**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**
**CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE**
**DEPARTAMENTO DE COMPUTAÇÃO**

**SILAS TURINI MACHADO DE SOUZA**

# IMPLEMENTAÇÃO E COMPARAÇÃO DE ALGORITMOS DE ROTEAMENTO COM BALANCEAMENTO DE CONSUMO DE ENERGIA

ALEGRE - ES
2016

**SILAS TURINI MACHADO DE SOUZA**

# IMPLEMENTAÇÃO E COMPARAÇÃO DE ALGORITMOS DE ROTEAMENTO COM BALANCEAMENTO DE CONSUMO DE ENERGIA

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. M.Sc. Helder de Amorim Mendes.

ALEGRE - ES

2016

SILAS TURINI MACHADO DE SOUZA

# IMPLEMENTAÇÃO E COMPARAÇÃO DE ALGORITMOS DE ROTEAMENTO COM BALANCEAMENTO DE CONSUMO DE ENERGIA

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 28 de novembro de 2016.

COMISSÃO EXAMINADORA

Prof. M.Sc. Helder de Amorim Mendes
Universidade Federal do Espírito Santo
Orientador

Prof. M.Sc. Edmar Hell Kampke
Universidade Federal do Espírito Santo

Prof. M.Sc. Rodrigo Freitas Silva
Universidade Federal do Espírito Santo

## DEDICATÓRIA

*“As dificuldades são como as montanhas. Elas só se aplainam quando avançamos sobre elas. ”*

Provérbio japonês

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me permitido avançar sobre as dificuldades e momentos que achei que seriam impossíveis.

Ao apoio da minha família, especialmente minha avó Marina Rosa que sempre acreditou em mim e me incentivou a ser uma pessoa melhor.

O Esforço dos meus pais em me manter estudando por todo esse tempo.

As minhas amizades Raphael Carletti, Caique Oliveira e Melissa Silva que me ajudaram em difíceis noites de estudo. E meus amigos Walace Newton, Yohan Salvador, Lenon Alonso e Ikaro James por terem se preocupado e cuidado de mim por todo esse tempo.

Aos Professores do departamento de computação que transmitiram a mim seus conhecimentos, e principalmente a meu orientador Helder de Amorim Mendes pela paciência e orientação.

A empresa Cinética Junior por permitir que eu aplicasse meus conhecimentos adquiridos ao longo do curso, e me proporcionar a oportunidade de conhecer o funcionamento do mercado de trabalho.

**RESUMO**

Redes de Sensores Sem Fio (RSSF) são compostas por sensores (nós) com restrições de recursos. Os nós são equipados com baterias tornando então o fator energético preocupante. Os eventos monitorados pelos nós acabam gerando um fluxo de informações a serem transmitidas e diversos protocolos são aplicados pela comunidade acadêmica com o intuito de diminuir a energia gasta pela rede, assim evitando que a energia dos nós se esgote, dessa forma aumentando o tempo de vida da rede. Este trabalho tem como objetivo a aplicação e comparação de algoritmos de roteamento para rede de sensores sem fio, com o foco em minimizar o consumo de energia dos sensores da rede. Será realizada a comparação entre os protocolos LEACH (Low - Energy Adaptive Clustering Hierarchy) e HISPEAR (High Speed Aggregation and Routing) usando o simulador Sinalgo.

Palavras chave: Redes de Sensores Sem Fio, Low - Energy Adaptive Clustering Hierarchy, High Speed Aggregation and Routing.

**ABSTRACT**

Wireless Sensor Networks (WSNs) are composed of sensors (nodes) with resource constraints. The nodes are equipped with batteries thus making the energy factor worrying. The events monitored by the nodes end up generating a flow of information to be transmitted and several protocols are applied by the academic community in order to increase the life time of the network. This work aims to the application and comparison of routing algorithms for wireless sensor network, with the focus on minimizing the power consumption of the network sensors. A comparison between LEACH (Low - Energy Adaptive Clustering Hierarchy) and HISPEAR (High Speed Aggregation and Routing) protocols will be performed using the Sinalgo simulator.

Keywords: Wireless Sensor Networks, Low - Energy Adaptive Clustering Hierarchy, High Speed Aggregation and Routing.

## LISTA DE FIGURAS

Figura 1 – Rede de Sensores Sem Fio ..... 12
Figura 2 – Estrutura do nó sensor ..... 13
Figura 3– Pseudocódigo de roteamento do algoritmo LEACH ..... 21
Figura 4 – Pseudocódigo de roteamento do algoritmo HISPEAR ..... 23
Figura 5 – Seleção de algoritmo implementado ..... 26
Figura 6 – Seleção do arquivo de coordenadas dos nós ..... 27
Figura 7 – Rede montada para simulação ..... 28
Figura 8 – Energia após a simulação com estação em X = 394.1, Y = 138.2 ..... 30
Figura 9 - Energia após a simulação com estação em X = 893.9, Y = 369.4 ..... 31
Figura 10 - Energia após a simulação com estação em X = 855.3, Y = 956.1 ..... 32

## LISTA DE TABELAS

Tabela 1– Cenários de simulação. ..... 24
Tabela 2 – Simulações com a estação base em X = 394.1, Y = 138.2. ..... 30
Tabela 3 – Simulações com a estação base em X = 893.9, Y = 369.4. ..... 31
Tabela 4 – Simulações com a estação base em X = 855.3, Y = 956.1. ..... 32

## LISTA DE ABREVIATURAS E SIGLAS

DESRO - *Dynamic Energy Saving Routing*

HISPEAR - *Wireless High Speed Routing*

LEACH - *Low-Energy Adaptive Clustering Hierarchy*

NS - *Network Simulator*

OMNET++ - *Objective Modular Network Testbed in C++*

RSSF - Rede de Sensores Sem Fio

Sinalgo - *Simulator for Network Algorithms*

TDMA - *Time Division Multiple Access*

**SUMÁRO**

1 INTRODUÇÃO ..... 12
1.1 O PROBLEMA E SUA IMPORTÂNCIA ..... 14
1.2 OBJETIVOS ..... 15
1.2.1 Objetivo geral ..... 15
1.2.2 Objetivos específicos ..... 15
1.3 ESTRUTURA DO TRABALHO ..... 15
2 REVISÃO DE LITERATURA ..... 16
2.1 SIMULAÇÃO DE REDES DE SENSORES ..... 18
3 METODOLOGIA ..... 19
3.1 IMPLEMENTAÇÃO ..... 19
3.2 LEACH ..... 20
3.2.1 Fase de anúncio ..... 20
3.2.2 Fase de início do grupo ..... 20
3.2.3 Escalonamento ..... 20
3.2.4 Fase de transmissão ..... 20
3.3 HISPEAR ..... 22
3.4 CENÁRIOS ..... 24
3.5 SIMULAÇÃO ..... 26
4 RESULTADOS ..... 29
5 CONCLUSÕES ..... 34
6 REFERÊNCIAS ..... 35

# 1 INTRODUÇÃO

Uma Rede de Sensores Sem Fio (RSSF) pode ser caracterizada pela alocação de pequenos dispositivos chamados nós sensores como visto na Figura 1, os quais com a capacidade de monitorar o ambiente que se encontram. Podendo também enviar informações para um ou mais nós utilizando a comunicação sem fio (TUBIELLO NETO, 2015).

