Análise de estratégias de *multi-hop* em protocolos baseados em aprendizagem de máquina não supervisionada

Diego Lima Pinheiro[1], **Otávio Alcântara de Lima Júnior**[1]

[1]Laboratório de Sistemas Digitais – Instituto Federal do Ceará (IFCE)
Maracanaú – CE – Brasil

diego.lima.ifce@gmail.com, otavio@ifce.edu.br

Abstract. *Although promising, Wireless Sensor Networks (WSN) have energy limitations that need to be resolved. The LEACH protocol is one of the most classic ways to reduce the energy consumption caused by the transmission of packets on the network. The use of unsupervised machine learning is another alternative, which uses a clustering methodology, such as LEACH, to organize the network into subnets. It is still possible to further improve the energy efficiency of the nodes, through the use of multi-hop, which is a network packet forwarding technique that optimizes energy expenditure, in which, in a WSN, it is possible use two different strategies, intra-cluster and inter-cluster. This paper proposes the use of multi-hop in conjunction with protocols based on LEACH that use K-means or K-medoids, performing a comparative study between each of the possible variations of these approaches. The use of multi-hop in the K-means and K-medoids protocols generated an increase in the network lifetime of up to 24,48%.*

Resumo. *Apesar de promissoras, as Redes de Sensores Sem Fio (RSSF) possuem limitações energéticas que precisam ser resolvidas. O protocolo LEACH é uma das formas mais clássicas de reduzir o gasto de energia causado pela transmissão dos pacotes na rede. O uso do aprendizado de máquina não supervisionado é uma outra alternativa, que utiliza uma metodologia de clustering, assim como o LEACH, para organizar a rede em sub-redes. Ainda é possível melhorar ainda mais a eficiência energética dos nós, através do uso do multi-hop, que é uma técnica de encaminhamento de pacotes da rede que otimiza o gasto de energia, na qual, em uma RSSF, é possível utilizar duas estratégias distintas, o intra-cluster e o inter-cluster. Este artigo propõe a utilização do multi-hop em conjunto com protocolos baseados no LEACH que utilizam o K-means ou o K-medoids, realizando um estudo comparativo entre cada uma das variações possíveis dessas abordagens. O uso do multi-hop nos protocolos K-means e K-medoids gerou um aumento no tempo de vida da rede de até 24,48%.*

1. Introdução

Redes de Sensores sem Fio (RSSFs) são um conjunto de sensores distribuídos, conectados em rede, que monitoram fenômenos físicos e enviam os dados coletados ao usuário. Seu desenvolvimento contribuiu significativamente para o surgimento da Internet das Coisas (IoT), que explora o conceito de conectar os mais variados objetos em rede para que estes coletem e enviem dados [Atzori et al. 2010]. As RSSFs podem ser utilizadas nas

mais variadas funções relacionadas a monitoramento e rastreamento, como aplicações médicas, ambientais, militares, entre outras [Rawat et al. 2014].

As RSSFs são formadas por um conjunto de sensores autônomos, com capacidade de sensoreamento e processamento de informações, espacialmente distribuídos e são utilizadas para fazer monitoramento de fenômenos em uma área e transmitir as informações coletadas [Rawat et al. 2014]. Essas informações são enviadas para uma estação base, também chamada de nó sorvedouro, que envia informações para o usuário final pela internet. Os nós sensores que compõem uma RSSF são formados por interface de rádio, sensores, processador e fonte de energia. Esses sensores captam informações do ambiente, como temperatura, umidade, luminosidade, movimento, dependendo da necessidade da aplicação. O processador faz o tratamento dos dados coletados, gerencia o rádio, que faz o envio das informações, e controla o gasto de energia.

Apesar das vantagens e das diversas possibilidades, as RSSFs possuem vários desafios que devem ser levados em conta durante o planejamento das aplicações. Um dos principais é a limitação de recursos. Como os sensores possuem tamanho reduzido, sua capacidade computacional e de armazenamento de energia é escassa, impactando diretamente no tempo de vida da rede. É necessário, então, adotar soluções que utilizem esses recursos de forma apropriada. O rádio é o recurso responsável pela maior parte do consumo de energia [Martinez et al. 2015], devido, principalmente, às trocas de mensagens em grandes distâncias entre os nós sensores e a estação base. Logo, é fundamental pensar em protocolos de comunicação que otimizem o consumo, para que não haja desperdício de energia, nem atrasos na entrega.

Algumas abordagens têm sido propostas para melhorar o desempenho, a comunicação e o tempo de vida das RSSFs, como visto em [Al-Karaki and Kamal 2004] e [Singh et al. 2015]. Uma técnica amplamente adotada para protocolos de comunicação em RSSFs é o *clustering* [Zeb et al. 2016]. Seu funcionamento consiste em dividir a rede em redes menores (*clusters*), cada uma delas com um líder (*cluster-head*), para quem os nós enviam as mensagens e que têm a função de agrupar e enviar essas mensagens para a estação base. A utilização do *clustering* na comunicação proporciona diversas vantagens, como economia de energia, balanceamento de carga, fusão de dados, melhoria na entrega dos pacotes, dentre outros [Shahraki et al. 2020], além de maior tolerância a falhas. Todas essas características auxiliam para que o tempo de vida da rede aumente.

