

Agregação de Dados com Desvio de Buracos para Redes de Sensores Sem Fio com *Sinks* de Alto Alcance

Moysés M. Lima¹, Horácio A. B. F. de Oliveira¹, Eduardo F. Nakamura¹
Raimundo Barreto¹ e Daniel L. Guidoni²

¹Instituto de Computação – Universidade Federal do Amazonas

²Departamento de Computação – Pontifícia Universidade Católica de Minas Gerais
{moyses.lima,horacio,nakamura,rbarreto}@icomp.ufam.edu.br, guidoni@ufs.br

Abstract. *In Wireless Sensor Networks (WSNs), Greedy Forwarding is a common technique used by most geographic routing algorithms. It uses location information of neighbors to choose the next hop geographically closer to the destination node. However, this strategy requires previous execution of a location system which is not always available in forest scenarios and in most cases is not able to overcome routing holes. In this work, we develop a geographic routing algorithm that performs data aggregation and is able to overcome routing holes in WSNs. Our approach, called ARESTA, uses a sink with long-range communication that sends an advertisement package to the entire network in a single hop. The ARESTA algorithm is also able to overcome holes through a new technique that uses RSSI (Received Signal Strength Indicator) to detect and bypass a hole. The RSSI values sent by the long-range sink node are used to create a time metric to guide in-network data aggregation. Our simulations show that the proposal has high delivery rate, scalability on densely deployed wireless networks, and also can identify and detect holes with all of the benefits of GF algorithms.*

Resumo. *Em Redes de Sensores sem Fio (RSSFs), a técnica de encaminhamento guloso é bastante utilizada pela maioria dos algoritmos de roteamento geográfico. A abordagem utiliza a informação de localização dos vizinhos para escolher o próximo salto, geograficamente mais próximo do destino. No entanto, a estratégia requer a utilização prévia de um sistema de localização, que nem sempre está disponível em cenários de floresta e na maioria dos casos não é capaz de lidar com buracos. Neste trabalho, desenvolvemos um algoritmo de roteamento geográfico que realiza agregação de dados e é capaz de desviar buracos de roteamento em RSSFs. Nossa abordagem, chamada ARESTA, utiliza um sink com alto alcance de comunicação que envia um pacote de anúncio a toda rede em um único salto. O algoritmo ARESTA é capaz de lidar com buracos de roteamento por meio de uma nova técnica que utiliza o RSSI (Received Signal Strength Indicator) dos nós para detectar e desviar buracos. Os valores do RSSI divulgados pelo nó sink de longo alcance são utilizados para criar um temporizador que auxilia na agregação de dados. As simulações realizadas mostram uma eficiente taxa de entrega, escalabilidade em redes densamente populadas, além de identificar e detectar buracos mantendo os benefícios de algoritmos de encaminhamento guloso.*

1. Introdução

Uma Rede de Sensores sem Fio (RSSF) consiste em um conjunto de nós sensores depositados em uma área de interesse que de forma colaborativa monitoram condições ambientais físicas e químicas em diferentes locais [Akyildiz et al. 2002b]. Por esta razão, as RSSFs são essenciais no monitoramento de condições ambientais tais como temperatura, luminosidade, movimento e presença de certos tipos de objetos. Na maior parte das aplicações, os nós sensores possuem limitações energéticas e de largura de banda, fazendo com que técnicas de otimização do consumo de energia e comunicação sejam necessárias.

Quando consideramos o número de saltos entre nós regulares e o nó *sink* em uma RSSF, podemos classificar as abordagens como redes de único salto e redes de múltiplos saltos [Akyildiz et al. 2002a]. Na primeira classificação, os nós regulares enviam suas informações diretamente ao *sink*, simplificando a implementação de soluções de envio de pacotes. Entretanto, é necessária a utilização de um dispositivo de comunicação sem fio de longo alcance, que normalmente é custoso em termos de consumo energético e hardware apropriado. Já na abordagem em múltiplos saltos, os nós regulares enviam suas informações ao *sink* utilizando comunicação de curto alcance estabelecida com um ou mais vizinhos em localização intermediária.

Avanços na área de comunicação sem fio demonstram que a propagação do sinal de longo alcance se mostra confiável e pode ser utilizada para estimar distâncias em uma RSSF. Tal estimativa é baseada nos valores de RSSI (Received Signal Strength Indicator) dos nós envolvidos na comunicação, considerando a relação entre a força do sinal recebido e a distância como uma função de distância entre o emissor e o receptor [Al Alawi 2011]. Como apresentado em [Lima et al. 2013], um possível cenário para a implementação destas tecnologias é o projeto ATTO (Amazonian Tall Tower Observatory), que consiste em uma torre de 320 m de altura e alta capacidade de comunicação, cujo objetivo é mapear eventos e fornecer medidas confiáveis de gases do efeito estufa [Tollefson 2010]. O sinal emitido pelo dispositivo de comunicação instalado na torre poderia alcançar um grande número de nós, como apresentado na Figura 1.

Os protocolos de roteamento geográficos têm sido vistos como uma das principais soluções de roteamento em Redes de Sensores Sem Fio (RSSFs) por serem escaláveis, dinâmicos e possuírem uma alta taxa de entrega de dados [Kim et al. 2005, You et al. 2009]. Contudo, em algumas situações os métodos envolvendo informações geográficas não podem ser utilizados, especialmente em cenários onde a informação de

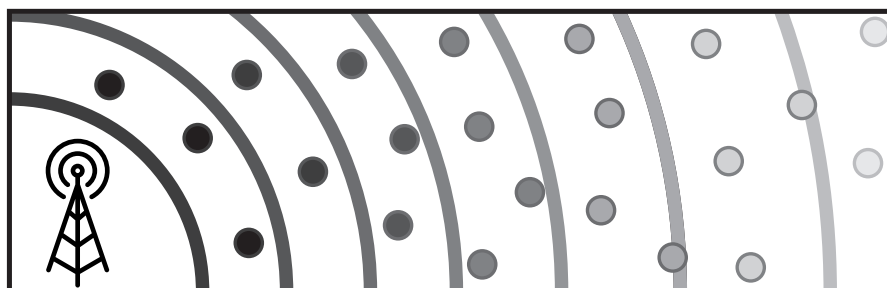


Figura 1. Valores de RSSI decrescendo à medida que a consulta do *sink* de alto alcance se propaga na rede. Nós mais distantes do *sink* receberão a mensagem com potência menor.

localização não está disponível ou não existe [Éfren Souza et al. 2012].