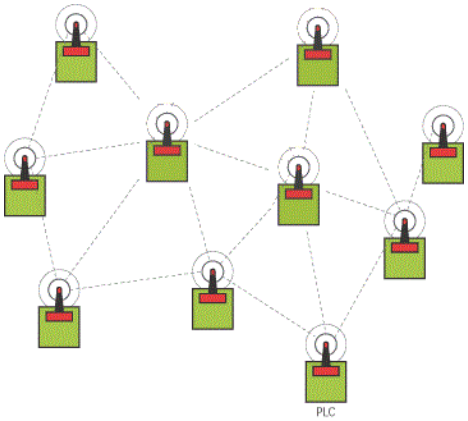


**Figura 1** – Rede de Sensores Sem Fio

Toda rede de sensores possui uma ou mais estações base, também conhecido como nó *sink* ou nó sorvedouro, sendo este, o responsável por receber todas as informações de outros nós sensores da rede, e repassá-las a uma rede externa ou para uma aplicação através de um link de internet (OLIVEIRA, 2011).

Cada sensor possui características bastante peculiares em relação à estrutura física, formada por outros pequenos dispositivos que possuem a capacidade de sensoriamento, processamento, atuação e comunicação como mostra a Figura 2. Cada sensor é sustentado por uma bateria, utilizada para alimentar todos os outros dispositivos. (NAKAMURA, 2003).

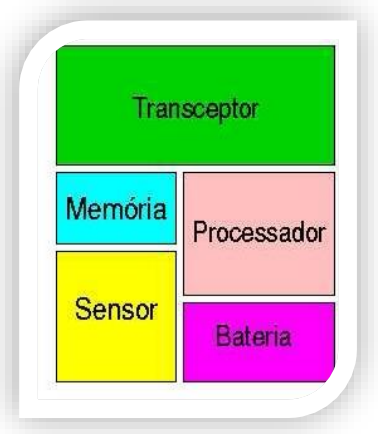


**Figura 2** – Estrutura do nó sensor

Fonte: Santos, Guimarães e Barbosa (2016)

Nós sensores possuem recursos limitados, como bateria e processamento, além de possuírem custos e tamanho reduzidos. Técnicas que visam minimizar o consumo de energia em RSSF, ou até mesmo um nó sensor, vêm sendo objetivo de pesquisas, uma vez que o fator energético representa um desafio. Outro aspecto é a maximização da vida útil da rede que pode ser obtida com a melhor gestão do trafego ou a redução da transição de pacotes entre os nós sensores (TUBIELLONETO, 2015).

## 1.1 O PROBLEMA E SUA IMPORTÂNCIA

Segundo Teixeira (2005), um dos mais importantes problemas encontrados nas redes de sensores é o consumo de energia, tendo em vista que sensores possuem baterias como fonte de alimentação.

Devido às circunstâncias das restrições de um nó sensor, principalmente quanto à quantidade de energia para transmitir informações, surgiu então a necessidade da aplicação de algoritmos de roteamento para encaminhar os dados coletados pelos sensores para um ou mais nós de maneira a economizar o máximo da energia da rede (TUBIELLO NETO, 2015).

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

Este trabalho tem como objetivo principal a implementação e comparação de dois protocolos de roteamento LEACH e HISPEAR.

### 1.2.2 Objetivos específicos

Os objetivos específicos são apresentados a seguir.

a) Estruturar a rede no simulador Sinalgo.
b) Implementar o protocolo LEACH.
c) Aplicar o protocolo LEACH a rede.
d) Realizar a simulação da rede com o simulador Sinalgo.
e) Implementar o protocolo HISPEAR.
f) Aplicar o protocolo HISPEAR a rede.
g) Realizar a simulação da rede com o simulador Sinalgo.
h) Avaliar o desempenho e comparação dos dois protocolos em relação a recursos energéticos.

## 1.3 ESTRUTURA DO TRABALHO

O restante deste trabalho está organizado da seguinte forma. No Capítulo 2 são apresentadas outras abordagens encontradas na literatura para solucionar o Problema Energético. O Capítulo 3 define como será a metodologia para realização deste trabalho. Os resultados obtidos estão descritos no Capítulo 4. Por fim, no Capítulo 5 são apresentadas as conclusões.

# 2 REVISÃO DE LITERATURA

Segundo Silva (2006) as redes de sensores eram motivadas inicialmente para aplicações militares, como por exemplo, na detecção de submarinos através de sensores acústicos. O avanço tecnológico proporcionou então a criação de sensores de baixo custo, tornando as redes de sensores uma ferramenta com potencial a outras aplicações.

As RSSF vêm sendo solução para diversos problemas de monitoramento, por exemplo, na área ambiental e agrícola. Problemas relacionados a áreas agrícolas são abordados por Torre-Neto e Guini (2011): enquanto ambientes industriais de petróleo e gás é abordado por Silva (2006).

Segundo Silva (2006) nós sensores possuem recursos limitados, como bateria. Alguns nós sensores podem falhar, serem bloqueados devido ao baixo nível da bateria, podem ser danificados fisicamente e ainda sofrerem interferência do ambiente em que estejam interagindo.

Problemas energéticos na rede de sensores são conhecidos pela sua dificuldade de tratamento onde, diferentes algoritmos são implementados e testados para escolher qual melhor se adapta ao problema (TAVARES, 2002).

Um dos trabalhos precursores que tratam do balanceamento de consumo de energia para redes de sensores sem fio é o de Teixeira (2005), onde foi utilizado o protocolo difusão direcionada. O mecanismo proposto estende o tempo de sobrevivência das RSSF através da descoberta de novos caminhos pela rede, distribuindo o gasto de energia.

Dantas (2010) abordou o problema energético aplicando o protocolo DESRO evitando que a energia do nó sensor se esgote realizando o gerenciamento de rotas.

Fassio (2013) utilizou um protocolo de múltiplas camadas envolvendo roteamento e controle de acesso ao meio, funções exercidas pelas camadas de rede e enlace respectivamente, com o intuito de manter alta eficiência energética e baixa latência.
Bastos (2011) trata a questão energética aplicando o protocolo NSINK originado do protocolo DESRO, permitindo então a utilização de mais de uma estação base e a realização do gerenciamento de rotas de forma dinâmica em uma rede de sensores sem fio, dividindo a sobrecarga para os nós sensores vizinhos a duas estações base.