Existem vários algoritmos de *clustering* para RSSFs. Um dos principais deles é o LEACH (*Low-Energy Adaptive Clustering Hierarchy*) [Heinzelman et al. 2000]. O LEACH é um protocolo de *clustering* que visa reduzir o desperdício e balancear o consumo de energia através da rotação aleatória de *cluster-heads*. Por ser um dos protocolos de *clustering* pioneiro para RSSFs, o LEACH inspirou diversos outros protocolos, que o utilizaram como base e adicionaram modificações. Entre eles, podemos citar o LEACH-C (LEACH-centralizado) [Heinzelman et al. 2002], o TL-LEACH (Two-Levels LEACH) [Loscri et al. 2005] e o Energy-LEACH [Xiangning and Yulin 2007].

Também é possível utilizar outras técnicas que reduzem o consumo de energia em conjunto com o *clustering*. Uma técnica muito utilizada é o *multi-hop*, que consiste no encaminhamento de pacotes na rede, na qual as mensagens transmitidas pelos nós sensores, ao invés de serem transmitidas diretamente para o destino, que pode estar muito

distante e consumir muita energia, podem passar pelos outros nós da rede, que fazem o repasse dessa mensagem até o destino.

Este artigo faz um estudo das estratégias de *multi-hop*, de acordo com [Pinheiro et al. 2019], mas aplicadas a protocolos baseados em aprendizagem de máquina não supervisionada. Os algoritmos que serão utilizados nesse estudo são o K-means e o K-medoids. O restante do artigo se organiza da seguinte forma. A seguir, na seção 2, será visto o *background* do LEACH, dos protocolos de aprendizagem de máquina não supervisionada, e da técnica de *multi-hop*. Na seção 3 é apresentada a metodologia adotada. A seção 4 descreve os experimentos e seus resultados. Por fim, na seção 5, temos as conclusões.

2. Background

2.1. LEACH

O LEACH [Heinzelman et al. 2000] é um protocolo de *clustering* distribuído, de nós estáticos e homogêneos, com *clusters* iguais e encaminhamento de mensagens *single-hop*. Nele, os *cluster-heads* são escolhidos de forma probabilística e há uma rotação da função de *cluster-head*, para que o consumo de energia seja balanceado entre todos os nós da rede. Por ser um protocolo distribuído, o LEACH não precisa de informações centralizadas sobre a rede, o que diminui o *overhead* e, consequentemente, o consumo de energia dos nós.

O funcionamento do LEACH é dividido em uma sequência consecutiva de rounds. Cada round possui uma breve fase de *setup*, onde os *clusters* são definidos, seguida por uma fase mais longa, na qual é feita a transmissão dos dados. No início da fase de *setup*, os nós vão decidir se serão ou não *cluster-heads* durante o round. Essa decisão é tomada de forma probabilística através da escolha de um número aleatório entre zero e um, que será comparado com o limiar dado pela seguinte equação:

$$T(n) = \frac{P}{1 - P * \left(r \frac{1}{P}\right)} \; se \; n \in G, 0 \; caso \; contrário \tag{1}$$

onde P é a porcentagem pretendida de *cluster-heads*, r é o round atual e G é a lista de nós que ainda não foram *cluster-heads* nos últimos $\frac{1}{P}$ rounds. Se o número aleatório for menor que $T(n)$, o nó se torna *cluster-head*. Após a eleição, todos os CHs irão enviar uma mensagem em *broadcast* se anunciando. Os nós não-*cluster-heads* (NCH) irão escolher o CH que tiver o melhor indicador de intensidade de sinal recebido (RSSI), ajustar sua potência de transmissão e enviar uma mensagem de adesão ao *cluster*. Quando o CH possui todas as confirmações, ele envia uma tabela TDMA com o cronograma de transmissão de cada nó do *cluster*.

Uma vez que todos os *clusters* estejam prontos, inicia-se a fase de transmissão de dados. Cada nó, em seu determinado *slot* TDMA irá transmitir seu pacote até o CH. Enquanto não estiverem transmitindo, os nós NCH podem manter o rádio desligado, para poupar energia. Após receber todos os pacotes, o CH irá realizar a agregação de dados, para comprimir os dados em um único sinal e fazer o envio desse pacote para a estação base.

2.2. Algoritmo K-means

O K-means pertence à família de algoritmos de aprendizado de máquina não supervisionado, que usa uma maneira simples de resolver o conhecido problema de agrupamento. O procedimento utilizado pelo K-means classifica o dado em um conjunto de dados através de um número pré-definido de *clusters* (assuma K *clusters*) estabelecidos inicialmente.

A ideia principal é definir K centroides, um para cada cluster. Esses centroides devem ser colocados o mais longe possível uns dos outros. O próximo passo é pegar cada ponto pertencente a um dado conjunto de dados e associá-lo ao centroide mais próximo. Quando nenhum ponto está pendente, a primeira etapa é concluída e os *clusters* são formados. Neste ponto, precisamos recalcular k novos centroides, como baricentros dos *clusters* resultantes da etapa anterior. Depois de termos esses k novos centroides, uma nova ligação deve ser feita entre os mesmos pontos de conjunto de dados e o novo centroide mais próximo. Um *loop* foi gerado, no qual podemos notar que os k centroides mudam sua localização passo a passo, até que não haja mais mudanças, ou seja, os centroides não se movam mais. Por fim, esta técnica visa minimizar uma função objetivo, neste caso uma função de erro ao quadrado.