O encaminhamento guloso é a forma mais simples de se implementar o roteamento geográfico e é particularmente atraente nas redes de sensores. A técnica utiliza uma estratégia simples de geração de rotas em que um nó sensor sempre encaminha um pacote ao vizinho geograficamente mais próximo ao nó *sink*. Um desafio importante na utilização dessa técnica é a ocorrência de buracos de roteamento durante o envio dos pacotes ao *sink*. Os buracos são normalmente associados à regiões onde os nós estão impossibilitados de encaminhar adiante informações recebidas. Sua ocorrência pode ser causada por obstáculos físicos, esgotamento energético, falhas na comunicação entre vizinhos eleitos, implantação incorreta e devido ao fenômeno dos mínimos locais [Chen and Varshney 2007].

Um outro desafio na utilização de algoritmos de roteamento geográfico é a necessidade de localização dos nós, que pode ser custosa e susceptível a grandes erros em RSSFs [Éfren Souza et al. 2012]. Além disso, a capacidade de processamento dos nós pode ser utilizada como estratégia para a redução do consumo de energia, durante o roteamento. Associada à solução de encaminhamento, a agregação de dados pode melhorar significativamente o desempenho de um algoritmo de encaminhamento guloso, pois provê a combinação de vários pacotes de dados de entrada em um único pacote de informação relevante.

Neste trabalho, propomos um algoritmo de roteamento geográfico, que provê agregação de dados e é capaz de lidar com buracos de roteamento em RSSFs, sem a necessidade de utilização de informações de localização, chamado ARESTA. A proposta é uma melhoria do algoritmo apresentado em [Lima et al. 2013], chamado RADR (Roteamento e Agregação de Dados baseado no RSSI). A idéia principal do algoritmo é aproveitar o grande alcance do nó *sink*, equipado com um dispositivo de comunicação de maior potência, de tal forma que em um único salto todos os nós da rede sejam alcançados. O retorno dos nós sensores para o nó *sink* é realizado através de múltiplos saltos, utilizando uma nova técnica de encaminhamento guloso que utiliza os valores do RSSI (*Received Signal Strength Indicator*) para calcular o próximo salto em direção ao nó *sink*, calcular o tempo de espera que possibilita a agregação de dados e detectar buracos de roteamento durante o processo de encaminhamento.

O restante deste trabalho está organizado da seguinte forma: na Seção 2, apresentamos e classificamos os trabalhos que serviram como base de pesquisa para o presente artigo, enquanto na Seção 3 descrevemos o algoritmo ARESTA, cuja avaliação de performance é mostrada na Seção 4. Uma breve discussão sobre a aplicabilidade da proposta é apresentada na Seção 5 e, finalmente, na Seção 6 apresentamos nossas conclusões e trabalhos futuros.

2. Trabalhos Relacionados

A técnica de encaminhamento guloso tem sido bastante utilizada pela maioria dos algoritmos de roteamento geográfico por se tratar de uma abordagem simples, eficiente e escalável [Boukerche et al. 2008]. A abordagem utiliza a informação de localização dos vizinhos para encaminhar pacotes ao nó que está geograficamente mais próximo do destino. Visando facilitar o entendimento sobre nossa proposta, classificamos os algoritmos de roteamento geográfico em três categorias principais de algoritmos: (1) baseados em

coordenadas geográficas; (2) baseados em coordenadas virtualis; e (3) livres de posicionamento. Um resumo dos artigos citados é mostrado na Tabela 1.

Na primeira categoria, os protocolos requerem uma estimativa da localização global dos nós (latitude/longitude). Tal estimativa pode ser obtida equipando-se todos os nós com receptores GPS (*Global Positioning System*), através da execução de um algoritmo de localização ou mesmo através do posicionamento manual dos nós [Boukerche et al. 2007]. Nesta categoria, um dos trabalhos mais relevantes da literatura sobre protocolos de roteamento geográfico é o *Greedy Perimeter Stateless Routing (GPSR)* [Karp and Kung 2000], que utiliza a posição dos nós e o destino do pacote para tomar decisões de encaminhamento guloso, através das informações sobre os vizinhos a um salto de distância. Sua utilização consiste de dois métodos de encaminhamento de pacotes: encaminhamento guloso, usado quando aplicável, e encaminhamento por perímetro, utilizado em regiões onde o primeiro falha. Neste último método, o algoritmo é capaz de lidar com a ocorrência de mínimos locais. O *Bounded Voronoi Greedy Forwarding (BVGF)* [Xing et al. 2004], é um algoritmo de localização que realiza decisões de encaminhamento guloso baseadas na localização dos vizinhos a um salto de distância. Neste algoritmo, uma rede é modelada através de um diagrama de Voronoi onde as regiões representam as localizações dos sensores. Cada nó conhece sua posição geográfica e mantém uma tabela com os endereços dos nós vizinhos e, para manter esta tabela, cada nó divulga periodicamente, uma mensagem de *broadcast* que inclui sua localização, bem como a localizações dos vértices das regiões de Voronoi. Já no algoritmo *Energy Aware Greedy Routing (EAGR)* [Haider et al. 2007], cada nó conhece sua localização geográfica e a de seus vizinhos, seu nível de energia e de seus vizinhos e executa uma decisão local para a escolha do próximo salto não somente baseado na localização dos nós, mas também nos seus níveis de energia, de modo a aumentar o tempo de vida da rede como um todo.

Tabela 1. Comparação dos algoritmos de roteamento geográficos.

