



UNIVERSIDADE ESTADUAL DO CEARÁ
CENTRO DE CIÊNCIAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

JOSÉ HENRIQUE BRANDÃO NETO

**UMA NOVA ABORDAGEM UTILIZANDO FLUXO EM REDES PARA O
MELHORAMENTO DE PROTOCOLOS DE ROTEAMENTO BASEADOS EM
CLUSTERIZAÇÃO EM REDES DE SENSORES SEM FIO**

FORTALEZA – CEARÁ

2016

JOSÉ HENRIQUE BRANDÃO NETO

UMA NOVA ABORDAGEM UTILIZANDO FLUXO EM REDES PARA O
MELHORAMENTO DE PROTOCOLOS DE ROTEAMENTO BASEADOS EM
CLUSTERIZAÇÃO EM REDES DE SENSORES SEM FIO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Joaquim Celestino Júnior

Co-Orientador: Prof. Dr. Leonardo Sampaio Rocha

FORTALEZA – CEARÁ

2016

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Brandão Neto, José Henrique.

Uma nova abordagem utilizando fluxo em redes para o melhoramento de protocolos de roteamento baseados em clusterização em redes de sensores sem fio [recurso eletrônico] / José Henrique Brandão Neto. - 2016.

1 CD-ROM: il.; 4 ¼ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 91 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2016.

Área de concentração: Ciência da Computação.

Orientação: Prof. Ph.D. Joaquim Celestino Júnior.

Coorientação: Prof. Dr. Leonardo Sampaio Rocha.

1. Redes de Sensores. 2. Clusterização. 3. Teoria dos Grafos. 4. Fluxo em Redes. I. Título.

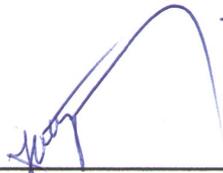
JOSÉ HENRIQUE BRANDÃO NETO

UMA NOVA ABORDAGEM UTILIZANDO FLUXO EM REDES PARA O
MELHORAMENTO DE PROTOCOLOS DE ROTEAMENTO BASEADOS EM
CLUSTERIZAÇÃO EM REDES DE SENSORES SEM FIO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 08 de Março de 2016

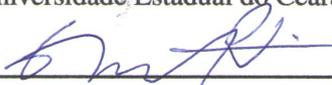
BANCA EXAMINADORA



Prof. Dr. Joaquim Celestino Júnior (Orientador)
Universidade Estadual do Ceará - UECE



Prof. Dr. Leonardo Sampaio Rocha (Co-Orientador)
Universidade Estadual do Ceará - UECE



Prof. Dr. Gerardo Valdisio Rodrigues Viana
Universidade Estadual do Ceará - UECE



Profa. Dra. Rossana Maria de Castro Andrade
Universidade Federal do Ceará - UFC

AGRADECIMENTOS

A Deus que sempre me conduziu pelos caminhos da sabedoria, da conquista e da paz.

A Nossa Senhora de Fátima, por me mostrar que no final tudo dá certo.

Aos meus pais, Fátima Lira e Henrique Brandão, que com cuidado, amor e carinho foram a base de tudo, apoiando-me em todas as decisões e ajudando-me a persistir e alcançar os meus objetivos.

A minha irmã, Renata Mendes, que sempre esteve à disposição, ajudando-me no que precisei.

Aos meus familiares que sempre acreditaram nessa vitória, dando atenção e apoio durante essa trajetória, especialmente à minha avó Terezinha Mendes, às minhas tias Eliene Brandão e Socorro Brandão, e aos meus tios Francisco Lira e Armando Brandão.

Ao professor Dr. Joaquim Celestino Jr., pelo aprendizado, incentivo, paciência, confiança, cobranças e sugestões nas orientações desta dissertação.

Aos professores do curso, que ao longo desses dois anos, compartilharam seus conhecimentos e contribuíram para o nosso desenvolvimento profissional.

Aos componentes da banca, que se propuseram a participar e acreditaram no meu trabalho.

Aos colegas da graduação, mestrado e do laboratório LARCES, pelo convívio, amizade e conselhos, estando comigo durante todo o percurso.

*“Entrega o teu caminho ao Senhor, confia nele e
o mais ele fará”*

(Salmos 37:5)

RESUMO

Redes de Sensores sem fio (RSSF) consistem de um conjunto de nós sensores que coletam informações do ambiente e enviam para uma estação base que processa os dados finais. Alguns desafios podem ser encontrados, como minimizar o consumo de energia e maximizar o tempo de vida da rede. Muitos protocolos alcançam a economia de energia através da clusterização da rede. Esse processo divide a rede em grupos e define para cada grupo um líder, chamado de cluster-head. Este trabalho apresenta uma nova modelagem por grafos, utilizando-se fluxo em redes, para melhorar os protocolos de roteamento que utilizam a técnica de clusterização, aprimorando o roteamento de dados entre os clusters-head, como também, definindo um tempo de uma rodada no processo de clusterização. Com esta técnica, nossa proposta consegue aumentar o tempo de vida da rede, balancear o consumo de energia entre os nós e distribuir melhor as transmissões de dados. Através de simulações no simulador de rede OMNET++, Castalia, a proposta é validada e sua eficiência é verificada comparando-a com outros protocolos que trabalham com clusterização.

Palavras-chave: Redes de Sensores. Clusterização. Teoria dos Grafos. Fluxo em Redes

ABSTRACT

Wireless Sensor Networks (WSN) consist of a set of sensor nodes that collect environmental information and send it to a base station that processes the final data. Some challenges can be found to minimize energy consumption and maximize the network lifetime. Many protocols achieve energy savings through clustering network. This process divides the network into groups and sets for each group a leader called cluster-head. This paper presents a new modeling by graphs that uses network flow to improve routing protocols that use clustering technique, improving the routing of data between the cluster-head, as well as, a new way to set the time of a round in the clustering process. With this technique, we propose can increase the network life time, balancing the energy consumption between the nodes and better distribute the data transmissions. Through simulations on Omnet++/Castalia network simulator, the proposal is validated and its efficiency is verified comparing it with other protocols that work with clustering.

Keywords: Wireless Sensor Networks. Clustering. Graph theory. Network Flow

LISTA DE ILUSTRAÇÕES

Figura 1 – Redes de Sensores sem Fio.	19
Figura 2 – Arquitetura Nó Sensor.	20
Figura 3 – Exemplos de Nós Sensores.	20
Figura 4 – Exemplo de um grafo.	25
Figura 5 – Grau de um vértice.	26
Figura 6 – Fluxo em rede.	27
Figura 7 – Fluxo máximo em um Fluxo em rede.	29
Figura 8 – Exemplo de uma Rede de Sensores no protocolo MCCR.	42
Figura 9 – Modelagem Final da Rede em um problema de circulação de custo mínimo.	42
Figura 10 – Diagrama de Estados de um Protocolo de Roteamento Centralizado baseado em Clusterização.	48
Figura 11 – Operação na Rede de um Protocolo de Roteamento Centralizado baseado em Clusterização.	48
Figura 12 – Diagrama de Estados de um Protocolo de Roteamento Centralizado baseado em Clusterização com a Abordagem proposta.	49
Figura 13 – Operação na Rede de um Protocolo de Roteamento Centralizado baseado em Clusterização com a Abordagem proposta.	50
Figura 14 – Grafo direcionado representando os clusters-head de uma RSSF.	51
Figura 15 – Relação entre as Capacidades e as Distâncias.	52
Figura 16 – Resultado do cálculo das capacidades a partir das distâncias.	55
Figura 17 – Adaptação do grafo para fluxo em redes.	57
Figura 18 – Resultado final da modelagem para fluxo em redes.	59
Figura 19 – Fluxo Máximo encontrado.	60
Figura 20 – Solução encontrada pela modelagem por fluxo em redes.	62
Figura 21 – Topologia e densidade do Cenário 1.	64
Figura 22 – Topologia e densidade do Cenário 2.	65
Figura 23 – Consumo médio de energia do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.	70
Figura 24 – Consumo médio de energia do GAEEP e do GAEEP-MaxFlow no Cenário 1.	71
Figura 25 – Consumo médio de energia de cada nó para o Cenário 2.	72

Figura 26 – Quantidade de dados recebidos pela Estação Base do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.	73
Figura 27 – Quantidade de dados recebidos pela Estação Base do GAEEP e do GAEEP-MaxFlow no Cenário 1.	74
Figura 28 – Quantidade de dados recebidos pela Estação Base no Cenário 2.	75
Figura 29 – Tempo de morte do primeiro nó do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.	76
Figura 30 – Tempo de morte do primeiro nó do GAEEP e do GAEEP-MaxFlow no Cenário 1.	78
Figura 31 – Tempo de morte do primeiro nó no Cenário 2.	79
Figura 32 – Tempo de Vida da Rede (50% de nós Mortos) do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.	80
Figura 33 – Tempo de Vida da Rede (50% de nós Mortos) do GAEEP e do GAEEP-MaxFlow no Cenário 1.	81
Figura 34 – Tempo de Vida da Rede (50% de nós Mortos) no Cenário 2.	82
Figura 35 – Tempo de Vida da Rede (90% de nós Mortos) do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.	84
Figura 36 – Tempo de Vida da Rede (90% de nós Mortos) do GAEEP e do GAEEP-MaxFlow no Cenário 1.	85
Figura 37 – Tempo de Vida da Rede (90% de nós Mortos) no Cenário 2.	86

LISTA DE QUADROS

Quadro 1 – Exemplo do funcionamento da equação de eleição dos clusters-head ($P_i(t)$).	32
Quadro 2 – Estrutura de uma abelha que representa uma solução.	34
Quadro 3 – Representação de um cromossomo (configuração dos CHs da rede).	36
Quadro 4 – Síntese dos Protocolos Leach, Leach-C, Bee-C e GAEEP.	40
Quadro 5 – Síntese dos Protocolos MCCR e SAMF.	45
Quadro 6 – Cenários	63
Quadro 7 – Parâmetros da Rede.	65
Quadro 8 – Parâmetros do Simulador.	66
Quadro 9 – Parâmetros do Protocolo Leach-C.	67
Quadro 10 – Parâmetros do Protocolo Bee-C.	67
Quadro 11 – Parâmetros do Protocolo GAEEP.	67
Quadro 12 – Resultados do consumo médio de energia em Joules de cada protocolo nos Cenários 1 e 2.	71
Quadro 13 – Resultados da Qt. de Pacotes entregues à Estação Base nos Cenários 1 e 2.	75
Quadro 14 – Resultados do Tempo de morte do primeiro nó nos Cenários 1 e 2.	79
Quadro 15 – Resultados do Tempo de Vida da Rede (50% de nós Mortos) nos Cenários 1 e 2.	82
Quadro 16 – Resultados do Tempo de Vida da Rede (90% de nós Mortos) nos Cenários 1 e 2.	85
Quadro 17 – Resultados obtidos pela Abordagem no Cenário 1.	88
Quadro 18 – Resultados obtidos pela Abordagem no Cenário 2.	88

LISTA DE ABREVIATURAS E SIGLAS

Bee-C	<i>A Bio-inspired Energy Efficient Cluster-based Algorithm.</i>
CH	<i>Cluster-head.</i>
GAEEP	<i>Genetic Algorithm-based Energy-Efficient Clustering Hierarchy Protocol.</i>
IP	<i>Internet Protocol.</i>
Leach	<i>Low-energy Adaptative Clustering Hierarchy.</i>
Leach-C	<i>Low-energy Adaptative Clustering Hierarchy - Centralized.</i>
MANET	<i>Mobile Ad hoc Network.</i>
MCCR	<i>Minimum-cost Capacity-constrained Routing.</i>
RSSF	<i>Redes de Sensores sem Fio.</i>
RSSI	<i>Received Signal Strenght Intensity.</i>
SAMF	<i>Self-adapting Maximum Flow.</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	MOTIVAÇÃO	16
1.2	OBJETIVOS	16
1.2.1	Objetivo Geral	16
1.2.2	Objetivos Específicos	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	REDES DE SENSORES SEM FIO	18
2.1.1	Definição	18
2.1.2	Nós Sensores	19
2.1.3	Estação Base	21
2.1.4	Características de Redes de Sensores sem Fio	21
2.2	TEORIA DOS GRAFOS	25
2.2.1	Fluxo em Redes	26
3	TRABALHOS RELACIONADOS	30
3.1	PROTOCOLOS QUE TRABALHAM COM CLUSTERIZAÇÃO	30
3.1.1	Protocolo Leach	30
3.1.2	Protocolo Leach-C	32
3.1.3	Protocolo Bee-C	33
3.1.4	Protocolo GAEEP	36
3.1.5	Síntese dos Protocolos Leach, Leach-C, Bee-C e GAEEP	39
3.2	PROTOCOLOS QUE TRABALHAM COM FLUXO EM REDES	40
3.2.1	Protocolo MCCR: Roteamento eficiente em energia com restrições de capacidade	40
3.2.2	SAMF: Algoritmo de roteamento de fluxo máximo auto-adaptativo	43
3.2.3	Síntese dos Protocolos MCCR e SAMF	45
4	ABORDAGEM USANDO FLUXO EM REDES	46
4.1	PROPOSTA	46
4.1.1	Modelagem da rede	50
4.1.2	Modelagem das capacidades do grafo	50
4.1.3	Adaptando o grafo para fluxo em redes	57

4.1.4	Solução para o problema	58
5	SIMULAÇÕES E RESULTADOS	63
5.1	CENÁRIOS E PARÂMETROS DA REDE E DO SIMULADOR	63
5.2	PARÂMETROS DOS PROTOCOLOS SIMULADOS	66
5.3	MÉTRICAS PARA AVALIAÇÃO	67
5.4	RESULTADOS	68
5.4.1	Consumo médio de energia por nó	69
5.4.2	Quantidade de dados recebidos pela Estação Base	70
5.4.3	Tempo de morte do primeiro nó	75
5.4.4	Tempo de vida da rede (50% de nós mortos)	78
5.4.5	Tempo de vida da rede (90% de nós mortos)	83
6	CONCLUSÕES E TRABALHOS FUTUROS	87
	REFERÊNCIAS	90

1 INTRODUÇÃO

Os avanços na tecnologia de dispositivos micro-sensores, com baixo consumo de energia que usam frequências de rádio para se comunicar, têm permitido o desenvolvimento de redes de sensores. Essas redes representam um novo modelo para a coleta de dados em um ambiente e permitem o monitoramento confiável em aplicações que incluem vigilância, detecção de falhas em máquinas, detecção de ameaças, entre outras aplicações (HEINZELMAN et al., 2002).

Redes de sensores são diferentes das redes tradicionais utilizadas pela maioria das pessoas. Elas possuem um grande número de nós com capacidades limitadas, possuindo mecanismos para auto-configuração em casos de falhas de comunicação e perda de nós (LOUREIRO et al., 2003). Uma RSSF opera de forma autônoma com o conjunto de nós colhendo informações do ambiente e enviando para uma estação base; computador de grande porte que recebe os dados.

No estado atual, redes de sensores enfrentam alguns problemas. Um deles está relacionado com a capacidade energética dos nós sensores, sendo que uma grande parte das pesquisas procuram alternativas eficientes para economizar a energia dos sensores, fazendo com que o tempo de vida da rede seja estendido. Outras pesquisas também procuram resolver outros problemas como cobertura e roteamento (REGO et al., 2012).

Devido a esse problema, a implementação de protocolos de roteamento, que tratam a maneira como o nó transmitirá a informação coletada e a forma como ela será transmitida, devem levar em consideração o consumo de energia dos nós. O gasto de energia de um nó está relacionado com o tamanho da informação e com a distância em que essa informação é transmitida de um nó para outro. Quanto maiores forem ambos, maior o consumo. Assim protocolos de roteamento devem ser projetados levando em consideração tal fato.

Muitos protocolos de roteamento foram sugeridos para tentar amenizar esse problema em redes de sensores sem fio. Uma vez que é uma tarefa difícil minimizar o consumo de energia, algoritmos de roteamento propostos na literatura empregam algumas técnicas como a agregação de dados para eliminar redundância, clusterização (divisão da rede em grupos), atribuição de regras para diferentes nós, métodos centrados em dados, algoritmos inspirados na biologia, entre outras (AL-KARAKI; KAMAL, 2004).

Alguns protocolos propostos utilizam a técnica de clusterização para alcançar a economia de energia. Essa técnica consiste em dividir a rede em grupos e eleger para cada grupo um líder. Esse líder se encarrega de enviar os dados coletados do seu grupo para a estação base.

Para balancear o consumo, o processo de clusterização é dividido em rodadas, ou seja, os nós líderes são trocados depois de um certo tempo fixado pelo protocolo de roteamento. Alguns protocolos encontrados em (REGO et al., 2012) (VIJAY; GUPTA, 2013) (JAVAID et al., 2013) (ABO-ZAHHAD et al., 2014) utilizam diferentes algoritmos para alcançar a divisão da rede em grupos (clusterização).

Dentre os protocolos que trabalham com clusterização, alguns deles também implementam uma comunicação multihop entre os clusters-head para repassar os dados para a Estação Base. Contudo, durante o tempo de uma rodada, esse detalhe pode ocasionar a sobrecarga de um cluster-head, visto que este terá que transmitir as suas informações assim como os dados de outros clusters-head vizinhos. Nessa situação, o cluster-head pode consumir rapidamente toda a sua energia.

Uma outra questão a considerar é que protocolos de clusterização estabelecem tempos fixos para a duração das rodadas. Esse fato não leva em consideração a capacidade energética atual do nó. Isso pode acarretar situações em que um cluster-head não possui mais energia para repassar os dados do seu cluster, porém o tempo da rodada ainda não expirou, ocasionando assim, perda de informação referente à área que tal cluster-head abrange.

Para evitar situações como essas citadas, este trabalho objetiva propor uma nova abordagem que pode ser incorporada a protocolos de roteamento centralizados que trabalham com clusterização. Essa incorporação irá habilitar uma melhor maneira para rotear as informações de forma multihop e também para definir um período para as rodadas.

Com essa abordagem, um cluster-head irá distribuir suas transmissões para outros nós na rede com a finalidade de não sobrecarregar nenhum cluster-head vizinho. Outra característica da abordagem é que, ao invés de estabelecer um tempo fixo para que a clusterização aconteça novamente, esse processo será engatilhado observando-se as capacidades energéticas de cada nó cluster-head. Como a configuração da rede estará ligada à capacidade de um nó, será evitado tanto o problema da sobrecarga como o do tempo fixo da rodada.

Na tentativa de alcançar os objetivos da proposta, *fluxo em redes* será utilizado (CORMEN, 2002), onde a rede será modelada como um problema de fluxo máximo. Essa técnica de otimização em grafos irá indicar para quem um cluster-head deve transmitir e também irá informar em que momento deve ocorrer a clusterização.

Assim, a proposta procura alcançar um melhoramento de protocolos de roteamento que enfrentam os problemas citados devido à utilização da técnica de clusterização. Esse

melhoramento consiste em tornar o protocolo mais eficiente em energia, balanceando o consumo de cada nó como também distribuindo melhor suas transmissões para obter um maior tempo de vida da rede.

A proposta pretende melhorar protocolos cujo domínio de aplicação seja uma rede com uma grande quantidade de sensores e com uma coleta periódica de dados. Isto é, os sensores constantemente coletam as informações do ambiente e enviam para a Estação Base. Um observador interessado nesses dados é que irá interpretar os valores medidos pelos sensores e inferir alguma informação do ambiente. Como exemplo desse domínio de aplicação, tem-se o monitoramento ambiental.

O corrente trabalho está dividido da seguinte forma: a Seção II apresenta a fundamentação teórica; a Seção III apresenta os trabalhos relacionados; a proposta é conceituada e detalhada na Seção IV; a validação e os resultados da proposta são indicados na Seção V e a conclusão do trabalho se encontra na Seção VI.

1.1 MOTIVAÇÃO

Com a crescente utilização de Redes de Sensores sem Fio para diversas aplicações, é de suma importância e relevância o aprimoramento de técnicas que resolvam ou amenizem os problemas encontrados nessas redes. Assim, estudos e tentativas, que melhoram a forma como os sensores se comunicam entre si e como eles economizam energia, podem expandir ainda mais as possíveis aplicações comerciais e de pesquisa que fazem uso de nós sensores.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho tem como objetivo geral propor uma abordagem que pode ser incorporada a protocolos de roteamento centralizados que trabalham com clusterização. Através dessa incorporação, será habilitada uma melhor maneira de rotear os dados e também definido um novo período para as rodadas. Espera-se então, tornar o protocolo mais eficiente, sem problemas de sobrecarga e morte prematura dos nós, alcançando assim um balanceamento no consumo de cada nó, como também uma melhor distribuição de suas transmissões.

1.2.2 Objetivos Específicos

Os objetivos específicos desse trabalho são listados a seguir:

- a) Fazer um levantamento bibliográfico sobre Redes de Sensores e Teoria dos Grafos;
- b) Fazer um Estudo sobre os trabalhos mais atuais que envolvem Redes de Sensores, clusterização e fluxo em redes;
- c) Propor uma abordagem usando fluxo em redes para o melhoramento de protocolos de roteamento baseados em clusterização;
- d) Implementar e validar a proposta através de simulações, comparando-a com as versões originais dos protocolos melhorados com a abordagem proposta.

2 FUNDAMENTAÇÃO TEÓRICA

A seguinte seção tem como objetivo apresentar os conceitos sobre Redes de Sensores sem Fio, como também as definições sobre grafos e fluxo em redes.

2.1 REDES DE SENSORES SEM FIO

2.1.1 Definição

Uma rede de sensores consiste de um conjunto de vários nós sensores e um computador de maior porte chamado de estação base. Os nós captam as informações do ambiente e enviam para a estação base, que pode se comunicar com outras redes e outros dispositivos para que os dados coletados sejam processados (LABRADOR; WIGHTMAN, 2009).

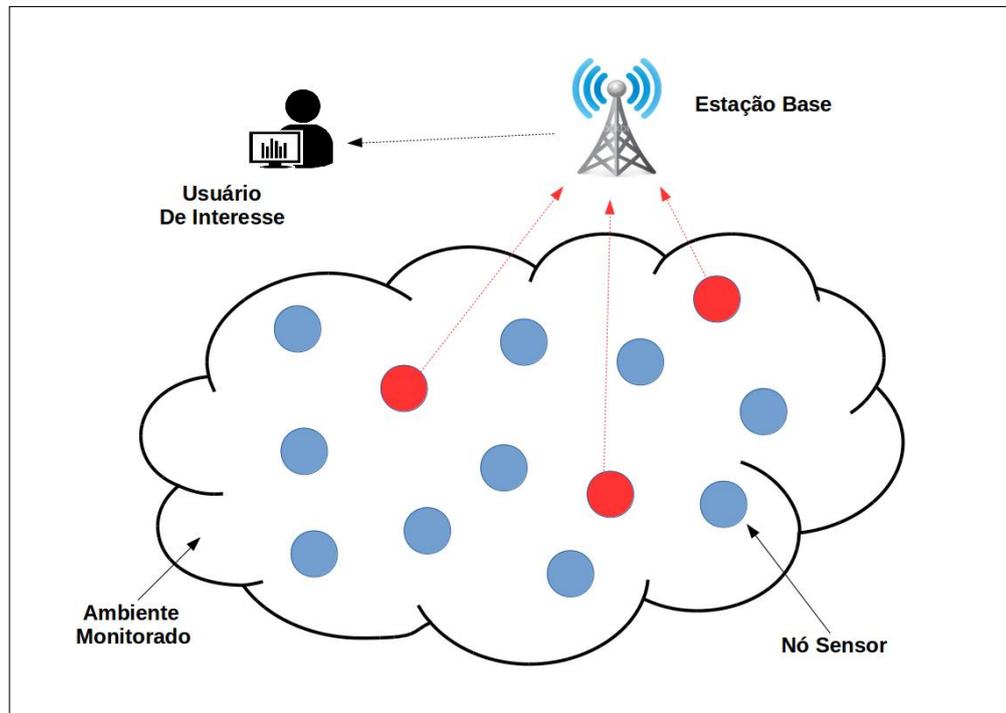
Os avanços na tecnologia de dispositivos micro-sensores, com baixo consumo de energia que usam frequências de rádio para se comunicar, têm permitido cada vez mais o desenvolvimento de Redes de Sensores. Elas representam um novo modelo para a coleta de dados em um ambiente e permitem o monitoramento confiável em aplicações que incluem vigilância, detecção de falhas em máquinas, detecção de ameaças, entre outras aplicações (HEINZELMAN et al., 2002). A Figura 1 ilustra o conceito de Redes de Sensores.

Redes de sensores são diferentes das redes tradicionais utilizadas pela maioria das pessoas. Elas possuem nós sensores com capacidades limitadas e mecanismos para auto-configuração em casos de falhas de comunicação e perda de nós. Uma RSSF opera de forma independente, onde não há interação humana direta nos sensores para que eles realizem a sua função de coleta de dados (LOUREIRO et al., 2003).

Outra informação sobre redes de sensores é que elas podem ser visualizadas como sendo uma variante de redes móveis *ad hoc* (MANET - *Mobile Ad hoc Network*). A comunicação feita entre os nós e a estação base não necessita de uma infraestrutura. Diferentemente de redes tradicionais como a Internet, os pontos de acesso ou entidades de roteamento não são necessários entre os nós sensores. A comunicação entre eles é feita diretamente.

Por causa da inexistência de infraestrutura, redes *ad hoc* demandam a contribuição distribuída dos nós da rede para o repasse dos pacotes. Nessas redes todos os nós colaboram na questão do roteamento, desempenhando funções de roteador (FERNANDES et al., 2006).

Figura 1 – Redes de Sensores sem Fio.



Fonte: Elaborado pelo autor

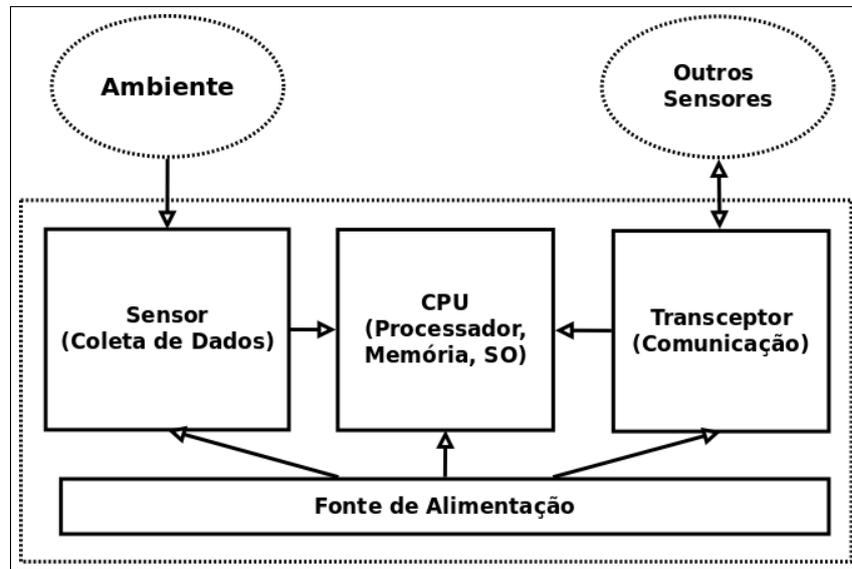
2.1.2 Nós Sensores

Os nós sensores são dispositivos eletrônicos equipados com capacidade de sensoria-mento, processamento e comunicação. Eles possuem componentes básicos como: transceptor, memória, processador, sensor e bateria. A redução do tamanho do sensor diminui a capacidade de seus componentes e também o seu preço. A bateria armazena a energia do sensor que tem capacidade limitada e pouca capacidade de reposição (RÊGO, 2011).

Estes sensores têm a capacidade de coletar, rotear e comunicar dados para outros nós sensores ou diretamente para a estação base. Uma grande quantidade desses nós, empregados em regiões bastante amplas, possibilita o sensoriamento abrangente da área, coletando dados com eficiente precisão (AL-KARAKI; KAMAL, 2004). A Figura 2 mostra a arquitetura de um nó sensor, relatando seus componentes principais. O módulo de Sensor é responsável pela coleta de dados. A entidade CPU efetua o processamento e armazenamento dos dados coletados. O hardware transceptor tem a função de transmitir e receber dados de outras fontes. A bateria é a fonte de energia que alimenta todos os módulos do sensor.