O uso do k-means em RSSFs tem o poder de reduzir as distâncias de transmissão entre os nós de um *cluster*. Isso se dá através da seleção de um ponto médio disposto em qualquer posição da rede, onde os nós irão se agrupar aos centroides mais próximos formando os *clusters*. Isso faz com que a distância total de transmissão seja mínima, o que reduz o consumo de energia da rede. Após encontrar os pontos ótimos da rede, é necessário encontrar o nó mais próximo a esses pontos e elegê-los como *cluster-heads* de um round específico.

Existem alguns algoritmos que utilizam o k-means e tem como base o protocolo LEACH, na qual podemos citar o KEAC [Razzaq et al. 2018], o CHSM-MBC [Sondes and Rim 2018], MENC [He 2019] e BENMAHDI [BENMAHDI and LEHSAINI 2020].

2.3. Algoritmo K-medoids

O K-medoids também pertence à mesma família que o k-means, e tem funcionamento similar. A diferença entre eles é relativo a escolha do centroide na rede. O K-means se baseia em centroides que consideram um ponto teórico na rede. Já o K-medoids, realiza os mesmos cálculos utilizando os nós reais que fazem parte da rede. Desta maneira, o cálculo das distâncias que é feito já traz exatamente as distâncias que serão utilizadas no cálculo para saber a distância total da transmissão de todos os nós de um *cluster*. Com o K-medoids, o ponto ótimo encontrado já se encontra como um nó dentro da rede, não precisando encontrá-lo posteriormente.

Existem propostas que utilizam o k-medoids em RSSFs, como exemplo temos o KCA [Wang et al. 2018], o EACA [Faid et al. 2021a], o IHEE [Faid et al. 2021b] e o EDMARA2 [Khademi Nori and Sharifian 2020].

2.4. *Multi-hop*

O encaminhamento das mensagens é uma parte crucial no projeto de um protocolo para RSSFs, pois quanto mais longa a distância de transmissão, mais elevado é o consumo de

energia. Uma forma de minimizar esse tipo de problema é utilizar a técnica de *multi-hop* para fazer a transmissão das mensagens. No *multi-hop*, ao invés da mensagem ser transmitida diretamente para o destino, que pode estar muito distante, a mensagem passa por nós mais próximos, que fazem a retransmissão. Protocolos de *clustering* utilizam a comunicação *single-hop*, como exemplificado na Figura 1. No entanto, para longas distâncias, o custo de energia de transmissão é muito alto.

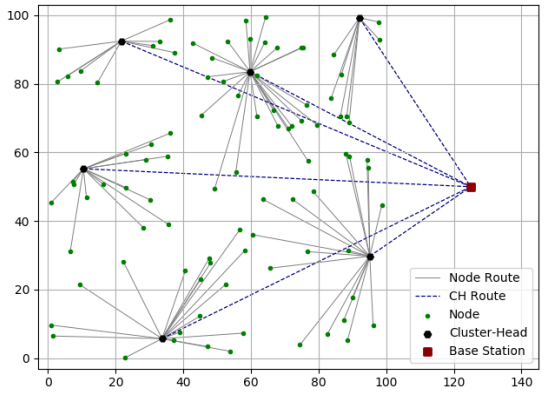

Figura 1. *Clusters* em um round do LEACH. [Pinheiro et al. 2019]

A aplicação do *multi-hop* à protocolos de *clustering* reduz os custos de comunicação, como visto em [Pinheiro et al. 2019]. Podemos dividir o *multi-hop* em três abordagens. No *intra-cluster*, os nós dentro do cluster se comunicam usando vários saltos. As mensagens são retransmitidas através de vários caminhos mais curtos para chegar ao *cluster-head*, como pode ser constatado na Figura 2. No *inter-cluster*, apenas os *cluster-heads* usam *multi-hop* para retransmitir mensagens para a estação base, como representado na Figura 2. A última abordagem usa *multi-hop* dentro do cluster e entre os *cluster-heads*, conforme representado na Figura 2.

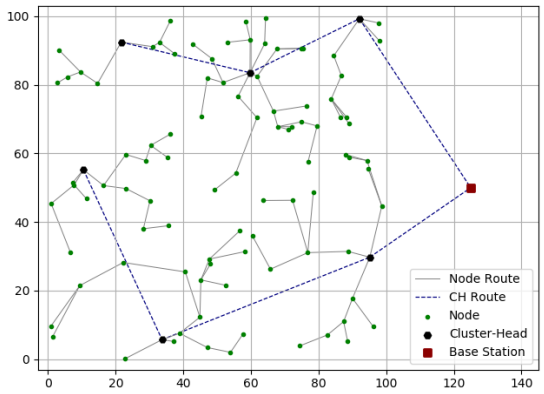

Figura 2. *Clustering* com uso do *multi-hop*. [Pinheiro et al. 2019]

Os algoritmos *multi-hop* usam vários critérios como RSSI [Loscri et al. 2005], distância [Hu et al. 2015] e energia residual [Chen and Chen 2013] para selecionar o caminho para retransmitir as mensagens para o destino. Entretanto, outros algoritmos utilizam técnicas de otimização e aproximação [Kaddi et al. 2017].

A Tabela 1 e a Tabela 2 lista alguns protocolos de *clustering* que utilizam aprendizagem de máquina não supervisionada. Nela podemos verificar os usos do *multi-hop* em cada abordagem, onde uma característica que se destaca é que, em todos os casos, nenhuma abordagem utiliza o *multi-hop intra-cluster*.