	A	B	C	D	E	F
BVGF [Xing et al. 2004], [Xing 2006]		✓				
EAGR [Haider et al. 2007]		✓		✓		
GF-RSSI [Pham et al. 2006]	✓	✓	✓			
GF-ViP [You et al. 2009]	✓					
GMFP [Panigrahi et al. 2009]	✓			✓		
GPSR [Karp and Kung 2000]		✓				✓
GRIP [Zhang et al. 2015]	✓					✓
RSSR [Boukerche et al. 2008]	✓	✓	✓			
RSSR with hole bypass capability [Oliveira et al. 2015]	✓	✓	✓			✓
RADR [Lima et al. 2013]	✓	✓	✓	✓	✓	
ARESTA	✓	✓	✓	✓	✓	✓

- A) Não requer coordenadas globais.
- B) Não requer coordenadas virtuais.
- C) Utiliza RSSI como métrica
- D) Ciente de energia.
- E) Realiza agregação de dados.
- F) Identifica buracos de roteamento.

Os algoritmos baseados em coordenadas virtuais estão na segunda categoria. Tais algoritmos tentam criar um outro tipo de sistema de coordenadas, não relacionado a um sistema de posicionamento global, onde a informação de localização não está disponível [Huang et al. 2011]. Neste contexto, o *Greedy Minimum energy consumption Forwarding Protocol (GMFP)* [Panigrahi et al. 2009] foi desenvolvido com o objetivo de aumentar o tempo de vida da rede, através da escolha ótima de um nó elegível em cada salto, utilizando encaminhamento geográfico guloso. O algoritmo *Greedy Forwarding with Virtual Position (GF-ViP)* [You et al. 2009], utiliza um esquema de encaminhamento guloso baseado no progresso do encaminhamento em direção ao *sink*. O algoritmo foi proposto com dois complementos, o *Virtual Multi-Level Position (MVP)* e o *Greedy Forwarding with Virtual Hierarchical Position (HVP)*. Cada algoritmo possui duas variações que utilizam algoritmos Gulosos baseados em progresso. A principal vantagem nestas abordagens é a aplicação do encaminhamento guloso durante todo o processo de roteamento, gerando resultados altamente eficientes na geração de rotas. Ainda nesta categoria, o *Greedy Routing through dIstributed Parametrization for guaranteed delivery in WSNs (GRIP)* proposto em [Zhang et al. 2015], utiliza a técnica de encaminhamento guloso para prover garantia de entrega através da parametrização do domínio de rede real em um domínio virtual. A abordagem utiliza um algoritmo de detecção de perímetro para identificar nós na borda da rede e aproveita essa informação para atribuir coordenadas iniciais.

Finalmente, na terceira categoria, os nós não precisam conhecer suas informações de localização nem as de seus vizinhos mas utilizam alguma outra informação para indicar a direção que o pacote deve seguir. Neste cenário, o *Greedy Forward with Received Signal Strength Indicator (GF-RSSI)* [Pham et al. 2006] utiliza a potência do RSSI recebido como um filtro usado para gerar rotas. No *Received Signal Strength Routing (RSSR)*, proposto em [Boukerche et al. 2008], o *sink* também é equipado com um dispositivo de comunicação de maior potência e seu pacote de consulta atinge todos os nós da rede em um único salto. O pacote de resposta dos nós sensores é encaminhado para o vizinho que recebeu a mensagem do *sink* com a maior potência que, na teoria, é o vizinho mais próximo do *sink*. Uma melhoria que permite ao algoritmo RSSR ser capaz de detectar e desviar buracos foi apresentada em [Oliveira et al. 2015]. Uma abordagem de encaminhamento guloso que explora a grande capacidade de comunicação do nó *sink* foi proposta em [Lima et al. 2013]. Neste trabalho, o algoritmo RADR utiliza uma técnica de encaminhamento guloso que é combinada aos valores obtidos pelo RSSI do *sink* e dos nós regulares para implementar roteamento geográfico e agregação de dados.

Nossa proposta atual, o ARESTA, difere das soluções citadas pois não requer a utilização de sistemas de coordenadas globais ou virtuais. Além disso, é capaz de lidar com buracos de roteamento durante o processo de encaminhamento das informações ao *sink*, mantendo todas as vantagens de um algoritmo de roteamento geográfico mas sem a necessidade de conhecer as informações sobre o posicionamento dos nós. Nosso algoritmo, ainda realiza agregação de dados utilizando para isso os valores de RSSI dos nós para estimar as distâncias entre vizinhos e o nó *sink*. Neste contexto, a agregação de dados foi utilizada para reduzir o tráfego de pacotes na rede e aumentar a vida útil dos nós.

3. ARESTA - Agregação de Dados com Desvio de Buracos para Redes de Sensores sem Fio com *Sinks* de Alto Alcance

Neste trabalho está sendo proposto um algoritmo de roteamento geográfico e agregação de dados, que utiliza um *sink* com grande capacidade de comunicação e é capaz de lidar com de buracos de roteamento chamado ARESTA (Agregação de Dados com desvio de buracos para Redes de Sensores sem Fio com *Sinks* de alto alcance). O algoritmo foi desenvolvido para ter todos os benefícios de um algoritmo de roteamento geográfico, como confiabilidade, escalabilidade e altas taxas de entrega, sem a necessidade da informação de posicionamento dos nós sensores, além de executar agregação de dados durante o processo de encaminhamento dos pacotes ao *sink*. Nosso algoritmo ainda é capaz de descobrir e desviar buracos de roteamento em RSSFs, através da construção de uma rota alternativa construída a partir de nossa regra de desvio de buracos executada até que um vizinho próximo ao destino seja encontrado.

Nossa solução considera a possibilidade de se equipar o nó *sink* com um dispositivo de comunicação de alto alcance de forma que em um único salto o nó *sink* pode enviar uma consulta para toda a rede, aproveitando o grande alcance na comunicação. O aspecto principal do algoritmo ARESTA é ser capaz detectar e desviar buracos de roteamento, traçando rotas alternativas até encontrar um vizinho que esteja geograficamente próximo ao destino, em nosso caso o *sink*. A motivação para o desenvolvimento desta melhoria surgiu a partir do projeto ATTO (*Amazonian Tall Tower Observatory*), que já está implementado na Amazônia e consiste em uma torre no meio da floresta com uma estrutura física de 320 m de altura e alta capacidade de comunicação. O objetivo do projeto é mapear eventos na região e fornecer medidas confiáveis de fontes e sumidouros de gases de efeito estufa como CO_2 , CH_4 e N_2O [Tollefson 2010].

