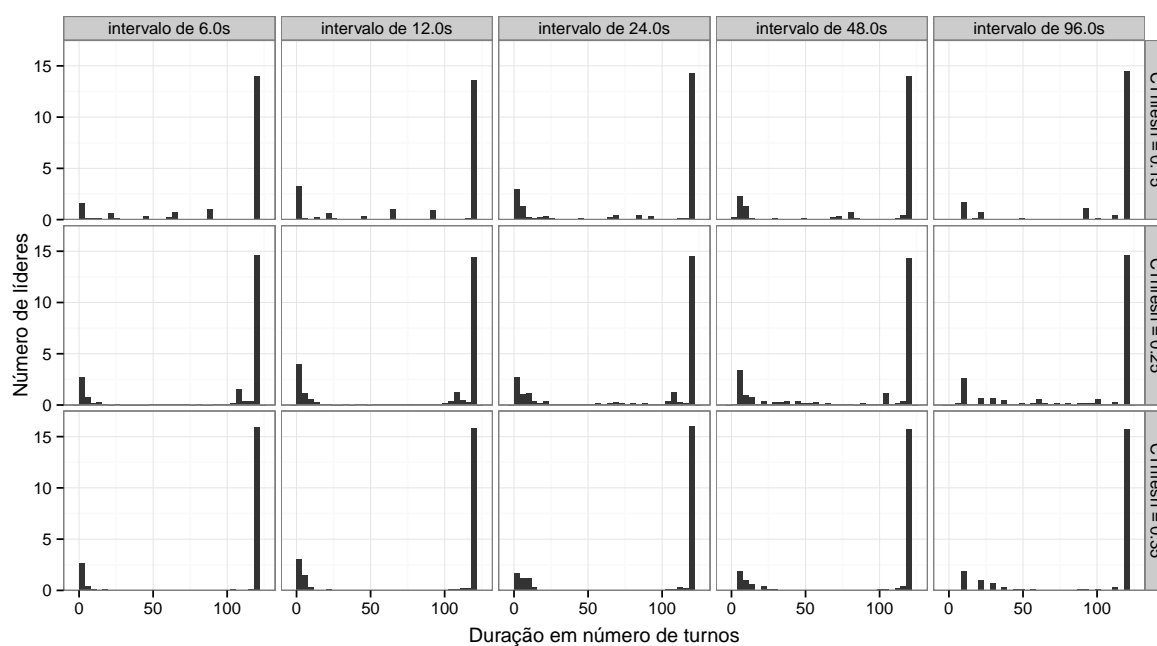
O comportamento da quantidade de nós vs. distância até o líder mais próximo é similar no DDFC, com a exceção de que com leituras de temperatura, o patamar de nós que levam apenas um salto para atingir um líder se demonstra mais alto no gráfico. Os patamares de um e dois saltos até o líder mais próximo se mostram mais próximos devido ao número

levemente superior de líderes eleitos para leituras de temperatura.