Tabela 1. Protocolos com uso do k-means

	Intra-cluster	***Inter-cluster***
KEAC [Razzaq et al. 2018]	*single-hop*	*single-hop*
CHSM-MBC [Sondes and Rim 2018]	*single-hop*	*multi-hop*
MENC [He 2019]	*single-hop*	*multi-hop*
[BENMAHDI and LEHSAINI 2020]	*single-hop*	*multi-hop*

Tabela 2. Protocolos com uso do k-medoids

	Intra-cluster	***Inter-cluster***
KCA [Wang et al. 2018]	*single-hop*	*single-hop*
EACA [Faid et al. 2021a]	*single-hop*	*single-hop*
IHEE [Faid et al. 2021b]	*single-hop*	*multi-hop*
EDMARA2 [Khademi Nori and Sharifian 2020]	*single-hop*	*multi-hop*

3. Metodologia

A maior parte das definições e configurações dos protocolos serão replicadas do LEACH, mas é necessário definir alguns pontos cruciais para esta proposta específica, principalmente as definições do funcionamento das modalidades do *multi-hop*.

O TDMA, que é parte essencial do funcionamento do LEACH, pode causar dessincronização de nós, pois *clusters* maiores precisam de mais tempo para receber as mensagens de todos os seus nós do que um cluster menor. Os nós devem ter baixa taxa de transmissão para sincronizar os *clusters*; assim, o período de transmissão é maior do que o pior caso para a programação TDMA, que é quando há apenas um *cluster-head* para toda a rede. Geralmente, em RSSFs, os nós possuem largura de banda e taxa de transmissão muito baixas.

O LEACH utiliza uma fórmula para definir a posição da estação base em relação ao centro da área de dispersão dos nós. Para este trabalho, esta localização é estabelecida de outra forma, na qual a coordenada x da estação base fica a uma distância de 25 metros de uma das bordas da área e a coordenada y fica no meio do comprimento da área, como é possível verificar na Figura 1.

Este artigo propõe a utilização das duas modalidades do *multi-hop* no encaminhamento de mensagens. Uma vez que é possível utilizar o *intra-cluster* e o *inter-cluster*, temos a possibilidade de aplicar arranjos de configuração distintas nos protocolos definidos, o que gera a criação de algumas variações destes protocolos. Existem quatro configurações possíveis que são obtidas através da ativação ou desativação de cada modalidade do *multi-hop*. A Tabela 3 nos mostra a nomenclatura das quatro variações possíveis, incluindo o *single-hop*, que é a operação padrão de comunicação.

A descrição de todos os passos do algoritmo pode ser verificado na Figura 3. O primeiro passo é selecionar o tipo de algoritmo que será utilizado na simulação, nesta etapa é definido apenas o modo como os *cluster-head* são selecionados. Os demais passos seguem a mesma estrutura para todos os protocolos. Os pontos chave do fluxograma é a ativação e inativação do modos de *multi-hop*, que seguem um caminho separado para realizar as divisões da rede ou do *cluster*.

Tabela 3. Modalidades do *multi-hop*

	Intra-cluster	***Inter-cluster***
Protocolo base	Inativo	Inativo
INTER	Inativo	Ativo
INTRA	Ativo	Inativo
HOP	Ativo	Ativo

As próximas sub-seções são destinadas à explicação da implementação das modalidades de *multi-hop*, *intra-cluster* e *inter-cluster*, nos protocolos deste estudo. O modelo apresentado é baseado em uma abordagem híbrida apresentada em [Bozorgi et al. 2017], tendo o seu funcionamento simplificado.

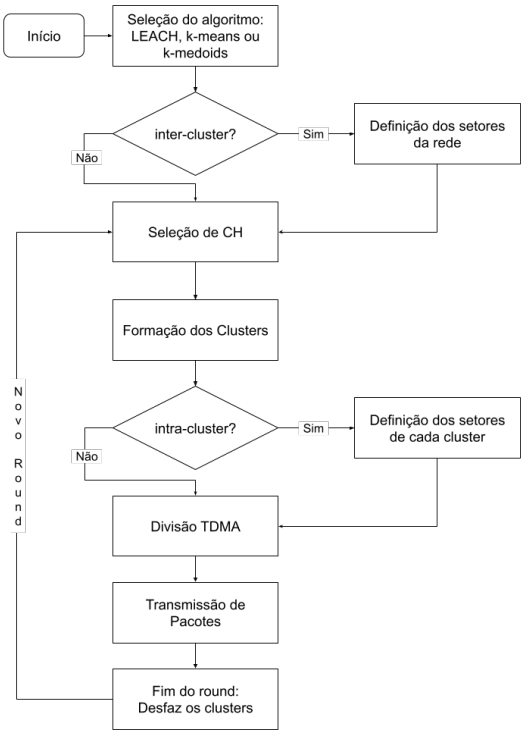

Figura 3. Fluxograma do algoritmo.

3.1. *Intra-cluster*

O roteamento *intra-cluster* é feito através do processo de setorização do *cluster*. Para isso, é necessário fazer um cálculo que irá definir quais são os setores do *cluster*. A primeira parte do cálculo define o raio da rede que é dado de acordo com a seguinte equação:

$$d = d_{max} - d_{min} \tag{2}$$

onde d_{max} e d_{min} são, respectivamente, as distâncias do nó mais distante e do nó mais próximo do *cluster-head*. A partir desta informação, nós dividimos a rede em setores como mostrado em:

$$L_{RL} = \frac{d}{N_s} \tag{3}$$

onde N_s é o número de setores que serão implementados no *cluster*. A Figura 4 nos mostra a divisão do campo de um *cluster* em quatro setores, onde L_{RL} estabelece o tamanho de cada setor, definindo assim limites inferiores e superiores para cada um. Um nó estará em um dado setor se sua distância até o CH estiver dentro dos limites daquele setor. Cada um irá receber uma numeração, em que o primeiro setor após o CH recebe o valor um, incrementando à medida que ficam mais distantes.

