

## Roteamento e Agregação de Dados baseado no RSSI em Redes de Sensores Sem Fio

Moisés M. Lima<sup>1,2</sup>, Horácio A. B. F. de Oliveira<sup>1</sup>, Eduardo F. Nakamura<sup>1,3</sup> e Antônio A. F. Loureiro<sup>4</sup>

<sup>1</sup>Instituto de Computação – Universidade Federal do Amazonas

<sup>2</sup>Coordenação de Tecnologia da Informação – Instituto Nacional de Pesquisas da Amazônia

<sup>3</sup>Centro de Análise, Pesquisa e Inovação Tecnológica – FUCAPI

Manaus – Amazonas – Brasil

<sup>4</sup>Departamento de Ciência da Computação – Universidade Federal de Minas Gerais  
Belo Horizonte – Minas Gerais – Brasil

{moyses.lima, horacio, nakamura}@icompu.ufam.edu.br, loureiro@dcc.ufmg.br

**Abstract.** *Geographic routing protocols have been considered as one of the most viable solution for routing in Wireless Sensor Networks (WSNs) due to their high scalability, dynamism, and high data delivery rate. These algorithms refer to nodes by position rather than address and use these coordinates to discover routes to the sink node. The main drawback of these algorithms is the need for position information of the nodes, which can be expensive in several ways. In this paper, we go further and propose the RADR (Roteamento e Agregação de Dados baseado no RSSI) algorithm, a new geographic-based routing algorithm that does not require position information while also adds data aggregation functionality to the network. Our algorithm takes advantage of the greater communication range of the sink node as well as the Received Signal Strength Indicator (RSSI) of the sensor nodes to configure routing paths and aggregation times back to the sink node. Our results indicate clearly that the proposed algorithm reduces the amount of redundant data and the number of transmissions in the network while maintaining all advantages of these kind of algorithms.*

**Resumo.** *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 possuem uma alta taxa de entrega de dados. Tais algoritmos referenciam a posição dos nós ao invés do endereço e utilizam essas coordenadas para descobrirem rotas para o nó sink. A principal desvantagem destes algoritmos é a necessidade de localização, que pode ser custoso em diversos aspectos. Neste trabalho, vamos além e propomos o algoritmo RADR (Roteamento e Agregação de Dados baseado no RSSI), um novo algoritmo de roteamento geográfico que não requer posição dos nós enquanto ainda provê a funcionalidade de agregação de dados na rede. Nosso algoritmo tira vantagem da possibilidade de maior alcance de comunicação do nó sink, bem como do indicador de potência do sinal recebido (RSSI) dos nós sensores para configurar rotas e tempos de agregação de dados em direção ao nó sink. Os resultados obtidos mostram claramente que o algoritmo proposto reduz a quantidade de dados redundantes e o número de transmissões na rede enquanto mantém todas as vantagens deste tipo de algoritmo.*

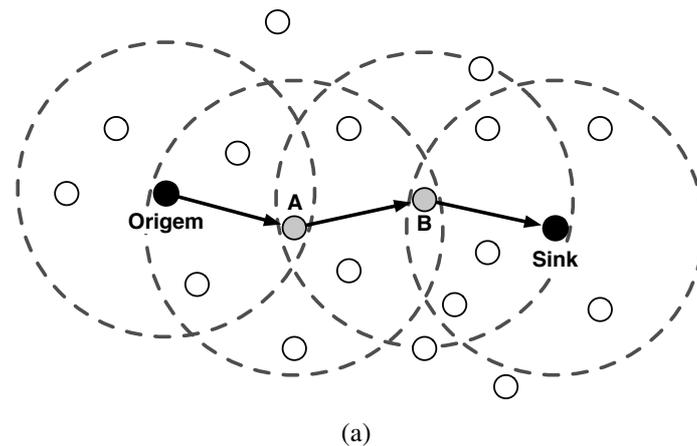


Figura 1. Algoritmo roteamento geográfico guloso. A cada passo, o pacote é repassado para o vizinho mais geograficamente próximo do nó *sink*. Baseado em [Park et al. 2010].

## 1. Introdução

Uma Rede de Sensores Sem Fio (RSSF) consiste em um conjunto de sensores, dispostos em um campo de sensoreamento em contato ou próximos a um evento ou fenômeno a ser sensoreado. Tais redes têm um grande impacto em aplicações que envolvem o monitoramento de condições ambientais tais como temperatura, luminosidade, movimento e presença de certos tipos de objetos. Nestas em na maioria das outras aplicações, os nós sensores possuem limitações de energia e largura de banda. Desta forma, técnicas de redução do consumo de energia e comunicação são necessárias em muitos cenários [Al-karaki and Kamal 2004].