Na arquitetura proposta, a mesma mensagem de consulta do nó *sink* alcançará os nós da rede com diferentes potências (RSSIs) de tal forma que os nós mais distantes possuirão um sinal mais baixo, ao contrário dos nós mais próximos, que irão receber a mensagem com um sinal mais forte, conforme ilustrado na Figura 1. A confiabilidade do RSSI, utilizado como métrica, é discutida em [Holland et al. 2006, Jacinto 2012].

A proposta, descrita no Algoritmo 1, inicia quando o nó *sink* envia uma consulta para toda a rede (linha 10). Os nós sensores, ao receberem essa consulta, estimam suas distâncias ao nó *sink* usando a técnica de RSSI (linhas 11 e 12), conforme ilustrado na Figura 2(a). Em seguida, cada nó irá enviar a seus vizinhos um pacote de anúncio, que contém a sua distância estimada para o *sink* (linha 13). Ao mesmo tempo, os nós receberão diversos pacotes deste tipo, um de cada vizinho, e salvarão tais dados de distâncias nas suas tabelas de roteamento (linha 19 e 20). Cada nó irá verificar se possui dados a serem retornados à consulta do *sink*. Se tiver, tais dados serão adicionados à lista de dados agregados (linhas 14–16). O próximo passo do algoritmo é calcular um tempo de espera para agregação dos dados antes de enviar a resposta ao *sink*. Um temporizador é iniciado com o tempo de espera calculado (linhas 17 e 18). O tempo para o envio dos pacotes é definido de acordo com a potência do RSSI recebido por cada nó, ou seja, quanto mais baixo o sinal do RSSI, menos tempo o nó terá que esperar para enviar seus pacotes em direção ao *sink*. Após a etapa de configuração, as informações agregadas a serem encaminhadas ao próximo salto são atualizadas (linha 21).

Antes do temporizador de um nó expirar, o nó corrente checa se há um vizinho

Algorithm 1 Algoritmo ARESTA

Variáveis:

- 1: $tabelaVizinhos_i = \emptyset$; {Distâncias dos vizinhos ao *sink*}
- 2: $dadosAgregados_i = \emptyset$; {Conjunto de dados agregados}
- 3: $dadosConsulta_i$; {Dados obtidos a partir da consulta do *sink*}
- 4: $tempoEnvio_i$; {Tempo de espera para envio da resposta}
- 5: $temporizador_i$; {Temporizador usado na agregação de dados}
- 6: $proximoSalto_i$; {Próximo salto para o envio dos dados agregados}
- 7: $saltoElegivel_i$; {Próximo salto elegível em caso de desvio de buraco}
- 8: $dist_i$; {Distância para o *sink*}

Evento:

- 9: *Sink* recebe da Central de Monitoramento uma $consulta(consultaId_k, consulta_k)$;

Ação:

- 10: Envia $consulta(consultaId_k, consulta_k)$ para todos os nós.

Evento:

- 11: $msg_i = consulta(consulta_k, consulta_k)$;
- 12: $dist_i = estimaDistancia(msg_i)$;

Ação:

- 13: Envia $anuncio(n_i, dist_i) \forall n_j \in vizinho_i$. {Divulgando a distância}
- 14: **Se** $dadosConsulta_i = recRetornoConsulta(consulta_k)$ **Então**
- 15: $dadosAgregados_i := dadosAgregados_i \cup dadosConsulta_i$; {Agrega informação ao pacote}
- 16: **Fim Se**
- 17: $tempoEnvio_i = estimaTempoEnvio(dist_i)$; {Tempo de agregação}
- 18: $temporizador_i.inicia(tempoEnvio_i)$; {Inicia temporizador}

Evento:

- 19: $msg_i = anuncio(vizinho_k, dist_k)$;
- 20: $tabelaVizinhos_i = tabelaVizinhos_i \cup (vizinho_k, dist_k)$;
- 21: $proximoSalto_i = vizinhoProximoAoSink()$;

Ação:

- 22: **Se** $dist_i < proximoSalto_i.dist$ **Então** {Estou em um buraco de roteamento?}
- 23: $saltoElegivel_i := desvioDeBuraco(tabelaVizinhos_i)$; {Qualifica o próximo salto como elegível}
- 24: Envia $resposta(dadosAgregados_i)$ para $saltoElegivel_i$;
- 25: **Fim Se**

Evento:

- 26: $temporizador_i.expirou()$; {Tempo para agregação expirou}

Ação:

- 27: Envia $resposta(dadosAgregados_i)$ para $proximoSalto_i$; {Encaminha dado agregado}
- 28: $dadosAgregados_i = \emptyset$;

Evento:

- 29: $msg_i = resposta(dadosAgregados_k)$;

Ação:

- 30: **Se** $n_i \neq 0$ **Então** {Sou um nó regular?}
 - 31: $dadosAgregados_i := dadosAgregados_i \cup dadosAgregados_k$;
 - 32: **Se** $temporizador_i.inativo()$ **Então** {O dado foi enviado?}
 - 33: Envia $resposta(dadosAgregados)$ para $proximoSalto_i$;
 - 34: $dadosAgregados_i = \emptyset$;
 - 35: **Fim Se**
 - 36: **Senão** {Eu sou o nó *Sink*}
 - 37: Envia o dado recebido à Central de Monitoramento.
 - 38: **Fim Se**
-

mais próximo ao *sink*. Caso contrário, isto significa que o nó está em uma região onde a utilização da técnica de encaminhamento guloso não é possível. O nó corrente então executa regra de detecção e contorno de buracos (linha 23), que seleciona um novo vizinho em direção ao *sink* a partir da tabela, ilustrado na Figura 2(b). A esquema de detecção de contorno de buracos permite que o próximo salto seja eleito de acordo com a distância do *sink* em relação a cada vizinho, considerando como elegível o vizinho geograficamente mais próximo do *sink* em relação ao nó corrente.