Cavalcanti (2006) defende o uso do protocolo LEACH para a resposta ao problema energético, onde o mesmo é um protocolo de roteamento hierárquico auto organizável para redes de sensores sem fio. A principal característica desse algoritmo é que os nós possuem uma visão limitada da rede. Sendo assim, os nós se organizam em grupos locais conhecidos como clusters e elegem um nó para ser a estação base local, esse nó é chamado de *cluster head*. Para que a energia desse nó escolhido não esgote, devido ao alto grau de transmissões que deverá fazer, inclui-se um revezamento de *cluster head*, assim quando o líder está com pouca energia é executada novamente a eleição de um novo *cluster head.*

Balico (2013) aborda o uso do algoritmo HISPEAR, cuja ideia principal é encaminhar dados da rede de sensores a estação base em movimento, fazendo com que a rede não crie tráfego desnecessário, mas apenas quando for feita uma solicitação pela estação base. Nesse modelo de algoritmo de roteamento, assim que a estação base entra na rede, ela dissemina por toda a rede um pacote de solicitação de informações com sua localização. Com a informação da localização e velocidade da estação base, calcula-se onde ele irá passar, assim efetua-se o roteamento nessa direção de maneira a interceptá-lo. O nó com informações monta um pacote e as envia a estação base. A cada salto a caminho da estação base, os nós podem olhar o conteúdo do pacote, adicionar as suas informações e assim encaminhar um único pacote pela rede.

## 2.1 SIMULAÇÃO DE REDES DE SENSORES

A simulação através do computador é uma ferramenta trivial para a construção de sistemas complexos ou com custos elevados, possibilitando a ajuda na avaliação e na solução de problemas do mundo real. Como no caso das redes de sensores sem fio, em que se trata de um número muito grande nós sensores, tornando necessário trabalhar com restrições energéticas e falha de comunicação entre os nós, o uso de simuladores de algoritmos de rede como NS-2, NS-3, Omnet++ e Sinalgo torna-se indispensável para se planejar um bom funcionamento da rede (NOGUEIRA, 2006).

Sinalgo é um framework de simulação para validação e testes de algoritmos de redes. O Sinalgo foca sua verificação aos algoritmos, abstraindo-se das camadas mais baixas. Portanto, pode-se focar na análise proposta, desconsiderando detalhes secundários na classificação de rotas. Algumas das principais características do Sinalgo são: Prototipagem rápida de seus algoritmos de rede em JAVA, Plug-ins ajustáveis embutidos, alto desempenho - executar simulações com 100.000 nós em um tempo aceitável, suporte para 2D e 3D e simulação síncrona e assíncrona (WATTENHOFER, 2016).

# 3 METODOLOGIA

A metodologia abordada neste trabalho consiste na implementação e aplicação de dois algoritmos de roteamento, LEACH e HISPEAR para uma RSSF. Os dados utilizados para criação da rede foram gerados de maneira randômica a partir do próprio simulador. Após essa etapa Inicial, foram implementados os algoritmos LEACH e HISPEAR. Por fim, os algoritmos implementados foram aplicados as redes geradas, e seus resultados comparados. Os algoritmos propostos foram implementados na linguagem de programação Java, usando a IDE Eclipse Neon.1a(4.6.1) e simulados no framework de simulação Sinalgo(0.75.3)

## 3.1 IMPLEMENTAÇÃO

No framework Sinalgo, a Implementação de um algoritmo é realizada, copiando a pasta “template” localizada em “src/projects”, sendo apenas uma estrutura básica pré-definida. Para implementar o comportamento de um nó, deve-se criar uma classe que herde o comportamento da classe “sinalgo.nodes.Node” e salvar o arquivo dentro da pasta “node/nodeImplementation” no projeto.

Para a implementação do algoritmo LEACH realiza-se a implementação de dois tipos de nós, a Estação base, chamada de “EstacaoBaseNode” e o nó monitor chamado de “LeachNode”, onde ambos estendem o comportamento da classe “sinalgo.nodes.Node” . O algoritmo HISPEAR também foi implementado da mesma forma, estendendo o comportamento das mesmas classes, tanto para a estação base (EstacaoBaseNode), quanto para o nó monitor (NormalNode).

Obrigatoriamente, implementa-se o método *handleMessages*() e, opcionalmente, qualquer um dos métodos abstratos da classe sinalgo.nodes.Node. Esse método implementado neste trabalho, consiste em tratar as mensagens que o nó recebe de maneira a obter o identificador do pacote, o tempo de vida do pacote, o nó de destino, o no de origem, o número de saltos até o destino e o estabelecimento da rota do pacote. Cada nó instanciado armazena as mensagens que recebe em uma instância da classe *Inbox*, onde a mesma provê um *iterator* para acessar cada mensagem armazenada.

## 3.2 LEACH

LEACH é um protocolo de roteamento hierárquico auto organizável para redes de sensores sem fio. Sendo assim, os nós se organizam em grupos locais conhecido como clusters e elegem um nó para ser a estação base local, chamado de *cluster head*. O nó escolhido para ser o *cluster head* é o nó com maior quantidade de energia dentre os nós do grupo. Para melhor compreensão, o algoritmo pode ser dividido em 4 fases como mostra em seu pseudocódigo nas linhas 6, 10, 11 e 13. A Figura 3 apresenta seu pseudocódigo.

### 3.2.1 Fase de anúncio

Quando os clusters estão sendo criados, cada nó é analisado para determinar qual será o *cluster head*. Após o nó ser eleito ele envia uma mensagem de anuncio a todos os nós informando que ele é o *cluster head* e, em seguida os nós que não são *cluster head* decidem se pertencem ou não àquele grupo.

### 3.2.2 Fase de início do grupo

Após a decisão dos nós sobre a qual grupo eles pertencem, eles informam ao seu *cluster head* que será um membro desse *cluster*.

### 3.2.3 Escalonamento

O *cluster head* recebe então todas as informações dos nós que querem participar do grupo e então cria um escalonamento TDMA, informando a cada nó do grupo quando ele pode transmitir informações.