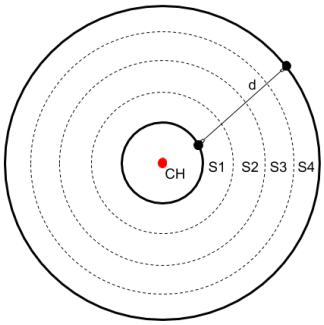

Figura 4. Divisão de um *cluster* em quatro setores. [Pinheiro et al. 2019]

O *cluster-head* é o responsável por realizar todo o processo de setorização no *intra-cluster*. Isso porque ele possui as informações de todos os nós do cluster, que foram recebidas durante a fase de *setup*. Para que o *multi-hop* seja eficiente, o CH deve organizar a tabela TDMA de modo que os nós que estão mais distantes devem receber os primeiros *slots* de tempo e os mais próximos receberão os últimos *slots*. Isso se torna necessário para que os nós tenham tempo de receber os pacotes dos nós vizinhos e repassá-los adiante.

O CH vai realizar o cálculo do L_{RL} e definir qual o setor de cada um dos nós. Uma vez que todos os setores estão definidos, o CH irá analisar, para cada nó, qual o melhor encaminhamento que deve ser feito. A seleção só leva em consideração os nós que estão em setores menores, fazendo assim com que os pacotes sigam na direção do CH. Todas essas informações que foram definidas devem ser enviadas para os nós. Dessa forma, uma outra modificação feita é na tabela TDMA, que será responsável por levar essas informações até os nós. Assim, cada linha da tabela deve conter quatro informações: o identificador do nó; o *slot* de tempo destinado ao envio do seu pacote; as informações do nó que receberá o pacote a ser enviado, podendo ser um nó vizinho ou o cluster-head; e os *slots* de tempo em que o nó deve ativar o seu rádio para receber algum pacote de um nó vizinho.

O último item da tabela é crucial para a eficiência do *intra-cluster*. Através dessa informação, cada *slot* de tempo terá um par de nós com o rádio ligado, cumprindo os

papéis de emissor e receptor. Essa definição é necessária para que os nós não tenham que ficar com o rádio ligado a todo instante, pois um nó tem que saber o momento exato que irá receber um pacote de um nó vizinho.

Quanto mais setores existem em um cluster, mais dividido ele fica e assim os pacotes podem realizar mais saltos até chegar ao destino. Tendo em vista disso, a probabilidade dos nós enviarem seus pacotes a um nó que esteja mais próximo é maior. Em um cenário em que há pelo menos um nó em cada setor, o pacote de um nó que esteja no último setor iria passar por todos os setores até o CH, tendo o número de saltos igual ao número de setores. Pode acontecer também que um dado nó em um setor distante tenha todos os seus vizinhos no lado oposto do CH, tornando o CH o nó mais próximo. Como o algoritmo analisa todos os nós que estão adiante do seu setor, o encaminhamento do pacote será feito diretamente ao CH, que seria a melhor escolha neste caso.

Quando um nó chega em seu tempo de *slot*, ele já deve ter recebido todos os pacotes de seus nós vizinhos, isso se existir algum encaminhamento para ele. De posse de todos esses pacotes, o nó irá fazer o processo de agregação de dados. Também podemos observar que o CH também economiza energia, pois o seu rádio só fica ligado no momento em que um nó irá enviar um pacote para ele.

3.2. *Inter-cluster*

O *inter-cluster* segue uma estrutura semelhante a do *intra-cluster*, porém possui algumas diferenças. Neste caso, torna-se necessária uma configuração prévia da rede pela estação base, antes mesmo do início do primeiro round. Na primeira parte da configuração, cada nó tem a responsabilidade de enviar seus dados de localização à estação base. Uma vez que a estação base conhece a localização de todos os nós da rede, ela irá realizar todo o cálculo de setorização que já foi descrito anteriormente, fazendo com que toda a área da rede seja setorizada.

Na segunda parte, após a definição dos setores, a estação base irá enviar um pacote em *broadcast* com uma tabela contendo o identificador do nó e seu devido setor na rede. Para que um CH saiba para qual outro CH ele deve enviar, cada um deles necessita das informações dos outros. Essa informação é adquirida durante a fase de *broadcast* dos *cluster-heads*. Dessa maneira, um CH conhecerá a localização dos possíveis nós pra quem ele pode encaminhar seus pacotes.

3.3. Algoritmos de Aprendizagem de Máquina

Tanto o K-means quanto o K-medoids utilizam uma metodologia de otimizar as distâncias entre os pontos e os centroides. Para esta aplicação, o LEACH irá realizar a seleção inicial dos *cluster-heads*, pois esta será a entrada do K para os dois algoritmos. Após isso, cada algoritmo, de acordo com seu funcionamento, irá movimentar os centroides ao longo da rede, mas no fim, ambos irão selecionar um nó da rede para torná-lo um *cluster-head*.