O nó da região de buracos não possui vizinhos elegíveis para o encaminhamento de pacotes através da técnica de encaminhamento guloso. Neste sentido, o encaminhamento de um pacote contornando o buraco (linha 24) é feito de acordo com as distâncias dos vizinhos armazenadas na tabela de roteamento de cada nó, de forma que o vizinho a ser escolhido como próximo salto esteja o mais distante possível em relação ao nó corrente, e o mais próximo possível em relação ao *sink*. Esta mesma mensagem contém o identificador de cada mensagem encaminhada ao nó que se encontra uma região de buracos. Tais informações serão utilizadas para atualizar a tabela de roteamento dos nós que permite a escolha do próximo salto. Este processo é repetido até que seja escolhido um vizinho apto a executar o encaminhamento guloso.

Quando o temporizador de um nó expira (linha 26), um pacote com todos os dados agregados é enviado ao vizinho mais próximo do nó *sink* (linha 27). Este pacote será recebido pelo vizinho escolhido (linha 29) que, provavelmente, não terá seu temporizador expirado ainda, por estar mais próximo do *sink*. Este último nó irá simplesmente agregar os dados recebidos em sua lista de dados agregados (linha 32) e continuar esperando que seu temporizador expire para enviar esses dados adiante (linhas 26–28). Finalmente, quando o *sink* recebe algum pacote de resposta, ele envia os dados recebidos de volta à central de monitoramento (linhas 37).

Em alguns casos raros, é possível que o temporizador de um determinado nó tenha expirado (e o seu dado agregado já tenha sido encaminhado) antes de receber todos os dados dos seus vizinhos. Isso pode ocorrer na presença de erros de RSSI elevados, propagados durante o processo estimativa de distância e configuração do temporizador para a agregação de dados. Caso isto ocorra, o pacote recebido é encaminhado imediatamente ao vizinho mais próximo do *sink*, através da técnica de encaminhamento guloso (linhas 32–35). Nesta proposta, utilizamos a agregação de dados com forma de reduzir o número de informações redundantes e aumentar o tempo de vida da rede, também proporcionado

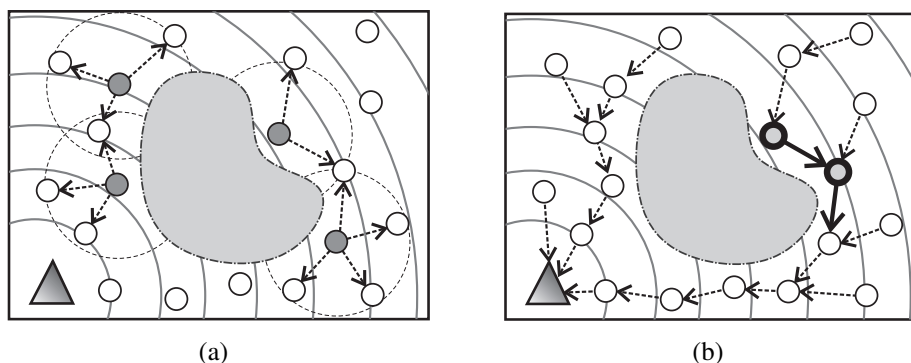
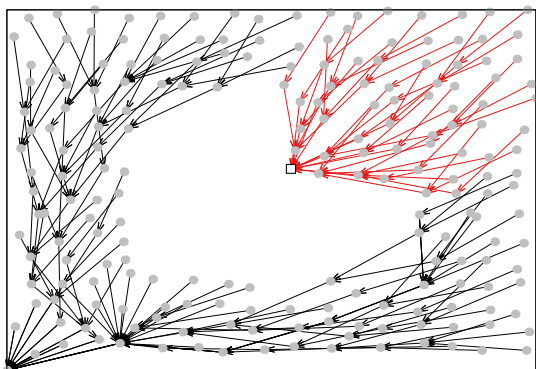


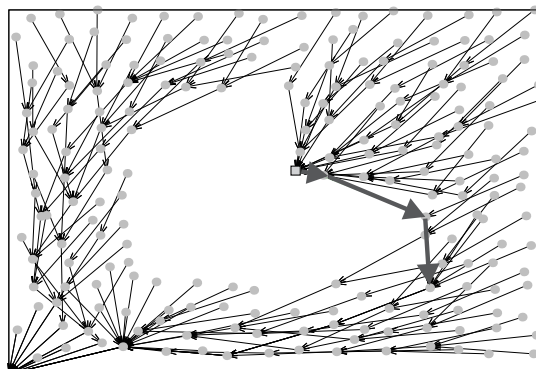
Figura 2. ARESTA: (a) Após a mensagem de anúncio do RSSI, cada nó estima sua distância em relação ao *Sink*. (b) Regra de desvio de buraco sendo aplicada.

pela economia de energia que é um benefício direto da agregação de dados.

Com o objetivo de visualizar e comparar as rotas geradas pelos algoritmos, a Figura 3(a) apresenta um gráfico de fluxo com as rotas geradas pelo algoritmo RADR, que não possui desvio de buracos, enquanto a Figura 3(b) apresenta as rotas geradas pelo algoritmo ARESTA. Os gráficos foram gerados a partir de uma simulação realizada e é possível ver claramente o fluxo com o escoamento dos pacotes formado pela abordagem proposta no cenário com a presença de um buraco (ao centro).



(a) Algoritmo RADR: nó não consegue identificar um buraco.



(b) Algoritmo ARESTA: nó detecta um buraco de roteamento e aplica a regra de desvio.

Figura 3. RADR e ARESTA: gráfico de fluxo de pacotes. Sink localizado no canto inferior esquerdo.

4. Avaliação de Performance

Na presente seção, avaliamos o algoritmo ARESTA proposto e comparamos sua performance com o algoritmo RADR apresentado em [Lima et al. 2013].

4.1. Metodologia

A avaliação de performance foi realizada através de simulações usando o simulador Sinalgo [ETH-Zurich 2014], que provê um ambiente completo para simulação de algoritmos distribuídos. As simulações foram realizadas em um campo de sensoriamento de $130\text{ m} \times 130\text{ m}^2$. O posicionamento dos nós foi feito com base em uma grade perturbada, mantendo a densidade de 0.03 nós/m^2 . Em todos os resultados, as curvas representam valores médios, enquanto as barras de erro representam intervalos de confiança para 95 % de confiança a partir de 33 instâncias independentes (sementes aleatórias).