### 3.2.4 Fase de transmissão

Baseado no escalonamento, os nós do *cluster* transmitem a informação coletada para o *cluster head* que, por sua vez, após receber a informação de todos os nós, processa e envia estas informações para a estação base.

```
1.  VARIAVEIS
2.  BOOLEAN: redeAtiva  TRUE
3.  INICIO
4.  ENQUANTO redeAtiva
5.          FAÇA
                    //FASE DE ANUNCIO
6.          ARMAZENAR estatísticas da rodada anterior
7.          ELIMINAR os nós sensores sem energia
8.          ELEGER os cluster heads da rodada atual de cada cluster
9.          ENVIAR mensagem a todos os nós informando que ele é o cluster head

                    //FASE DE INÍCIO DO GRUPO
10.         ENVIAR mensagem ao cluster head, informando que pertencem ao cluster

                    //ESCALONAMENTO
11.         RECEBER mensagem dos participantes, informando que pertencem ao cluster
12.         CRIAR escalonamento TDMA

                    //FASE DE TRANSMISSÃO
13.         TRANSMITIR informações coletadas ao cluster head
14.         RECEBER mensagem dos participantes do cluster, com informações coletadas
15.         ENVIAR mensagem com informações coletadas a estação base
16.         SE (conectividade com estação base for zero)
17.         ENTAO
18.         redeAtiva  FALSE
19.         FIMSE
20. FIMENQUANTO
21. ESCREVA "Rede Inativa"
22. FIM
```

Figura 3– Pseudocódigo de roteamento do algoritmo LEACH

## 3.3 HISPEAR

HISPEAR é um protocolo de roteamento para rede de sensores sem fio, cujo pseudocódigo está apresentado na Figura 4. Neste algoritmo a estação base envia um pacote de consulta para a rede, onde a mesma, é disseminada em *flooding*, ou seja, cada vez que um nó recebe a consulta, se ele próprio não for o destino da comunicação, ele repassa o pacote para todos os nós que está ligado diretamente a ele, menos para o nó por onde recebeu o pacote, assim garantindo que a consulta vai chegar a todos os nós da rede. Ao receber essa consulta, os nós com informações montam um pacote e o enviam para a estação base. Para encaminhar esse pacote até a estação base, realiza-se a transmissão ao seu nó vizinho mais próximo da estação base, e assim sucessivamente, já que todos os nós conhecem sua localização, pois a mesma foi passada dentro do pacote de solicitação. Para cada nó onde a informação passa, o próprio nó pode abrir e ver o conteúdo do pacote, adicionar mais conteúdo e, assim, encaminhar um único pacote com informações pela rede até a estação base.

```
1.  VARIAVEIS
2.  BOOLEAN: redeAtiva  TRUE
3.  INICIO
4.  ENQUANTO redeAtiva
5.          FAÇA
6.          ELIMINAR os nós sensores sem energia
7.          ENVIAR consulta a rede
8.          DISSEMINAR a consulta em flooding por toda rede
9.          FORMAR pacote com informação de monitoramento
10.         ENVIAR pacote a estação base
11.         SE (conectividade com aestação base for zero)
12.         ENTAO
13.         redeAtiva  FALSE
14.         FIMSE
15. FIMENQUANTO
16. ESCREVA "Rede Inativa"
17. FIM
```

Figura 4 – Pseudocódigo de roteamento do algoritmo HISPEAR

## 3.4 CENÁRIOS

Foram elaborados cenários combinando-se diferentes quantidades de nós sensores, diferentes posições da estação base e diferentes quantidades de eventos. A Tabela 1 apresenta todos os cenários obtidos, sendo que para todos a área possui 1000 metros de largura por 1000 de comprimento.

Tabela 1– Cenários de simulação.

| Num. Sensores | Quant. Eventos | Pos. Estação Base |
|---|---|---|
| 20 | 3 | X = 394.1, Y = 138.2 |
| 20 | 3 | X = 893.9, Y = 369.4 |
| 20 | 3 | X = 855.3, Y = 956.1 |
| 20 | 5 | X = 394.1, Y = 138.2 |
| 20 | 5 | X = 893.9, Y = 369.4 |
| 20 | 5 | X = 855.3, Y = 956.1 |
| 20 | 10 | X = 394.1, Y = 138.2 |
| 20 | 10 | X = 893.9, Y = 369.4 |
| 20 | 10 | X = 855.3, Y = 956.1 |
| 40 | 3 | X = 394.1, Y = 138.2 |
| 40 | 3 | X = 893.9, Y = 369.4 |
| 40 | 3 | X = 855.3, Y = 956.1 |
| 40 | 5 | X = 394.1, Y = 138.2 |
| 40 | 5 | X = 893.9, Y = 369.4 |
| 40 | 5 | X = 855.3, Y = 956.1 |
| 40 | 10 | X = 394.1, Y = 138.2 |
| 40 | 10 | X = 893.9, Y = 369.4 |
| 40 | 10 | X = 855.3, Y = 956.1 |
| 80 | 3 | X = 394.1, Y = 138.2 |
| 80 | 3 | X = 893.9, Y = 369.4 |
| 80 | 3 | X = 855.3, Y = 956.1 |
| 80 | 5 | X = 394.1, Y = 138.2 |
| 80 | 5 | X = 893.9, Y = 369.4 |
| 80 | 5 | X = 855.3, Y = 956.1 |

| | | |
|---|---|---|
| 80 | 10 | X = 394.1, Y = 138.2 |
| 80 | 10 | X = 893.9, Y = 369.4 |
| 80 | 10 | X = 855.3, Y = 956.1 |
| 100 | 3 | X = 394.1, Y = 138.2 |
| 100 | 3 | X = 893.9, Y = 369.4 |
| 100 | 3 | X = 855.3, Y = 956.1 |
| 100 | 5 | X = 394.1, Y = 138.2 |
| 100 | 5 | X = 893.9, Y = 369.4 |
| 100 | 5 | X = 855.3, Y = 956.1 |
| 100 | 10 | X = 394.1, Y = 138.2 |
| 100 | 10 | X = 893.9, Y = 369.4 |
| 100 | 10 | X = 855.3, Y = 956.1 |

## 3.5 SIMULAÇÃO

Para realizar a simulação da rede, inicialmente foi escolhido um dos algoritmos implementados no simulador como mostra a Figura 5. Após a escolha do algoritmo foi possível então, visualizar o campo onde os nós são adicionados para a simulação. Para adicionar os nós em coordenadas específicas, seleciona-se as opções “Gerar Nodes” e “PositionFile”, especificando o arquivo com a posição dos nós como apresenta a Figura 6, também é necessário selecionar e a quantidade e o tipo de nó que deseja-se adicionar a simulação. A figura 7 apresenta a rede montada pronta para simulação.