Esta operação deve ser centralizada, assim, a estação base deve realizar esses cálculos e enviar os dados para os nós com as informações dos novos *cluster-heads* da rede. Os algoritmos de aprendizagem de máquina não interferem nas outras operações do LEACH ou do *multi-hop*, apenas na seleção de *cluster-heads*.

4. Resultados

As implementações dos protocolos LEACH, K-means e K-medoids, e os respectivos modos de *multi-hop*, foram feitas com a linguagem Python[1]. Este artigo avalia o desempenho dos protocolos através do tempo de vida da rede, que é medido em rounds. Para cada round, é quantificado o total de nós que ainda estão ativos na rede, ou seja, que ainda possuem nível de bateria suficiente para transmissão de dados. Ao longos dos rounds, os nós vão esgotando as baterias e a rede vai reduzindo sua eficiência. Desta maneira, podemos realizar um comparativo de qual protocolo tem a capacidade de manter a rede em funcionamento por mais tempo.

O modelo do rádio adotado é o mesmo utilizado pelo protocolo LEACH. Este trabalho definiu dois tamanhos de pacote diferentes, os pacotes trocados durante a fase de *setup*, que serão chamados de pacotes de configuração e os pacotes de dados. Todos os parâmetros de configuração que são utilizados na simulação estão listados na Tabela 4.

Tabela 4. Parâmetros de configuração

Energia inicial	0,5 J
Consumo de recepção	50 nJ/bit
Consumo do amplificador	100 pJ/bit
Consumo da agregação de dados	5 nJ/bit
P (% de *cluster-heads*)	5%
Tamanho do pacote de configuração	300 bits

Na Tabela 5 temos a listagem dos parâmetros que foram utilizados em todas as simulações. Para este trabalho, os dados presentes nas Tabelas 4 e 5 foram adotados a partir das definições do LEACH encontrados em [Heinzelman et al. 2000] e [Pinheiro et al. 2019]. Os parâmetros foram utilizados por serem facilmente encontrado em vários artigos da área de estudo, como visto em [Wang et al. 2018], [Faid et al. 2021a], [BENMAHDI and LEHSAINI 2020] e [Faid et al. 2021b].

Tabela 5. Parâmetros das simulações

Número de nós	100
Tamanho do pacote de dados	2000 bits
Número de setores	2
Tamanho da rede	$100m$x$100m$
Posição da estação base	(125, 50)

Os gráficos de linha que comparam o desempenho dos protocolos ao longo dos rounds foram gerados a partir de uma única execução de cada protocolo. Já os gráficos de barra com os resultados finais e os devidos intervalos de confiança, foram obtidos através de uma média aritmética de 33 repetições, quantidade essa que gera um intervalo de confiança de 95%. Primeiramente é apresentada uma comparação inicial dos protocolos base, sem o uso de nenhum tipo de *multi-hop*, para se ter uma métrica inicial de desempenho de cada um dos três protocolos escolhidos para o estudo.

[1]https://github.com/DiegoL1M4/LEACH-K-HOP

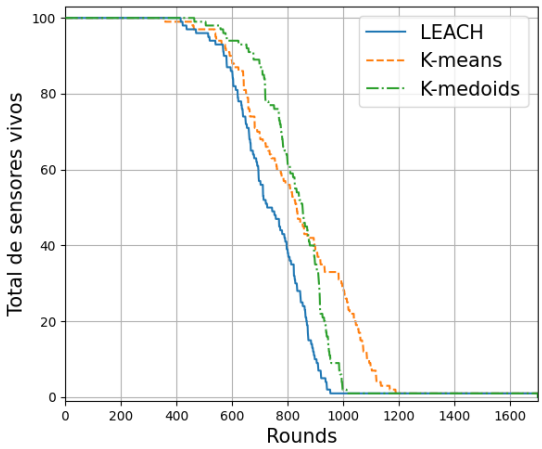

Figura 5. Comparativo entre o LEACH, K-means e K-medoids.

De acordo com a Figura 5, podemos verificar que o protocolo LEACH possui um desempenho inferior aos demais. Os protocolos baseados em aprendizagem de máquina tiveram um desempenho médio de 11,27% a mais que o protocolo LEACH. Entre o K-means e o K-medoids, temos que ambos possuem desempenho similar, considerando o intervalo de confiança, entretanto, nos experimentos realizados, foi obtido uma superioridade do K-means de 0,85% nos tempos médios calculados. O melhor desempenho dos algoritmos de aprendizado de máquina se dá pela otimização das distâncias entre os nós e os *cluster-heads*, por meio da movimentação dos centroides.

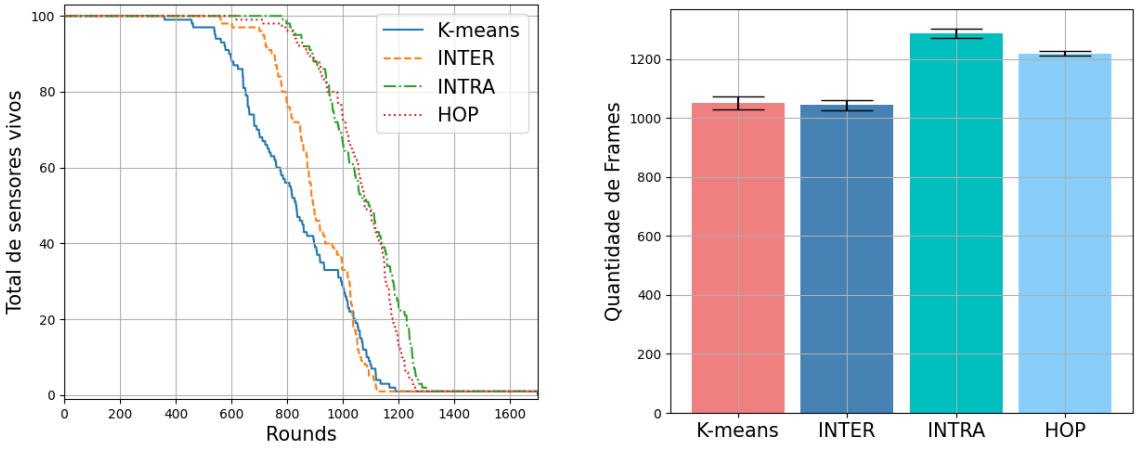

Figura 6. Análise do *multi-hop* no K-means.