Os processadores de um nó sensor são geralmente de 8 bits com frequência de 10 Mhz, responsáveis por fazer algum processamento prévio em cima dos dados, tais como:

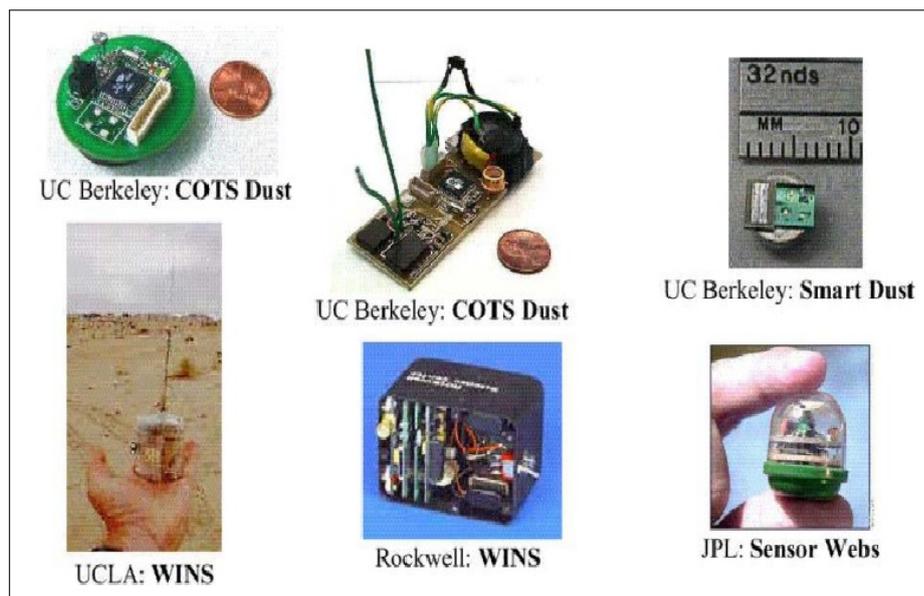
Figura 2 – Arquitetura Nó Sensor.



Fonte: Elaborado pelo autor

contagem, compressão e agregação (RUIZ et al., 2004). Os transceptores têm largura de banda de 1 kbit/s a 1 Mbit/s e são responsáveis pela transmissão e recepção dos pacotes. Para cada bit transmitido ou recebido há um consumo de energia relacionado. As transmissões provocam um consumo maior de energia. A capacidade de memória pode ser de 128 Kbytes a 1 Mbyte. Dependendo das diferentes tecnologias de fabricação de baterias, têm-se diferentes consumos de energia (LOUREIRO et al., 2003). A Figura 3 apresenta alguns exemplos de nós sensores.

Figura 3 – Exemplos de Nós Sensores.



Fonte: (RUIZ et al., 2004)

2.1.3 Estação Base

Uma estação base é um dispositivo de grande porte para onde os nós sensores enviam as informações coletadas do ambiente. Ela pode ser um computador pessoal, um servidor ou um *gateway*, que fornece ligação com outras redes. Não possui as restrições de um nó sensor, podendo ter um grande poder de processamento, alta capacidade de armazenamento e sem preocupações com o consumo de energia (KARL; WILLIG, 2007).

Outra característica da estação base é que ela pode atuar como um *gateway*, responsável por apenas encaminhar o pacote para outras redes até chegar a um observador. A informação coletada pode ser transmitida diretamente por um nó para o *gateway* ou ir sendo repassada por sensores intermediários (LOUREIRO et al., 2003).

2.1.4 Características de Redes de Sensores sem Fio

Como apresentado nos trabalhos (LOUREIRO et al., 2003) e (RUIZ et al., 2004), redes de sensores têm características de acordo com as áreas em que são aplicadas. A aplicação tem grande importância na definição da rede.

Ela ditará aspectos como a arquitetura dos nós sensores, a quantidade de armazenamento, a capacidade de processamento, e o tipo do sensoriamento; quantidades e colocação dos nós na rede; na topologia da rede etc. Algumas características do projeto e objetivos de redes de sensores são listadas a seguir:

- **Implantação dos sensores:** Os nós podem ser densamente empregados no ambiente para coleta de dados. A quantidade de entidades implantadas pode ser maior do que em uma rede MANET (SINGH et al., 2010a). A implantação pode ser caracterizada como determinística ou randômica (AL-KARAKI; KAMAL, 2004).

Quando a situação é determinística, os sensores são colocados de forma manual no ambiente seguindo uma organização definida, onde os dados são encaminhados por rotas já estabelecidas. Em estratégias randômicas, os nós são jogados em uma região de forma aleatória através de aviões ou helicópteros, onde a comunicação entre eles é determinada de forma autônoma e ocorre de maneira não estruturada.

A rede pode ter uma concentração de nós em uma região considerada apropriada para os objetivos dessa rede. Quando a quantidade de sensores é muito grande por unidade, a rede é considerada densa. A rede é definida como esparsa se o número de nós por área é menor

(RUIZ et al., 2004).

- **Coleta e Entrega dos Dados:** A coleta pode acontecer de forma periódica, onde os nós coletam informações de um ambiente em intervalos definidos de tempo. O monitoramento pode ocorrer de forma contínua, onde os sensores verificam os parâmetros de uma área. Um exemplo é uma aplicação de vigilância que necessita constantemente verificar se há movimentação de intrusos.

Coletas em tempo real também são realizadas. Uma grande quantidade de dados é colhida em um pequeno período. Esse tipo de coleta pode ser vista em aplicações que envolvem risco de vida para pessoas, como monitoramento em desastres de prédio, tráfego de veículos etc.

A aplicação induzirá o modelo de entrega de dados à estação base que pode ser classificada em contínua, dirigida à ocorrência de eventos, dirigida à requisição, ou híbrida (AKKAYA; YOUNIS, 2005). Em abordagens contínuas, cada nó sensor recolhe dados do ambiente e envia para outro nó de maneira periódica.

Em situações dirigidas a eventos, os nós transmitem informações somente quando um evento de interesse acontece. Um exemplo é uma aplicação no tráfego, onde um evento de interesse pode ser um veículo passando em determinada rua.

No modelo dirigido às requisições, a estação base faz pedidos sobre parâmetros de um local e os nós implantados nessa região respondem com os dados. Em ambientes híbridos, ocorre a mistura dos modelos contínuos, dirigidos a eventos e às requisições.

- **Endereçamento dos sensores:** Endereçamento baseado em IP não é utilizado em RSSF (AL-KARAKI; KAMAL, 2004), visto que manter endereços para uma grande quantidade de nós se torna um problema para a rede, e o fato de transmitir bits relacionados a endereços IP's influencia no consumo de energia (RUIZ et al., 2004).

Além do mais, de acordo com a aplicação, cada nó terá uma identificação ou não. Sensores colocados em peças numa linha de montagem ou dentro do corpo humano devem ser endereçados unicamente, pois a aplicação deseja saber o local exato de onde o dado foi coletado. Por outro lado, sensores monitorando o ambiente numa dada região não necessitam de uma identificação única, pois o objetivo é saber a ocorrência de algum fato nessa região.

Algumas formas de endereçamento consistem em atribuir um identificador ao nó, que podem ser as coordenadas geográficas do nó. Em outras abordagens, os nós são endereçados

de acordo com o dado que é coletado.

- **Mobilidade dos sensores:** Representa a capacidade de movimentação dos nós nas áreas em que são empregados. Em ambientes variados, agentes externos como o ritmo das marés, as correntes eólicas ou de rios, e a movimentação de animais, determinam a mobilidade dos nós na rede (JABOUR et al., 2007).

Sensores em uma floresta para extrair informações sobre a umidade ou temperatura são estáticos, enquanto sensores ligados a veículos para monitorar condições de tráfego são considerados móveis (TILAK et al., 2002).

- **Agregação dos dados:** Representa a capacidade da rede de fazer agregação (junção) ou não das informações coletadas pelos nós. Com isso a rede é capaz de reduzir o número de mensagens transmitidas (LOUREIRO et al., 2003). Essa capacidade tem o objetivo de eliminar redundância dos dados coletados, devido a um problema chamado *overlap* que acontece quando dois nós monitoram a mesma área e enviam informações iguais para o mesmo nó vizinho que irá repassar a mensagem (AKKAYA; YOUNIS, 2005).
- **Limitação da energia disponível:** Geralmente os sensores são colocados em áreas de difícil acesso, dificultando a sua manutenção e troca. O tempo de vida de um sensor está ligado a quantidade de energia presente em sua bateria. Assim, protocolos para redes de sensores devem ser escolhidos levando em consideração o modelo e a quantidade de energia consumida.
- **Auto-organização da rede:** Capacidade que a rede apresenta quando acontece alguma modificação em sua topologia. Tais mudanças podem ocorrer devido a problemas no canal de comunicação dos nós que os torna inoperantes, ou a inclusão de novos nós na rede. Assim, RSSF devem apresentar algum mecanismo de auto-organização para que a coleta de dados não seja interrompida.

Uma questão importante em redes de sensores é o roteamento. Segundo (AKKAYA; YOUNIS, 2005), roteamento em redes de sensores encontra muitos desafios devido a algumas características que diferem redes de sensores de outras redes sem fio. Muitas vezes não é possível construir um modelo de endereçamento para a instalação de um grande número de nós. Portanto, protocolos de roteamento que utilizam um endereço não podem ser utilizados em redes de sensores.

Uma questão referente ao roteamento é que, diferente do que acontece em redes tradicionais, quase todas as aplicações de redes de sensores requerem o fluxo de dados coletados

de múltiplas regiões para um único ponto central: a estação base.

Mais uma questão é que, o tráfego de dados contém redundância uma vez que muitos sensores podem gerar o mesmo dado coletado por seus nós vizinhos. Tal redundância precisa ser amenizada e resolvida pelos protocolos de roteamento realizando alguma agregação de dados, melhorando o consumo de energia e a largura de banda utilizada (AKKAYA; YOUNIS, 2005).

Devido a tais questões, novos algoritmos foram propostos para realizarem o roteamento em redes de sensores. Eles consideram as características dos nós sensores e da aplicação em que irão atuar.

A classificação de roteamento em RSSF é dividida, de acordo com a estrutura da rede, em: roteamento centrado em dados, roteamento hierárquico ou roteamento geográfico (AKKAYA; YOUNIS, 2005).

No roteamento centrado em dados, a estação base faz requisições a certas regiões e espera por dados dos sensores localizados nestas regiões. Como os dados estão sendo pedidos através de requisições, é necessária a nomeação do dado para especificar as suas propriedades.

No roteamento hierárquico o objetivo é manter o consumo baixo de energia dos nós sensores, atribuindo na rede uma comunicação multihop entre um conjunto de nós, realizando agregação e fusão de dados na ordem de diminuir o número de dados redundantes transmitidos para a estação base.

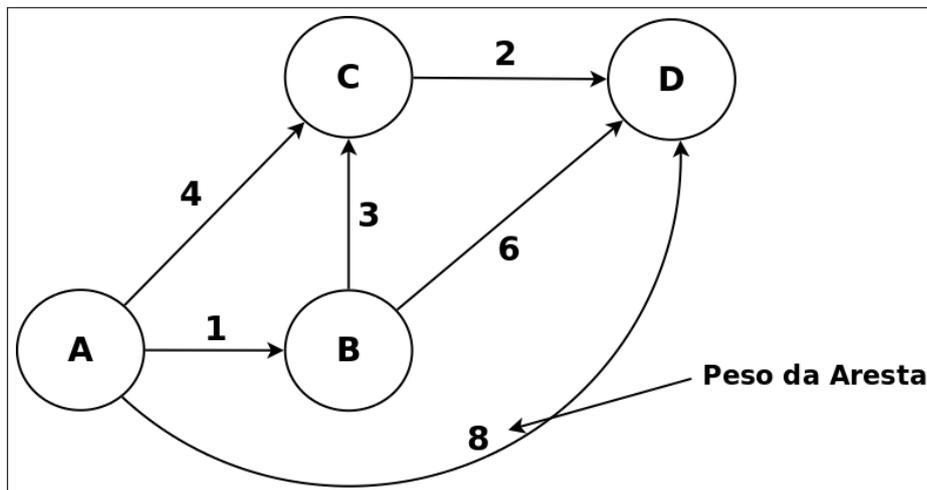
Segundo (REGO et al., 2012), nesse roteamento, os nós podem se organizar em clusters (grupos), ou seja, os protocolos desse tipo de roteamento realizam a clusterização da rede para alcançar a economia de energia. Dois tipos de nós sensores podem ser classificados: os nós fontes e os nós líderes de cada grupo, chamados de cluster-head, que comandam as atividades dentro de um grupo. Os nós fontes recolhem as informações do ambiente e enviam para o seu líder (cluster-head) que agrega os dados, diminuindo a redundância das informações e enviando as mensagens finais para a estação base.

E por último, tem-se o roteamento geográfico. Ele utiliza a localização geográfica dos sensores. Nele, todos os nós têm conhecimento sobre as coordenadas dos outros sensores. Assim, um nó origem pode achar o melhor caminho até um nó destino, pois serão conhecidas as posições de tais nós.

2.2 TEORIA DOS GRAFOS

Conforme apresentado em (CORMEN, 2002), um grafo G é um par (V, E) , onde V é um conjunto finito e E é uma relação binária em V . O conjunto V é chamado de conjunto de vértices de G , e seus elementos são chamados vértices. O conjunto E é chamado de conjunto de arestas de G , e seus elementos são chamados de arestas. Um valor ou peso pode ser atribuído a cada aresta. A Figura 4 ilustra um grafo com o conjunto $V = \{A, B, C, D\}$ e o conjunto $E = \{AB, AC, AD, BC, BD, CD\}$.

Figura 4 – Exemplo de um grafo.



Fonte: Elaborado pelo autor

Um grafo pode ser classificado como direcionado e não direcionado. Em um grafo direcionado, cada aresta está associada a um par ordenado de vértices que são chamados de nós terminais. Assim, se uma aresta e é associada ao par (u, v) de vértices, em um grafo direcionado $G = (V, E)$, então se diz que e é uma aresta dirigida de u para v .

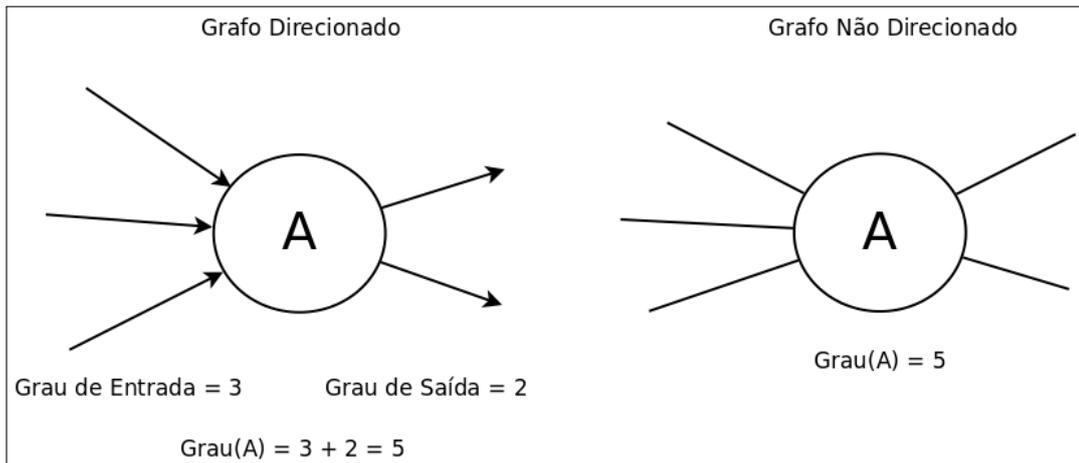
Em um grafo não direcionado $G = (V, E)$, o conjunto de arestas E consiste em pares de vértices ordenados tais que $(u, v) = (v, u)$. Isto é, uma aresta é um conjunto $\{u, v\}$, onde $u, v \in V$ e $u \neq v$. Por convenção, usa-se a notação (u, v) para uma aresta.

Se (u, v) é uma aresta em um grafo $G = (V, E)$, diz-se que o vértice v é adjacente ao vértice u . Quando o grafo é não direcionado, a relação de adjacência é simétrica. Se v é adjacente a u em um grafo direcionado, sua representação pode ser escrita da forma $u \rightarrow v$.

O *grau* de um vértice em um grafo não direcionado é o número de arestas incidentes nele. Em um grafo direcionado, o *grau de saída* de um vértice é o número de arestas que saem dele, e o *grau de entrada* de um vértice é o número de arestas que entram nele. O *grau* de um

vértice em um grafo direcionado é seu grau de entrada somado a seu grau de saída. A Figura 5 mostra o grau de um vértice em um grafo direcionado e não direcionado.

Figura 5 – Grau de um vértice.



Fonte: Elaborado pelo autor

2.2.1 Fluxo em Redes

Segundo (CORMEN, 2002), uma rede de fluxos $G = (V, E)$ é um grafo direcionado em que cada aresta $(u, v) \in E$ tem uma capacidade não negativa $c(u, v) \geq 0$. Se $(u, v) \notin E$, $c(u, v) = 0$. Distinguem-se dois vértices em um fluxo de redes: uma origem s e um sorvedouro t . Supõe-se que cada vértice faça parte de algum caminho entre a origem até o sorvedouro. Isto é, para todo vértice $v \in V$, existe um caminho $s \rightarrow v \rightarrow t$. Assim, o grafo é conectado, e $|E| \geq |V| - 1$.

Seja $G = (V, E)$ um fluxo em rede com uma função de capacidade c . Seja s a origem da rede, e seja t o sorvedouro. Um fluxo em G é uma função de valor real $f : V \times V \rightarrow R$ que satisfaz as três propriedades seguintes:

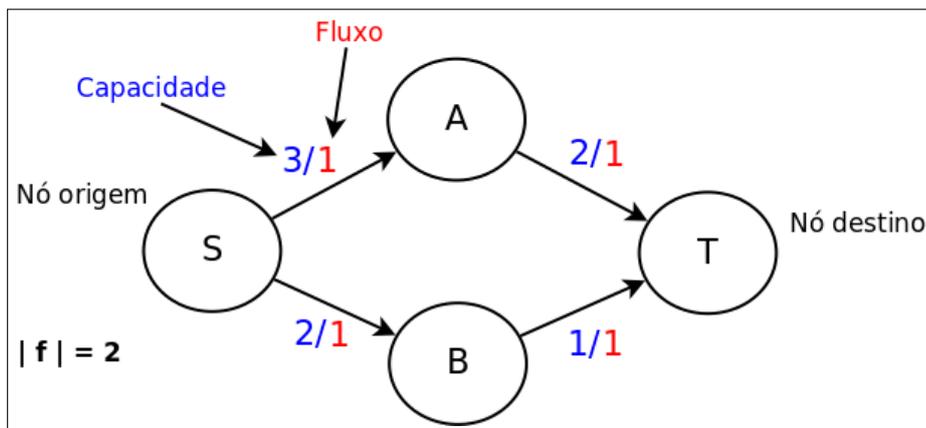
- Restrição de capacidade: Para todo $u, v \in V$, tem-se $f(u, v) \leq c(u, v)$.
- Anti-simetria: Para todo $u, v \in V$, tem-se $f(u, v) = -f(v, u)$
- Conservação do fluxo: Para todo $u \in V - \{s, t\}$, tem-se $\sum_{v \in V} f(u, v) = 0$.

A quantidade $f(u, v)$, que pode ser positiva, 0 ou negativa, é chamada *fluxo* do vértice u até o vértice v . O valor de um fluxo f é definido como $|f| = \sum_{v \in V} f(s, v)$, ou seja, o fluxo total que sai da origem. A notação $|\cdot|$ denota valor de fluxo e não valor absoluto ou cardinalidade.

A restrição de capacidade simplesmente afirma que o fluxo de um vértice até outro

não deve exceder a capacidade dada. A anti-simetria é uma conveniência de notação que afirma que o fluxo de um vértice u até um vértice v é o valor negativo do fluxo no sentido inverso. A propriedade de conservação de fluxo afirma que o fluxo total para fora de um vértice que não seja a origem ou o sorvedouro é 0. Por anti-simetria, pode-se reescrever a propriedade de conservação de fluxo como $\sum_{u,v \in V} f(u,v) = 0$ para todo $v \in V$ menos os vértices s e t . Isto é, o fluxo total em um vértice é 0. A Figura 6 ilustra um exemplo de fluxo em rede.

Figura 6 – Fluxo em rede.



Fonte: Elaborado pelo autor

Um fluxo em uma rede de fluxos pode atingir um valor máximo que não desobedece nenhuma restrição imposta. A esse fato, dá-se o nome de *fluxo máximo*. Na literatura, essa característica também pode ser referenciada como o *problema do fluxo máximo*.

No problema do fluxo máximo, pretende-se encontrar o maior valor de um fluxo que pode ser enviado desde a origem até o sorvedouro sem violar nenhuma restrição. Um fluxo máximo em uma rede de fluxos pode ser encontrado por algoritmos eficientes como Ford-Fulkerson, Edmonds-Karp e Dinic (CORMEN, 2002). Uma outra maneira de se encontrar o fluxo máximo em um fluxo em redes é representá-lo como um problema de programação linear formulado da seguinte forma:

Maximizar:

$$\sum_{j=1}^m x_{sj} - \sum_{k=1}^m x_{ks} (= F) \quad (2.1)$$

Sujeito a:

$$\sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} \begin{cases} \geq 0 & \text{se } i = 1 = s \\ = 0 & \text{se } i \neq 1(s) \text{ ou } i \neq m(t) \\ \leq 0 & \text{se } i = m = t \end{cases} \quad (2.2)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall i, j = 1, \dots, m \quad (2.3)$$

onde,

F = função objetivo ou variável de decisão (Fluxo Máximo);

s = nó de origem da rede;

t = nó sorvedouro da rede;

x_{ij} = quantidade a fluir do vértice i para o vértice j ;

u_{ij} = capacidade de uma aresta com origem no vértice i e destino no vértice j ;

m = quantidade de vértices do grafo.

O fluxo em rede da Figura 6, por exemplo, pode ser modelado e resolvido como um problema de programação linear de acordo com as fórmulas 2.4 2.5 2.6. Uma solução para essa modelagem pode ser encontrada utilizando-se alguma ferramenta de resolução de problemas de otimização. O *solver* utilizado para resolver o modelo dado pelas equações 2.4 e 2.5 foi o *glsol* da biblioteca GLPK (MAKHORIN, 2008). A Figura 7 apresenta o fluxo máximo da rede de fluxos representada na Figura 6.

Maximizar:

$$x_{SA} + x_{SB} \quad (2.4)$$

Sujeito a:

$$\begin{aligned} x_{SA} + x_{SB} &\geq 0 \\ -x_{SA} + x_{AT} &= 0 \\ -x_{SB} + x_{BT} &= 0 \\ -x_{AT} - x_{BT} &\leq 0 \end{aligned} \quad (2.5)$$

$$0 \leq x_{SA} \leq 3; \quad 0 \leq x_{SB} \leq 2;$$

$$0 \leq x_{AT} \leq 2; \quad 0 \leq x_{BT} \leq 1$$

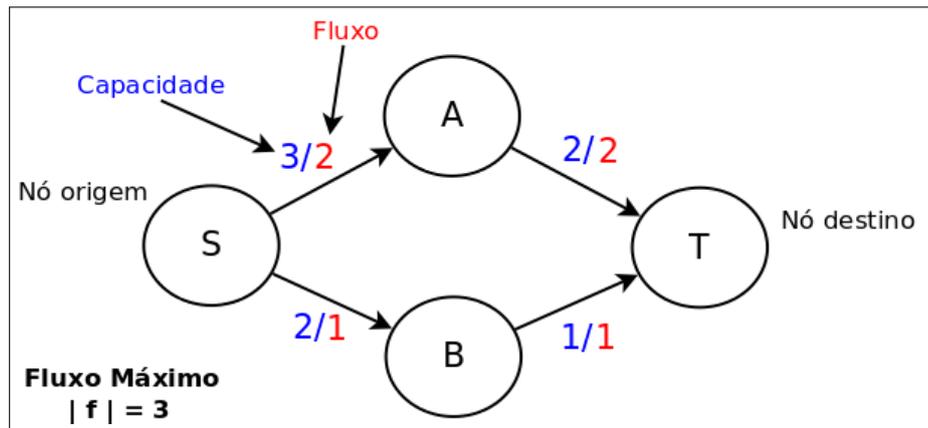
Solução:

$$\text{Fluxo Máximo: } F = x_{SA} + x_{SB} = 3$$

$$\text{Fluxos: } x_{SA} = 2; \quad x_{SB} = 1; \quad (2.6)$$

$$x_{AT} = 2; \quad x_{BT} = 1$$

Figura 7 – Fluxo máximo em um Fluxo em rede.



Fonte: Elaborado pelo autor

3 TRABALHOS RELACIONADOS

Esta seção apresenta trabalhos que utilizam técnicas de clusterização como também o conceito de fluxo em redes para resolver problemas inerentes em redes de sensores.

3.1 PROTOCOLOS QUE TRABALHAM COM CLUSTERIZAÇÃO

3.1.1 Protocolo Leach

O Leach (HEINZELMAN et al., 2002) é um protocolo de roteamento distribuído e hierárquico que tem o objetivo de estender o tempo de vida dos nós sensores, aumentando assim o tempo de vida da rede. Nele, os nós se organizam entre si para a formação de grupos (clusters) e para a eleição de um líder (cluster-head) para cada grupo. Assim, a rede é clusterizada de forma distribuída, não tendo nenhuma participação da Estação Base ou cálculo centralizado para a formação dos clusters.

Quando os clusters da rede são definidos, cada nó pertencente a um grupo envia seus dados para o seu líder. Ao receber os dados de todos os membros do seu grupo, o cluster-head executa procedimentos em cima dos dados recebidos (por exemplo, fusão dos dados) e transmite as informações para a Estação Base (HEINZELMAN et al., 2002).

Como os líderes têm um papel importante dentro de cada grupo, eles acabam gastando mais energia do que os demais. Para evitar que eles fiquem inoperantes rapidamente, o Leach funciona por rodadas. Em cada rodada, os líderes são mudados para distribuir o consumo na rede. Existem duas fases que compõem uma rodada: A fase de formação dos grupos (clusters) e a fase de coleta e transmissão de dados.

Na fase de formação dos clusters, um algoritmo distribuído elege os líderes e constrói os grupos. O objetivo é encontrar um certo número k de clusters durante cada rodada. Os nós não precisam ter conhecimento global da rede, podendo fazer decisões autônomas sem nenhum controle centralizado.

Cada sensor da rede decide se ele será ou não um cluster-head de acordo com uma fórmula probabilística $P_i(t)$ no tempo t . A fórmula leva em consideração um número k desejável de clusters-head na rede e o fato de que um nó somente pode se tornar um líder a cada N/k rodadas, onde N é o número total de nós na rede. A probabilidade $P_i(t)$ de um nó se tornar um

cluster-head é definida pela equação 3.1:

$$P_i(t) = \begin{cases} \frac{k}{N - k*(r \bmod \frac{N}{k})}, & \text{Se } i \in G \\ 0, & \text{Caso Contrário} \end{cases} \quad (3.1)$$

onde,

N = Quantidade de nós da rede;

k = Número de clusters-head desejáveis para a rede;

i = Nó sensor que calcula a probabilidade;

r = ID do *round* corrente indexado a partir do zero (0, 1, 2, 3, ...);

G = Conjunto de sensores que podem se tornar cluster-head na rodada r .

Os nós sensores selecionam um valor aleatório T entre 0 e 1. Caso T seja menor do que $P_i(t)$, então o nó i será um cluster-head. O grupo G , especificado pela equação 3.1, contém os nós que ainda não foram clusters-head nas N/k rodadas. Os nós que já foram líderes dentro dessas N/k rodadas ficam fora do grupo G . À medida que os *rounds* aumentam, a quantidade de nós em G vai diminuindo e a probabilidade de os nós se tornarem líderes aumenta. Quando se chega ao final das N/k rodadas, todos os nós já foram líderes pelo menos uma vez na rede. Assim, um ciclo de N/k rodadas se inicia novamente, onde cada nó pode se tornar cluster-head uma outra vez.

O quadro 1 exemplifica o funcionamento da equação 3.1. Considerando $N = 100$ e $k = 5$, o quadro mostra o valor de $P_i(t)$ que os nós calculam à medida que as rodadas aumentam. Conforme apresentado no quadro, quando são atingidas as N/k rodadas ($r = 19$), o grupo G possui apenas 5 nós (quantidade desejável de CHs) e assim, a probabilidade que cada um tem para se tornar cluster-head é 100%. A partir do *round* 20, todos os nós podem se tornar líderes novamente nas próximas N/k rodadas.

Os nós que se tornam clusters-head devem enviar uma mensagem para os outros nós da rede indicando que eles são os líderes naquela rodada. Assim, eles enviam um mensagem de anúncio (ADV-CH) de forma broadcast. Cada nó não-líder que recebe esse anúncio determina quem será seu cluster-head baseado na intensidade do sinal recebido (*RSSI - Received Signal Strength Intensity*), ou seja, nós CHs com maior força de sinal indicam maior proximidade e requerem menor gasto de energia na comunicação.

Uma vez que cada nó decide em qual cluster irá ficar, ele cria e envia uma mensagem de ingresso (JOIN-REQ) para o seu respectivo cluster-head. Assim, os CHs armazenam os nós

Quadro 1 – Exemplo do funcionamento da equação de eleição dos clusters-head ($P_i(t)$).