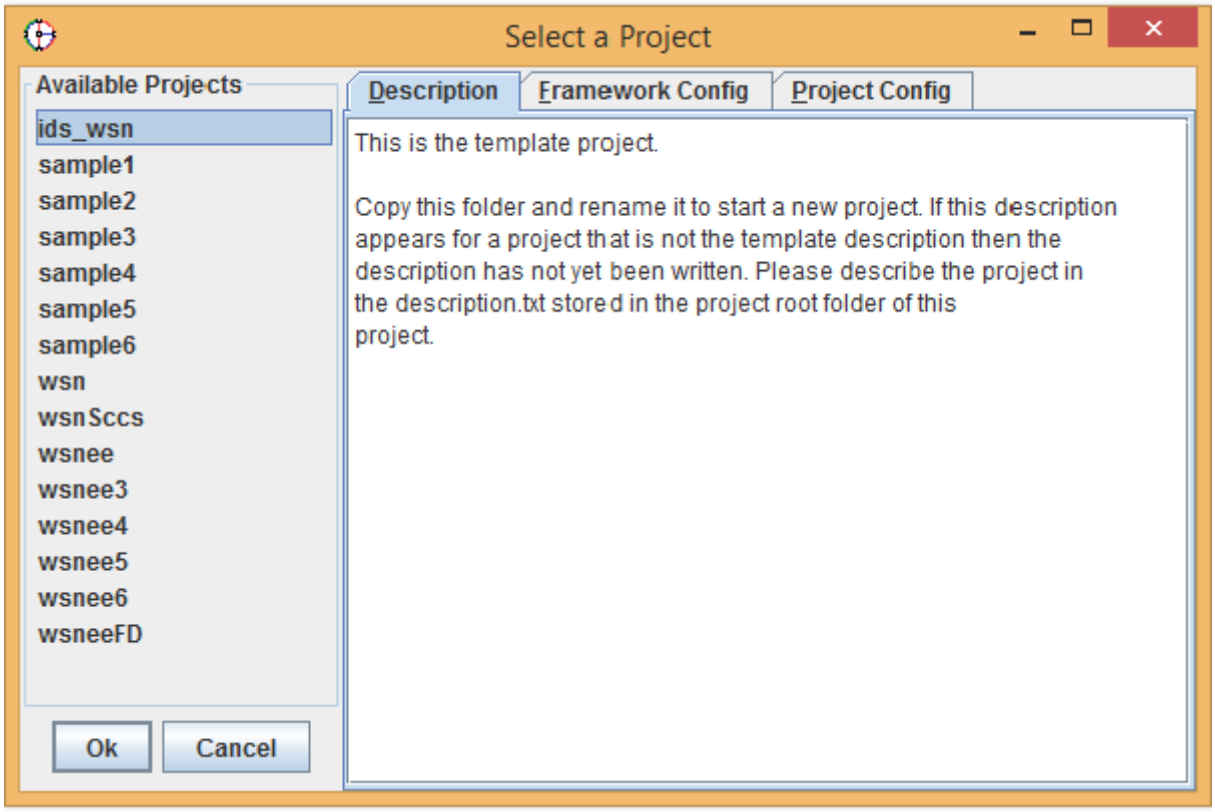


Figura 5 – Seleção de algoritmo implementado

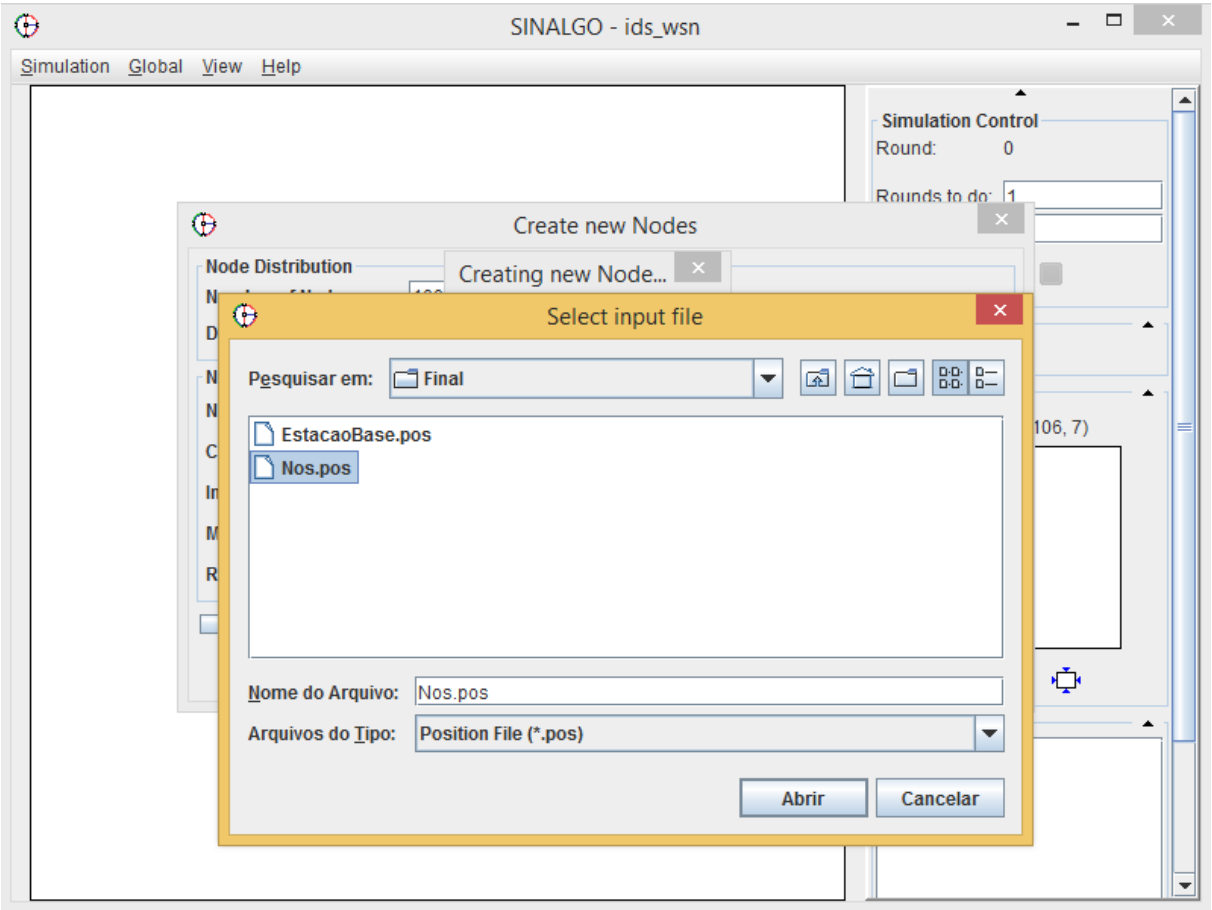


Figura 6 – Seleção do arquivo de coordenadas dos nós

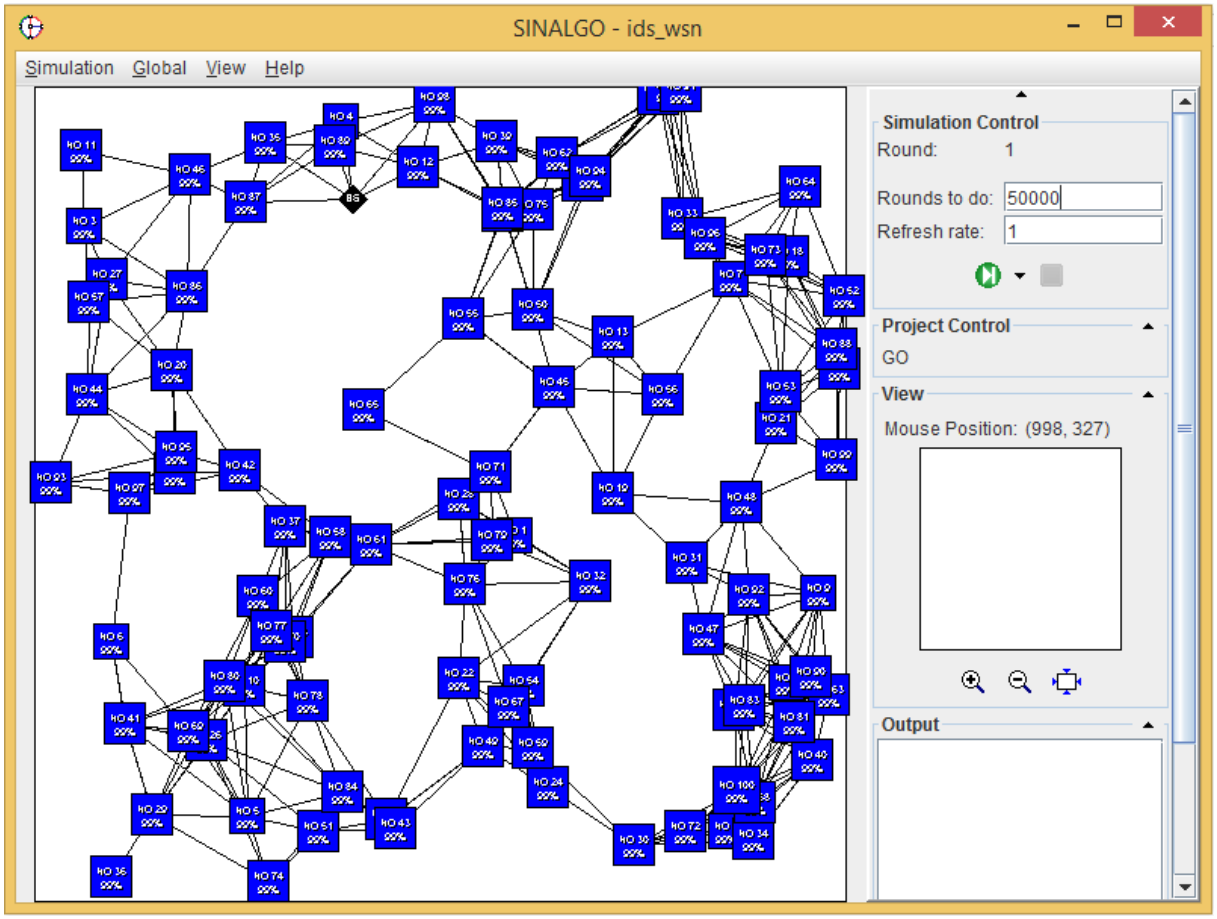


Figura 7 – Rede montada para simulação

# 4 RESULTADOS

Após a adição dos nós, entra-se com a quantidade de rodadas no campo “Rounds to do” que irão ser executadas na simulação, como mostra a Figura 7. Sendo que, para este trabalho, a quantidade de rodadas foi fixada em 50.000. No primeiro conjunto de simulações, considerou-se os cenários com a estação base localizada em X = 394.1, Y = 138.2. Foram realizadas simulações a ocorrência de 3, 5 e 10 eventos. Na sequência foram simulados os cenários para as mesmas quantidades de eventos, mas alterando a posição da estação base para as coordenadas X = 893.9, Y = 369.4 e X = 855.3, Y = 956.1.

Ao final das simulações foi feito o registro do total de energia gasto na simulação e o cálculo do total de energia restante para cada nó.

Nas Tabelas 2, 3 e 4 a seguir, são mostrados os resultados da simulação para a estação base nas posições S1(394.1,138.2), S2(893.9, 369.4) e S3(855.3, 956.1) de cada algoritmo, com a respectiva quantidade de nós, eventos e a porcentagem de energia que foi gasta na simulação. A coluna *Qtd. de Nós Sensores* indica a quantidade