Seguindo a proposta do estudo, foram aplicadas as duas modalidades do *multi-hop* no K-means, cujo os resultados podem ser verificados na Figura 6. No gráfico é possível constatar que o uso dos métodos de *multi-hop* potencializa o desempenho energético da rede, fazendo com que os nós durem por mais tempo. Destes resultados, podemos observar que a modalidade de *intra-cluster* possui o melhor desempenho, obtendo um aumento no tempo de vida da rede de 22,44% se comparado com o K-means base. O uso do *inter-cluster* não obteve resultados satisfatórios, na qual foi obtida uma redução no tempo de vida da rede de 0,70% em relação ao K-means base.

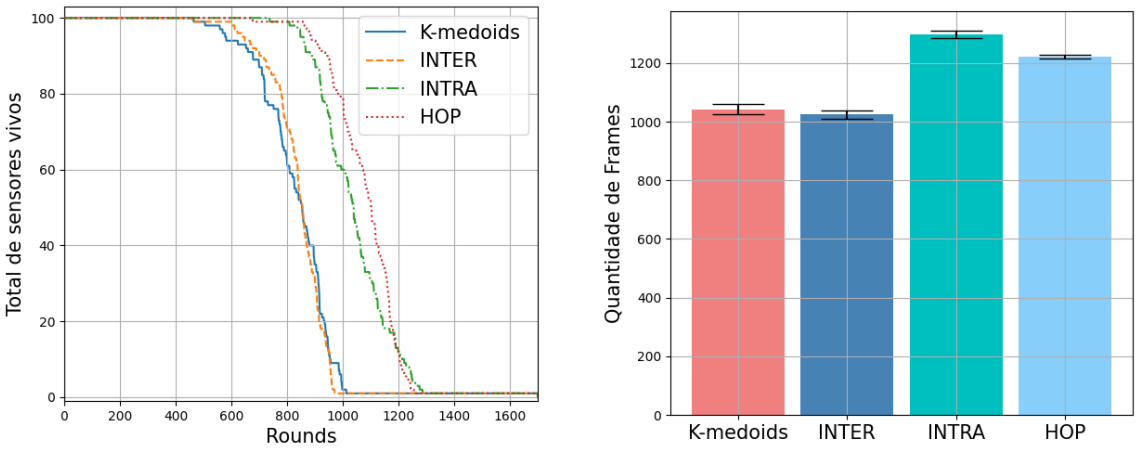

Figura 7. Análise do *multi-hop* no K-medoids.

A Figura 7 mostra os resultados da análise do K-medoids com o uso do *multi-hop*. Como no experimento anterior, também temos uma melhoria progressiva com o uso do *multi-hop*, tendo também o *intra-cluster* com o melhor desempenho, obtendo um aumento do tempo da rede de 24,48%. O uso do *inter-cluster* também não obteve resultados satisfatórios, na qual foi obtida uma redução no tempo de vida da rede de 1,70%, que considerando o intervalo de confiança, temos praticamente ambas as modalidade em ponto de igualdade, tanto para o K-medoids quanto para o K-means.

Tabela 6. Resultados das simulações com os valores do intervalo de confiança

Protocolo	Base	Inter	Intra	Hop
LEACH	940,81 rounds $\pm 12,13$	902,15 rounds $\pm 11,12$	1177,66 rounds $\pm 11,20$	1118,48 rounds $\pm 12,83$
K-means	1051,42 rounds $\pm 21,55$	1044,15 rounds $\pm 17,32$	1287,42 rounds $\pm 16,15$	1218,51 rounds $\pm 8,30$
K-medoids	1042,48 rounds $\pm 16,89$	1024,66 rounds $\pm 15,28$	1297,78 rounds $\pm 13,78$	1221,15 rounds $\pm 6,52$

Com base nos resultados da Tabela 6, temos que o LEACH com o *intra-cluster* tem o melhor desempenho dentre as variações do *multi-hop*. Realizando um comparativo entre as variações com melhores desempenhos, temos que o LEACH-INTRA é superado pelo K-means-INTRA com um aumento de 9,32% no tempo de vida da rede. Já em um comparativo entre o K-means e o K-medoids, temos que o K-medoids-INTRA obteve um aumento de 0,80% no tempo de vida da rede.

A Tabela 6 também possui os valores do intervalo de confiança para valor médio. A partir destes resultados, podemos ver que alguns protocolos estão com resultados bastante similares, como a variação Intra nos protocolos K-means e K-medoids.