Round r	$r \bmod \frac{N}{k}$	Qt. de nós em $G = \{ N - k * (r \bmod \frac{N}{k}) \}$	Valor de $P_i(t)$ (entre 0 e 1)
0	0	100	0.0500
1	1	95	0.0526
2	2	90	0.0555
3	3	85	0.0588
...
19	19	5	1
20	0	100	0.0500
21	1	95	0.0526
22	2	90	0.0555
23	3	85	0.0588
...

Fonte: Elaborado pelo autor

que pertencem ao grupo em que ele é líder. Os sensores que não recebem a mensagem de anúncio (ADV-CH) do cluster-head acabam enviando seus dados diretamente para a Estação Base. Para evitar colisões entre as mensagens de dados, um cluster-head aloca *slots* de tempo para cada membro de seu cluster. Assim, cada nó-membro transmite seus dados somente no seu *slot* e vai para um modo de baixo consumo (modo *sleep*) quando o tempo desse *slot* termina.

Após a fase de formação dos clusters, a fase de coleta de dados é iniciada. Nesse momento, os nós coletam as informações do ambiente, enviam para o seu líder (cluster-head) que por sua vez envia para a Estação Base. Quando o tempo de rodada finaliza, todo o processo (fase de formação dos clusters e coleta de dados) se inicia novamente.

3.1.2 Protocolo Leach-C

O Leach-C (HEINZELMAN et al., 2002) é um protocolo também hierárquico baseado no Leach. A diferença entre eles é que o Leach-C usa um algoritmo centralizado para a formação dos clusters. O Leach-C busca encontrar melhores clusters para tentar amenizar um ponto fraco do protocolo Leach. Neste último, os líderes de cada grupo são escolhidos somente de forma probabilística, o que pode formar clusters-head mal distribuídos e com pouca energia. Assim, o Leach-C parte da ideia de usar um algoritmo centralizado para produzir melhores clusters, localizando os clusters-head de forma mais eficiente na rede.

O funcionamento do protocolo é por rodadas divididas em duas fases: formação de clusters e coleta de informações. Na primeira fase, cada nó envia as suas informações de coordenadas geográficas, colhidas através de um sistema GPS, e nível de energia, para a Estação Base que tomará conhecimento de cada nó da rede que enviou a informação. Como ponto de

partida para selecionar bons clusters, a Estação Base calcula a média de energia de todos os nós conhecidos. Após esse passo, ela seleciona somente os nós que estejam acima da média para serem possíveis clusters-head. A partir desses nós, ela usa o algoritmo de *Têmpera Simulada* para encontrar k clusters ótimos.

Uma vez que os líderes e os grupos (clusters) são definidos, a Estação Base envia uma mensagem de forma broadcast contendo o ID do cluster-head para cada nó. Caso o ID enviado seja igual ao ID do nó, esse nó saberá que ele é um cluster-head. Caso contrário, o nó que recebeu o ID saberá que este representa o seu cluster-head. Os sensores que se tornam líderes criam pacotes TDMA e definem os slots que os nós vão utilizar para transmitir.

A fase de coleta de informações é praticamente a mesma do protocolo Leach, ou seja, os nós coletam as informações dentro do cluster em que estão e enviam os dados coletados para o seu líder, que envia para a Estação Base. Quando o tempo da rodada finaliza, todo o processo descrito acima começa novamente.

3.1.3 Protocolo Bee-C

O protocolo Bee-C (REGO et al., 2012) foi desenvolvido baseado no Leach-C com a diferença de que a clusterização da rede é alcançada através de um algoritmo bioinspirado no comportamento de abelhas. O Bee-C foi idealizado para redes com a disseminação constante de dados.

Além de usar um algoritmo bioinspirado em abelhas, o Bee-C implementa novas funções objetivos que visam otimizar o consumo de energia e assim obter melhores soluções de clusters-head para a rede. Dessa forma, o objetivo desse trabalho é estender o tempo de vida da rede, prolongando a duração da bateria dos nós sensores. O protocolo trabalha com rodadas e a duração de cada uma delas é definida como parâmetro de entrada. Cada rodada é composta de duas fases: formação dos clusters e comunicação de dados.

A formação dos clusters se inicia com a Estação Base recebendo as informações geográficas e os níveis de energia de cada nó. De posse dessas informações, ela calcula a média de energia da rede e aqueles nós que estão com a energia acima dessa média se tornam elegíveis para serem possíveis clusters-head. Esse conjunto de nós elegíveis representa uma abelha do algoritmo. Existem dois tipos principais de abelhas: a rainha e os zangões. A abelha rainha tem a função de gerar novas abelhas descendentes e os zangões a de realizar o cruzamento com a rainha. O quadro 2 apresenta a estrutura de uma abelha (rainha ou zangão) que representa uma

possível solução de clusters-head na rede. Cada posição dessa abelha informa o *id* e a posição do cluster-head que faz parte da solução. O tamanho da estrutura da abelha é a quantidade de CHs desejáveis na rede, definida como parâmetro de entrada para o protocolo.

Quadro 2 – Estrutura de uma abelha que representa uma solução.

Nó CH 1	Nó CH 2	...	Nó CH N
id CH	id CH	id CH	id CH
posição x	posição x	posição x	posição x
posição y	posição y	posição y	posição y

Fonte: Elaborado pelo autor

O Bee-C implementa duas funções objetivos para avaliar a qualidade de uma abelha. A primeira função tenta estimar, durante determinada rodada, o montante médio de energia de cada cluster. A segunda função tenta encontrar nós comuns mais próximos de seu líder como também clusters-head com maior energia e mais próximos da Estação Base. Como a primeira função objetivo trouxe melhores resultados para o Bee-C, quando comparado com outros protocolos, ela representa a principal função objetivo do protocolo. Assim, o algoritmo busca maximizar o valor dessa função, representada pela equação 3.2:

$$F_{obj} = \sum_{k=1}^p \frac{E_{T_k} - E_{C_k}}{N_k} \quad (3.2)$$

onde,

p = Quantidade de clusters-head na rede;

E_{T_k} = Energia total do cluster k , considerando os nós fontes e o CH;

E_{C_k} = Energia total gasta pelo cluster k , considerando os nós fontes e o CH;

N_k = Quantidade de nós do cluster k .

A energia total, E_{T_k} , de um cluster k é dada pela equação 3.3:

$$E_{T_k} = \sum_{j=1}^{N_k} E_{no_j} \quad (3.3)$$

onde,

E_{no_j} = Energia corrente de um nó j ;

N_k = Quantidade de nós do cluster k .

O total de energia gasta, E_{C_k} , por um cluster k é representada pela equação 3.4:

$$E_{C_k} = \underbrace{N_k (E_{elec} + \varepsilon_{tr} d_{CH-toBS}^4)}_{\text{CH enviando pacotes do cluster para a Sink}} + \underbrace{\sum_{j=1}^{N_k} (E_{elec} + \varepsilon_{fs} d_{j-toCH}^2)}_{\text{nós do cluster enviando dados para o seu CH}} + \underbrace{N_k E_{da}}_{\text{CH realiza fusão dos dados}} + \underbrace{(N_k - 1) E_{elec}}_{\text{CH recebe dados dos nós do cluster}} \quad (3.4)$$

onde,

E_{elec} = Energia elétrica, em Joules, necessária para transmitir um bit;

ε_{tr} = Energia de amplificação do sinal. Modelo de consumo de energia de dois raios, utilizado quando a distância entre a origem e o destino é muito grande;

ε_{fs} = Energia de amplificação do sinal. Modelo de consumo de energia de espaço livre, utilizado para pequenas distâncias entre a origem e o destino;

E_{da} = Energia, em Joules, para agregação/fusão de dados;

$d_{CH-toBS}$ = Distância de um CH para a Estação Base;

d_{j-toCH} = Distância de um nó j para o seu cluster-head;

N_k = Quantidade de nós do cluster k .

A partir do momento que são definidos os nós elegíveis na rede, uma população inicial de abelhas é gerada. A melhor solução dessa população, indicada pela função objetivo 3.2, é escolhida para ser a abelha rainha. As outras abelhas são definidas como abelhas zangões. O método da roleta é usado para selecionar determinada quantidade de zangões para cruzarem com a rainha. Uma geração de descendentes é formada a partir do cruzamento de cada zangão selecionado com a rainha. O processo de cruzamento utilizado é o de máscara binária.

Uma vez gerados os descendentes, realiza-se uma mutação em cada um deles de acordo com uma probabilidade (taxa de mutação). Caso a abelha descendente sofra mutação, um cluster-head de sua estrutura é selecionado de forma aleatória e substituído por um outro nó elegível da rede. Se o novo descendente gerado apresentar um valor melhor do que a sua versão sem mutação, este último é substituído pelo novo descendente. Após a mutação, verifica-se a melhor solução presente na população corrente. Se essa solução apresentar valor melhor do que a rainha corrente, ela será a nova rainha para a próxima geração.

Todas as etapas do algoritmo (seleção dos zangões, cruzamento com a rainha, mutação dos descendentes e escolha da nova rainha) são repetidas até um número máximo de gerações, definida como parâmetro de entrada para o algoritmo.

3.1.4 Protocolo GAEEP

O protocolo GAEEP foi proposto por (ABO-ZAHHAD et al., 2014), sendo baseado em clusterização, com o objetivo de prolongar o tempo de vida da rede encontrando um número ótimo de clusters-head e suas localizações. A clusterização é realizada através de *Algoritmos Genéticos*, onde essa técnica tenta minimizar o consumo de energia de cada nó. O funcionamento do GAEEP acontece em rodadas, onde estas são divididas em: formação dos clusters e coleta de dados.

Na fase de formação dos clusters, todos os nós começam enviando suas posições geográficas e níveis de energia para a Estação Base. Quando esta obtém conhecimento dos nós na rede, ela aplica *Algoritmos Genéticos* para encontrar uma quantidade ótima de CHs e suas localizações na rede. Além de encontrar os líderes de cada grupo, a Estação Base também determina em qual grupo um nó que não é líder irá ficar. Caso um sensor esteja mais próximo da Estação Base do que qualquer outro nó CH, esse nó transmitirá diretamente para ela.

Os passos iniciais do protocolo GAEEP se iniciam com a Estação Base construindo um conjunto de sensores $S = \{s_1, s_2, s_3, \dots, s_{n_S}\}$. Somente fazem parte desse conjunto os nós que têm energia maior do que a média de energia de todos os nós na rede. Assim, um cromossomo é criado com tamanho n_S , onde cada gene desse cromossomo representa um nó sensor do conjunto S . Uma vez definido o cromossomo, cada gene é configurado com o valor 0 ou 1, gerado aleatoriamente. Um gene contendo o valor 1 indica que o nó s_i é um cluster-head. Caso o valor seja 0, s_i será um nó comum. Como exemplo, pode-se considerar um conjunto contendo 10 sensores com energia acima da média, $S = \{s_1, s_2, s_3, \dots, s_{10}\}$. Considerando também que s_3, s_5, s_6 e s_9 são sensores CHs e os outros são nós comuns, um exemplo de cromossomo inicial que representa a configuração dos clusters-head da rede é mostrado no quadro 3.

Quadro 3 – Representação de um cromossomo (configuração dos CHs da rede).

Conjunto de Sensores (S)	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Cromossomo dos CHs	0	0	1	0	1	1	0	0	1	0

Fonte: Elaborado pelo autor

Uma vez definido o cromossomo, o GAEEP implementa uma função objetivo que indicará se tal cromossomo é uma boa solução, ou seja, se a configuração de clusters-head é ótima. Essa função tenta minimizar a energia dissipada no processo de comunicação entre os sensores. A função de avaliação também leva em consideração a quantidade de clusters-head,

visto que poucos líderes resultam em um consumo eficiente, enquanto que muitos CHs consomem um volume maior de energia comparado aos nós não líderes. Portanto, a função objetivo $F(X)$ usada pelo *Algoritmo Genético* para determinar quão bom é um cromossomo, é determinada pela equação 3.5.

$$F(X) = w \left(\frac{E_{disp}}{E_{live}} \right) + (1 - w) \left(\frac{L}{N_{live}} \right) \quad (3.5)$$

onde,

w = Fator de peso ($0 \leq w \leq 1$);

E_{disp} = Total de energia dissipada por todos os nós vivos na rede;

E_{live} = Total de energia de todos os nós vivos;

L = Quantidade de clusters-head;

N_{live} = Quantidade de nós vivos na rede.

A energia total de todos os nós E_{live} é descrita pela equação 3.6:

$$E_{live} = \sum_{j=1}^{N_{live}} E_o(j) = N_{live} E_o \quad (3.6)$$

onde,

E_o = Energia inicial de um sensor;

N_{live} = Quantidade de nós vivos na rede.

A energia de dissipação de todos os nós vivos E_{disp} é dada pela equação 3.7:

$$E_{disp} = \sum_{i=1}^L \left[(E_{CH}(i) + E_{CH_{CP}}(i)) + \sum_{k=1}^{N_{live}-L} (E_{mem_i}(k) + E_{mem_{CP_i}}(k)) \right] \quad (3.7)$$

onde,

E_{CH} = Energia dissipada por um cluster-head para transmitir pacotes de dados;

$E_{CH_{CP}}$ = Energia dissipada por um cluster-head para transmitir pacotes de controle;

E_{mem_i} = Energia dissipada por um nó membro do cluster-head i para transmitir pacotes de dados;

$E_{mem_{CP_i}}$ = Energia dissipada por um nó membro do cluster-head i para transmitir pacotes de controle;

N_{live} = Quantidade de nós vivos na rede;

L = Quantidade de clusters-head.

As energias de dissipação dos clusters-head, E_{CH} e $E_{CH_{CP}}$, e dos nós membros, E_{mem_i} e $E_{mem_{CP_i}}$, são descritas, respectivamente, pelas equações 3.8, 3.9, 3.10 e 3.11:

$$E_{CH}(i) = \underbrace{k E_{elec} m_i}_{\text{recebendo sinal dos nós membros}} + \underbrace{k E_{da} (m_i + 1)}_{\text{agregando } m_i + 1 \text{ sinais}} + \underbrace{k E_{elec} + E_{amp} k d_{i-toBS}^p}_{\text{transmitindo o sinal agregado para a Sink}} \quad (3.8)$$

$$E_{CH_{CP}}(i) = \underbrace{k_{CP} E_{elec}}_{\text{recebendo requisição da Sink}} + \underbrace{k_{CP} E_{elec} + E_{amp} k_{CP} d_{i-toBS}^p}_{\text{transmite posição e energia para a Sink}} + \underbrace{k_{CP} E_{elec}}_{\text{recebendo resultado da clusterização}} + \underbrace{m_i (k_{CP} E_{elec} + E_{amp} k_{CP} d_{j-toCH_i}^p)}_{\text{transmite TDMA schedule para } m_i \text{ nós membros}} \quad (3.9)$$

$$E_{mem_i}(j) = k E_{elec} + E_{amp} k d_{j-toCH_i}^p \quad (3.10)$$

$$E_{mem_{CP_i}}(j) = \underbrace{k_{CP} E_{elec}}_{\text{recebendo requisição da Sink}} + \underbrace{k_{CP} E_{elec} + E_{amp} k_{CP} d_{j-toBS}^p}_{\text{transmite posição e energia para a Sink}} + \underbrace{k_{CP} E_{elec}}_{\text{recebendo resultado da clusterização}} + \underbrace{k_{CP} E_{elec} m_i}_{\text{recebendo TDMA schedule do } i^{th} \text{ CH}} \quad (3.11)$$

onde,

k = Quantidade de bits de um pacote de dados;

k_{CP} = Quantidade de bits de um pacote de controle;

E_{elec} = Energia, em Joules, necessária para transmitir um bit;

E_{da} = Energia, em Joules, necessária para realizar a agregação dos dados;

d_0 = Distância limite, calculada de acordo com os valores de E_{elec} e E_{amp} ;

d_{i-toBS} = Distância de um nó i para a Estação Base;

d_{j-toCH_i} = Distância de um nó j para o seu cluster-head i ;

E_{amp} = Energia de amplificação do sinal. Caso $d_{i-toBS} < d_0$ utiliza-se o modelo de espaço livre (ϵ_{fs}). Caso contrário, utiliza-se o modelo multipath (ϵ_{mp});

p = Caso $d_{i-toBS} < d_0$, $p = 2$. Caso contrário, $p = 4$;

m_i = Quantidade de membros do cluster-head i ;

A população inicial consiste de uma quantidade p_s de cromossomos CHs gerados de forma aleatória, onde o tamanho de cada cromossomo é igual a quantidade de nós vivos na

rede. O próximo passo consiste no processo de seleção que determina quais cromossomos da população corrente irão realizar o cruzamento. O método usado nesse processo foi o método da roleta. Como método de *crossover* foi usado o cruzamento de único ponto. Após a etapa de cruzamento, cada cromossomo da população é selecionado para a fase de mutação com uma probabilidade p_m . Caso o cromossomo seja selecionado para a mutação, ele terá um de seus genes escolhidos de forma aleatória, cujo valor será trocado de 0 para 1, ou vice-versa.

O último passo executado pelo *Algoritmo Genético* consiste em selecionar os p_s melhores cromossomos (valores mínimos de $F(X)$) da população corrente para formar a população da próxima geração. O algoritmo finaliza quando a função objetivo não muda por um certo número de gerações ou quando a quantidade de gerações excede um valor máximo definido como parâmetro do protocolo.

Quando a Estação Base encontra os melhores clusters, ela envia o resultado para todos os nós da rede. Cada sensor verifica se ele será um líder ou um nó membro do cluster. Uma vez que essa configuração é definida, os nós não líderes enviam os seus dados coletados para o seu respectivo cluster-head que por sua vez envia direto para a Estação Base. Essa comunicação permanece até o tempo da rodada expirar. Quando esse tempo termina, a fase de definição dos clusters e de coleta de dados se inicia novamente.

3.1.5 Síntese dos Protocolos Leach, Leach-C, Bee-C e GAEEP

Esta subseção apresenta algumas limitações e características dos protocolos apresentados na seção 3.1. Como limitações, pode-se dizer que nenhum deles implementam uma comunicação multihop entre os clusters-head, ou seja, estes transmitem diretamente para a Estação Base. Esse fato influencia no consumo, visto que os CHs enviam os dados a uma distância maior. Outra limitação está no fato de que eles não possuem nenhum método ou tratamento para evitar a morte de um cluster-head antes que o tempo de uma rodada expire.

Algumas características de cada protocolo são resumidas no quadro 4. Como pode ser visto, o único com operação distribuída é o protocolo Leach. Todos realizam uma agregação em cima dos dados, ou seja, os líderes de cada grupo efetuam uma fusão/agregação das informações transmitidas pelos sensores membros do cluster.

Quadro 4 – Síntese dos Protocolos Leach, Leach-C, Bee-C e GAEEP.

Protocolos	Modos de Operação					
	Comunicação		Distribuído	Centralizado	Clusterizado	Agregação-dados
	Single-Hop	Multi-hop				
Leach	X		X		X	X
Leach-C	X			X	X	X
Bee-C	X			X	X	X
GAEEP	X			X	X	X

Fonte: Elaborado pelo autor

3.2 PROTOCOLOS QUE TRABALHAM COM FLUXO EM REDES

3.2.1 Protocolo MCCR: Roteamento eficiente em energia com restrições de capacidade

Os autores desse trabalho propuseram um protocolo de roteamento com restrições de capacidade (MCCR - *Minimum-cost Capacity-constrained Routing*) que minimiza o total de energia consumida no roteamento enquanto garante que a carga total de cada nó sensor em enlaces sem fio, não exceda sua capacidade (PATEL et al., 2007). Esse protocolo é derivado de algoritmos de fluxo de custo mínimo que possuem complexidade polinomial.

A rede de sensores é modelada como um grafo direcionado $G(V,A)$ onde V é representado por $V = V_s \cup V_b$, sendo V_s o conjunto de nós sensores e V_b o conjunto de estações base. O conjunto A representa todas as arestas direcionadas (i, j) , onde $i \in V$ e $j \in V$. Cada aresta direcionada (i, j) possui um custo (C_{ij}) associado por pacote e uma capacidade U_{ij} que representa o número de pacotes por unidade de tempo. A variável D_i representa a taxa em que os pacotes são gerados no nó sensor i por unidade de tempo.

O fluxo de pacotes do nó i para seu vizinho j sobre enlaces sem fio (i, j) é representado por f_{ij} . Assim, o objetivo é entregar todos os pacotes de dados gerados pelos sensores para a Estação Base sempre com um custo baixo e sem exceder as capacidades dos enlaces. O problema pode ser formalizado da seguinte forma:

$$Obj = Min \sum_{(i,j) \in A} C_{ij} f_{ij} \quad (3.12)$$

Sujeito a:

$$\sum_{j:(i,j) \in A} f_{ij} - \sum_{j:(j,i) \in A} f_{ji} = D_i \quad \forall i \in V_s, \quad (3.13)$$

$$\sum_{k:k \in V_b} \left(\sum_{j:(k,j) \in A} f_{kj} - \sum_{j:(j,k) \in A} f_{jk} \right) = - \sum_{i:i \in V_s} D_i, \quad (3.14)$$

$$0 \leq f_{ij} \leq U_{ij} \quad \forall (i,j) \in A, \quad (3.15)$$

$$f_{ij} \in \mathbb{Z}^+ \quad \forall (i,j) \in A. \quad (3.16)$$

O primeiro conjunto de restrições (3.13) garante a conservação do fluxo em cada nó sensor. A segunda restrição (3.14) assegura que as estações base recebam todos os pacotes gerados por todos os sensores. O fluxo de pacotes no *link* não deve exceder sua capacidade e isto é assegurado pela terceira restrição (3.15). A quarta restrição (3.16) garante que os valores dos fluxos de pacotes sejam inteiros.

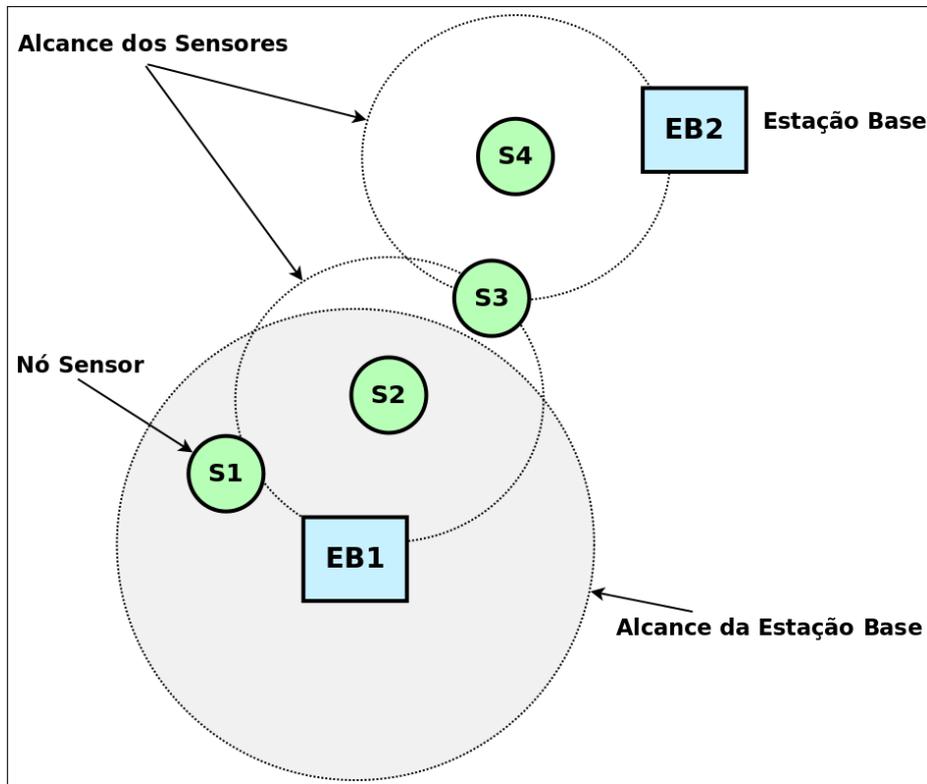
A Figura 8 mostra um exemplo de uma rede de sensores que é considerada nesse trabalho. Os raios de alcance de todos os nós são considerados iguais. As Estações Base possuem um alcance de transmissão maior do que o dos sensores.

O problema definido pelas equações 3.12 3.13 3.14 3.15 3.16 é semelhante ao problema de fluxo de custo mínimo (AHUJA et al., 1993). Assim, os autores desse trabalho modelaram a rede como um problema de circulação de custo mínimo realizando os seguintes passos:

1. Adicionar uma fonte x e um sorvedouro y ao grafo $G(V,A)$;
2. Adicionar *links* direcionais (x,i) , conectando o nó x a um sensor i , $\forall i \in V_s$. Configurar os custos desses *links* para 0 e as capacidades para D_i ;
3. Adicionar *links* direcionais (j,y) , conectando a estação base j ao sorvedouro y , $\forall j \in V_b$. Configurar os custos desses *links* para 0 e as capacidades para infinito;
4. Adicionar um *link* direcionado (y,x) , conectando o sorvedouro y à fonte x . Configurar o custo do *link* (y,x) para $-|V|C_{max}$ e a capacidade para infinito;
5. O grafo modificado fica definido como $G'(V \cup \{x,y\}, A \cup A')$, onde $A' = \{(x,i) : i \in V_s\} \cup \{(j,y) : j \in V_b\} \cup \{(y,x)\}$.

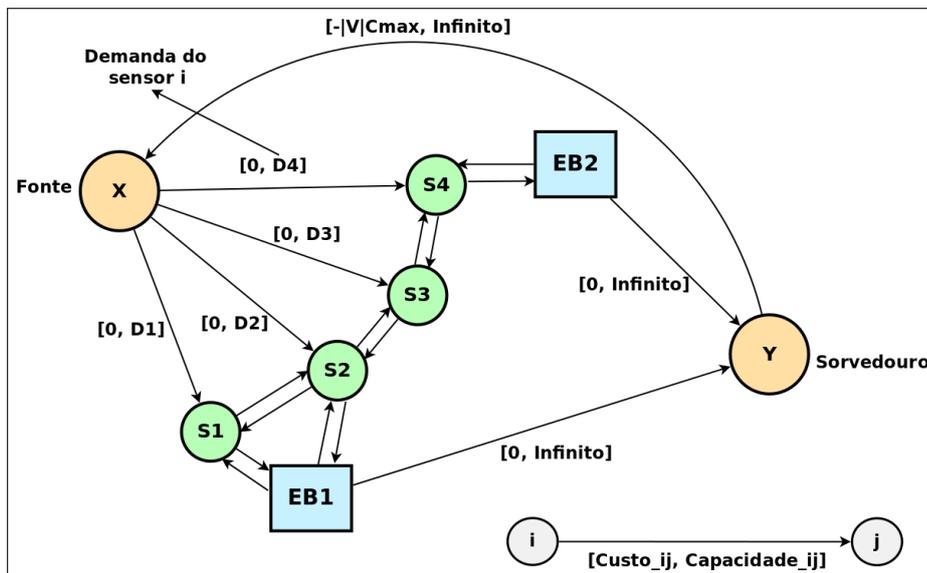
A Figura 9 apresenta a modelagem final da rede, depois da aplicação desses passos na rede de exemplo definida na Figura 8.

Figura 8 – Exemplo de uma Rede de Sensores no protocolo MCCR.



Fonte: Elaborado pelo autor

Figura 9 – Modelagem Final da Rede em um problema de circulação de custo mínimo.



Fonte: Elaborado pelo autor

O problema de circulação de custo mínimo pode ser resolvido de forma eficiente por algoritmos de fluxo de custo mínimo encontrados na literatura (GOLDBERG; TARJAN, 1987) (AHUJA et al., 1992) (ORLIN, 1993). Uma vantagem desses algoritmos é a integralidade dos

fluxos. Uma vez que as capacidades dos *links* e as taxas de dados dos nós são valores inteiros, os algoritmos de fluxo de custo mínimo podem encontrar caminhos com valores de fluxos inteiros. Assim, de posse dessa modelagem, os autores aplicaram um algoritmo de fluxo de custo mínimo e encontraram uma solução para a rede. Esta solução representa uma forma eficiente, com custo mínimo, de entregar as demandas de cada nó sensor para as Estações Base.

3.2.2 SAMF: Algoritmo de roteamento de fluxo máximo auto-adaptativo

Esse trabalho apresenta uma estratégia de roteamento de fluxo máximo auto-adaptativo (SAMF - *Self-adapting maximum flow*) para redes de sensores autônomas (BOGLIOLO et al., 2011). A proposta dos autores consiste em um algoritmo que é capaz de rotear uma carga útil máxima sob limitações de energia, largura de banda e outros recursos, enquanto se adapta de forma automática às variações do ambiente.