de nós sensores da rede, a coluna *Qtd. de Eventos* indica a quantidade de eventos da rede, as colunas *LEACH* e *HISPEAR* indicam o percentual de energia gasta após a simulação da rede em 50.000 rodadas. Destaca-se em negrito o melhor resultado dentre os dois algoritmos em cada simulação realizada.

Dos resultados da Tabela 2, verifica-se que o algoritmo LEACH foi superior em todas as simulações, com destaque ao cenário com 100 nós sensores e 10 eventos onde obteve seu melhor resultado. O algoritmo HISPEAR gastou 34,61% da energia da rede enquanto o LEACH gastou apenas 25,34%, proporcionando então uma economia de 9,27% de energia. A Figura 8 representa de maneira mais clara os níveis de energia da rede após a simulação dos algoritmos referente a Tabela 2, porem apenas para o cenário com 100 nós sensores.

Tabela 2 – Simulações com a estação base em X = 394.1, Y = 138.2.

| Qtd. de Nós Sensores | Qtd. de Eventos | LEACH | HISPEAR |
|---|---|---|---|
| 20 | 3 | **31,08** | 33,62 |
| 20 | 5 | **33,68** | 36,28 |
| 20 | 10 | **36,94** | 44,64 |
| 40 | 3 | **26,48** | 28,83 |
| 40 | 5 | **27,71** | 33,65 |
| 40 | 10 | **30,76** | 39,92 |
| 80 | 3 | **28,17** | 28,81 |
| 80 | 5 | **29,16** | 29,73 |
| 80 | 10 | **30,09** | 36,12 |
| 100 | 3 | **24,59** | 30,08 |
| 100 | 5 | **24,96** | 31,75 |
| 100 | 10 | **25,34** | 34,61 |