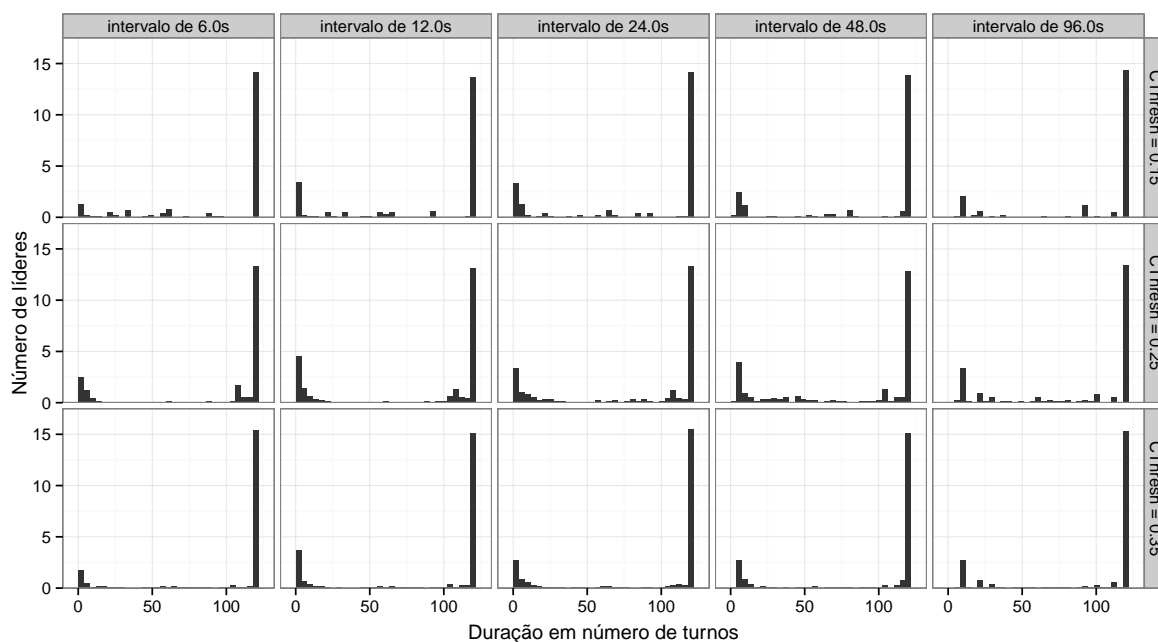
Quantidade de nós vs. distância até o líder mais próximo

O protocolo DDFC e sua variante DDFC-K apresentaram comportamentos similares quanto à duração dos líderes eleitos e suas estabilidades, confirmando que o DDFC também elegeu líderes mais estáveis para as leituras de temperatura, quando comparado à variante DDFC-K.



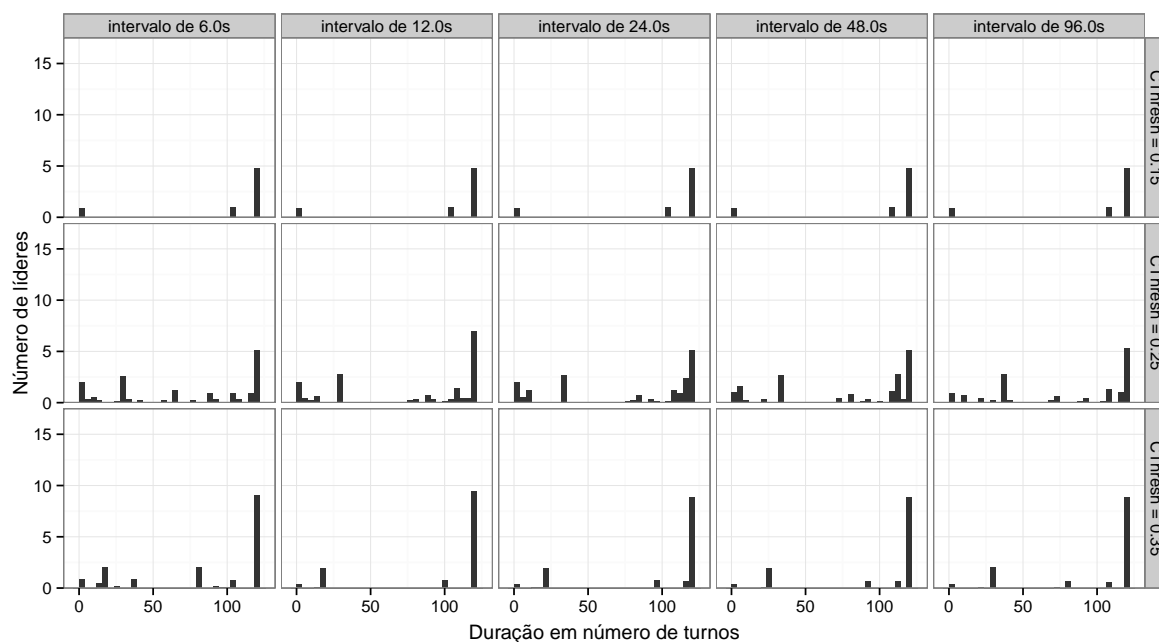
Duração dos líderes em número de turnos de 10s