Algumas contribuições podem ser elencadas com esse trabalho. A primeira delas indica uma modelagem generalizada de uma rede de fluxo para representar RSSF autônomas, de forma que restrições como largura de banda, processamento e energia possam ser modeladas como restrições de capacidade. A segunda apresenta a vantagem de uma estratégia de roteamento auto-adaptativa capaz de suportar uma máxima carga útil de dados coletados. A terceira contribuição representa uma implementação válida da estratégia de roteamento. A última contribuição dos autores se refere às discussões levantadas com os resultados das simulações que avaliam a escalabilidade e a aplicabilidade da proposta.

Os autores modelaram uma RSSF autônoma como um grafo direcionado. Os sensores representam os vértices (indicados por v) do grafo e as arestas (indicadas por e) representam os *links* entre eles. Dois vértices quaisquer v_i e v_j são conectados por uma aresta e_{ij} , se e somente se, existe uma conexão entre os nós i e j .

Cada nó possui duas variáveis: $P(v)$, que representa o poder disponível no nó e $CPU(v)$, que representa seu poder computacional expresso como o número de pacotes que podem ser processados em uma unidade de tempo. Cada aresta é representada pelas variáveis $C(e)$, que representa a capacidade ou largura de banda do *link* e $E(v, e)$, que representa a energia requerida no sensor v para processar um dado e encaminhar para uma aresta e .

A quantidade máxima de pacotes que podem ser enviados por um nó v através de e é representada por $cap(e)$, sendo limitada por sua largura de banda $C(e)$, pelo poder de processamento $CPU(v)$ e pela taxa obtida pela razão do poder disponível e a energia gasta para

processar e enviar um pacote ($P(v)/E(v, e)$). $F(e)$ representa o fluxo de pacotes que passam pela aresta e . A equação 3.17 mostra a capacidade máxima de pacotes.

$$F(e) \leq cap(e) = \min\{C(e), CPU(v), \frac{P(v)}{E(v, e)}\} \quad (3.17)$$

Cada nó da rede possui arestas de saída, com as quais ele distribui sua energia e sua força computacional. Assim, um fluxo $F(e)$ de pacotes que sai pelas arestas de um nó qualquer (v) deve respeitar as seguintes restrições:

$$\sum_{e=1}^{|EdgesOut| \text{ of node } v} F(e) \leq CPU(v) \quad (3.18)$$

$$\sum_{e=1}^{|EdgesOut| \text{ of node } v} F(e)E(v, e) \leq P(v) \quad (3.19)$$

onde,

$|EdgesOut|$ of node v = Quantidade de arestas que saem de um nó v .

O total de pacotes que podem ser processados por um nó v é representado por $cap(v)$, sendo limitado por seu poder computacional e pela razão do poder do nó $P(v)$ e sua energia $E(v)$. A equação 3.20 mostra esse total.

$$cap(v) = \min\{CPU(v), \frac{P(v)}{E(v)}\} \quad (3.20)$$

As equações 3.17 e 3.20 desmonstram que limitações como poder, capacidade de processamento e largura de banda podem ser expressas como restrições de capacidades para vértices (sensores) e arestas (*links*). Modelando-se dessa forma a rede é transformada em uma rede de fluxo com nós e arestas limitadas. Assim, a carga máxima que pode ser sustentada pela rede pode ser determinada achando-se o fluxo máximo na rede de fluxo correspondente.

O trabalho define 4 tipos de nós sensores: *sensores comuns*, que monitoram o ambiente, *sinks* que geram pacotes de controle e pacotes de dados, *routers* que repassam pacotes de acordo com um algoritmo de roteamento e *sensor-routers* que se comportam como sensores comuns e *routers*. Os autores também propuseram um algoritmo de roteamento que implementa uma estratégia gulosa que segue o seguinte critério: sempre encaminhar pacotes para a *sink* através do caminho com a maior capacidade residual.

3.2.3 Síntese dos Protocolos MCCR e SAMF

Esta subseção apresenta algumas limitações e características dos protocolos apresentados na seção 3.2. Como limitações, pode-se dizer que ambos os protocolos, MCCR e o SAMF, não realizaram simulações com outros protocolos da literatura. As simulações realizadas são somente entre variações de parâmetros de cada protocolo. Outro aspecto ruim é que ambos os trabalhos não deixam claro se eles são protocolos distribuídos ou centralizados. Não é possível definir se são delegadas algumas funções para os nós sensores ou se todos os cálculos são realizados pela Estação Base.

Uma característica comum entre esses dois protocolos é que eles não trabalham com clusterização. Isto é, eles não escolhem líderes para determinados grupos de nós e a operação da rede não é dividida em rodadas. Outras características de cada protocolo são resumidas no quadro 5. Como pode ser visto, o MCCR encontra uma solução para a rede utilizando algoritmos de fluxo de custo mínimo. Já o SAMF, busca o fluxo máximo que pode ser gerado pelos sensores. O objetivo é entregar esse fluxo até a Estação Base. O SAMF determina funções diferentes para determinados sensores, sendo assim um protocolo hierárquico. No MCCR, cada sensor se comporta de maneira igual, não contendo nenhum nível de hierarquia.

Quadro 5 – Síntese dos Protocolos MCCR e SAMF.

Protocolos	Modos de Operação					
	Comunicação		Clusterizado	Hierárquico	Fluxo Máximo	Fluxo Custo Mínimo
	Single-Hop	Multi-hop				
MCCR		X				X
SAMF		X		X	X	

Fonte: Elaborado pelo autor

4 ABORDAGEM USANDO FLUXO EM REDES

Esta seção apresenta a proposta do trabalho e enfatiza seus objetivos. É apresentada também a metodologia que será seguida para alcançar as finalidades do trabalho.

4.1 PROPOSTA

O foco deste trabalho consiste em estabelecer uma abordagem que utiliza *fluxo em redes* para melhorar protocolos de roteamento que utilizam a técnica de clusterização. O objetivo é *acoplar* os processos (passos) da abordagem aos protocolos que já existem e assim alcançar um melhoramento de suas operações. Como parte desse melhoramento, procura-se evitar problemas que podem surgir quando se utiliza a clusterização (explicados mais à frente). Outros refinamentos da abordagem visam o aumento do tempo de vida da rede, um balanceado consumo de energia entre os nós e uma melhor distribuição das transmissões de dados.

Antes de prosseguir com o detalhamento da proposta, é necessário levantar algumas dificuldades que podem surgir com a técnica de clusterização. Uma delas é que os cluster-heads podem se comunicar entre si, ou seja, eles podem enviar seus dados coletados diretamente para a Estação Base como também para outro cluster-head realizando assim uma comunicação multihop. Esse fato pode acarretar a sobrecarga de algum cluster-head, visto que este terá que repassar as informações de seu cluster e as informações de outro líder, consumindo assim mais energia.

Outra consideração é que um cluster-head poderá consumir toda a sua energia, enquanto que o tempo de rodada ainda está sendo contado. Esse fato ocasionaria perda de dados na área em que tal cluster-head abrange.

Assim, é preciso um método que possa evitar as situações apresentadas acima e ao mesmo tempo atingir uma melhor operação do protocolo. Com tais argumentos levantados, podem ser elencados os seguintes questionamentos:

1. Como evitar que um nó seja sobrecarregado com dados de outros, consumindo mais do que possui?
2. Existe uma forma melhor de transmitir os dados de um nó e ao mesmo tempo balancear o consumo de energia?
3. Como indicar um percentual de energia máximo que um cluster-head pode consumir para engatilhar o processo de clusterização?

Para responder a essas perguntas e alcançar uma possível solução para os problemas,

será utilizado o conceito de *fluxo em redes*, abordando-se o problema do fluxo máximo. Porém, por que fluxo em redes? Como em uma rede de fluxos as arestas representam capacidades, então pode-se evitar a questão da sobrecarga. Se a capacidade da aresta representar a energia de um nó, é garantido que ele não transmitirá mais do que sua energia/capacidade evitando consumir mais do que possui. Assim, pode-se responder a primeira questão.

Um algoritmo de fluxo em redes tenta distribuir um fluxo de forma que ele passe por caminhos (arestas) livres até o destino não excedendo a capacidade de nenhuma aresta. Logo, utilizando-se essa característica, as comunicações entre clusters-head podem ser distribuídas respeitando-se as capacidades de cada um, evitando que muitas transmissões sejam enviadas a somente um nó. Esse fato também contribui para que ocorra um balanceamento de energia entre os sensores. Dessa forma, é possível responder a segunda questão.

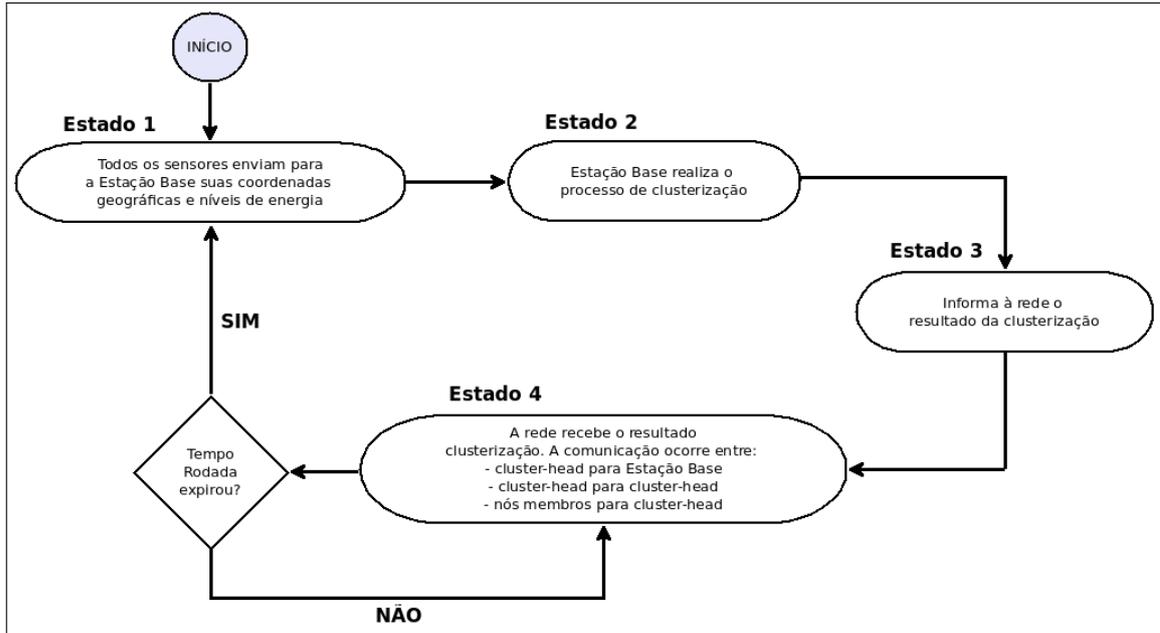
A quantidade máxima de energia que um cluster-head irá consumir para engatilhar uma nova clusterização pode ser visualizada como um valor máximo de energia que ele pode gastar sem sobrecarregar outro cluster-head da rede com seus dados transmitidos. Outro fato a considerar é que o fluxo máximo, em uma rede de fluxos, indica o maior valor que pode sair de uma fonte e chegar totalmente a um destino. Assim, se a quantidade de energia que um cluster-head pode consumir representar fluxos que podem sair dele para outro nó, então o fluxo máximo irá indicar quanto de energia cada cluster-head pode gastar. Logo, a solução de um algoritmo de fluxo máximo pode responder a terceira questão.

Portanto, as respostas para as três perguntas apresentadas indicam que a abordagem por fluxo em redes (fluxo máximo) pode ser utilizada para alcançar os objetivos do trabalho. Agora, onde essa abordagem se encaixaria nos protocolos que queiram fazer uso dela? É possível perceber que vários protocolos de roteamento centralizados e baseados em clusterização seguem determinados passos (explicados mais à frente). Alguns protocolos que seguem esses passos podem ser verificados em (HEINZELMAN et al., 2002) (MAIA et al., 2004) (MURUGANATHAN et al., 2005) (SINGH et al., 2010b) (REGO et al., 2012) (AMMARI; DAS, 2012) (ABO-ZAHHAD et al., 2014).

Inicialmente cada nó da rede envia sua energia e coordenadas geográficas para a Estação Base. Ela realiza o processo de clusterização e informa aos nós quem é cluster-head e quem é nó-fonte. Após isso, ocorre a fase de coleta de dados, em que os nós-fontes enviam seus dados para o cluster-head que por sua vez repassa para outro cluster-head ou direto para a Estação Base. Essa última operação ocorre dentro de um período fixo de rodada que uma vez

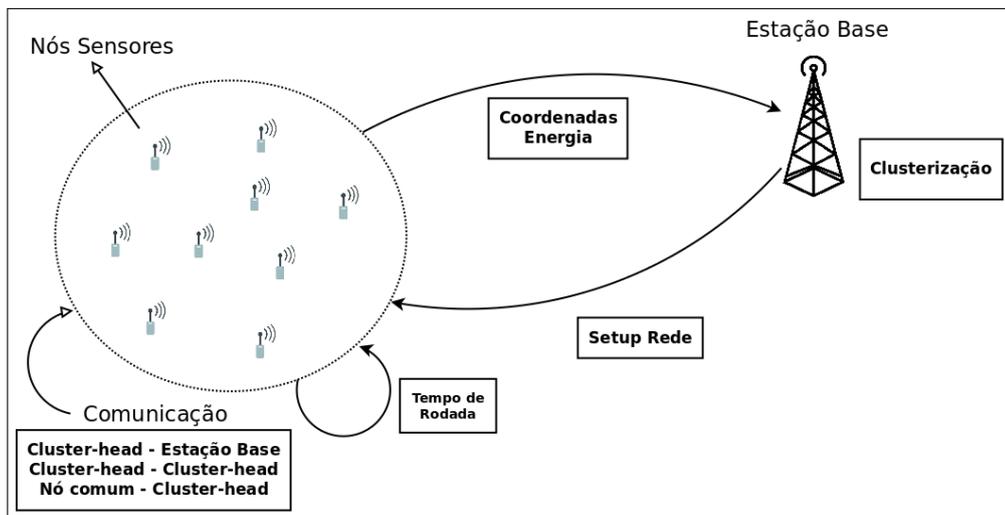
atingido, engatilha novamente o processo de clusterização. A figura 10 apresenta um diagrama de estados que representa o funcionamento de protocolos dessa categoria. A Figura 11 reforça, em um aspecto de rede, o comportamento descrito pelo diagrama da Figura 10.

Figura 10 – Diagrama de Estados de um Protocolo de Roteamento Centralizado baseado em Clusterização.



Fonte: Elaborado pelo autor

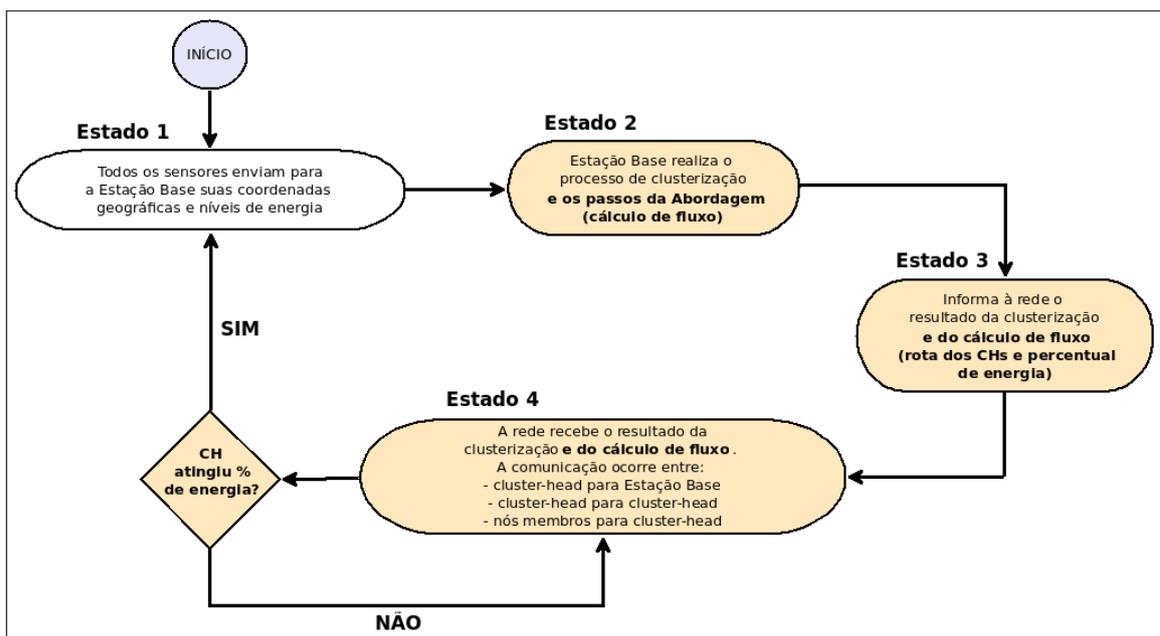
Figura 11 – Operação na Rede de um Protocolo de Roteamento Centralizado baseado em Clusterização.



Fonte: Elaborado pelo autor

A Figura 12 apresenta um diagrama de estados que representa as operações do protocolo utilizando a abordagem por fluxo. Os estados com a cor laranja indicam adições feitas pela abordagem. A Figura 13 reforça, em um aspecto de rede, o comportamento descrito pelo diagrama da Figura 12. Nota-se que nenhum passo é alterado, apenas alguns passos da abordagem são adicionados. O único item retirado é o tempo fixo de rodada, sendo agora a clusterização engatilhada pelo percentual de energia de algum cluster-head.

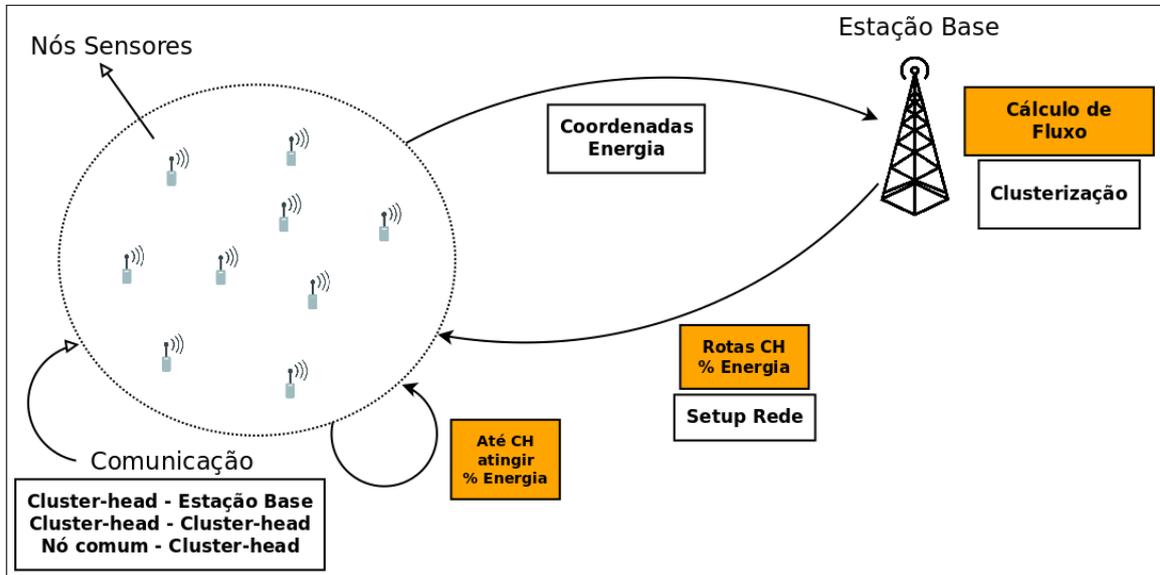
Figura 12 – Diagrama de Estados de um Protocolo de Roteamento Centralizado baseado em Clusterização com a Abordagem proposta.



Fonte: Elaborado pelo autor

Assim, o trabalho busca uma modelagem utilizando fluxo em redes (fluxo máximo). Pode-se definir a abordagem proposta como um conjunto de passos, onde cada passo representa uma modelagem. Esses passos são realizados de forma centralizada, ou seja, a Estação Base efetua todos os cálculos que serão apresentados mais a frente. Quando uma solução para a rede é encontrada, a Estação Base informa esse resultado para os nós. As próximas seções mostram os seguintes passos para a criação da abordagem: modelagem da rede, modelagem das capacidades, adaptação do grafo para fluxo em redes e cálculo do fluxo máximo (solução para o problema).

Figura 13 – Operação na Rede de um Protocolo de Roteamento Centralizado baseado em Clusterização com a Abordagem proposta.



Fonte: Elaborado pelo autor

4.1.1 Modelagem da rede

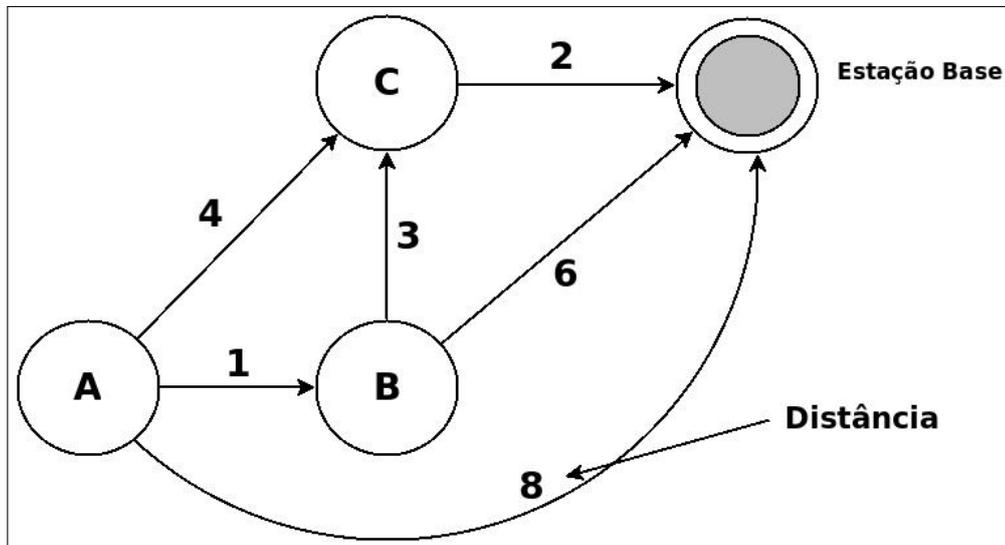
Esse passo consiste em criar um grafo direcionado com os nós clusters-head escolhidos no processo de clusterização. O grafo da Figura 14 representa um exemplo de rede de sensores já clusterizada. Esse exemplo será utilizado durante a explicação deste trabalho. Nele, os vértices A, B e C representam os clusters-head da rede. Os pesos das arestas representam as distâncias geográficas entre os nós.

Calcular o grafo direcionado da Figura 14 é uma tarefa simples. Basta verificar as distâncias dos nós em relação a Estação Base. Um nó A com menor distância da Estação Base em relação a um outro nó B, será o destino deste nó B, que possui maior distância em relação a Estação Base. Como em redes de sensores, o objetivo é encaminhar os dados para a Estação Base, então os dados devem fluir na direção dela. Logo as arestas ficam em direção a Estação Base. O algoritmo 1 apresenta o pseudocódigo utilizado para criar um grafo direcionado somente com os nós clusters-head (resultado da clusterização).

4.1.2 Modelagem das capacidades do grafo

Para criar um fluxo em redes do grafo, as arestas devem representar capacidades. A capacidade de uma aresta direcionada representa o total de energia. Assim, a capacidade de

Figura 14 – Grafo direcionado representando os clusters-head de uma RSSF.



Fonte: Elaborado pelo autor

Algoritmo 1: Criar Grafo Direcionado com os Clusters-Head.

Entrada: Id dos nós clusters-head (CH)

Saída: Grafo G direcionado com os clusters-head

início

Defina o grafo G considerando os clusters-head e a Sink (Estação Base);

Inicialize G ;

para cada CH_i **faça**

para cada CH_j **faça**

se ($Distância$ do CH_i para a Sink) > ($Distância$ do CH_j para a Sink) **então**

 Defina a aresta $a(CH_i, CH_j)$ em G ;

$a(CH_i, CH_j) := Distância$ do CH_i para o CH_j ;

fim

fim

fim

para cada CH **faça**

 Defina a aresta $a(CH, Sink)$ em G ;

$a(CH, Sink) := Distância$ do CH para a Sink;

fim

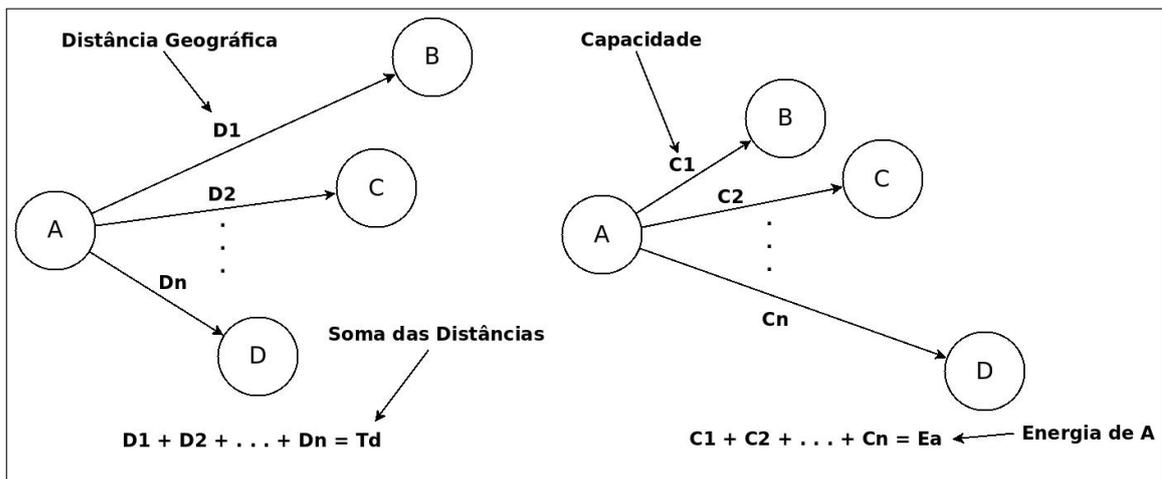
Retorne G

fim

uma aresta $A \rightarrow B$ representa a energia do nó A disponível para comunicação com o nó B. Isso representa o quanto que ele pode transmitir/consumir por aquela aresta. Para nós com mais de um vizinho, sua energia é dividida nas arestas entre esses vizinhos.

As capacidades do fluxo em redes serão modeladas da seguinte forma: arestas com *menores distâncias* geográficas entre os nós irão representar *maiores capacidades* no grafo de fluxo em redes e *maiores distâncias* irão indicar *menores capacidades*. Isso fará com que um fluxo de dados que passa por maiores capacidades, ou seja, passa por menores caminhos, economize energia, uma vez que menores distâncias implicam em menores gastos de energia. A Figura 15 exemplifica a modelagem das capacidades.

Figura 15 – Relação entre as Capacidades e as Distâncias.



Fonte: Elaborado pelo autor

Logo, é feito um cálculo que converte D_{ij} para um valor maior, caso esse valor seja menor e vice-versa. Porém, esse procedimento deve respeitar o modelo de consumo de um nó conforme apresentado em (HEINZELMAN et al., 2002). Nesse modelo, um sensor consome energia de acordo com as equações 4.1 e 4.2.

A transmissão de uma mensagem de k bits a uma distância d tem o seguinte consumo de energia:

$$E_{Tx}(k, d) = \begin{cases} E_{elec} k + \epsilon_{fs} k d^2, & \text{Se } d < d_0 \\ E_{elec} k + \epsilon_{mp} k d^4, & \text{Se } d \geq d_0 \end{cases} \quad (4.1)$$

e para receber uma mensagem, o sensor tem o seguinte consumo:

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} k, \quad (4.2)$$

onde:

$E_{Tx-elec}$ = Energia gasta na transmissão;

$E_{Rx-elec}$ = Energia gasta na recepção;

d = Distância entre dois sensores;

d_0 = Distância limite, calculada de acordo com os valores de E_{elec} , ϵ_{fs} e ϵ_{mp} ;

ϵ_{fs} = Parâmetro chamado de modelo de espaço livre (fs); é utilizado se a distância da origem ao destino for menor que d_0 ;

ϵ_{mp} = Parâmetro chamado de modelo multipath (mp); é utilizado se a distância da origem ao destino for maior ou igual a d_0 ;

E_{elec} = Energia gasta por bit transmitido ou recebido.

Verifica-se através desse modelo que o consumo é proporcional ao quadrado da distância entre dois sensores. Essa característica indica que o sinal de uma transmissão enfraquece na medida que a distância aumenta ao quadrado. Logo, o cálculo da distância deve considerar o seu valor ao quadrado para que a modelagem das capacidades seja coerente com o consumo de um sensor. Assim, dada uma distância D_{ij} , o seu valor convertido pode ser encontrado pela equação 4.3.

$$D_{ij} \rightarrow \left(\sum_{k=1}^n D_{ik}^2 - D_{ij}^2 \right) / (n - 1) \quad (4.3)$$

onde,

D_{ij} = Distância do nó i para o nó j ;

D_{ik} = Distância de um nó i para algum vizinho k ;

n = Quantidades de vizinhos de i .