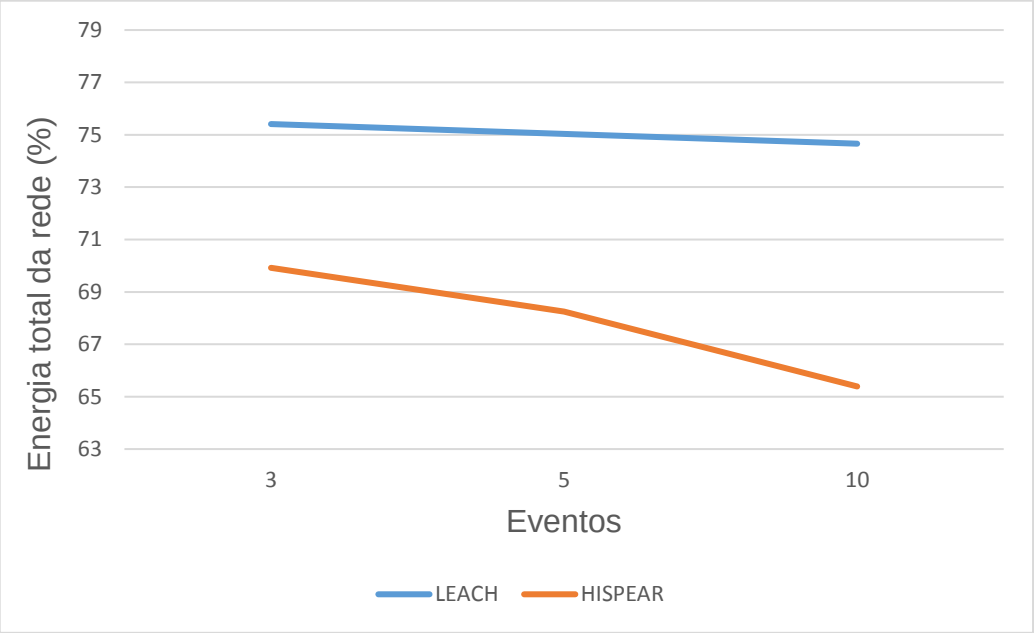


Figura 8 – Energia após a simulação com estação em X = 394.1, Y = 138.2

Para os resultados da Tabela 3, o algoritmo LEACH também foi superior ao HISPEAR em todas as simulações, com destaque ao cenário com 100 nós sensores e 10 eventos onde obteve seu melhor resultado. O algoritmo HISPEAR gastou 34,14% da energia da rede enquanto o LEACH gastou apenas 24,79%, proporcionando então uma economia de 9,35% de energia. A Figura 9 representa de maneira mais clara os níveis de energia da rede após a simulação dos algoritmos referente a Tabela 3, porem apenas para o cenário com 100 nós sensores.

Tabela 3 – Simulações com a estação base em X = 893.9, Y = 369.4.

| Qtd. de Nós Sensores | Qtd. de Eventos | LEACH | HISPEAR |
|---|---|---|---|
| 20 | 3 | **30,84** | 32,06 |
| 20 | 5 | **32,97** | 34,5 |
| 20 | 10 | **36,85** | 43,94 |
| 40 | 3 | **25,26** | 30 |
| 40 | 5 | **26,39** | 33,02 |
| 40 | 10 | **28,98** | 36,21 |
| 80 | 3 | **26,53** | 28,72 |
| 80 | 5 | **26,99** | 29,27 |
| 80 | 10 | **29,69** | 34,52 |
| 100 | 3 | **22,11** | 28,46 |
| 100 | 5 | **22,83** | 29,51 |
| 100 | 10 | **24,79** | 34,14 |

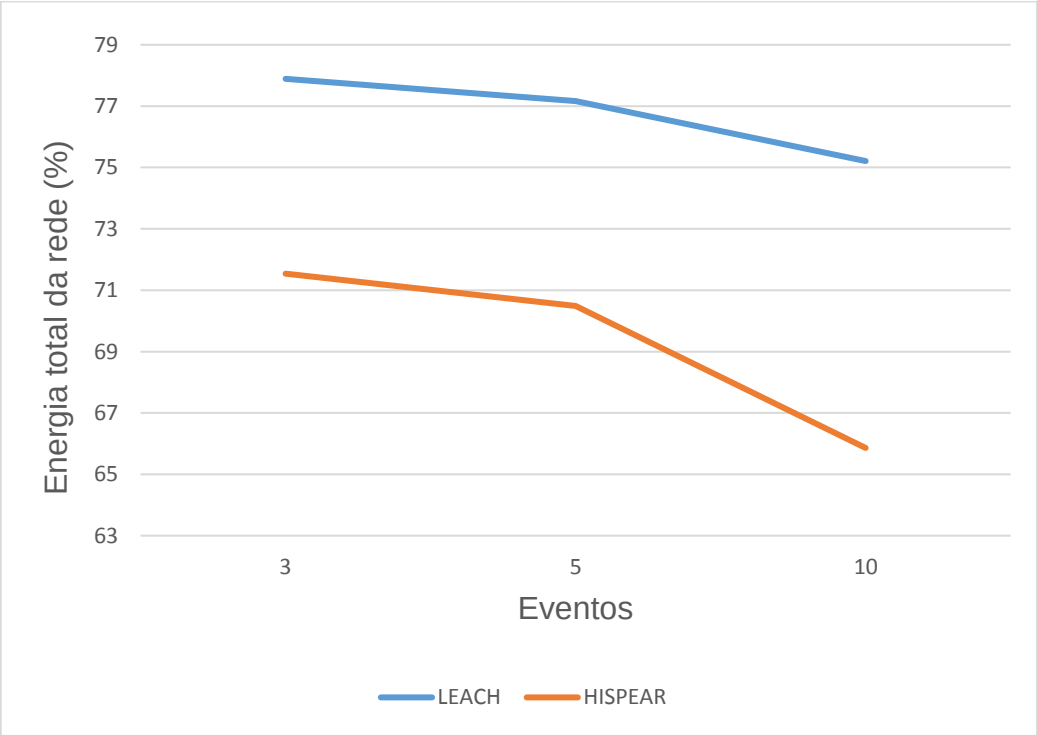


Figura 9 - Energia após a simulação com estação em X = 893.9, Y = 369.4

Na da Tabela 4, são mostrados os resultados da simulação, onde o algoritmo LEACH também foi superior ao HISPEAR em todas as simulações. Seu melhor resultado também foi sobre o cenário com 100 nós sensores e 10 eventos. O algoritmo HISPEAR gastou 32,96% da energia da rede enquanto o LEACH gastou apenas 25,18%, proporcionando então uma economia de 7,78% de energia. A Figura 10

representa de maneira mais clara os níveis de energia da rede após a simulação dos algoritmos referente a Tabela 4, porem apenas para o cenário com 100 nós sensores.

Tabela 4 – Simulações com a estação base em X = 855.3, Y = 956.1.

| Qtd. de Nós Sensores | Qtd. de Eventos | LEACH | HISPEAR |
|---|---|---|---|
| 20 | 3 | **28,59** | 30,55 |
| 20 | 5 | **31,21** | 33,03 |
| 20 | 10 | **35,76** | 39,94 |
| 40 | 3 | **26,93** | 31,85 |
| 40 | 5 | **28,76** | 34,25 |
| 40 | 10 | **32,06** | 40,02 |
| 80 | 3 | **26,61** | 28,3 |
| 80 | 5 | **27,14** | 29,77 |
| 80 | 10 | **30,33** | 34,82 |
| 100 | 3 | **22,98** | 26,65 |
| 100 | 5 | **24,11** | 29,32 |
| 100 | 10 | **25,18** | 32,96 |