Tabela 2. Configuração do cenário de simulação.

Parâmetro	Valor padrão
Área monitorada	$130\text{ m} \times 130\text{ m}^2$
Quantidade de nós	512 nós (grade perturbada)
Densidade	0.03 nós/m^2
Alcance da comunicação	20 m
Imprecisão do RSSI a curta distância	15 % da dist. real
Posição do Sink	Canto inferior esquerdo

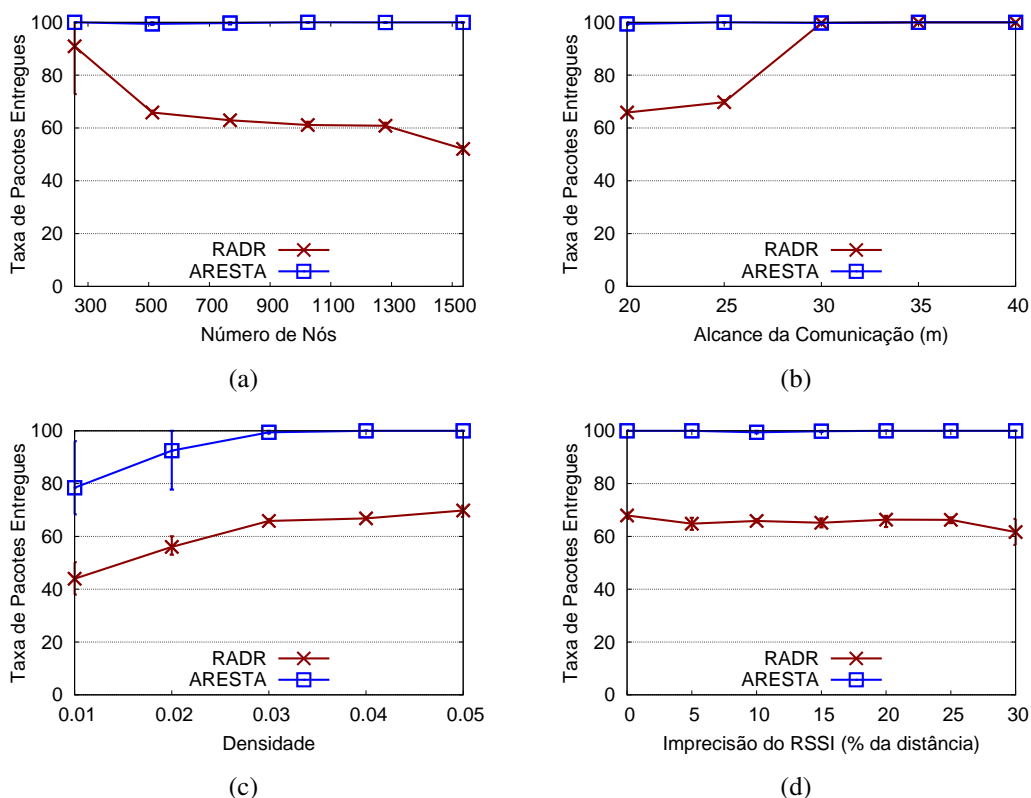


Figura 4. Resultados obtidos ao se avaliar a taxa de entrega.

O nó *sink* foi posicionado no canto inferior esquerdo enquanto que os nós regulares ocupam a grade uniformemente. Também introduzimos um buraco de roteamento na rede, conforme apresentado na Figura 2, configurado de acordo o alcance máximo dos nós vizinhos. Neste cenário, os nós em uma extremidade do buraco não podem se comunicar com nós da outra extremidade. Os parâmetros de simulação foram baseados no sensor MicaZ e os valores utilizados são mostrados na Tabela 2.

4.2. Impacto da Escalabilidade da Rede

A escalabilidade de rede foi avaliada através do aumento do número de nós na rede de 256 para 1536 nós, aumentando-se o tamanho da área monitorada de modo a manter a densidade constante. Na Figura 4(a), podemos observar que o algoritmo RADR não consegue entregar todos os dados, uma vez que não é capaz de lidar com buracos de roteamento. Por outro lado, o algoritmo ARESTA consegue entregar todos os pacotes na maioria das vezes. O resultado demonstra claramente que nossa solução é capaz de operar na presença de buracos, até mesmo quando aumentamos o número de nós.

4.3. Impacto do Raio de Comunicação

Para avaliarmos o impacto do raio de comunicação dos nós em nosso algoritmo, aumentamos este parâmetro de 20 m até 40 m. Raios de comunicação pequenos reduzem o número de nós vizinhos, o que conseqüentemente reduz o número de possibilidades para o próximo salto [Boukerche et al. 2008]. A Figura 4(b), mostra que o número de pacotes entregues com sucesso do algoritmo RADR diminui para cerca de 65 % quando o raio de comunicação é 20 m e fica em torno de 70 % quando o alcance aumenta para 25 m.

Neste caso, os nós possuem poucos vizinhos elegíveis, gerando ainda mais buracos de rede. Neste cenário, o algoritmo RADR interrompe o processo de encaminhamento guloso. Contudo, em todos os casos nosso algoritmo ARESTA é capaz de entregar todos os dados agregados.

4.4. Impacto da Densidade da Rede

O impacto da densidade da rede em nosso algoritmo foi avaliado começando com uma densidade relativamente baixa de 0.01 nós/ m^2 e aumentando tal densidade até 0.05 nós/ m^2 . Em densidades baixas, um nó terá poucos vizinhos, dificultando a agregação e consequentemente o encaminhamento do pacote adiante, enquanto que ao se aumentar a densidade, o número de vizinhos de cada nó aumentará. Podemos observar pela Figura 4(c) que o algoritmo ARESTA continua funcionando corretamente e é capaz de operar de forma eficiente mesmo em baixas densidades ao contrário do algoritmo RADR, que em baixas densidades consegue entregar cerca de 40 % dos pacotes. Por fim, observamos que número de pacotes entregues com sucesso pelo algoritmo ARESTA é de cerca de 80 % quando a densidade está em 0.01 nós/ m^2 . Neste caso, os nós possuem poucos vizinhos e isso aumenta as chances de escolha de um salto elegível para o encaminhamento dos pacotes.