Caso o valor encontrado tenha parte fracionária maior do que 0.5, pega-se o teto do valor. Caso contrário, pega-se o piso do valor.

A equação 4.3 pode ser encontrada pelas seguintes equações 4.4, 4.5 e 4.6.

$$D_{i1}^2 + D_{i2}^2 + D_{i3}^2 + \dots + D_{in}^2 = \sum_{k=1}^n D_{ik}^2 \quad (4.4)$$

A equação 4.4 considera que um nó qualquer i tenha n vizinhos, obtendo assim a soma das distâncias ao quadrado para cada um deles. Subtraindo o montante encontrado $\sum_{k=1}^n D_{ik}^2$

por cada parcela da soma D_{ij} ($j = 1, 2, \dots, n$), obtem-se o seguinte resultado conforme mostrado na equação 4.5. Desse resultado, pode-se verificar a seguinte relação indicada pela equação 4.6.

$$\begin{aligned} & \left(\sum_{k=1}^n D_{ik}^2 - D_{i1}^2 \right) + \left(\sum_{k=1}^n D_{ik}^2 - D_{i2}^2 \right) + \dots + \left(\sum_{k=1}^n D_{ik}^2 - D_{in}^2 \right) = \\ & n * \sum_{k=1}^n D_{ik}^2 - \underbrace{\left(D_{i1}^2 + D_{i2}^2 + D_{i3}^2 + \dots + D_{in}^2 \right)}_{\text{equação 4.4}} = \end{aligned} \quad (4.5)$$

$$n * \sum_{k=1}^n D_{ik}^2 - \sum_{k=1}^n D_{ik}^2 = (n-1) * \sum_{k=1}^n D_{ik}^2$$

$$\begin{aligned} & \left(\sum_{k=1}^n D_{ik}^2 - D_{i1}^2 \right) + \left(\sum_{k=1}^n D_{ik}^2 - D_{i2}^2 \right) + \dots + \left(\sum_{k=1}^n D_{ik}^2 - D_{in}^2 \right) = (n-1) * \sum_{k=1}^n D_{ik}^2 \\ & \frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{i1}^2 \right)}{(n-1)} + \frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{i2}^2 \right)}{(n-1)} + \dots + \frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{in}^2 \right)}{(n-1)} = \sum_{k=1}^n D_{ik}^2 \end{aligned} \quad (4.6)$$

A relação que se deseja mostrar entre as equações 4.4 e 4.6 está representada na equação 4.7:

$$\underbrace{D_{i1}^2 + D_{i2}^2 + \dots + D_{in}^2}_{\text{equação 4.4}} = \underbrace{\frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{i1}^2 \right)}{(n-1)} + \frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{i2}^2 \right)}{(n-1)} + \dots + \frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{in}^2 \right)}{(n-1)}}_{\text{equação 4.6}} = \sum_{k=1}^n D_{ik}^2 \quad (4.7)$$

Como pode ser visto na equação 4.7, a soma das parcelas da equação 4.6 gera o mesmo total de distâncias encontrado na equação 4.4 e essas parcelas representam um valor alternativo para cada distância D_{ij} . Ou seja, quanto menor D_{ij} , maior será o valor de $\left(\sum_{k=1}^n D_{ik}^2 - D_{ij}^2 \right) / (n-1)$ e vice-versa. Dessa forma, confirma-se o que é indicado na equação 4.3.

Assim, o cálculo da capacidade se resume a uma regra de 3 simples utilizando-se o valor encontrado pela equação 4.3. As equações 4.8 e 4.9 mostram como o valor da capacidade é encontrado.

$$\frac{C_{ij}}{E_i} = \frac{\left(\sum_{k=1}^n D_{ik}^2 - D_{ij}^2 \right) / (n-1)}{\sum_{k=1}^n D_{ik}^2} \quad (4.8)$$

$$C_{ij} = \frac{E_i}{\sum_{k=1}^n D_{ik}^2} * \frac{\sum_{k=1}^n D_{ik}^2 - D_{ij}^2}{(n-1)} \quad (4.9)$$

onde,

C_{ij} = Capacidade de uma aresta de um nó origem i a um nó destino j ;

D_{ik} = Distância de um nó i para algum vizinho k ;

D_{ij} = Distância do nó i ao nó j ;

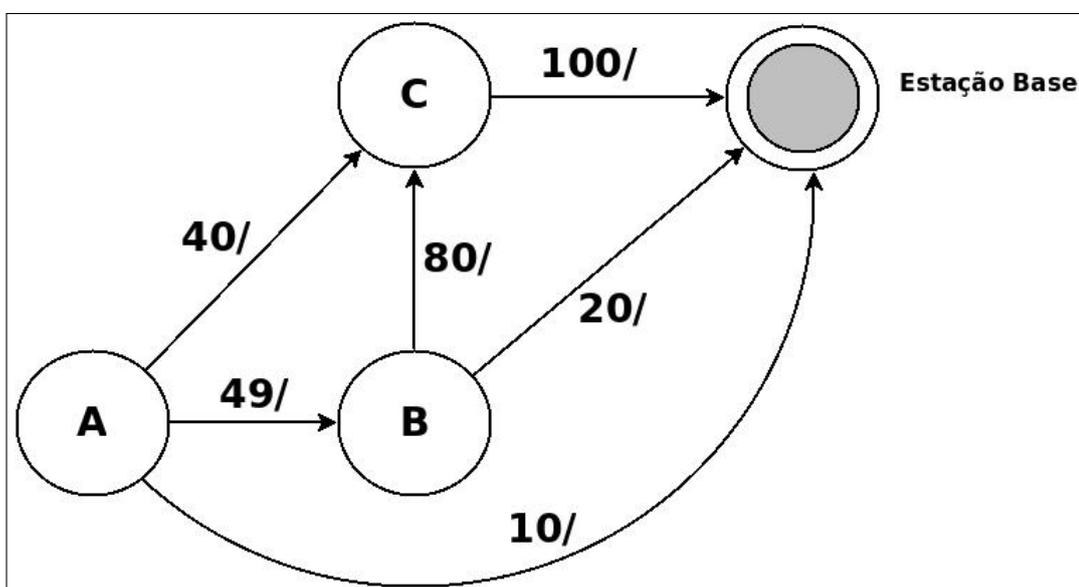
E_i = Energia Total do nó i ;

n = Quantidades de vizinhos de i .

O Algoritmo 2 apresenta o pseudocódigo que cria um grafo de capacidades a partir do gráfico direcionado de clusters-head.

O grafo da Figura 16 representa o resultado final do cálculo das capacidades proporcionais às distâncias. Considerou-se para esse exemplo que cada nó contém 100 Joules de energia. É possível verificar essa proporção comparando-se com o grafo original da Figura 14, que possui as distâncias. Arestas com distâncias menores representam arestas com maior capacidade no grafo final. Não é necessário efetuar o cálculo de capacidade para um nó que contém somente um vizinho. A capacidade para um nó com esta característica é configurada com a sua capacidade energética.

Figura 16 – Resultado do cálculo das capacidades a partir das distâncias.



Fonte: Elaborado pelo autor

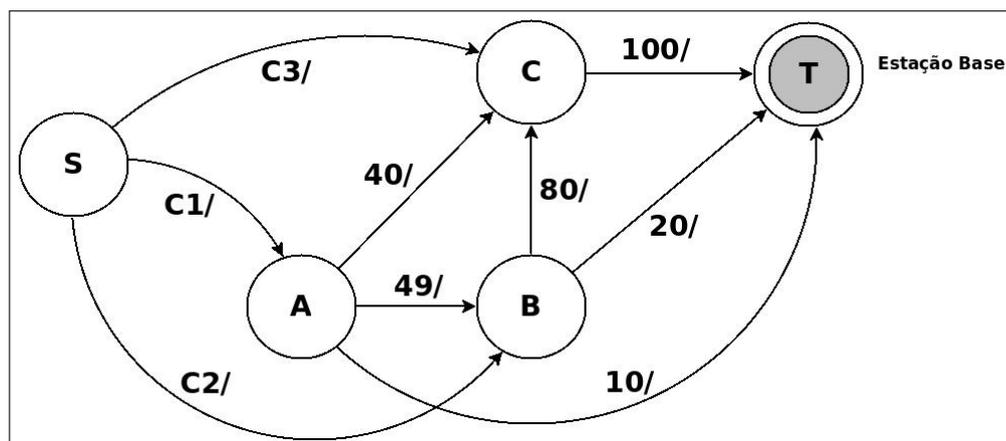
Algoritmo 2: Criar Grafo de Capacidades.**Entrada:** Grafo direcionado G de Clusters-head que contém as distâncias entres eles**Saída:** Grafo $G_{capacidades}$ **início**Defina o grafo $G_{capacidades}$ a partir de G ;Inicialize cada aresta $a(u, v) := 0$ em $G_{capacidades}$;Calcule a quantidade de vizinhos n_i de cada CH_i em G ;Calcule $\sum_{k=1}^n D_{ik}^2$; // D_{ik} representa a distância do CH_i aos seus vizinhos CH_k **para** cada CH_i em $G_{capacidades}$ **faça****para** cada CH_j em $G_{capacidades}$ **faça**Calcule a energia E_i do CH_i ;**se** CH_i tem mais de um vizinho ($n_i > 1$) **então**Defina a aresta $a(CH_i, CH_j)$ em $G_{capacidades}$;Calcule $C_{ij} = \frac{E_i}{\sum_{k=1}^n D_{ik}^2} * \frac{\sum_{k=1}^n D_{ik}^2 - D_{ij}^2}{(n_i - 1)}$; $a(CH_i, CH_j) := C_{ij}$;**fim****senão** $a(CH_i, CH_j) := E_i$;**fim****fim****fim**Retorne $G_{capacidades}$ **fim**

É possível notar que a capacidade energética de cada nó é distribuída entre os enlaces com seus vizinhos. A soma desses enlaces é igual a sua capacidade energética ($E_i = 100$ Joules), que foi definida anteriormente a nível de exemplo. Nota-se também, nesse exemplo, que o nó C só possui um vizinho e portanto, sua capacidade é a sua energia.

4.1.3 Adaptando o grafo para fluxo em redes

Para que o problema seja resolvido como um problema de fluxo máximo (fluxo em redes), o grafo precisa ter um nó fonte (s) e um nó destino (t). O nó fonte irá gerar uma quantidade de fluxo com o objetivo de que este fluxo chegue no destino. Logo, é necessário adaptar o grafo criando-se um nó fonte e destino. O vértice que representa a Estação Base será o nó destino. Porém, um novo nó será adicionado para representar a fonte acrescentando uma quantidade de energia em cada nó. Esta quantidade irá representar o quanto que cada nó pode consumir, e será um valor que tenta considerar o gasto de energia para repassar os dados que o nó recebeu de outros clusters-head. Como as capacidades representam energia, as arestas do nó origem para os demais nós da rede também irão representar energia. A Figura 17 ilustra o que foi exposto.

Figura 17 – Adaptação do grafo para fluxo em redes.



Fonte: Elaborado pelo autor

Como mencionado anteriormente, essas capacidades devem levar em conta se o nó está recebendo informações de outros nós. Assim, se um nó possui arestas de entrada, significa que ele pode receber dados para repassar e assim consumir energia. Logo, sua energia deve ser distribuída para transmitir dados que ele mesmo coletou e para dados que ele recebeu e deve repassar.

Portanto, as capacidades que saem da fonte para os nós representam as energias que cada um possui para transmitir dados que eles mesmos coletaram. Para achar essa quantidade de energia, foi dividida a capacidade energética de cada nó pelo seu grau no grafo de fluxo. O Algoritmo 3 apresenta o pseudocódigo para transformar o grafo de capacidades em um grafo de

fluxo em redes.

Algoritmo 3: Criar Grafo de Fluxo em Redes.

Entrada: Grafo $G_{capacidades}$

Saída: Grafo $G_{fluxo-em-redes}$

início

Defina o grafo $G_{fluxo-em-redes}$ igual ao grafo $G_{capacidades}$;

Adicione um vértice origem s em $G_{fluxo-em-redes}$;

Defina o vértice destino t como sendo a Estação Base em $G_{fluxo-em-redes}$;

para cada CH_i em $G_{fluxo-em-redes}$ **faça**

 | Calcule $grauEntrada_{CH_i}$;

fim

para cada aresta $a(s, CH_i)$ em $G_{fluxo-em-redes}$ **faça**

 | Calcule a energia E_i do CH_i ;

 | $a(s, CH_i) := E_i / grauEntrada_{CH_i}$;

fim

Retorne $G_{fluxo-em-redes}$;

fim

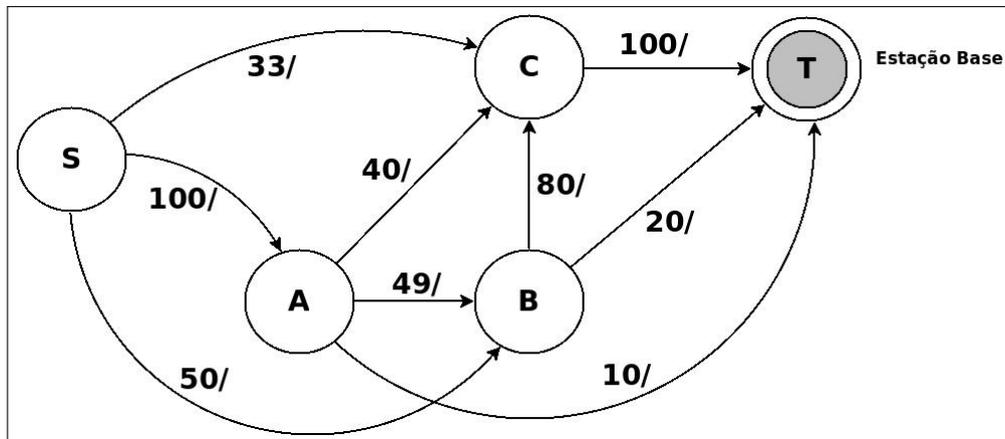
Assim, no exemplo da figura 17, as capacidades C1, C2 e C3 ficam com os seguintes valores $EA/1 = 100$, $EB/2 = 50$ e $EC/3 = 33$ respectivamente. Como o nó A não possui aresta de entrada, ele não terá que repassar dados de ninguém, logo sua energia pode ser usada completamente para coleta de dados. Já o nó B tem que considerar o que A está enviando para ele. Logo a energia de B é dividida, sendo 50% para repassar os dados de A e 50% para coletar seus próprios dados. A mesma ideia para o nó C, 33% de sua energia é para transmitir seus próprios dados e 66% (33% + 33%) para repassar o que A e B enviou para ele. A modelagem final do grafo para fluxo em redes é representado na Figura 18.

4.1.4 Solução para o problema

De posse desse fluxo em redes, pode-se executar algum algoritmo de fluxo máximo para encontrar o quanto de energia A, B e C podem consumir de forma que essa quantidade de energia não sobrecarregue nenhum nó e que, uma vez atingida, engatilha o processo de clusterização. A solução também indica por quais enlaces do nó os dados vão fluir.

Para encontrar o fluxo máximo e uma possível solução para a rede, este trabalho

Figura 18 – Resultado final da modelagem para fluxo em redes.



Fonte: Elaborado pelo autor

utilizou o algoritmo de Ford-Fulkerson (CORMEN, 2002). Esse algoritmo foi escolhido por ter implementação fácil e a sua complexidade não ocasiona impacto no desempenho da abordagem. O algoritmo 4 apresenta o pseudocódigo do algoritmo de Ford-Fulkerson utilizado neste trabalho.

Algoritmo 4: Algoritmo de Ford-Fulkerson.

Entrada:

G : Grafo do Fluxo em Redes

s : vértice origem

t : vértice destino

Saída: Fluxo Máximo f_{max}

início

Inicia $f(u, v) = 0$ para cada aresta (u, v) de G ;

Defina $G_{residual}$ a partir de G ;

enquanto existir caminho de aumento P de s para t em $G_{residual}$ **faça**

 Defina $C_p :=$ capacidade do arco de menor capacidade em P ;

para cada (u, v) em P **faça**

$f(u, v) := \text{fluxo}(u, v) + C_p$;

$f(v, u) := \text{fluxo}(v, u) - C_p$;

fim

$f_{max} := f_{max} + C_p$;

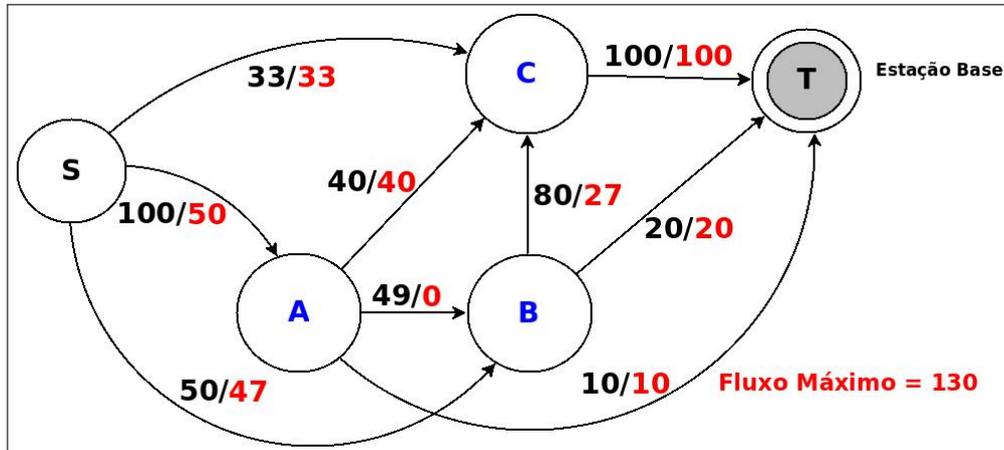
fim

Retorne f_{max}

fim

A Figura 19 mostra o fluxo máximo encontrado através do algoritmo de Ford-Fulkerson para o exemplo em questão.

Figura 19 – Fluxo Máximo encontrado.



Fonte: Elaborado pelo autor

Uma maneira alternativa que também pode ser escolhida para encontrar uma solução para esse fluxo em redes da Figura 18, é através da modelagem por otimização linear, conforme feito na subseção 2.2.1. As equações 4.10, 4.11 e 4.12 representam a modelagem por programação linear para o fluxo em redes definido na Figura 18. Cada variável x_{ij} da modelagem representa uma aresta do grafo com origem no vértice i e destino no vértice j . Para se encontrar uma solução para um modelo de programação linear, pode-se utilizar alguns programas *solvers* que utilizam algoritmos como, por exemplo, o *simplex* (KLEE; MINTY, 1970) para achar uma solução para o modelo. O *solver* utilizado para resolver o modelo dado pelas equações 4.10 e 4.11 foi o *glpsol* da biblioteca GLPK (MAKHORIN, 2008).

Maximizar:

$$x_{SA} + x_{SB} + x_{SC} \quad (4.10)$$

Sujeito a:

$$\begin{aligned}
 x_{SA} + x_{SB} + x_{SC} &\geq 0 \\
 -x_{SA} + x_{AB} + x_{AC} + x_{AT} &= 0 \\
 -x_{SB} - x_{AB} + x_{BC} + x_{BT} &= 0 \\
 -x_{SC} - x_{AC} - x_{BC} + x_{CT} &= 0 \\
 -x_{AT} - x_{BT} - x_{CT} &\leq 0
 \end{aligned} \tag{4.11}$$

$$\begin{aligned}
 0 \leq x_{SA} \leq 100; \quad 0 \leq x_{SB} \leq 50; \quad 0 \leq x_{SC} \leq 33; \\
 0 \leq x_{AB} \leq 49; \quad 0 \leq x_{AC} \leq 40; \quad 0 \leq x_{AT} \leq 10; \\
 0 \leq x_{BC} \leq 80; \quad 0 \leq x_{BT} \leq 20; \quad 0 \leq x_{CT} \leq 100;
 \end{aligned}$$

Solução:

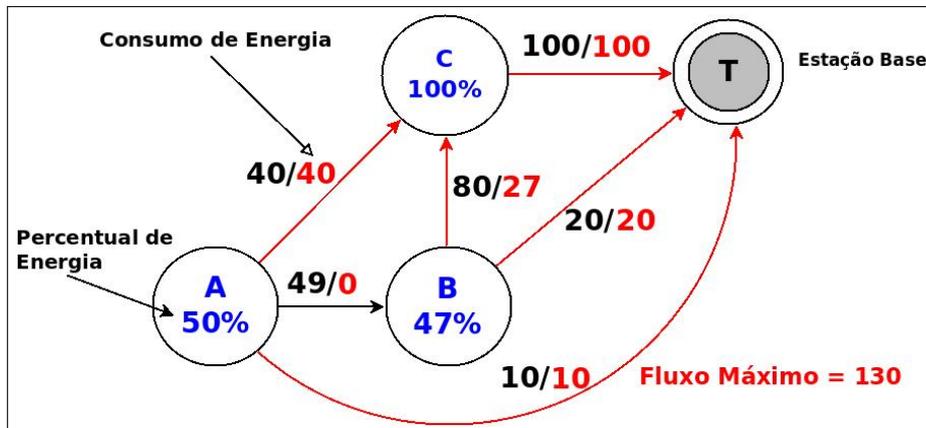
$$\begin{aligned}
 \text{Fluxo Máximo: } F &= x_{SA} + x_{SB} + x_{SC} = 130 \\
 \text{Fluxos: } x_{SA} &= 47; \quad x_{SB} = 50; \quad x_{SC} = 33; \\
 x_{AB} &= 0; \quad x_{AC} = 37; \quad x_{AT} = 10; \\
 x_{BC} &= 30; \quad x_{BT} = 20; \quad x_{CT} = 100;
 \end{aligned} \tag{4.12}$$

O grafo da Figura 20 apresenta uma solução para o problema. Nele é indicado o quanto que cada nó está gastando de energia por suas arestas e por onde eles devem transmitir os dados. A solução indica que o nó A deve encaminhar todos os seus dados através de todos os seus enlaces, porém dividindo a quantidade de dados que vão fluir em cada aresta. Mesma ideia para o nó B. Como o nó C só possui um enlace, ele transmite todos os seus dados por ele. Os nós A e B enviam dados para seus vizinhos como também diretamente para a Estação Base conforme calculado pelo fluxo. Isso representa um balanceamento no consumo geral dos nós, visto que, se o nó B enviasse todos os seus dados coletados (mais o agregado) para o nó C, sobrecarregaria tal nó fazendo com que ele gastasse mais energia. Para evitar isso, um nó consome um pouco mais para que outro nó não consuma muita energia.

O grafo solução também fornece o quanto que cada cluster-head pode consumir até ser engatilhada a clusterização. Basta contar o consumo que cada nó obteve em cada um de seus enlaces. No exemplo, a quantidade máxima que A, B e C podem consumir seria 50J, 47J e 100J respectivamente. Isso diz o seguinte, o nó que atingir algum desses valores primeiro, engatilhará o processo de clusterização.

Para o exemplo da Figura 20, foi considerado para o cálculo das capacidades a energia completa do nó ($E_i = 100$ Joules). Porém, nas simulações deste trabalho, o cálculo das

Figura 20 – Solução encontrada pela modelagem por fluxo em redes.



Fonte: Elaborado pelo autor

capacidades foi feito considerando 70% da energia corrente do nó. Isso foi feito para evitar que um nó consuma toda a sua energia dentro de uma rodada. Esse fato pode acontecer considerando um cenário de pior caso da Figura 20, onde nem o nó A e nem o nó B atingem os seus percentuais de energia antes que o nó C atinja o dele (100%). Nesse caso, o nó C engatilha a clusterização, mas morre dentro do período de uma rodada. Outro motivo para essa porcentagem foi evitar que as primeiras rodadas durem um período muito longo, visto que com uma energia total, a abordagem calcula um percentual de energia alto para engatilhar a clusterização.

5 SIMULAÇÕES E RESULTADOS

A proposta foi simulada utilizando-se um programa simulador de rede, com o objetivo de verificar a sua eficiência e validade. O simulador de rede foi o Castalia (BOULIS, 2009), que é um simulador de eventos discretos específico para redes de sensores sem fio, sendo uma extensão do simulador de redes Omnet++ (VARGA, 2007). O Castalia oferece modelagens realísticas do canal sem fio, gerenciamento da bateria e módulos de rádio de cada nó sensor.

A proposta foi incorporada nos seguintes protocolos da literatura que trabalham com clusterização: Leach-C (HEINZELMAN et al., 2002), BEE-C (REGO et al., 2012) e GAEEP (ABO-ZAHHAD et al., 2014). A finalidade é comparar os protocolos integrados com a abordagem proposta (MaxFlow) contra suas versões originais. O *link* para o *download* dos protocolos implementados para este trabalho, como também a implementação da abordagem, é o seguinte: <https://dl.dropboxusercontent.com/u/31794086/Codigos_Dissertacao.rar>

5.1 CENÁRIOS E PARÂMETROS DA REDE E DO SIMULADOR

O quadro 6 mostra os cenários utilizados para as simulações. As Figuras 21 e 22 apresentam a topologia e a densidade da rede em cada cenário.

Quadro 6 – Cenários

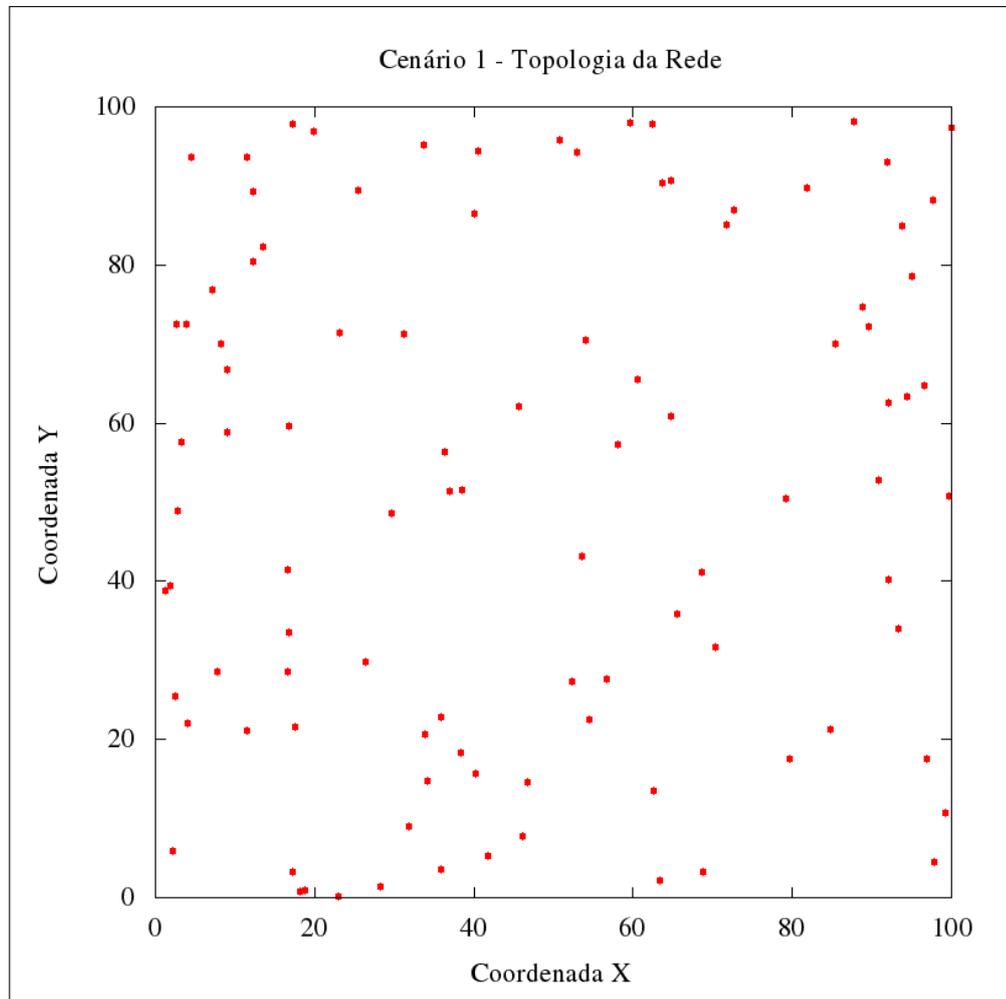
Cenários	Quantidades de nós	Área (m x m)	Posição da Estação Base
1	100	100 X 100	(0, 0)
2	300	500 X 500	(0, 0)

Fonte: Elaborado pelo autor

Cada cenário foi simulado um total de 33 vezes, tirando-se a média dos valores e atingindo-se um intervalo de confiança de 95%. Os quadros 7 e 8 apresentam os parâmetros da rede montada para os cenários e alguns parâmetros do simulador Castalia, respectivamente.