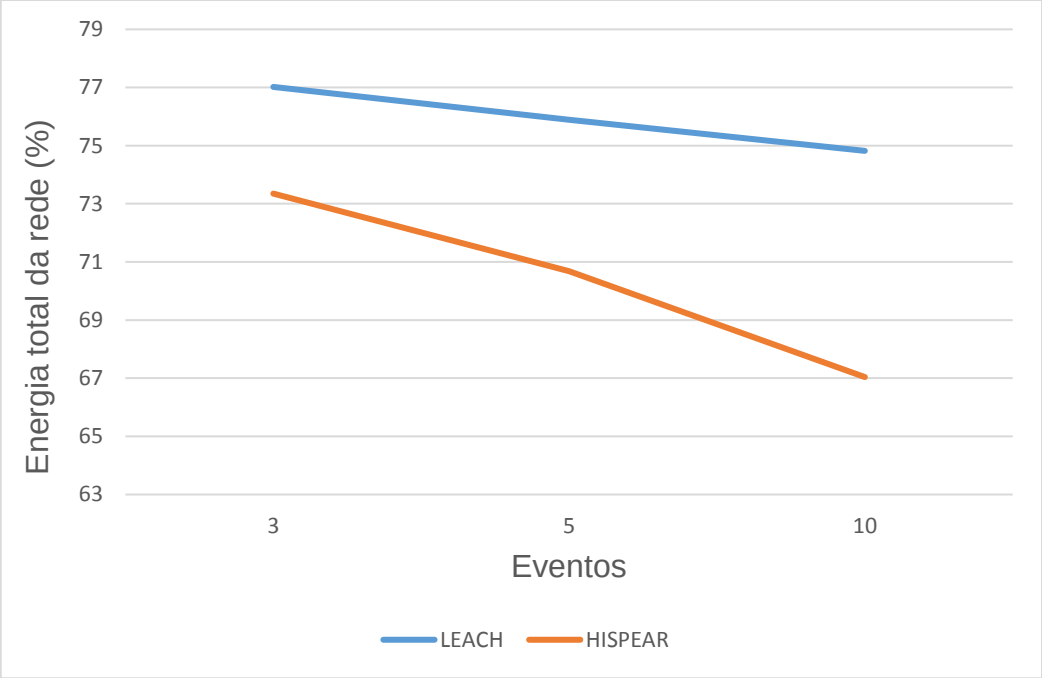


Figura 10 - Energia após a simulação com estação em X = 855.3, Y = 956.1

Pelos resultados apresentados nas simulações, percebe-se que, o algoritmo LEACH obteve o melhor resultado para todas as simulações, onde seu melhor resultado de

todas as simulações realizadas foi uma economia de 9,35% de energia para uma rede de 100 nós sensores e 10 eventos com a estação base localizada em X = 893.9, Y = 369.4. Essa superioridade em economia se deu devido ao algoritmo HISPEAR precisar disseminar um pacote com informações em *flooding* por toda a rede, ocasionando um gasto de energia elevado. Tornando então, o algoritmo LEACH mais eficiente em questões energéticas para uma RSSF.

# 5 CONCLUSÕES

Neste trabalho foi tratado o problema de consumo de energia para rede de sensores sem fio, sendo o principal objetivo a implementação e comparação de dois algoritmos de roteamento de forma a minimizar o consumo de energia da rede. Para solução foram utilizados dois algoritmos, LEACH e HISPEAR tendo como *framework* de simulação o Sinalgo.

Pode-se concluir por meios dos resultados obtidos no trabalho que: ao comparar o desempenho em relação a questões energéticas de cada algoritmo, afirma-se que o LEACH apresentou melhor desempenho, comparado com o algoritmo HISPEAR, proporcionando então uma maior economia de energia, que por consequência um aumento no tempo de vida da rede.

# 6 REFERÊNCIAS

BALICO, Leandro N. **Roteamento e agregação de dados usando sinks em alta velocidade em redes de sensores sem fio**. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Manaus, 2013.

BASTOS, Alex Vidigal. **NSINKS: protocolo de roteamento para redes de sensores sem fio em redes com vários sorvedouros**. 2011. 45 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Viçosa, Viçosa, 2011.

CAVALCANTI, Sávio Rodrigues. **LEACH - low - energy adaptive clustering hierarchy**. Rio de Janeiro: Grupo de Teleinformática e Automação, 2006. 5 p.

DANTAS, Arthur Vinícius Maciel. **DESRO: um protocolo de roteamento com gerenciamento dinâmico de energia para redes de sensores sem fio**. 2010. 51 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Viçosa, Viçosa, 2010.

FASSIO, Alexandre Victor**. Um protocolo de multiplas camadas para rede de sensores sem fio com baixo cunsumo de energia e latencia reduzida.** 2013. 55 f. Monografia (Especialização) - Curso de Ciência da Computação, Universidade Federal de Lavras, Lavras, 2013.

NAKAMURA, Fabíola Guerra. **Planejamento dinâmico para controle de cobertura e conectividade, em redes de sensores sem fio planas**. 2003. 61 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2003.

NOGUEIRA, Jose Marcos Silva. **Simulação de redes de sensores sem fio.** 2006. Disponível em: <http://www.dcc.ufmg.br/dcc/?q=pt-br/node/695>. Acesso em: 13 nov. 2016.

OLIVEIRA, Carlos Eder Ramos de**. Utilização de nó sorvedouro móvel para balancear a carga de energia em rede de sensores sem fio**. 2011. 69 f. Dissertação

(Mestrado) - Curso de Ciência da Computação, Universidade Federal de Viçosa, Viçosa, 2011.

SANTOS, Raphael Rocha; GUIMARÃES, Fernanda Rodrigues; BARBOSA, Raphael Ferreira. **Redes de Computadores I.** Disponível em: <http://www.gta.ufrj.br/grad/09_1/versao-final/rssf/de_nos_sensores.html>. Acesso em: 07 dez. 2016.

SILVA, Ivanovitch Medeiros Dantas da. **Redes de sensores sem fio aplicadas em ambientes industriais de petróleo e gás**. 2006. 83 f. Monografia (Especialização) - Curso de Engenharia de Computação, Universidade Federal do Rio Grande do Norte, Natal, 2006.

TAVARES, Pedro Lemos. **Redes de sensores sem-fio**. 2002. 26 f. Monografia (Especialização) - Curso de Engenharia Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002.

TEIXEIRA, Ingrid. **Roteamento com balanceamento de consumo de energia para redes de sensores sem fio**. 2005. 94 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

TORRE-NETO, André; GHINI, Raquel. **Rede de sensores sem fio para monitoramento e controle de processos em ambiente agrícola**. In: INAMASU, Ricardo Yassushi. Agricultura de Precisão: um novo olhar. São Paulo: Cubo, 2011. Cap. 11. p. 115-119.

TUBIELLO NETO, Francesco. **Técnica de otimização de energia para rssf.** 2015. 145 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2015.

WATTENHOFER, Roger. **Sinalgo - simulator for network algorithms.** 2016. Disponível em: <http://www.dcg.ethz.ch/projects/sinalgo/>. Acesso em: 13 nov. 2016.