Comparação com DDFC-K e SCCS



Duração dos líderes para a variação DDFC-K

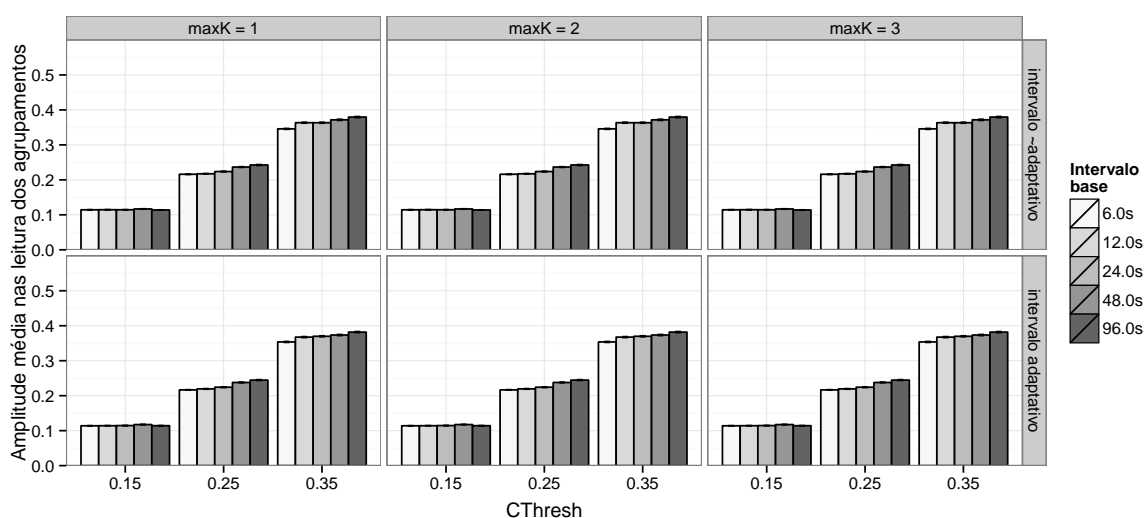
O protocolo SCCS apresentou resultados distintos quanto à duração dos líderes eleitos. Para as leituras de temperatura, os líderes existem em maior quantidade na duração máxima, indicando maior estabilidade em relação à leituras de umidade. Contudo, este comportamento é decorrente do número muito superior de nós solitários no cenário de leituras de temperatura, existindo menos líderes e agrupamentos na rede.



Duração dos líderes para o SCCS

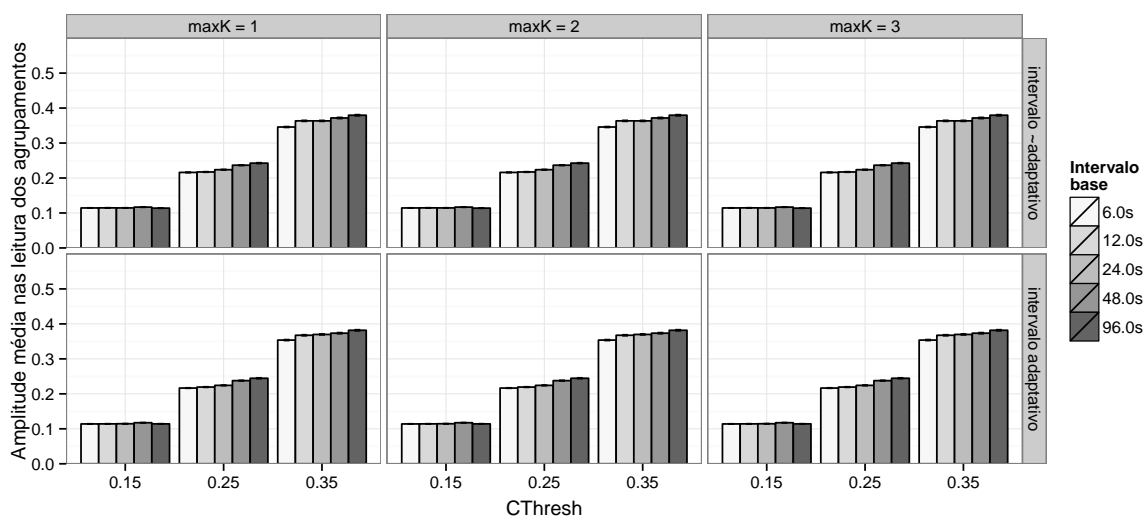
Similaridade de leituras dos nós agrupados