Os nós sensores foram depositados no ambiente de forma aleatória seguindo uma distribuição uniforme e não possuindo nenhum aspecto de mobilidade. Cada nó foi configurado com uma energia inicial de 10 Joules para que as simulações não demorassem muito tempo. Na camada MAC, foi escolhido o protocolo TMAC (BOULIS, 2011) por ser adequado em aplicações com transmissão contínua ou periódica de dados e também por que o simulador já oferece uma implementação do mesmo. Todos os protocolos foram simulados com o tempo total de execução de 600s, com a exceção do protocolo GAEEP com a abordagem (GAEEP-MaxFlow)

Figura 21 – Topologia e densidade do Cenário 1.



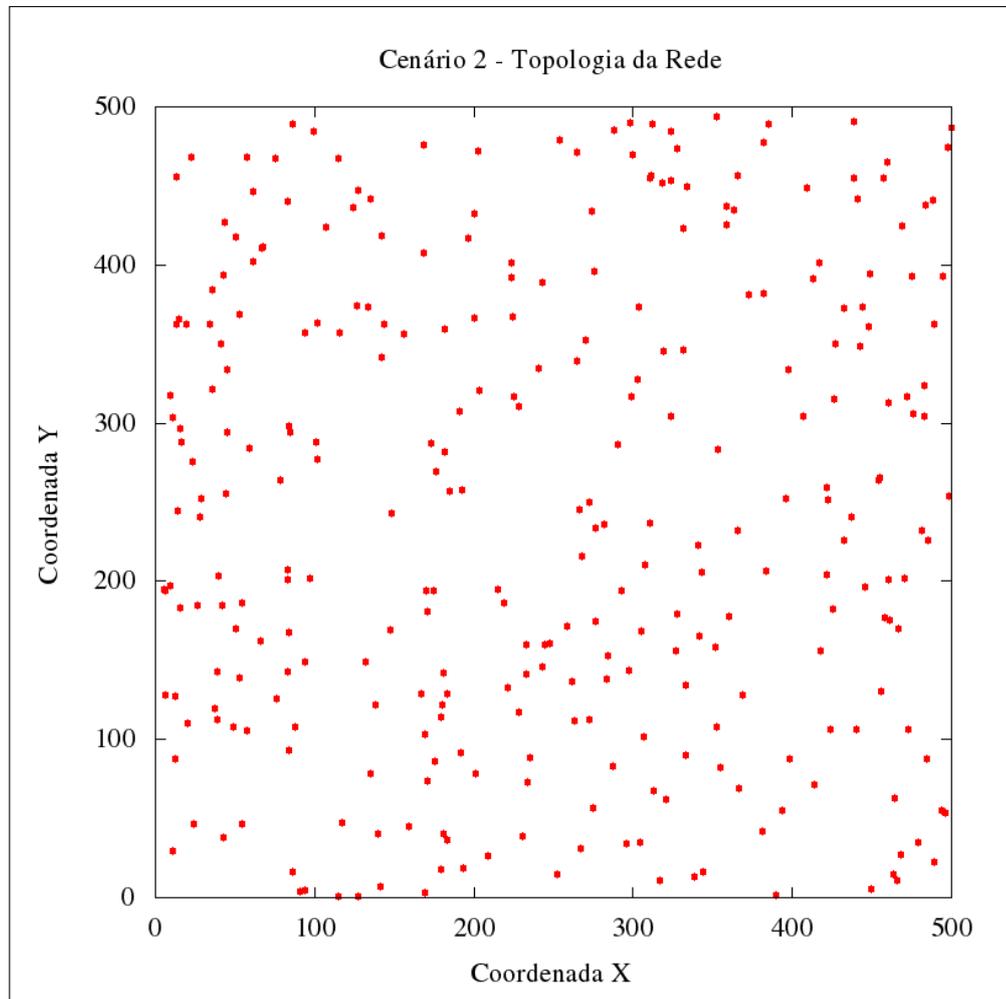
Fonte: Elaborado pelo autor

que precisou ser simulado com 700s para que todas as métricas pudessem ser coletadas.

O parâmetro do simulador *Sigma* foi configurado com o valor zero. O parâmetro *BidirectionalSigma* também foi configurado com valor zero. A configuração desses dois parâmetros com esse valor indica que os enlaces entre dois nós quaisquer apresentam a mesma força de sinal e são perfeitamente bidirecionais, ou seja, a qualidade de um *link* $A \rightarrow B$ é a mesma do *link* $B \rightarrow A$ (DAM; LANGENDOEN, 2003). O parâmetro *PathLossExponent* foi configurado com 1.6 para o primeiro Cenário. No segundo Cenário foi configurado o valor 1.1. Esse parâmetro indica quão rápido a força do sinal é enfraquecida.

O simulador também apresenta modelagens de módulos de rádios reais que podem ser utilizados na simulação carregando-se determinados arquivos. O arquivo CC2420.txt define um rádio real de mesmo nome desenvolvido pela empresa *Texas Instruments*. Nesse arquivo são definidas, por exemplo, as potências máxima e mínima que um sensor possui para transmitir

Figura 22 – Topologia e densidade do Cenário 2.



Fonte: Elaborado pelo autor

um dado. O parâmetro *CollisionModel* foi configurado com o valor 1. Esse parâmetro define o modelo de colisão usado pelo rádio para calcular a interferência de um sinal recebido. O valor 1 indica um modelo simplista para colisões, ou seja, se dois transmissores estão concorrentemente transmitindo e um receptor pode receber ambos os sinais, então sempre existirá uma colisão no receptor (DAM; LANGENDOEN, 2003).

Quadro 7 – Parâmetros da Rede.

Posição dos Nós	Energia Inicial	Protocolo Camada MAC	Tempo de Simulação	Mobilidade
Aleatória	10 Joules	TMAC	600s a 700s	Estático

Fonte: Elaborado pelo autor

Quadro 8 – Parâmetros do Simulador.

Sigma	BidirectionalSigma	PathLossExponent	RadioParametersFile	CollisionModel
0	0	1.1 a 1.6	"../Parameters/Radio/CC2420.txt"	1

Fonte: Elaborado pelo autor

5.2 PARÂMETROS DOS PROTOCOLOS SIMULADOS

Os protocolos utilizados nas simulações foram os seguintes: Leach-C (HEINZELMAN et al., 2002), BEE-C (REGO et al., 2012) e GAEEP (ABO-ZAHHAD et al., 2014). Todos esse protocolos trabalham com técnicas para alcançar a clusterização da rede. O leach-C utiliza o algoritmo de Têmpera Simulada (TS). O Bee-C implementa um algoritmo baseado no cruzamento de abelhas, enquanto que o GAEEP aplica Algoritmos Genéticos para selecionar os clusters.

Os quadros 9, 10 e 11 resumem os parâmetros utilizados em cada um desse protocolos, respectivamente. Os valores dos parâmetros *Tempo de rodada*, *Tempo de Slot* e *Tamanho do pacote de dados* foram configurados de acordo com uma implementação do protocolo Leach (HEINZELMAN et al., 2002) desenvolvida pelo Grupo de Pesquisa em Redes de Computadores e Comunicação Multimídia (GERCOM). A quantidade desejável de clusters para cada cenário foi calculada de acordo com a equação 5.1 que está definida em (HEINZELMAN et al., 2002). Considerando as energias de amplificação $E_{fs} = 10\text{pj}$ e $E_{mp} = 0.0013\text{pj}$, foram encontrados os valores aproximados de 5 clusters para o primeiro cenário e 10 clusters para o segundo cenário.

$$k = \frac{\sqrt{N}}{\sqrt{2\pi}} \sqrt{\frac{E_{fs}}{E_{mp}} \frac{M}{d_{toBS}}} \quad (5.1)$$

onde:

N = Número de nós da rede;

E_{fs} e E_{mp} = Energias de amplificação do rádio transmissor;

M = Dimensão de uma rede $M \times M$;

d_{toBS} = Distância do nó mais distante da Estação Base;

Os valores dos parâmetros *Tempo de Rodada*, *Tempo de Slot* e *Tamanho do pacote de dados* foram configurados os mesmos para todos os protocolos simulados. As simulações do Leach-C e do Bee-C apresentaram também a mesma quantidade de clusters-head em ambos os cenários. O número de clusters no GAEEP é definido de acordo com a melhor configuração de

nós clusters-head encontrada pelo Algoritmo Genético. Os demais parâmetros dos protocolos BEE-C e GAEEP foram configurados de acordo com os seus artigos-base.

Quadro 9 – Parâmetros do Protocolo Leach-C.

Tempo de Rodada	Tempo do Slot	Qt. de clusters	Tamanho do pacote de dados
30s	0.2s	5 a 10	250 bytes

Fonte: Elaborado pelo autor

Quadro 10 – Parâmetros do Protocolo Bee-C.

Parâmetros	Valor
Qt. inicial de Zangões	100
Tamanho da Espermateca da rainha	70
Qt. de zangões a cruzar com as Rainhas	100
Quantidade de Abelhas Rainhas	1
Taxa de Mutação	0.05
Taxa de Cruzamento	0.20
Máximo de Gerações	20
Tempo de Rodada	30s
Tempo do Slot	0.2s
Qt. de clusters	5 a 10
Tamanho do pacote de dados	250 bytes

Fonte: Elaborado pelo autor

Quadro 11 – Parâmetros do Protocolo GAEEP.

Parâmetros	Valor
Tamanho da População Inicial	50
Máximo de Gerações	150
Qt. de filhos gerados	20
Taxa de Mutação	0.07
Taxa de Cruzamento	0.7
Fator w	0.95
Tempo de Rodada	30s
Tempo do Slot	0.2s
Tamanho do pacote de dados	250 bytes

Fonte: Elaborado pelo autor

5.3 MÉTRICAS PARA AVALIAÇÃO

Os resultados foram comparados de acordo com as seguintes métricas:

- **Consumo médio de energia por nó:** Nesta métrica, é avaliado se a proposta proporciona economia de energia para os nós sensores. O objetivo é que a abordagem possa minimizar essa métrica.

- **Quantidade de dados recebidos pela Estação Base:** Esta métrica representa a quantidade de dados que foram corretamente entregues à Estação Base. Quanto maior essa quantidade de dados, melhor a avaliação dos eventos que acontecem no ambiente. Assim, é esperado que a proposta consiga maximizar essa métrica.
- **Tempo de morte do primeiro nó:** Esta métrica checa o tempo de cobertura da rede, ou seja, o tempo em que a rede trabalhou 100% na coleta de dados. Uma vez que um nó morre, a cobertura deixa de ser total. Dessa forma, pretende-se maximizar esse tempo de cobertura da rede.
- **Tempo de vida da rede (50% e 90% de nós mortos):** Com esta métrica, é possível ver se a proposta conseguiu estender o tempo de vida da rede. O propósito é maximizar essa métrica.

5.4 RESULTADOS

Esta seção apresenta os resultados das simulações nos dois cenários definidos. A abordagem proposta foi incorporada nos protocolos Leach-C, Bee-C e GAEEP e estes foram comparados contra as suas versões originais. Os protocolos com a abordagem foram chamados de Leach-C-MaxFlow, Bee-C-MaxFlow e GAEEP-MaxFlow, respectivamente. É importante fazer uma observação para o Cenário 1 com relação ao protocolo GAEEP. A sua versão original (GAEEP) e a versão com a abordagem (GAEEP-MaxFlow), ao contrário dos demais protocolos, foram executados com tempo de simulação de 700s para todas as métricas.

Esse aumento no tempo foi necessário para que todas as métricas pudessem ser coletadas no GAEEP-MaxFlow, principalmente o tempo de morte de 90% dos nós. Simulando o GAEEP-MaxFlow com 600s, em algumas simulações, a rede não continha 90% de nós mortos. Para deixar ambos os protocolos nas mesmas condições, o protocolo GAEEP (versão original) também foi simulado com 700s. Os demais protocolos (Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow) foram simulados com 600s no Cenário 1. Assim, os resultados obtidos nesse Cenário 1 foram divididos em dois gráficos para cada métrica. Um deles apresenta os resultados dos protocolos Leach-C e Bee-C (versão original e com a abordagem) com 600s de simulação. O outro gráfico mostra os resultados do protocolo GAEEP (versão original e com a abordagem) com 700s de simulação.

Esse fato do protocolo GAEEP, descrito acima, só acontece no Cenário 1. Isto é, no Cenário 2, todos os protocolos (Leach-C, Leach-C-MaxFlow, Bee-C, Bee-C-MaxFlow,

GAEEP e GAEEP-MaxFlow) foram simulados com 600s. As métricas e os parâmetros utilizados nas simulações foram mostrados nas seções anteriores. Todos os gráficos apresentados foram gerados pela ferramenta GNUPLOT (WILLIAMS; HECKING, 2003). As próximas subseções apresentam os resultados para cada métrica definida.

5.4.1 Consumo médio de energia por nó

A Figura 23 apresenta o consumo médio de energia dos protocolos Leach-C e Bee-C (versão original e versão com a abordagem) no Cenário 1. Nota-se que os protocolos com a abordagem proposta obtiveram menor consumo médio de energia em relação às suas versões originais. O ganho em economia de energia do Leach-C-MaxFlow em relação ao Leach-C foi de aproximadamente 0.25%. O Bee-C-MaxFlow obteve um ganho em relação ao Bee-C de aproximadamente 0.28%. A Figura 24 apresenta o consumo médio de energia dos protocolos GAEEP e GAEEP-MaxFlow no Cenário 1. O ganho em economia de energia do GAEEP-MaxFlow em relação ao GAEEP foi pequeno, sendo de 0.0006%. Porém, nas outras métricas, o GAEEP-MaxFlow obteve resultados melhores em relação a sua versão original.

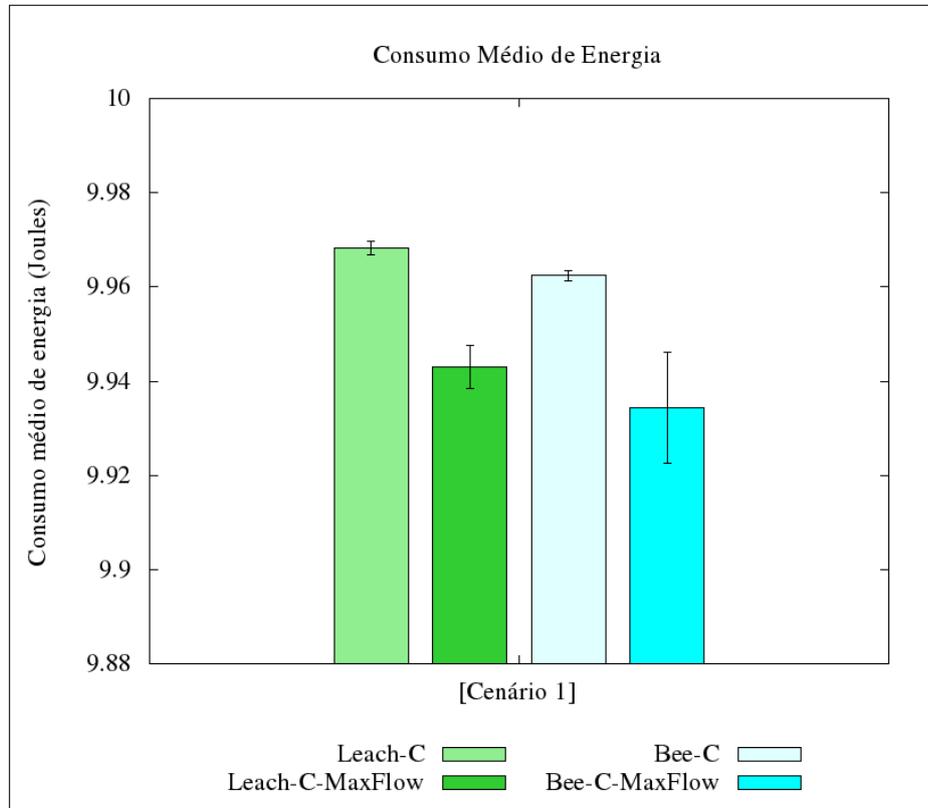
O ganho em economia alcançado se justifica pelo fato da abordagem integrar ao protocolo uma comunicação multihop, o que contribui para a economia. Outra questão é o fato dela proporcionar aos nós uma melhor utilização de sua energia, ou seja, cada sensor distribui de forma inteligente sua bateria entre seus nós vizinhos evitando sobrecarregar algum deles. Essa distribuição inteligente também evita que um nó transmita muitos dados para nós mais distantes.

A Figura 25 apresenta o consumo médio de energia de cada protocolo no Cenário 2. A quantidade de nós nesse cenário é bem maior do que no Cenário 1, o que ocasiona uma maior quantidade de clusters-head na rede consumindo um montante a mais de energia, visto que a quantidade de nós em um cluster também aumenta. Dessa forma, todos os protocolos consumiram quase que completamente toda a sua energia durante o período de tempo definido para as simulações.

A diferença de consumo entre o Leach-C-MaxFlow e o Leach-C foi de apenas 0.00055J. A diferença de energia consumida entre o Bee-C-MaxFlow e o Bee-C foi de 0.00561J. O GAEEP-MaxFlow e o GAEEP apresentaram uma diferença de 0.00804J. Porém, nas outras métricas, os protocolos com a abordagem, a exceção do GAEEP-MaxFlow, obtiveram resultados melhores do que suas versões originais no Cenário 2.

O quadro 12 apresenta a média do consumo de cada nó e os seus respectivos limites

Figura 23 – Consumo médio de energia do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.



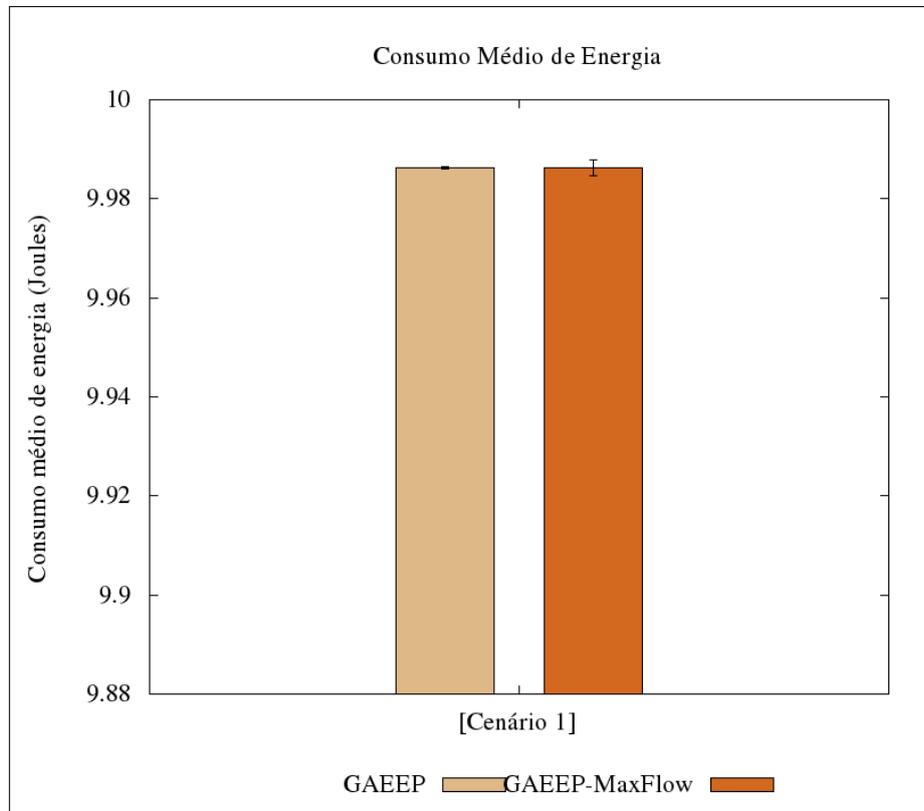
Fonte: Elaborado pelo autor

(inferior e superior), considerando o intervalo de confiança de 95%, para cada protocolo nos Cenários 1 e 2.

5.4.2 Quantidade de dados recebidos pela Estação Base

A Figura 26 apresenta a quantidade de dados recebidos pela Estação Base que os protocolos Leach-C e Bee-C (versão original e versão com a abordagem) obtiveram no Cenário 1. Verifica-se a partir dos gráficos que os protocolos com a abordagem apresentaram maior quantidade de dados entregues corretamente à Estação Base. O protocolo Leach-C-MaxFlow conseguiu um aumento de aproximadamente 26.84% em relação a sua versão original. O Bee-C-MaxFlow também obteve melhorias, conseguindo um aumento de 26.33% em relação ao Bee-C. A Figura 27 apresenta a quantidade de dados recebidos pela Estação Base dos protocolos GAEEP e GAEEP-MaxFlow no Cenário 1. A abordagem também trouxe melhorias para o protocolo GAEEP. A versão desse protocolo com a abordagem (GAEEP-MaxFlow) apresentou um aumento aproximado de 37.98% em relação a sua versão original.

Figura 24 – Consumo médio de energia do GAEEP e do GAEEP-MaxFlow no Cenário 1.



Fonte: Elaborado pelo autor

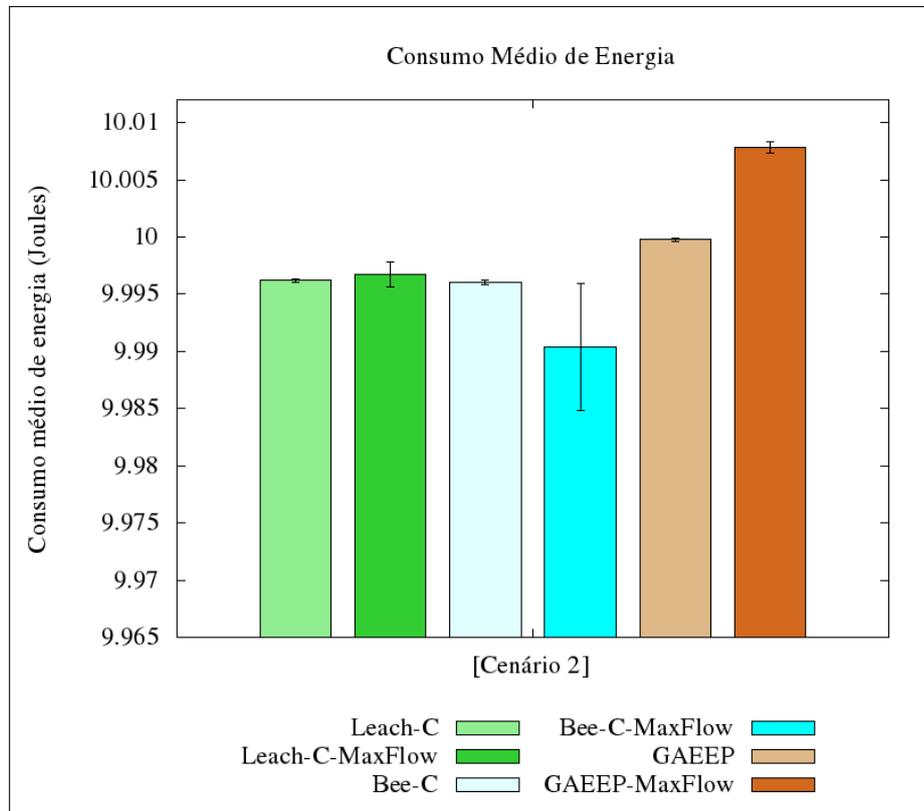
Quadro 12 – Resultados do consumo médio de energia em Joules de cada protocolo nos Cenários 1 e 2.

Cenário 1			
Protocolo	Consumo Médio	Limite Inferior	Limite Superior
Leach-C	9.96836	9.96691	9.96980
Leach-C-MaxFlow	9.94303	9.93850	9.94755
Bee-C	9.96239	9.96132	9.96345
Bee-C-MaxFlow	9.93439	9.92253	9.94624
GAEEP	9.98627	9.98611	9.98642
GAEEP-MaxFlow	9.98621	9.98460	9.98781
Cenário 2			
Protocolo	Consumo Médio	Limite Inferior	Limite Superior
Leach-C	9.99621	9.99606	9.99635
Leach-C-MaxFlow	9.99676	9.99568	9.99783
Bee-C	9.99603	9.99584	9.99621
Bee-C-MaxFlow	9.99042	9.98487	9.99596
GAEEP	9.99976	9.99961	9.99990
GAEEP-MaxFlow	10.0078	10,0073	10.0082

Fonte: Elaborado pelo autor

O aumento da quantidade de pacotes enviados é explicada por alguns fatores. Um deles consiste no fato de que os nós vivem por um período maior de tempo, devido à economia de energia que eles alcançam, e assim acabam por transmitir mais informações. Uma outra

Figura 25 – Consumo médio de energia de cada nó para o Cenário 2.



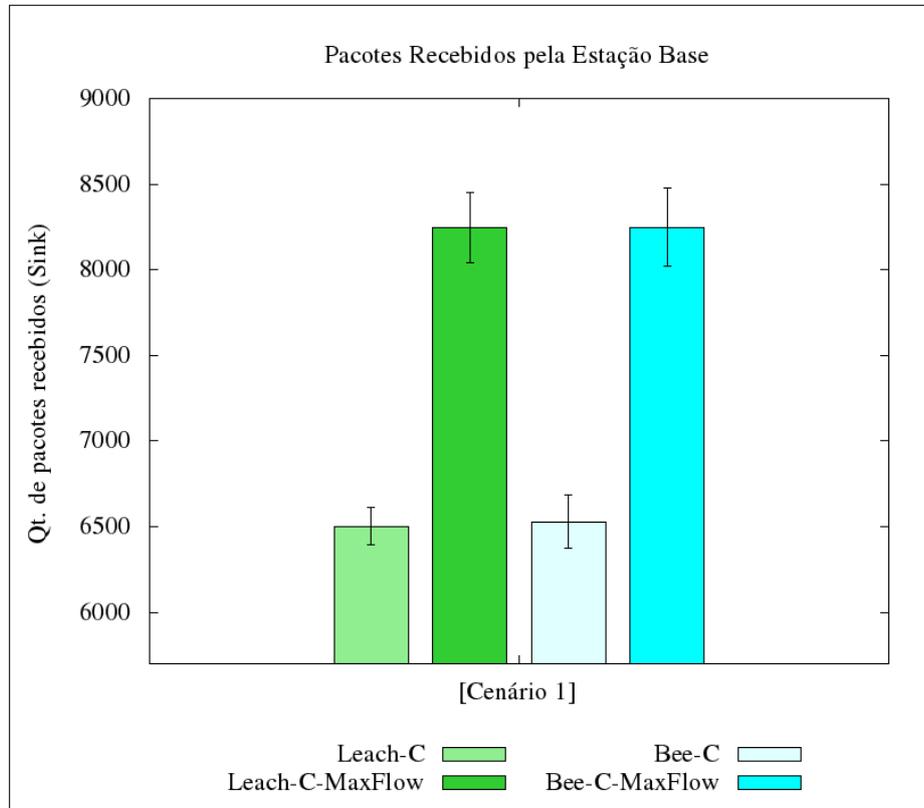
Fonte: Elaborado pelo autor

questão está relacionada ao tempo de duração da rodada. Na versão original dos protocolos, esse tempo é fixo e os nós acabam sendo limitados a somente transmitirem durante esse intervalo. Uma vez que esse tempo expira, a transmissão de pacotes é encerrada e uma nova clusterização deve começar.

Na versão dos protocolos com a abordagem, o tempo de rodada apresenta uma duração maior, visto que uma nova clusterização só será engatilhada quando algum nó atingir um percentual de sua energia. Como nos períodos iniciais da rede, os sensores possuem bastante bateria; o percentual de energia leva certo tempo para ser atingido, o que estende a duração do *round*, proporcionando assim a transmissão de mais dados.

A Figura 28 mostra a quantidade de dados recebidos pela Estação Base que cada protocolo obteve no Cenário 2. Através dos resultados apresentados no gráfico, observou-se que o protocolo Leach-C-MaxFlow obteve um ganho na quantidade de pacotes enviados de aproximadamente 37.35% em relação ao Leach-C. O ganho do Bee-C-MaxFlow em relação ao Bee-C foi de aproximadamente 26.83%. Nesse cenário, a quantidade de pacotes enviados por cada protocolo foi maior em relação ao Cenário 1. Isso acontece devido a uma maior quantidade

Figura 26 – Quantidade de dados recebidos pela Estação Base do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.