4.5. Impacto da Imprecisão do RSSI a curta distância

Neste trabalho avaliamos o impacto da imprecisão do RSSI a curta distância (estimativa de distância entre os nós regulares) aumentando essa imprecisão de 0 % para 30 % da distância real entre os nós. Como é possível observar na Figura 4(d), o algoritmo ARESTA não é afetado de forma significativa pela imprecisão do RSSI, além de se mostrar confiável no encaminhamento de pacotes ao *sink*. Por outro lado, o algoritmo RADR é afetado durante o processo de estimativa de distância entre os vizinhos prejudicando as decisões de próximo salto a partir de uma área com a presença de um buraco.

4.6. Impacto da Regra de Desvio de Buracos na escolha do próximo salto.

O impacto da solução de desvio de buracos em relação à escolha do próximo salto também foi avaliado. Neste sentido, comparamos os resultados obtidos através da execução dos algoritmos RADR e ARESTA na presença de um buraco de roteamento. A grade foi configurada com a inserção de um polígono côncavo irregular em forma de *L*, ao centro conforme Figura 2, de forma que os nós de um lado do buraco não se comunicassem com os nós do outro lado, portanto a parte oposta do buraco. Esta configuração pode ser considerada como o pior cenário, já que um buraco baseado em uma circunferência não cria um buraco de roteamento nem mesmo denota o fenômeno dos mínimos locais. Outros tipos de polígonos não podem criar cenários de pior caso para o problema explorado neste trabalho.

O número de escolhas erradas em relação ao próximo salto no algoritmo ARESTA é um pouco maior do que no algoritmo RADR com poucos nós, como mostram as Figuras 5(a), 5(b), 5(c) e 5(d), pois no desvio de buracos os pacotes são encaminhados a vizinhos que não estão geograficamente mais próximos do *sink* e sim na borda do buraco. Apesar disto, o algoritmo ARESTA é capaz de gerar árvores de escoamento eficientes enquanto em direção *sink*, contornando a borda do buraco como já mostrado na Figura 3(b).

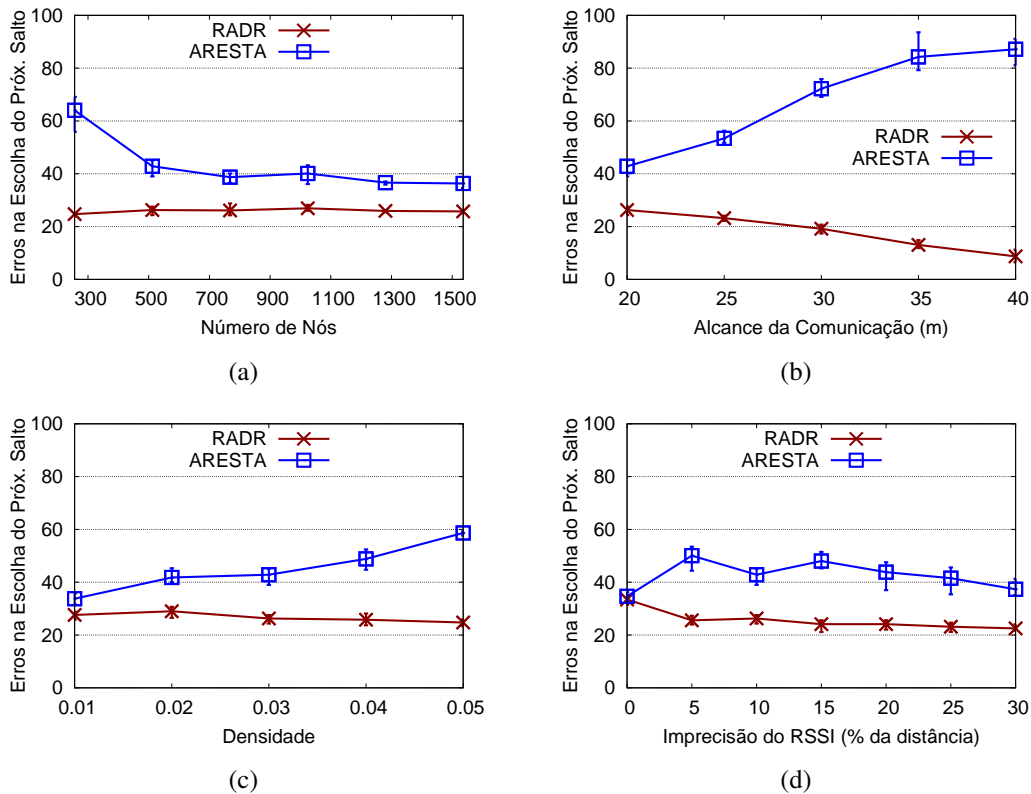


Figura 5. Resultados obtidos ao se avaliar escolha do próximo salto.

5. Aplicabilidade da Solução Proposta

Em cenários de Redes de Sensores em Fio, alguns obstáculos naturais como lagos, construções ou regiões sem a presença de sensores podem causar a interrupção do encaminhamento de pacotes. Em algoritmos de roteamento geográfico, a existência de uma área em que os nós regulares não conseguem encaminhar adiante seus pacotes de dados é problema dos mínimos locais, amplamente estudado em [Choi and Choo 2011, Chang et al. 2012]. Neste trabalho consideramos a possibilidade de se equipar o nó *sink* com um dispositivo de comunicação de maior potência, de forma que em um único salto toda a rede seja alcançada por uma consulta, também única do *sink*. Esta é uma suposição razoável em cenários como o do projeto ATTO (Amazonian Tall Tower Observatory), em que pretendemos utilizar e melhorar esta arquitetura de comunicação. Neste contexto, o algoritmo ARESTA pode ser utilizado como uma solução de roteamento guloso para o encaminhamento de pacotes, a partir de nós sensores implantados ao redor da torre. Por exemplo, caso haja obstáculos naturais no ambiente sensoreado, o ARESTA estaria apto a detectar e aplicar a regra de desvio. Neste sentido, o algoritmo ARESTA proporciona um aumento do tempo de vida dos nós que participam da agregação de dados.