O protocolo DDFC e sua variante DDFC-K apresentaram comportamentos similares quanto à amplitude das leituras dos agrupamentos formados, também sempre respeitando o limite máximo de $2 * CThresh$, como estabelecido no Capítulo 5. Neste caso não se observam casos anômalos quanto à variação do intervalo entre *beacons* para $MaxK = 3$, pois para o cenário de temperaturas, o $CThresh$ máximo não foi grande o bastante para que a rede fosse composta de apenas 2 agrupamentos, como no cenário de umidades.



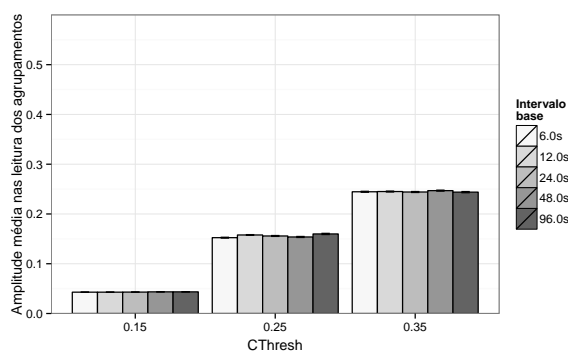
Similaridade de leituras dentro do mesmo agrupamento

Comparação com DDFC-K e SCCS



Similaridade de leituras na vertente DDFC-K

Para o protocolo SCCS, as amplitudes alcançadas foram sempre inferiores às do protocolo DDFC e sua variante DDFC-K, correspondendo aos resultados do Capítulo 5.