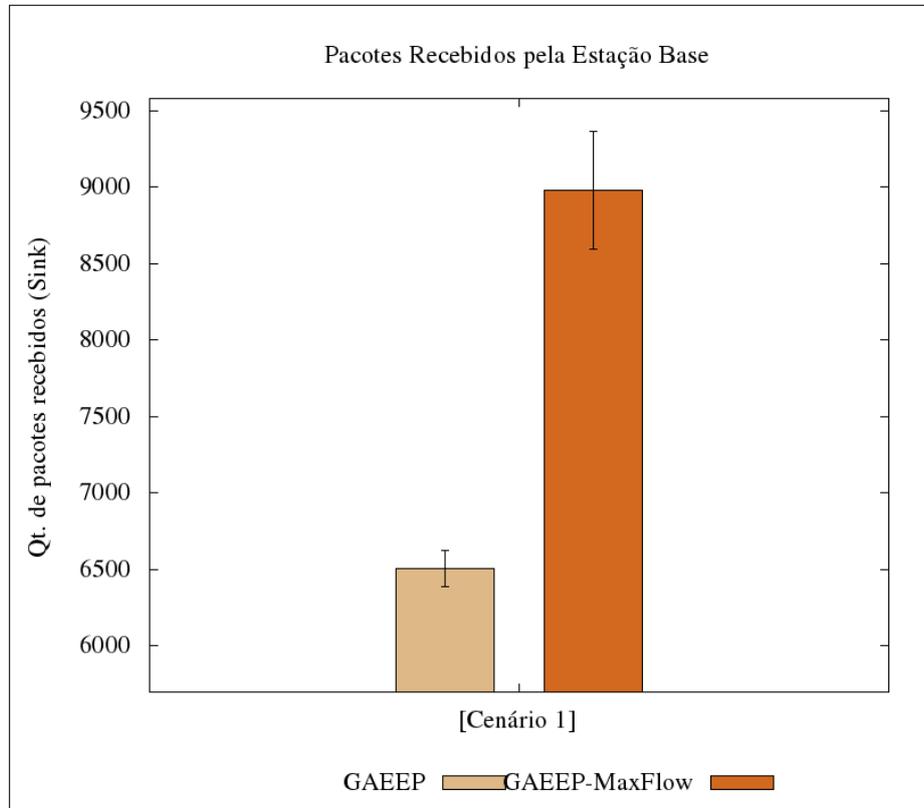
Fonte: Elaborado pelo autor

de nós e de clusters-head gerando dados na rede.

O protocolo GAEEP-MaxFlow apresentou resultados piores do que sua versão original no Cenário 2. A razão disso está relacionada com a quantidade de clusters-head que o GAEEP pode formar na rede. Observou-se através das simulações que, nesse Cenário 2, o GAEEP encontra uma quantidade média de 90 clusters-head. Então, a abordagem constrói um grafo direcionado dessa quantidade de clusters-head, calculando o quanto de energia cada nó pode gastar por cada um de seus vizinhos (enlaces).

O problema é que como o grafo de clusters-head possui uma grande quantidade de nós, cada CH possui muitos vizinhos para dividir sua energia, fazendo com que a abordagem calcule uma fração dela para cada aresta (vizinho) que esse CH tem no grafo. Acrescenta-se a essa afirmação o fato de que, o grafo de clusters-head pode apresentar nós com muitas arestas de entrada (vizinhos transmitindo dados para ele) e muitas arestas de saída (repasse de informação para outros nós). Para os nós com essa característica, a abordagem encontra um percentual de energia bem pequeno, ficando uma fração também bem pequena de energia que ele pode gastar

Figura 27 – Quantidade de dados recebidos pela Estação Base do GAEEP e do GAEEP-MaxFlow no Cenário 1.



Fonte: Elaborado pelo autor

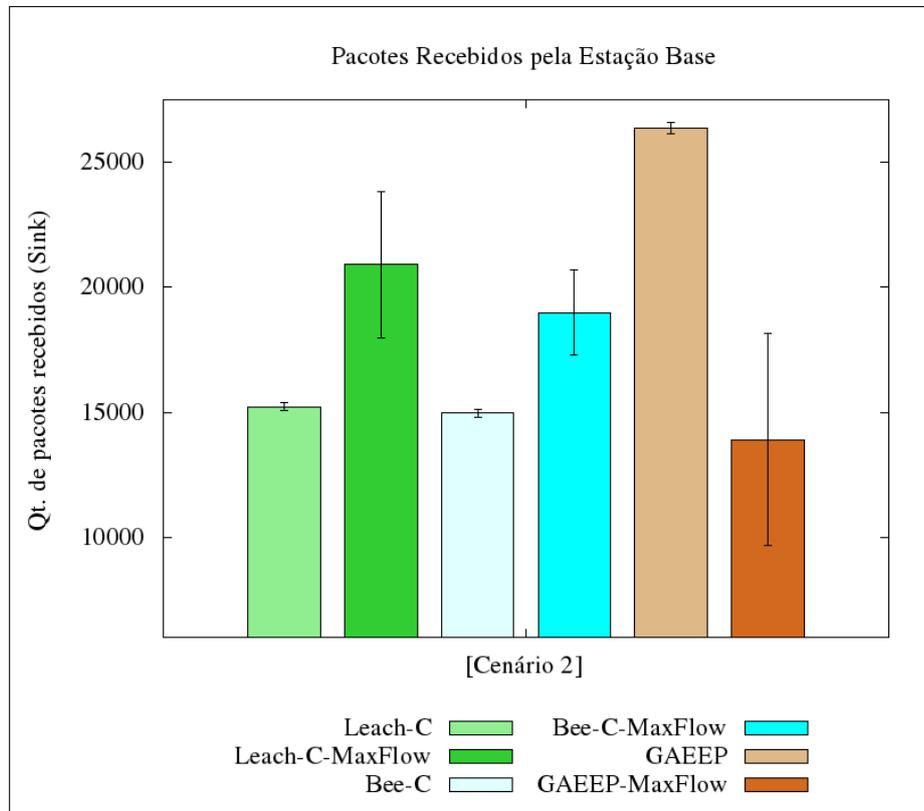
com os seus vizinhos.

Dessa forma, os nós atingem de forma mais rápida o percentual máximo de energia, engatilhando assim o processo de clusterização. Esse fato ocasiona alguns fatores que contribuem para o mau desempenho do protocolo. O primeiro deles é um tempo de *round* mais curto, o que gera uma baixa taxa de entrega de pacotes à Estação Base, visto que se tem menos tempo para ficar transmitindo. O outro fato consiste em realizar repetidas vezes e em intervalos curtos de tempo o processo de clusterização, o que contribui bastante no aumento do consumo de energia.

Por esses motivos, o GAEEP-MaxFlow apresentou um desempenho abaixo de sua versão original. Isto leva a conclusão de que em uma rede com muitos clusters-head, a abordagem acaba contribuindo para um maior consumo e diminuindo a quantidade de pacotes corretamente entregues à Estação Base.

O quadro 13 apresenta a média da quantidade de dados entregues à Estação Base e os seus respectivos limites (inferior e superior), considerando o intervalo de confiança de 95%, para cada protocolo nos Cenários 1 e 2.

Figura 28 – Quantidade de dados recebidos pela Estação Base no Cenário 2.



Fonte: Elaborado pelo autor

Quadro 13 – Resultados da Qt. de Pacotes entregues à Estação Base nos Cenários 1 e 2.

Cenário 1			
Protocolo	Qt. de Pacotes Enviados	Limite Inferior	Limite Superior
Leach-C	6503	6395.1	6610.9
Leach-C-MaxFlow	8248.64	8043.775	8453.505
Bee-C	6529.58	6374.498	6684.662
Bee-C-MaxFlow	8248.94	8018.736	8479.144
GAEEP	6508.97	6389.875	6628.065
GAEEP-MaxFlow	8981.3	8596.506	9366.094
Cenário 2			
Protocolo	Qt. de Pacotes Enviados	Limite Inferior	Limite Superior
Leach-C	15223.4	15062.512	15384.288
Leach-C-MaxFlow	20910.2	17972.78	23847.62
Bee-C	14965.8	14796.24	15135.36
Bee-C-MaxFlow	18982.5	17287.21	20677.79
GAEEP	26369.5	26143.468	26595.532
GAEEP-MaxFlow	13895.4	9652.02	18138.78

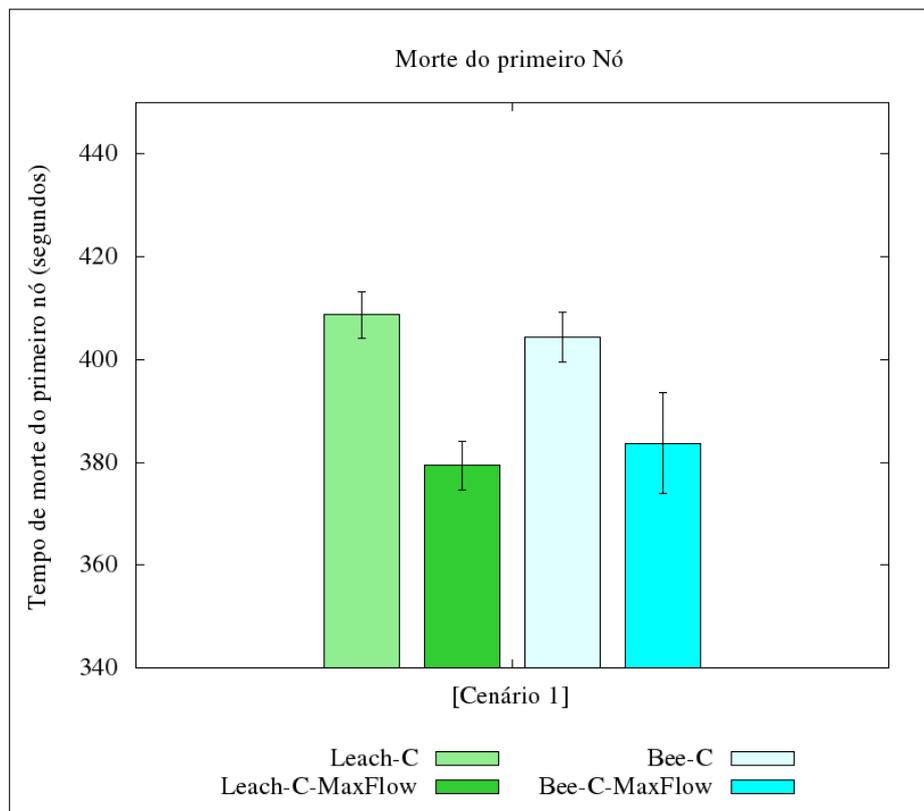
Fonte: Elaborado pelo autor

5.4.3 Tempo de morte do primeiro nó

A Figura 29 apresenta o tempo de morte do primeiro nó obtido pelos protocolos Leach-C e Bee-C (versão original e com a abordagem) no Cenário 1. Para essa métrica, nesse

cenário, os resultados dos protocolos com a abordagem, com a exceção do GAEEP-MaxFlow, não foram melhores quando comparados com as suas versões originais. A morte do primeiro nó no Leach-C-MaxFlow aconteceu no tempo 379.366s, enquanto que no Leach-C ocorreu no tempo 408.674s. O protocolo Bee-C-MaxFlow apresentou o tempo de 383.684s e sua versão original (Bee-C) apresentou tempo de 404.316s.

Figura 29 – Tempo de morte do primeiro nó do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.



Fonte: Elaborado pelo autor

A abordagem proposta ocasiona uma morte mais rápida do primeiro nó pelo fato das primeiras rodadas durarem mais tempo. Assim os sensores clusters-head que participam das primeiras rodadas acabam por consumir mais energia do que os demais. Dessa forma, esses nós acabam morrendo mais cedo. Contudo, à medida que as rodadas aumentam, o tempo de duração delas tende a ser menor. Esse fato acontece por que os sensores alcançam de forma mais rápida o percentual de energia calculado pela abordagem, uma vez que eles não estão mais com a sua energia total. Desse modo, a troca de clusters-head acontece em um período de tempo menor, evitando que estes nós fiquem na posição de líder do grupo, o que consome mais energia, por um longo período na rede.

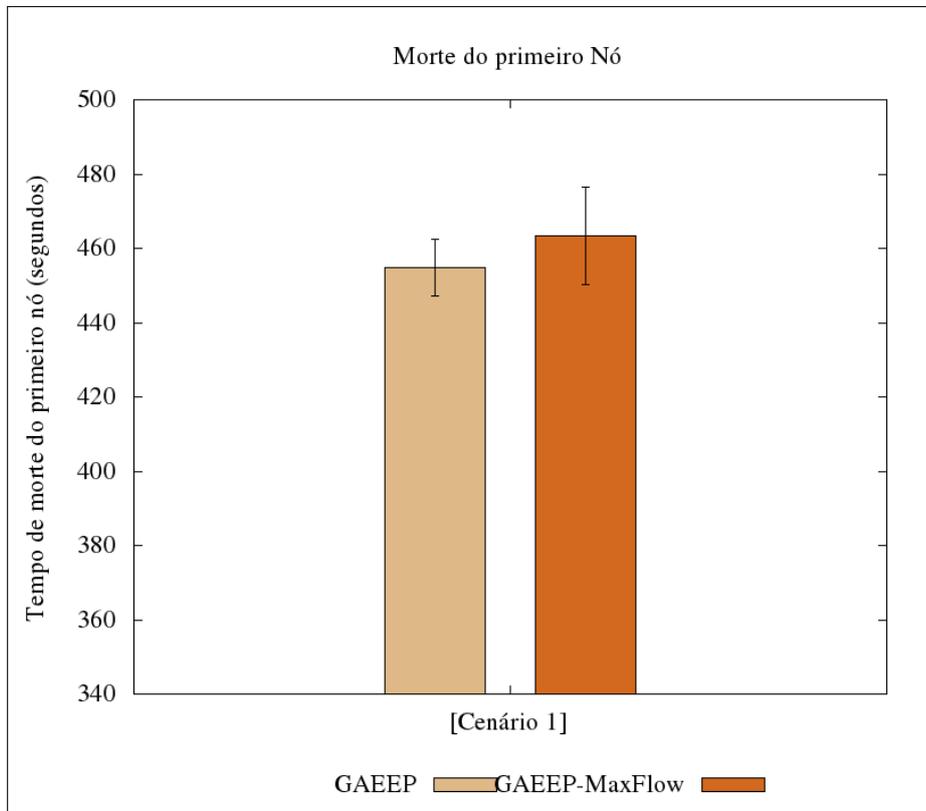
Outro fato que também contribui para a morte mais rápida do primeiro nó nos protocolos com a abordagem, é o fato de os nós mais próximos à Estação Base consumirem bastante energia, uma vez que eles são responsáveis por repassarem os dados da maioria dos nós na rede. Isso acontece devido à comunicação multihop entre os sensores ser implementada pela abordagem, ou seja, os nós que estão mais distantes da Estação Base escolhem os nós mais próximos a ela para encaminhar os dados, assim esses nós acabam consumindo mais energia e geralmente alcançam um tempo de morte mais rápido.

A Figura 30 apresenta o tempo de morte do primeiro nó obtido pelos protocolos GAEEP e GAEEP-MaxFlow no Cenário 1. O protocolo GAEEP-MaxFlow obteve o tempo 463.327s, enquanto que o GAEEP apresentou um tempo de 454.749s. A versão do protocolo com a abordagem conseguiu um tempo maior em relação a sua versão original pois a quantidade de clusters que o GAEEP pode formar é variável. Com essa afirmação, verificou-se que o protocolo, em algumas rodadas, encontra uma quantidade pequena de clusters-head e em algumas vezes inferior à quantidade encontrada pelos protocolos Leach-C e Bee-C. Com isso, alguns *rounds* possuem menos nós clusters-head gastando uma carga maior de energia, o que evita uma morte rápida do primeiro nó. Porém, esse fato não possui um impacto muito grande, visto que parte do intervalo de confiança do GAEEP está contido no intervalo de confiança do GAEEP-MaxFlow, conforme pode ser visto no gráfico da Figura 29.

A Figura 31 apresenta o tempo de morte do primeiro nó obtido por cada protocolo no Cenário 2. O Leach-C-MaxFlow apresentou o tempo de 427.361s, enquanto que o Leach-C obteve o tempo 416.561s. Como pode ser visto, os dois protocolos apresentaram tempos parecidos, sendo que o primeiro nó demora mais a morrer no Leach-C-MaxFlow. Esse fato deve-se a grande densidade de nós apresentada no Cenário 2. Como os nós estão mais próximos, a possibilidade de colisões e perdas de pacotes aumenta, podendo variar de protocolo para protocolo. Assim, nas etapas de formação dos clusters, a Estação Base pode não receber pacotes de determinados nós da rede. Por esse motivo, esses nós acabam se tornando auto-clusters-head. Dessa forma, o Leach-C apresentou o tempo de morte do primeiro nó mais rápido, visto que a rede contém mais nós líderes, o que contribui para um maior consumo de energia. Porém, esse fato não apresenta tanto impacto, visto que parte do intervalo de confiança do Leach-C está contido no intervalo do Leach-C-MaxFlow. Esse fato indica que os tempos de ambos os protocolos (Leach-C e Leach-C-MaxFlow) foram semelhantes.

O protocolo Bee-C-MaxFlow obteve o tempo de 365.597s e o Bee-C alcançou o

Figura 30 – Tempo de morte do primeiro nó do GAEEP e do GAEEP-MaxFlow no Cenário 1.



Fonte: Elaborado pelo autor

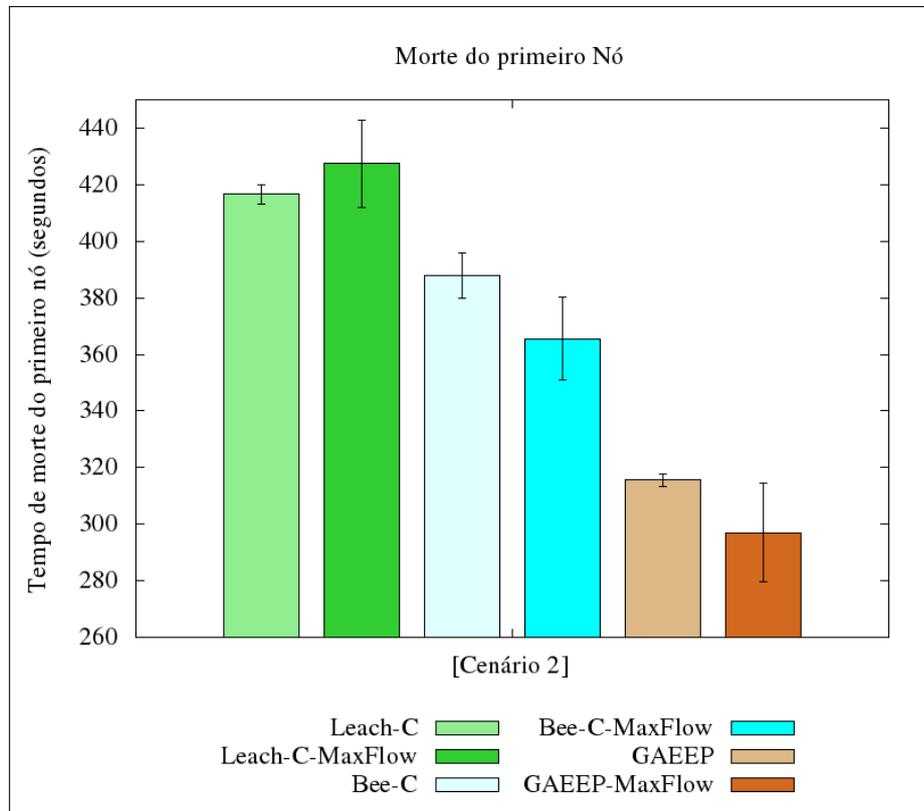
tempo de 387.751s. O GAEEP-MaxFlow apresentou um tempo de 297.046s e sua versão original conseguiu o tempo de 315.518s. Nesses últimos casos, os tempos dos protocolos também foram semelhantes, porém as versões com a abordagem apresentaram tempo menor. Conforme explicado anteriormente nessa seção para o Cenário 1, a abordagem contribuiu para um tempo mais rápido de morte do primeiro nó. Esse fato se deve a uma duração mais longa das rodadas iniciais e ao maior consumo dos CHs próximos da Estação Base que têm que repassar os dados de vários sensores da rede.

O quadro 14 exibe a média do tempo de morte do primeiro nó e os seus respectivos limites (inferior e superior), considerando o intervalo de confiança de 95%, para cada protocolo nos Cenários 1 e 2.

5.4.4 Tempo de vida da rede (50% de nós mortos)

A Figura 32 apresenta o tempo de vida da rede, considerando 50% de nós mortos, alcançado pelos protocolos Leach-C e Bee-C (versão original e com a abordagem) no Cenário 1. Nesse cenário, a abordagem conseguiu aumentar o tempo de vida dos sensores em cada

Figura 31 – Tempo de morte do primeiro nó no Cenário 2.



Fonte: Elaborado pelo autor

Quadro 14 – Resultados do Tempo de morte do primeiro nó nos Cenários 1 e 2.

Cenário 1			
Protocolo	Tempo de Morte	Limite Inferior	Limite Superior
Leach-C	408.674	404.166	413.181
Leach-C-MaxFlow	379.366	374.552	384.179
Bee-C	404.316	399.518	409.113
Bee-C-MaxFlow	383.684	373.898	393.469
GAEEP	454.749	447.198	462.299
GAEEP-MaxFlow	463.327	450.302	476.351
Cenário 2			
Protocolo	Tempo de morte	Limite Inferior	Limite Superior
Leach-C	416.561	413.290	419.831
Leach-C-MaxFlow	427.361	411.826	442.895
Bee-C	387.751	379.758	395.743
Bee-C-MaxFlow	365.597	350.932	380.261
GAEEP	315.518	313.458	317.577
GAEEP-MaxFlow	297.046	279.511	314.580

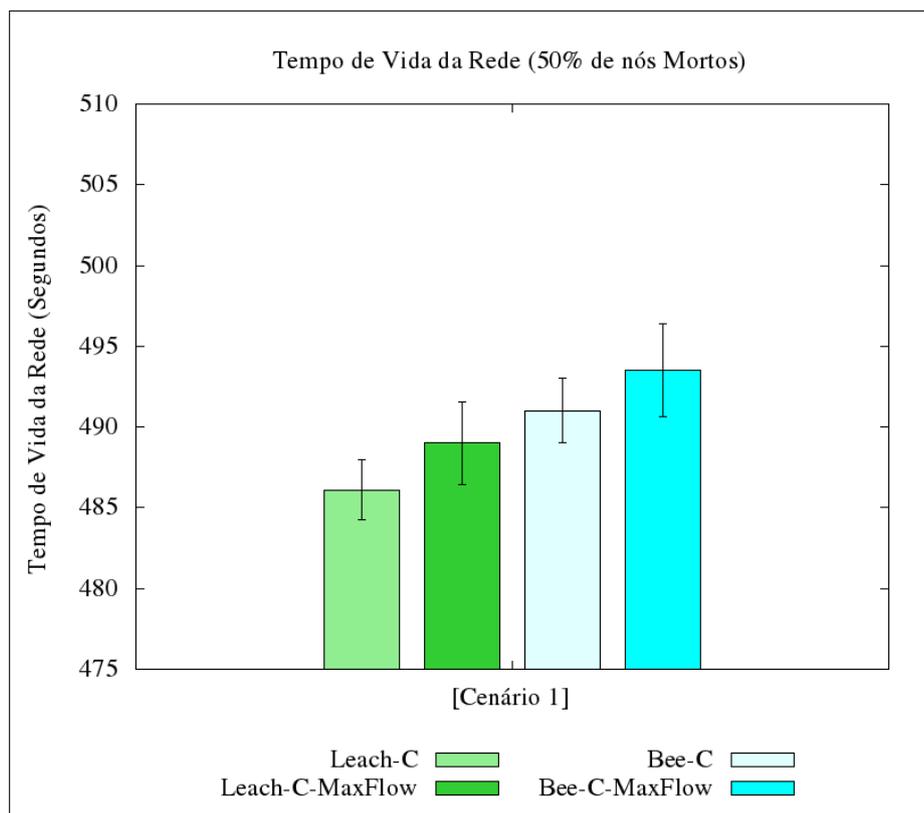
Fonte: Elaborado pelo autor

protocolo. O Leach-C-MaxFlow apresentou um tempo de 488.994s, enquanto que a sua versão original (Leach-C) alcançou um tempo de 486.098s. O percentual de ganho atingido pelo Leach-C-MaxFlow foi de aproximadamente 0.59%. O protocolo Bee-C-MaxFlow obteve um tempo de 493.503s e o Bee-C apresentou o tempo 491.027s. Nesse caso, o protocolo com a

abordagem também conseguiu um percentual de aumento de 0.5042% em relação a sua versão original. A Figura 33 apresenta o tempo de vida da rede obtido pelos protocolos GAEEP e GAEEP-MaxFlow no Cenário 1. O GAEEP-MaxFlow apresentou um percentual de aumento de aproximadamente 7.67% em relação ao GAEEP. Os tempos para ambos os protocolos, GAEEP e GAEEP-MaxFlow, foram de 525.746s e 566.073s, respectivamente.

O aumento no tempo de vida dos sensores está relacionado com a economia de energia que cada nó alcança devido às características da abordagem, ou seja, por causa da comunicação multi-hop, balanceamento de energia, etc. Assim, os nós vivem mais e acabam por demorar a alcançar um percentual de 50% de nós mortos.

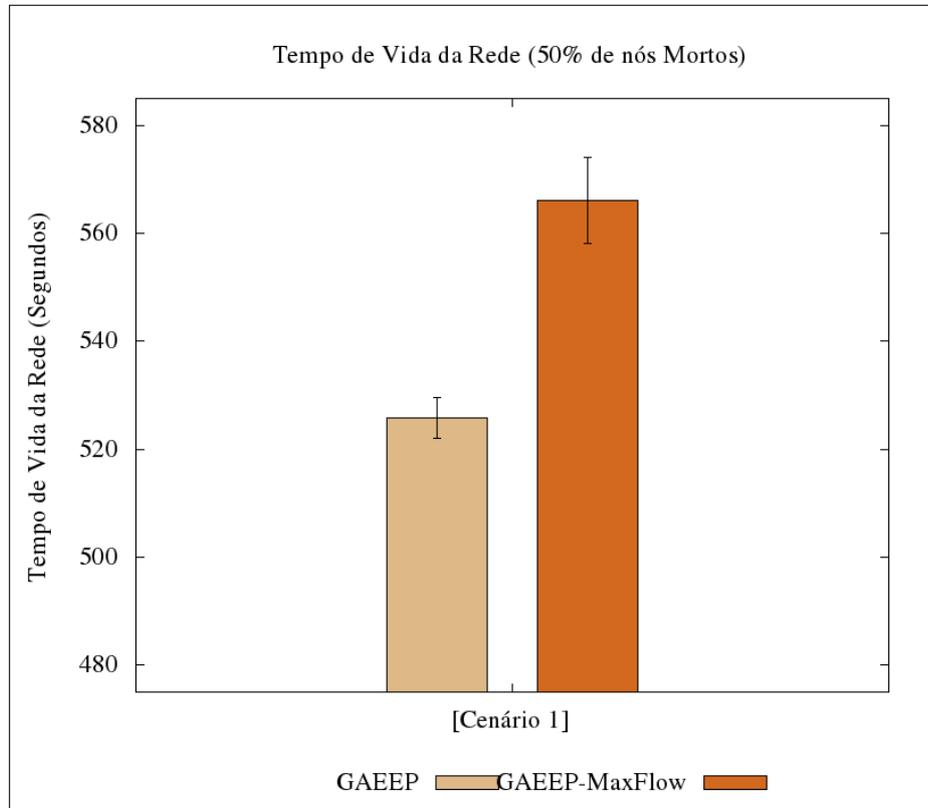
Figura 32 – Tempo de Vida da Rede (50% de nós Mortos) do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.



Fonte: Elaborado pelo autor

A Figura 34 apresenta o tempo de vida da rede alcançado por cada protocolo no Cenário 2. O protocolo Leach-C-MaxFlow apresentou um ganho percentual de 5.58% em relação ao Leach-C. Os tempos desses protocolos (Leach-C-MaxFlow e Leach-C) foram 499.793s e 473.359s, respectivamente. O Bee-C-MaxFlow obteve um aumento de aproximadamente 3.99% quando comparado com o Bee-C e os seus tempos foram de 506.427s e 486.979s,

Figura 33 – Tempo de Vida da Rede (50% de nós Mortos) do GAEEP e do GAEEP-MaxFlow no Cenário 1.