5. Conclusão

Este trabalho avaliou o uso do *multi-hop* em conjunto com o roteamento baseado em *clustering* do K-means e do K-medoids. Vimos que o *multi-hop* é uma alternativa viável de

implementação nos protocolos de *clustering* baseados em aprendizado de máquina não supervisionado, pois a estruturação que é criada entre os nós realiza uma maior economia das baterias dos dispositivos. A partir da análise dos resultados, concluímos que os protocolos podem ter ganhos significativos com o uso do modo *intra-cluster*, na qual os dois protocolos de aprendizado de máquina obtiveram um aumento médio do tempo de vida da rede de 23,46%. Assim, os resultados sugerem que os protocolos apresentados nas Tabelas 1 e 2, podem ter ganhos de desempenho significativos, com o uso do intra-cluster. Contudo, é necessário que sejam realizados experimentos adicionais

O uso do *inter-cluster* reduz o tempo de vida da rede, porém, para o LEACH, a diferença entre o protocolo base e com o uso do *inter-cluster* foi de 4,11%. Já para o K-means, a redução da eficiência foi de 0,70% e do K-medoids foi de 1,70%. Assim, podemos concluir que protocolos com algoritmos diferentes podem ter eficiências diferentes quanto ao uso do *multi-hop*.

Referências

Al-Karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *IEEE wireless communications*, 11(6):6–28.

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

BENMAHDI, M. B. and LEHSAINI, M. (2020). A ga-based multihop routing scheme using k-means clustering approach for wireless sensor networks. In *2020 Second International Conference on Embedded & Distributed Systems (EDiS)*, pages 155–160. IEEE.

Bozorgi, S. M., Rostami, A. S., Hosseinabadi, A. A. R., and Balas, V. E. (2017). A new clustering protocol for energy harvesting-wireless sensor networks. *Computers & Electrical Engineering*, 64:233–247.

Chen, Z. and Chen, K. (2013). An improved multi-hop routing protocol for large-scale wireless sensor network based on merging adjacent clusters. *JSW*, 8(8):2080–2086.

Faid, A., Sadik, M., and Sabir, E. (2021a). Eaca: An energy aware clustering algorithm for wireless iot sensors. In *2021 28th International Conference on Telecommunications (ICT)*, pages 1–6. IEEE.

Faid, A., Sadik, M., and Sabir, E. (2021b). Ihee: An improved hybrid energy efficient algorithm for wsn. In *Future of Information and Communication Conference*, pages 283–298. Springer.

He, W. (2019). Energy-saving algorithm and simulation of wireless sensor networks based on clustering routing protocol. *IEEE Access*, 7:172505–172514.

Heinzelman, W. B., Chandrakasan, A. P., Balakrishnan, H., et al. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, 1(4):660–670.

Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE.

Hu, S., Han, J., Wei, X., and Chen, Z. (2015). A multi-hop heterogeneous cluster-based optimization algorithm for wireless sensor networks. *Wireless Networks*, 21(1):57–65.

Kaddi, M., Benahmed, K., and Omari, M. (2017). Leach-kang: A new routing protocol for wsn based on leach protocol and kangaroo method. In *2017 International Conference on Mathematics and Information Technology (ICMIT)*, pages 273–278. IEEE.

Khademi Nori, M. and Sharifian, S. (2020). Edmara2: a hierarchical routing protocol for eh-wsns. *Wireless Networks*, 26(6):4303–4317.

Loscri, V., Morabito, G., and Marano, S. (2005). A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach). In *IEEE vehicular technology conference*, volume 62, page 1809. IEEE; 1999.

Martinez, B., Monton, M., Vilajosana, I., and Prades, J. D. (2015). The power of models: Modeling power consumption for iot devices. *IEEE Sensors Journal*, 15(10):5777–5789.

Pinheiro, D. L., Garça, J. L., de Lima, O. A., and Furtado, C. G. (2019). Analysis of multi-hop strategies in leach protocol. In *2019 IX Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–8. IEEE.

Rawat, P., Singh, K. D., Chaouchi, H., and Bonnin, J. M. (2014). Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of supercomputing*, 68(1):1–48.

Razzaq, M., Ningombam, D. D., and Shin, S. (2018). Energy efficient k-means clustering-based routing protocol for wsn using optimal packet size. In *2018 International Conference on Information Networking (ICOIN)*, pages 632–635. IEEE.

Shahraki, A., Taherkordi, A., Haugen, Ø., and Eliassen, F. (2020). Clustering objectives in wireless sensor networks: A survey and research direction analysis. *Computer Networks*, 180:107376.

Singh, J., Mishra, A. K., et al. (2015). Clustering algorithms for wireless sensor networks: a review. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 637–642. IEEE.

Sondes, D. and Rim, H. (2018). Chsm-mbc cluster head selection method with multi hop balanced clustering routing protocol for heterogeneous wireless sensors networks. In *2018 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–6. IEEE.

Wang, J., Wang, K., Niu, J., and Liu, W. (2018). A k-medoids based clustering algorithm for wireless sensor networks. In *2018 international workshop on advanced image technology (IWAIT)*, pages 1–4. IEEE.

Xiangning, F. and Yulin, S. (2007). Improvement on leach protocol of wireless sensor network. In *2007 International Conference on Sensor Technologies and Applications (SENSORCOMM 2007)*, pages 260–264. IEEE.

Zeb, A., Islam, A. M., Zareei, M., Al Mamoon, I., Mansoor, N., Baharun, S., Katayama, Y., and Komaki, S. (2016). Clustering analysis in wireless sensor networks: the ambit of performance metrics and schemes taxonomy. *International Journal of Distributed Sensor Networks*, 12(7):4979142.