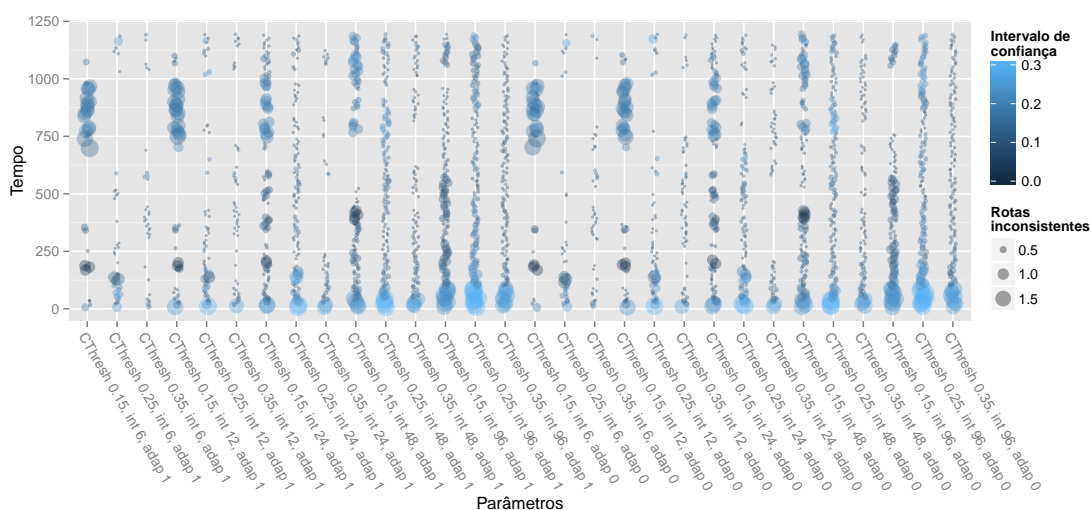


Similaridade de leituras no SCCS

Para as leituras de temperatura, nota-se que em todos os casos o intervalo entre os *beacons* não acarretou em uma mudança grande na amplitude das leituras. Isso demonstra que a variação da temperatura é muito mais gradual do que a variação da umidade. Desta forma, a rede pode operar com intervalos *int* maiores, sem comprometer os resultados.

Inconsistência das rotas

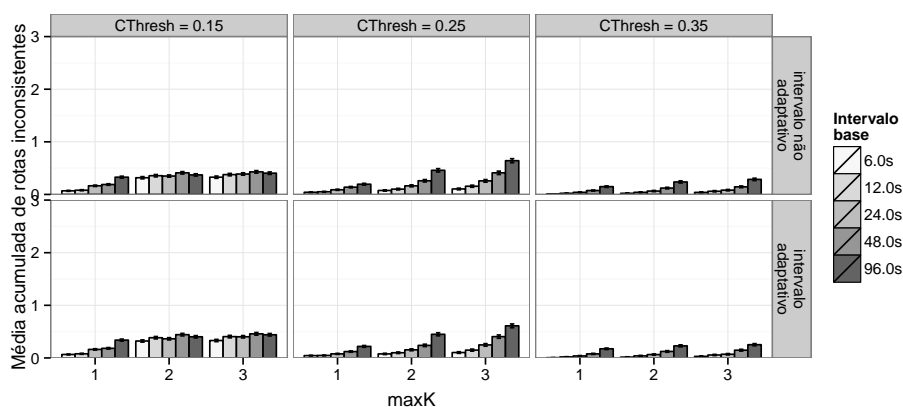
No cenário de temperaturas, a inconsistência das rotas é também maior no começo do tempo de vida da rede. Percebe-se também que neste cenário a valoração $CThresh = 0.15$ apresentou picos de rotas inválidas em alguns instantes, devido ao intervalo de similaridade muito estrito para tal valoração.



Inconsistência das rotas através do tempo

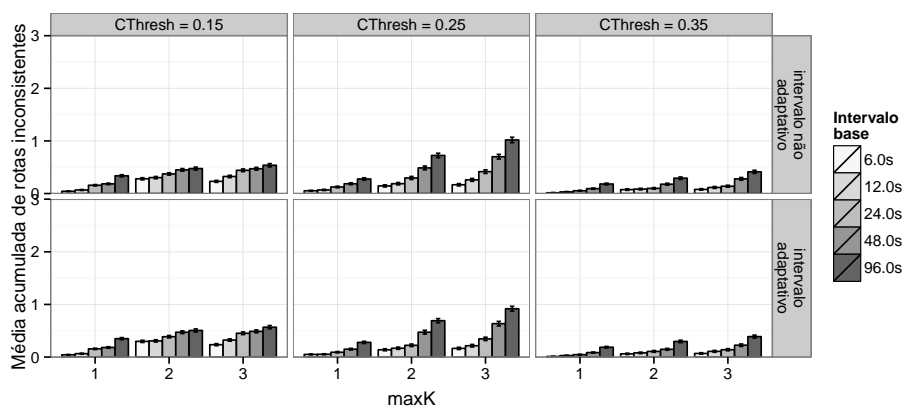
A inconsistência acumulada das rotas é similar ao cenário de umidade para o protocolo DDFC e sua variante DDFC-K, atingindo um patamar máximo próximo de 0.5 rotas inválidas para o DDFC durante o tempo total da simulação. Para as leituras de temperatura,

o uso de intervalos adaptativos não aumentou o número de rotas inválidas, como no caso de leituras de umidade, devido à variação mais gradual das leituras de temperatura.