6. Conclusão

Neste trabalho propomos um novo algoritmo de roteamento geográfico com agregação de dados para redes de sensores, que chamamos de ARESTA (Agregação de Dados com desvio de buracos para Redes de Sensores sem Fio com *Sinks* de alto alcance). O aspecto essencial desta abordagem é equipar o nó *sink* com um potente dispositivo de comunicação de forma que em um único pacote de anúncio todos os nós da rede sejam alcançados.

Com base apenas nos valores de RSSI obtidos pelos nós, propomos uma nova e eficiente abordagem o roteamento e agregação de dados, com a capacidade de lidar com buracos de roteamento.

O algoritmo ARESTA, é capaz de detectar e superar buracos em uma RSSF através da construção de uma rota alternativa até o vizinho mais próximo do *sink*. A agregação de dados ocorre durante o processo de encaminhamento dos pacotes e é possível graças à utilização de um temporizador configurado em cada nó regular após a consulta inicial recebida pelo nó *sink*. Uma série de experimentos foi realizada para avaliar a solução proposta. Os resultados demonstram claramente os benefícios introduzidos pela técnica de desvio de buracos, somada a abordagem simples de encaminhamento guloso e agregação de dados.

Referências

- Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002a). A survey on sensor networks. *IEEE, Communications Magazine*, 40:102–114.
- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002b). Wireless sensor networks: a survey. *Computer Networks*, 38:393–422.
- Al Alawi, R. (2011). Rssi based location estimation in wireless sensors networks. In *17th IEEE International Conference on Networks (ICON), 2011*, pages 118–122.
- Boukerche, A., Oliveira, H., Nakamura, E., and Loureiro, A. (2007). Localization Systems for Wireless Sensor Networks. *IEEE Wireless Communications, 2007*, 14(6):6–12.
- Boukerche, A., Oliveira, H., Nakamura, E., and Loureiro, A. (2008). A Novel Location-Free Greedy Forward Algorithm for Wireless Sensor Networks. In *IEEE International Conference on Communications (ICC), 2008*.
- Chang, C.-Y., Hung, L.-L., Wang, T.-C., and Wang, T.-L. (2012). Orzbp: An obstacle-resistant zone-based broadcasting protocol for wireless sensor networks. In *14th International Conference on High Performance Computing and Communication, 2012*, pages 714–721.
- Chen, D. and Varshney, P. (2007). A survey of void handling techniques for geographic routing in wireless networks. *Communications Surveys Tutorials, IEEE*, 9(1):50–67.
- Choi, M. and Choo, H. (2011). Bypassing hole scheme using observer packets for geographic routing in wsns. In *Information Networking (ICOIN), 2011 International Conference on*, pages 435–440.
- Éfren Souza, Nakamura, E., Oliveira, H., and Figueiredo, C. M. S. (2012). Reducing the impact of location errors for target tracking in wireless sensor networks. *Journal of the Brazilian Computer Society*.
- ETH-Zurich, D. C. G. a. (2014). Sinalgo - simulator for network algorithms. In <http://dcg.ethz.ch/projects/sinalgo/>.
- Haider, R., Javed, M., and Khattak, N. (2007). Eagr: Energy aware greedy routing in sensor networks. In *Future Generation Communication and Networking (FGCN 2007)*, volume 2, page 344349.

- Holland, M., Aures, R., and Heinzelman, W. (2006). Experimental investigation of radio performance in wireless sensor networks. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pages 140–150.
- Huang, P., Wang, C., and Xiao, L. (2011). Improving end-to-end routing performance of greedy forwarding in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, PP(99):1.
- Jacinto, R. M. P. (2012). Modelação da Propagação numa Rede de Sensores sem Fios. Master's thesis, Universidade Nova de Lisboa.
- Karp, B. and Kung, H. T. (2000). Gpsr: Greedy perimeter stateless routing for wireless networks. In *MOBICOM*, pages 243–254.
- Kim, Y.-J., Govindan, R., Karp, B., and Shenker, S. (2005). Geographic routing made practical. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 217–230, Berkeley, CA, USA. USENIX Association.
- Lima, M., Oliveira, H., Nakamura, E., and Loureiro, A. (2013). Roteamento e Agregação de Dados baseado no RSSI em Redes de Sensores sem Fio. In *31^o Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC 2013)*, Brasilia, DF.
- Oliveira, H. A., Boukerche, A., Guidoni, D. L., Nakamura, E. F., Mini, R. A., and Loureiro, A. A. (2015). An enhanced location-free greedy forward algorithm with hole bypass capability in wireless sensor networks. *J. Parallel Distrib. Comput.*, 77(C):1–10.
- Panigrahi, B., De, S., and Sun Luk, J.-D. (2009). A greedy minimum energy consumption forwarding protocol for wireless sensor networks. In *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, pages 1–6.
- Pham, N., Youn, J., and Won, C. (2006). A comparison of wireless sensor network routing protocols on an experimental testbed. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006*, volume 2, pages 276–281.
- Tollefson, J. (2010). A towering experiment an ambitious project to track greenhouse gases from a perch high above the amazon forest will provide crucial data - but only if scientists can get it built. *Nature*, 467(386).
- Xing, G. (2006). *Unified power management in wireless sensor networks*. PhD thesis, St. Louis, MO, USA. AAI3238702.
- Xing, G., Lu, C., Pless, R., and Huang, Q. (2004). On greedy geographic routing algorithms in sensing-covered networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '04*, pages 31–42, New York, NY, USA. ACM.
- You, J., Lieckfeldt, D., Han, Q., Salzmann, J., and Timmermann, D. (2009). Look-ahead geographic routing for sensor networks. In *IEEE International Conference on Pervasive Computing and Communications (PerCom), 2009*, pages 1–6.
- Zhang, M., Li, F., He, Y., Lin, J., Gu, X., and Luo, J. (2015). GRIP: Greedy Routing through dIstributed Parametrization for guaranteed delivery in WSNs. *Springer Wireless Networks*, 21(1):67–80.