Fonte: Elaborado pelo autor

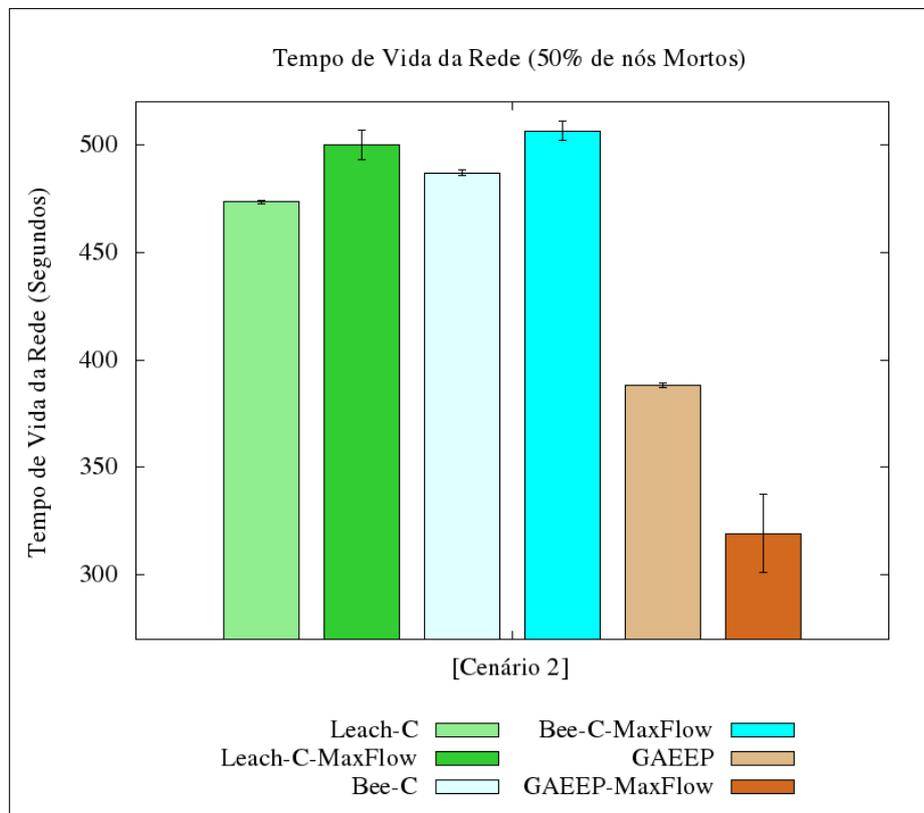
respectivamente.

Como pode ser visto nesse cenário, os protocolos com a abordagem, com a exceção do GAEEP-MaxFlow, também apresentaram maior tempo de vida em relação às suas versões originais. Apesar de todos eles consumirem praticamente toda a sua bateria no Cenário 2, a abordagem proporcionou uma economia aos sensores, prolongando os seus tempos de vida.

O protocolo GAEEP-MaxFlow apresentou o tempo 319.18s, enquanto que o GAEEP obteve o tempo de 388.05s. Como pode ser visto, o protocolo com a abordagem atingiu de forma mais rápida o percentual de 50% de nós mortos. Conforme foi explicado na subseção 5.4.2, quando a rede possui muitos clusters-head, a abordagem acaba piorando o desempenho do protocolo. Como no Cenário 2 o GAEEP forma muitos CHs, a abordagem contribui para um maior consumo de energia dos nós, pois ocasiona muitas clusterizações em um período curto de tempo. Esse fato ocasiona uma morte prematura dos sensores, onde a rede alcança rapidamente um percentual de 50% de nós mortos.

O quadro 15 apresenta a média do tempo de vida da rede e os seus respectivos limites

Figura 34 – Tempo de Vida da Rede (50% de nós Mortos) no Cenário 2.



Fonte: Elaborado pelo autor

(inferior e superior), considerando o intervalo de confiança de 95%, para cada protocolo nos Cenários 1 e 2.

Quadro 15 – Resultados do Tempo de Vida da Rede (50% de nós Mortos) nos Cenários 1 e 2.

Cenário 1			
Protocolo	Tempo de Vida da Rede	Limite Inferior	Limite Superior
Leach-C	486.098	484.253	487.942
Leach-C-MaxFlow	488.994	486.462	491.525
Bee-C	491.027	489.040	493.013
Bee-C-MaxFlow	493.503	490.640	496.365
GAEEP	525.746	521.928	529.563
GAEEP-MaxFlow	566.073	558.126	574.019
Cenário 2			
Protocolo	Tempo de Vida da Rede	Limite Inferior	Limite Superior
Leach-C	473.359	472.356	474.361
Leach-C-MaxFlow	499.793	492.918	506.667
Bee-C	486.979	485.486	488.471
Bee-C-MaxFlow	506.427	501.869	510.984
GAEEP	388.05	387.121	388.978
GAEEP-MaxFlow	319.18	301.101	337.258

Fonte: Elaborado pelo autor

5.4.5 Tempo de vida da rede (90% de nós mortos)

A Figura 35 apresenta o tempo de vida da rede, considerando 90% de nós mortos, alcançado pelos protocolos Leach-C e Bee-C (versão original e com a abordagem) no Cenário 1. Essa métrica é importante, pois ela representa quase o tempo total de vida da rede, ou seja, o máximo de tempo que os sensores ficam monitorando o ambiente. Nessa métrica e nesse cenário, a abordagem também conseguiu aumentar o tempo de vida dos sensores em cada protocolo.

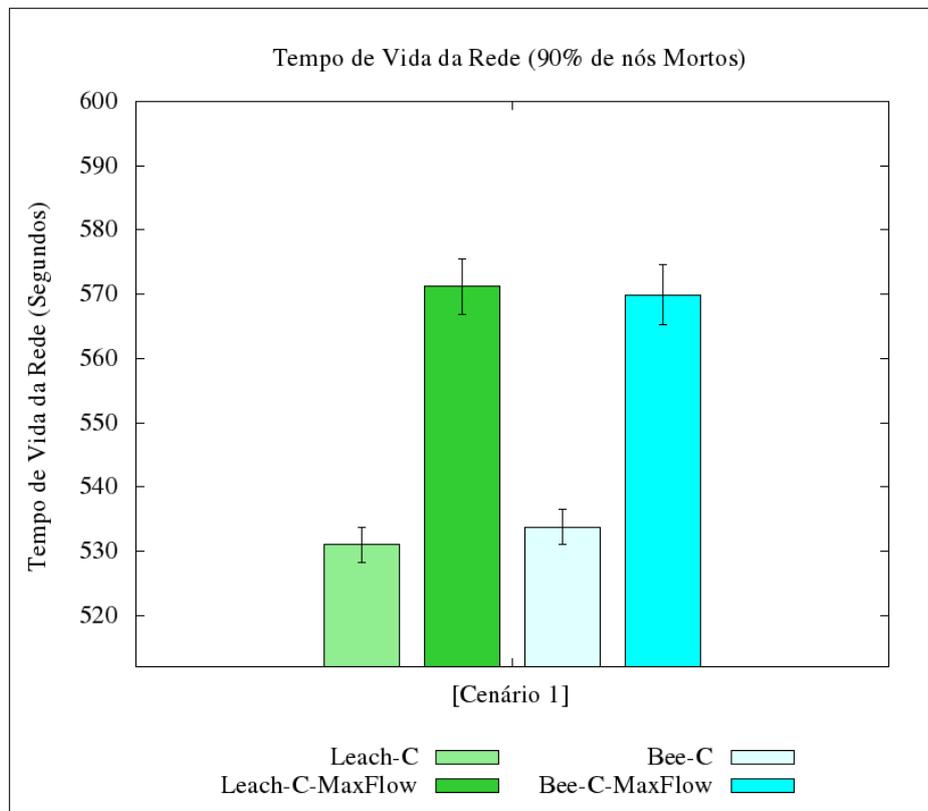
O protocolo Leach-C-MaxFlow atingiu um tempo de 571.173s, enquanto que o Leach-C apresentou um tempo de 530.965s. A diferença dos tempos obtidos por esses protocolos proporcionou um aumento de 7.5726% para o Leach-C-MaxFlow. O Bee-C-MaxFlow conseguiu um tempo de 569.91s enquanto que o tempo do Bee-C foi de 533.774s. O ganho do Bee-C-MaxFlow em relação a sua versão original foi de aproximadamente 6.769%. A Figura 36 apresenta o tempo de vida da rede obtido pelos protocolos GAEEP e GAEEP-MaxFlow no Cenário 1. O GAEEP-MaxFlow conseguiu um tempo de 636.13s e o GAEEP apresentou o tempo 557.258s. Nesse último caso, a versão do protocolo com a abordagem obteve um ganho percentual de aproximadamente 14,15%.

Conforme dito anteriormente, o aumento no tempo de vida dos sensores ocorre por causa da economia de energia que cada nó alcança devido às características que a abordagem proporciona. Assim, a rede leva um longo período de tempo para alcançar um percentual de 90% de nós mortos.

A Figura 37 mostra o tempo de vida da rede que cada protocolo atingiu no Cenário 2. O Leach-C-MaxFlow conseguiu um aumento de 7.3445% quando comparado com o Leach-C e os seus tempos foram de 531.889s e 495.497s, respectivamente. O protocolo Bee-C-MaxFlow apresentou um ganho de 5.64% em relação ao Bee-C. Os tempos de ambos os protocolos (Bee-C-MaxFlow e Bee-C) foram de 539.06s e 507.408s, respectivamente. Nesse cenário, a abordagem incorporada aos protocolos Leach-C e Bee-C, também propiciou uma economia de energia aos nós sensores. Dessa forma, eles atingiram um percentual de 90% de nós mortos em um período maior de tempo.

O GAEEP-MaxFlow obteve o tempo de 327.03s e sua versão original (GAEEP) obteve o tempo de 425.703s. Nessa métrica e nesse cenário, o protocolo GAEEP com a abordagem também apresentou um desempenho ruim. Conforme dito anteriormente na subseção 5.4.2, o GAEEP encontra uma grande quantidade de clusters-head no Cenário 2 e quando se aplica a abordagem a um número maior de CHs, ela acaba prejudicando a execução do protocolo.

Figura 35 – Tempo de Vida da Rede (90% de nós Mortos) do Leach-C, Leach-C-MaxFlow, Bee-C e Bee-C-MaxFlow no Cenário 1.

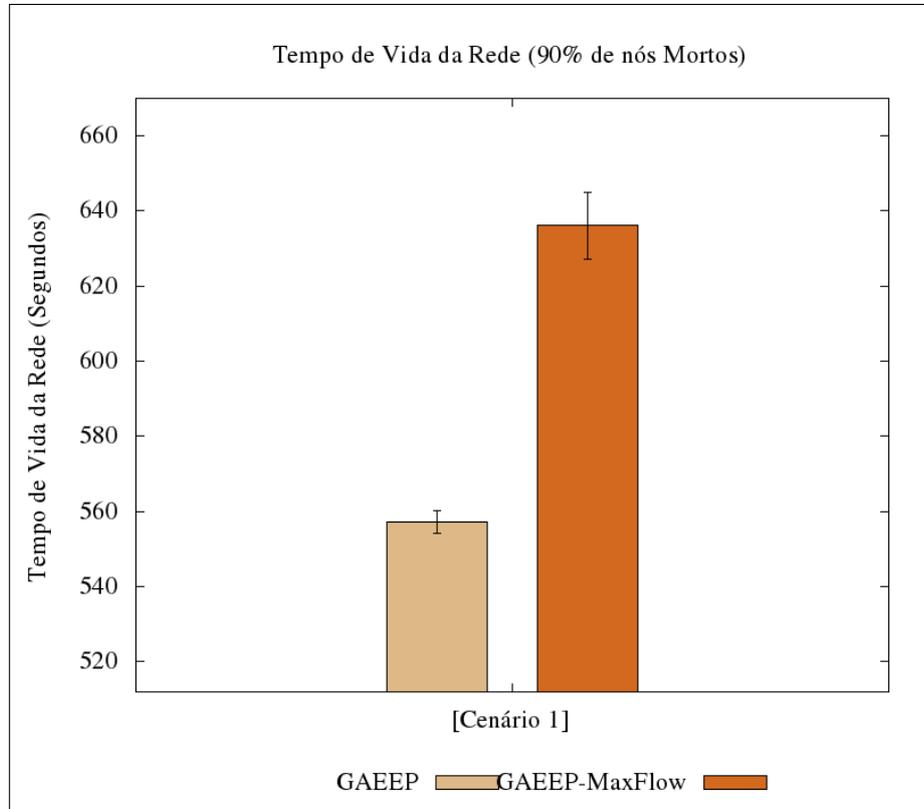


Fonte: Elaborado pelo autor

Essa *performance* ruim é explicada pelo maior consumo de energia que os sensores apresentam, ocasionada por muitas clusterizações que acontecem em intervalos curtos de tempo. Nesse cenário, outro fator que contribui para um tempo de vida menor no GAEEP-MaxFlow é o gasto maior de energia dos sensores mais próximos à Estação Base, que repassam dados de uma quantidade maior de nós sensores. Assim, eles acabam morrendo mais rapidamente.

O quadro 16 apresenta a média do tempo de vida da rede e os seus respectivos limites (inferior e superior), considerando o intervalo de confiança de 95%, para cada protocolo nos Cenários 1 e 2.

Figura 36 – Tempo de Vida da Rede (90% de nós Mortos) do GAEEP e do GAEEP-MaxFlow no Cenário 1.



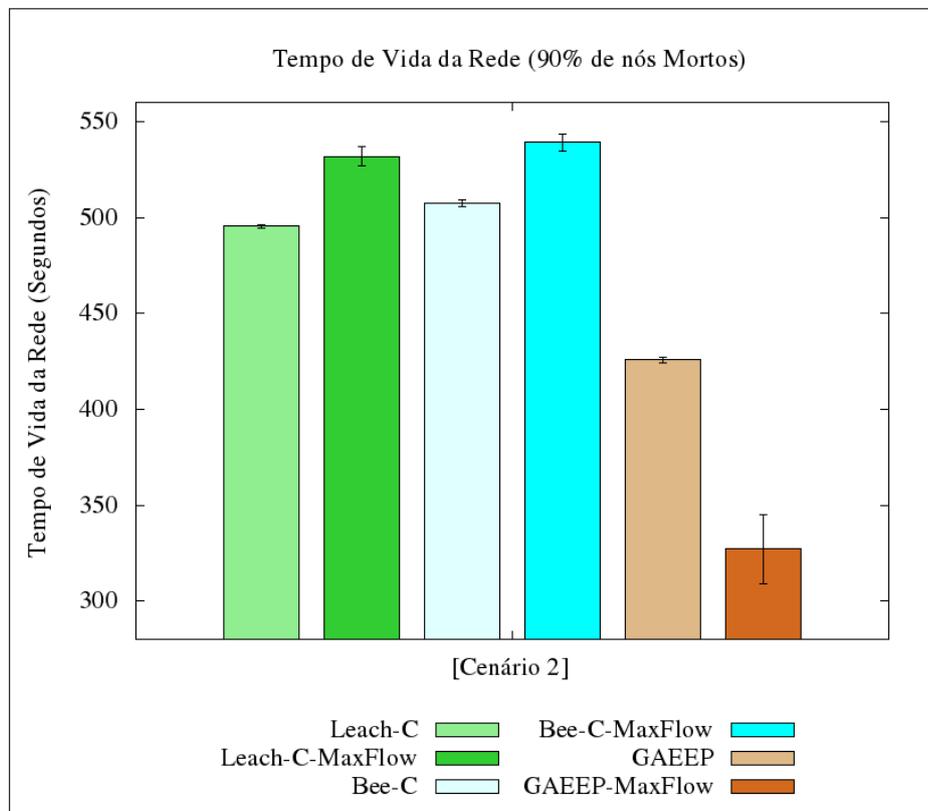
Fonte: Elaborado pelo autor

Quadro 16 – Resultados do Tempo de Vida da Rede (90% de nós Mortos) nos Cenários 1 e 2.

Cenário 1			
Protocolo	Tempo de Vida da Rede	Limite Inferior	Limite Superior
Leach-C	530.965	528.190	533.739
Leach-C-MaxFlow	571.173	566.920	575.425
Bee-C	533.774	531.104	536.443
Bee-C-MaxFlow	569.91	565.282	574.537
GAEEP	557.25	554.212	560.303
GAEEP-MaxFlow	636.13	627.306	644.953
Cenário 2			
Protocolo	Tempo de Vida da Rede	Limite Inferior	Limite Superior
Leach-C	495.497	494.564	496.429
Leach-C-MaxFlow	531.889	526.949	536.828
Bee-C	507.408	505.873	508.942
Bee-C-MaxFlow	539.06	534.558	543.561
GAEEP	425.703	424.108	427.297
GAEEP-MaxFlow	327.03	308.967	345.092

Fonte: Elaborado pelo autor

Figura 37 – Tempo de Vida da Rede (90% de nós Mortos) no Cenário 2.



Fonte: Elaborado pelo autor

6 CONCLUSÕES E TRABALHOS FUTUROS

O trabalho teve como objetivo propor uma nova abordagem para melhorar protocolos de roteamento que utilizam a técnica de clusterização. A abordagem procura trazer benefícios na maneira de rotear os dados entre os clusters-head como também uma nova maneira de definir o tempo de uma rodada no processo de clusterização. Para alcançar os objetivos do trabalho, foi apresentada uma modelagem através de grafos utilizando-se fluxo em redes.

A proposta foi incorporada nos protocolos Leach-C, Bee-C e GAEEP, que trabalham com clusterização, e avaliada através de simulações no simulador de rede Omnet++/Castalia. O objetivo foi comparar os protocolos incorporados com a abordagem com as suas versões originais (sem a abordagem). As simulações foram realizadas em dois cenários e os resultados foram comparados através das seguintes métricas: consumo médio de energia do nó; quantidade de dados recebidos pela Estação Base; morte do primeiro nó e tempo de vida da rede (50% e 90% de nós mortos).

Através dos resultados obtidos, observou-se que a abordagem melhorou o desempenho dos protocolos em uma rede menos densa (Cenário 1), proporcionando ganho em quase todas as métricas. Nesse cenário, a abordagem conseguiu alcançar uma economia de energia para os nós, aumentar a quantidade de pacotes entregues a Estação Base e prolongar o tempo de vida da rede. Na métrica que mede o tempo de morte do primeiro nó, a abordagem obteve, no geral, tempos inferiores em relação às versões originais dos protocolos.

Assim, pode-se concluir que se um protocolo objetivar um maior tempo de vida da rede e uma maior quantidade de pacotes transmitidos, então a abordagem pode ser uma boa escolha para melhorar suas operações. Porém, se o objetivo for um tempo maior de cobertura da rede, tal protocolo deve procurar outras alternativas. O quadro 17 resume os resultados obtidos pela abordagem no Cenário 1.

Em uma rede maior e mais densa (Cenário 2), a proposta também trouxe benefícios para os protocolos Leach-C e Bee-C. Embora esses protocolos (versão original e versão com a abordagem) tenham apresentado praticamente o mesmo consumo nesse cenário, a abordagem conseguiu aumentar a quantidade de pacotes entregues à Estação Base, como também conseguiu prolongar o tempo de vida da rede. A abordagem não foi muito adequada com o protocolo GAEEP nesse Cenário 2. Como esse protocolo formou muitos clusters em uma rede com um grande número de nós, a abordagem acabou prejudicando a sua *performance*. Isto se deve ao grafo bastante denso que a proposta acaba criando quando a quantidade de clusters-head é bem

Quadro 17 – Resultados obtidos pela Abordagem no Cenário 1.

Métricas	Protocolos		
	Leach-C-MaxFlow em relação ao Leach-C	Bee-C-MaxFlow em relação ao Bee-C	GAEEP-MaxFlow em relação ao GAEEP
Consumo Médio	Economia de 0.25 %	Economia de 0.28%	Economia de 0.0006%
Pacotes Transmitidos	Aumento de 26.84%	Aumento de 26.33%	Aumento de 37.98%
Morte do primeiro nó	Perda de 7.17%	Perda de 5.10%	Aumento de 1.8%
50% de nós Mortos	Aumento de 0.59%	Aumento de 0.5042%	Aumento de 7.67%
90% de nós Mortos	Aumento de 7.5726%	Aumento de 6.769%	Aumento de 14.15%

Fonte: Elaborado pelo autor

elevada. Com esse grafo denso, a abordagem acaba calculando um percentual pequeno de energia para cada nó, o que influencia na baixa taxa de entrega de pacotes e no alto consumo de energia.

Assim, conclui-se que a abordagem conseguiu trazer melhorias aos protocolos em uma rede maior e mais densa, porém se a quantidade de clusters-head não for adequada de acordo com o número de nós da rede, a proposta acaba piorando o desempenho de tais protocolos. Dessa forma, um protocolo que queira usar a abordagem para melhorar as suas operações, deve verificar se sua forma de clusterizar a rede se enquadra com as características da abordagem. O quadro 18 resume os resultados obtidos pela abordagem no Cenário 2.

Quadro 18 – Resultados obtidos pela Abordagem no Cenário 2.

Métricas	Protocolos		
	Leach-C-MaxFlow em relação ao Leach-C	Bee-C-MaxFlow em relação ao Bee-C	GAEEP-MaxFlow em relação ao GAEEP
Consumo Médio	Praticamente o mesmo consumo	Praticamente o mesmo consumo	Praticamente o mesmo consumo
Pacotes Transmitidos	Aumento de 37.35%	Aumento de 26.83%	Perda de 47%
Morte do primeiro nó	Aumento de 2.5%	Perda de 5.71%	Perda de 5.85%
50% de nós Mortos	Aumento de 5.58%	Aumento de 3.99%	Perda de 17.7%
90% de nós Mortos	Aumento de 7.3445%	Aumento de 5.64%	Perda de 23.17%

Fonte: Elaborado pelo autor

A principal contribuição do trabalho consiste em fornecer mecanismos para melhorar uma categoria de protocolos encontrados em Redes de Sensores sem Fio. Ou seja, não é mais

um protocolo sendo definido e que supera algum já existente, e sim uma abordagem que pode ser incorporada a qualquer protocolo dessa categoria trazendo benefícios e ajudando ainda mais a amenizar problemas como o consumo de energia e o roteamento de dados.

O trabalho possui certa limitação, pois não é aplicável a qualquer protocolo existente em Redes de Sensores Sem Fio. O protocolo que queira fazer uso da abordagem proposta terá que ser enquadrado na categoria de protocolos de roteamento centralizados, baseados em clusterização. Isso indica que o protocolo organiza a rede em grupo com os respectivos líderes, sendo a Estação Base responsável por realizar todo o processo de clusterização. Um protocolo que não possua essas características não pode fazer uso da abordagem por fluxo proposta.

Para trabalhos futuros, a abordagem será aplicada em protocolos somente centralizados. A ideia é montar um grafo de nós sensores onde a vizinhança seria definida apenas com vizinhos de primeiro ou segundo salto, ou de acordo com o raio de alcance do sensor. Em posse desse grafo, a abordagem indicará uma solução para onde os nós devem encaminhar os seus dados sem sobrecarregar nenhum sensor e sempre respeitando os seus limites de energia.

REFERÊNCIAS

- ABO-ZAHHAD, M.; AHMED, S. M.; SABOR, N.; SASAKI, S. A new energy-efficient adaptive clustering protocol based on genetic algorithm for improving the lifetime and the stable period of wireless sensor networks. **International Journal of Energy, Information & Communications**, v. 5, n. 3, 2014.
- AHUJA, R. K.; GOLDBERG, A. V.; ORLIN, J. B.; TARJAN, R. E. Finding minimum-cost flows by double scaling. **Mathematical programming**, Springer, v. 53, n. 1-3, p. 243–266, 1992.
- AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. Network flows. 1993. **Pg**, v. 39, p. 193–196, 1993.
- AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. **Ad hoc networks**, Elsevier, v. 3, n. 3, p. 325–349, 2005.
- AL-KARAKI, J. N.; KAMAL, A. E. Routing techniques in wireless sensor networks: a survey. **Wireless communications, IEEE, IEEE**, v. 11, n. 6, p. 6–28, 2004.
- AMMARI, H. M.; DAS, S. K. Centralized and clustered k-coverage protocols for wireless sensor networks. **Computers, IEEE Transactions on, IEEE**, v. 61, n. 1, p. 118–133, 2012.
- BOGLIOLO, A.; DELPRIORI, S.; LATTANZI, E.; SERAGHITI, A. Self-adapting maximum flow routing for autonomous wireless sensor networks. **Cluster Computing**, Springer, v. 14, n. 1, p. 1–14, 2011.
- BOULIS, A. Castalia. **A simulator for wireless sensor networks and body area networks. User's manual.**[Online, 2009.
- BOULIS, A. Castalia user manual. **NICTA. Marzo del**, 2011.
- CORMEN, T. H. **Algoritmos: teoria e prática**. [S.l.]: Elsevier, 2002.
- DAM, T. V.; LANGENDOEN, K. An adaptive energy-efficient mac protocol for wireless sensor networks. In: ACM. **Proceedings of the 1st international conference on Embedded networked sensor systems**. [S.l.], 2003. p. 171–180.
- FERNANDES, N. C.; MOREIRA, M. D.; VELLOSO, P. B.; COSTA, L.; DUARTE, O. Ataques e mecanismos de segurança em redes ad hoc. **Minicursos do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'2006)**, p. 49–102, 2006.
- GOLDBERG, A.; TARJAN, R. Solving minimum-cost flow problems by successive approximation. In: ACM. **Proceedings of the nineteenth annual ACM symposium on Theory of computing**. [S.l.], 1987. p. 7–18.
- HEINZELMAN, W. B.; CHANDRAKASAN, A. P.; BALAKRISHNAN, H. An application-specific protocol architecture for wireless microsensor networks. **Wireless Communications, IEEE Transactions on, IEEE**, v. 1, n. 4, p. 660–670, 2002.
- JABOUR, F.; GIANCOLI, E.; PEDROZA, A. Um esquema em duas camadas para suporte a mobilidade em redes de sensores sem fio. **XXV Simpósio Brasileiro de Telecomunicações-Recife, PE**, 2007.

- JAVAID, N.; QURESHI, T.; KHAN, A.; IQBAL, A.; AKHTAR, E.; ISHFAQ, M. Enhanced developed distributed energy-efficient clustering for heterogeneous wireless sensor networks. **Procedia Computer Science**, Elsevier, v. 19, p. 914–919, 2013.
- KARL, H.; WILLIG, A. **Protocols and architectures for wireless sensor networks**. [S.l.]: John Wiley & Sons, 2007.
- KLEE, V.; MINTY, G. J. **How good is the simplex algorithm**. [S.l.], 1970.
- LABRADOR, M. A.; WIGHTMAN, P. M. **Topology control in wireless sensor networks**. [S.l.]: Springer, 2009.
- LOUREIRO, A. A.; NOGUEIRA, J. M. S.; RUIZ, L. B.; MINI, R. A. d. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S. Redes de sensores sem fio. In: **Simpósio Brasileiro de Redes de Computadores (SBRC)**. [S.l.: s.n.], 2003. p. 179–226.
- MAIA, E. H. B.; CÂMARA, D.; LOUREIRO, A. A. F. Ica: Um novo algoritmo de roteamento para redes de sensores. 2004.
- MAKHORIN, A. **GLPK (GNU linear programming kit)**. 2008.
- MURUGANATHAN, S. D.; MA, D. C.; BHASIN, R.; FAPOJUWO, A. O. et al. A centralized energy-efficient routing protocol for wireless sensor networks. **Communications Magazine, IEEE**, IEEE, v. 43, n. 3, p. S8–13, 2005.
- ORLIN, J. B. A faster strongly polynomial minimum cost flow algorithm. **Operations research, INFORMS**, v. 41, n. 2, p. 338–350, 1993.
- PATEL, M.; VENKATESON, S.; CHANDRASEKARAN, R. Energy-efficient capacity-constrained routing in wireless sensor networks. **International Journal of Pervasive Computing and Communications**, Emerald Group Publishing Limited, v. 2, n. 2, p. 69–80, 2007.
- RÊGO, A. da S. **Um algoritmo de roteamento bio-inspirado para economia de energia em redes de sensores sem fio**. Dissertação (Mestrado) — Universidade Estadual do Ceará, 2011.
- REGO, A. da S.; CELESTINO, J.; SANTOS, A. dos; CERQUEIRA, E. C.; PATEL, A.; TAGHAVI, M. Bee-c: a bio-inspired energy efficient cluster-based algorithm for data continuous dissemination in wireless sensor networks. In: **IEEE. Networks (ICON), 2012 18th IEEE International Conference on**. [S.l.], 2012. p. 405–410.
- RUIZ, L. B.; CORREIA, L. H. A.; VIEIRA, L. F. M.; MACEDO, D. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M.; VIEIRA, M. A. M.; BECHELANE, E. H.; CAMARA, D.; LOUREIRO, A. A. Arquiteturas para redes de sensores sem fio. In: **Simpósio Brasileiro de Redes de Computadores (SBRC)**. [S.l.: s.n.], 2004. v. 22, p. 167–218.
- SINGH, S. K.; SINGH, M.; SINGH, D. Routing protocols in wireless sensor networks—a survey. **International Journal of Computer Science & Engineering Survey (IJCSSES) Vol**, v. 1, p. 63–83, 2010.
- SINGH, S. K.; SINGH, M.; SINGH, D. A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks. **International Journal of Advanced Networking and Application (IJANA)**, Citeseer, v. 2, n. 02, p. 570–580, 2010.

TILAK, S.; ABU-GHAZALEH, N. B.; HEINZELMAN, W. A taxonomy of wireless micro-sensor network models. **ACM SIGMOBILE Mobile Computing and Communications Review**, ACM, v. 6, n. 2, p. 28–36, 2002.

VARGA, A. Omnet++ user manual. **Disponível na Internet: <http://www.omnetpp.org/doc/manual/usman.html>. Último acesso em Setembro de, 2007.**

VIJAY, U.; GUPTA, N. Clustering in wsn based on minimum spanning tree using divide and conquer approach. 2013.

WILLIAMS, T.; HECKING, L. Gnuplot. 2003.