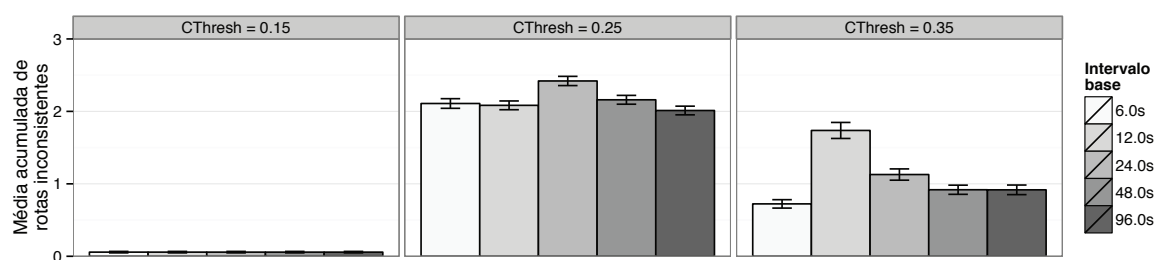
Inconsistência acumulada das rotas

Comparação com DDFC-K e SCCS



Inconsistência das rotas para a variante DDFC-K

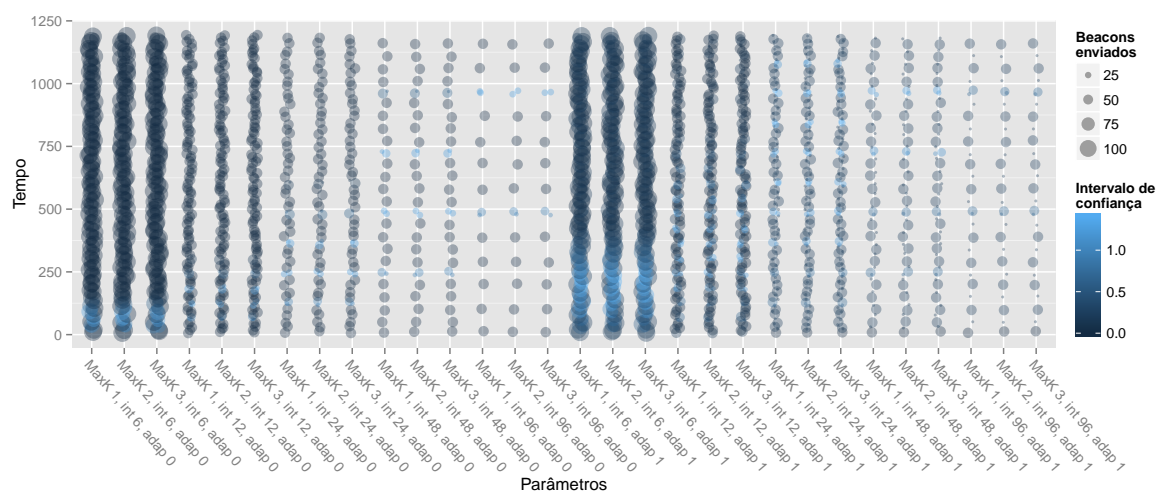
No protocolo SCCS, a valoração $CThresh = 0.15$ resultou em um número quase nulo de rotas inválidas absolutas durante o tempo de simulação. Isso ocorre pois com tal valoração, existem 40 nós solitários na rede e apenas cerca de sete líderes e agrupamentos. Desta forma, o número de rotas é muito inferior e, como as leituras de temperatura mudam mais gradualmente, o número de vezes que as rotas devem ser reparadas é também muito inferior. Para as demais valorações, o comportamento das rotas inválidas do SCCS no cenário de temperatura é similar ao cenário de umidade, apresentando sempre um número superior de rotas quando comparado ao protocolo DDFC e sua variante DDFC-K.



Inconsistência das rotas para o SCCS

Mensagens *beacon*

A troca de mensagens *beacon* para leituras de temperatura é similar ao cenário de umidade, para todos os protocolos e variantes considerados.

Mensagens *beacon* enviadas através do tempo