Um dos objetivos principais no projeto de uma RSSF é prover a comunicação de dados entre os nós sensores e o nó *sink*. O desafio é prolongar a vida útil da rede mantendo uma qualidade na comunicação e na entrega dos dados [Akkaya and Younis 2005]. Um dos pontos chaves observados em diversos estudos é que os protocolos de roteamento em RSSFs irão diferir de acordo com a aplicação e a arquitetura da rede.

Recentemente, algoritmos que exploram informações geográficas, chamados Algoritmos de Roteamento Geográfico [Rao et al. 2003], têm sido propostos para o transporte de dados nas RSSFs por serem escaláveis, dinâmicos e possuírem uma alta taxa de entrega de dados. Estes algoritmos se baseiam na posição física dos nós da rede ao invés de seus endereços para criar rotas em direção ao nó *sink*. Dentre os algoritmos mais usados no roteamento geográfico está o algoritmo guloso de roteamento geográfico (*Greedy Geographic Routing*) [Xing et al. 2004]. Tal técnica realiza o roteamento dos pacotes através do encaminhamento do pacote ao nó vizinho mais geograficamente próximo do nó *sink* (Figura 1).

Um dos pontos fracos do roteamento geográfico é a necessidade de localização dos nós, que pode ser custoso e susceptível a grandes erros em RSSFs [Souza et al. 2010]. Além disso, tais algoritmos em geral não possuem agregação de dados, o que os tornam inviáveis para RSSFs em que grande parte dos nós possuem dados a serem enviados ao *sink* [Al-karaki and Kamal 2004]. Desta forma, neste trabalho propomos um novo algoritmo de roteamento geográfico com agregação de dados que não requer informações de posicionamento dos nós: O RADR (Roteamento e Agregação de Dados baseado no

RSSI). A ideia principal do algoritmo proposto é considerar a possibilidade de se equipar o nó *sink* com um dispositivo de comunicação de maior potência para alcançar todos os nós da rede em um único salto. Entretanto, a resposta/retorno dos nós sensores para o nó *sink* é realizada através de múltiplos saltos utilizando nossa abordagem de roteamento guloso que tira proveito do RSSI (*Received Signal Strength Indicator*) para calcular o próximo salto do roteamento em direção ao nó *sink* bem como para calcular o tempo de espera para possibilitar a agregação de dados.

O restante deste trabalho está organizado como segue. Na Seção 2, apresentamos os trabalhos relacionados bem como uma classificação dos algoritmos estudados. A Seção 3 descreve o algoritmo RADR, cuja avaliação de performance é mostrada na Seção 4. Na Seção 5, realizamos uma breve discussão sobre a aplicabilidade do algoritmo proposto e, finalmente, na Seção 6 apresentamos nossas conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

Os protocolos de roteamento geográficos podem ser divididos em três categorias: (1) baseados em coordenadas geográficas; (2) baseados em coordenadas virtuais; e (3) livres de posicionamento.

Na primeira categoria, os protocolos requerem uma estimativa da localização global dos nós (latitude/longitude). Tal estimativa deve ser obtida (1a) equipando-se todos os nós com receptores GPS (*Global Positioning System*); (1b) ou através da execução de um algoritmo de localização; (1c) ou mesmo através do posicionamento manual dos nós [Boukerche et al. 2007]. Nesta categoria, 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 *Geographic Random Forwarding (GeRaF)* [Zorzi and Rao 2003], quando um nó possui um pacote a ser enviado, este envia o dado através de uma mensagem de *broadcast*, contendo sua própria localização e a localização do destino pretendido para só então utilizar o método de transmissão pelo menor esforço. 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.

Na segunda categoria estão os algoritmos baseados em coordenadas virtuais. Esses algoritmos tentam criar um outro tipo de sistema de coordenadas, que não está relacionado ao sistema de posicionamento global, onde a informação de localização não está disponível [Rao et al. 2003]. Neste cenário, o *Greedy Minimum energy consumption Forwarding Protocol (GMFP)* [Panigrahi et al. 2009] foi concebido 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,

Takagi and Kleinrock 1984], utiliza o 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 baseado em progresso *Most Forwarding Within Radius (MFR)*. 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 Embedding Spring Coordinates (GSpring)* [Leong et al. 2007] gera coordenadas virtuais úteis e viáveis, com o objetivo de gerar rotas eficientes e detectar buracos de cobertura. A abordagem usa um algoritmo de detecção de perímetro para identificar nós no perímetro da rede e usa 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 contexto, 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. Já no *Received Signal Strength Routing (RSSR)* [Boukerche et al. 2008], algoritmo que serviu de base para o proposto no presente trabalho, 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*.

Nossa proposta difere das duas primeiras categorias, pois nossa abordagem não necessita de informações sobre localização dos nós. Assim como o algoritmo *RSSR*, a ideia principal do algoritmo proposto no presente trabalho é aproveitar a grande capacidade do nó *sink* e equipá-lo com um dispositivo de comunicação de maior potência. Entretanto, nossa abordagem se diferencia desta última por implementar um esquema de agregação de dados durante o envio das mensagens. Atualmente não existem soluções que explorem essa combinação de capacidades. Um resumo dos artigos citados é mostrado na Tabela 1.

### **3. RADR - Roteamento e Agregação de Dados baseado no RSSI**

Neste artigo, está sendo proposto um novo algoritmo de roteamento geográfico com agregação de dados para redes de sensores sem fio que não requer informações de posicionamento dos nós: o RADR (Roteamento e Agregação de Dados baseado no RSSI). A proposta do algoritmo é possuir todas as vantagens de um algoritmo de roteamento geográfico tais como robustez, escalabilidade, dinamicidade e entrega confiável dos dados, mas sem precisar das informações de posicionamento dos nós ao mesmo tempo que provê o recurso de agregação de dados na rede.

A presente proposta considera a possibilidade de se equipar o nó *sink* com um dispositivo de comunicação de maior potência. Desta forma, em um único salto, o nó *sink* pode enviar uma consulta para toda a rede. A ideia desta arquitetura de comunicação 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 foco do projeto é mapear

Tabela 1. Comparação dos algoritmos de roteamento geográficos.

	A	B	C	D	E
BVGF [Xing et al. 2004], [Xing 2006]		✓			
EAGR [Haider et al. 2007]		✓		✓	
GeRaF [Zorzi and Rao 2003]		✓			
GF-RSSI [Pham et al. 2006]	✓	✓	✓		
GF-ViP [You et al. 2009]	✓				
GMFP [Panigrahi et al. 2009]	✓			✓	
GSpring [Kermarrec and Tan 2010]	✓				
RSSR [Boukerche et al. 2008]	✓	✓	✓		
<b>RADR</b>	✓	✓	✓	✓	✓

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

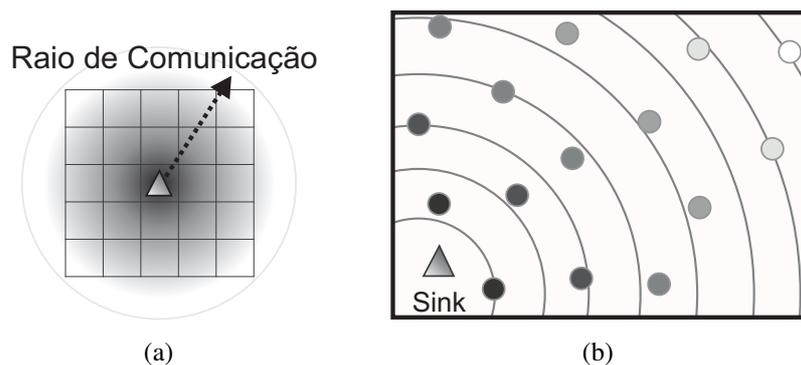


Figura 2. Potência do sinal decrescendo à medida que a mensagem de consulta do *sink* se propaga na rede. Nós sensores localizados mais distantes do *sink* receberão a mensagem com RSSI menor que os nós mais próximos. Baseado em [Boukerche et al. 2008].

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), sendo 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 nas Figuras 2(a) e 2(b).

Neste trabalho, o RSSI foi utilizado para estimar a distância entre os nós regulares e o *sink*. A aferição é baseada na potência do sinal recebido, de forma que nós mais distantes do *sink* receberam menor valor de RSSI (devido à propagação do sinal). A confiabilidade do RSSI, utilizado como métrica, é discutida em [Holland et al. 2006] e [Jacinto 2012].

O algoritmo proposto (definido no Algoritmo 1) inicia quando o nó *sink* envia uma consulta para toda a rede (linha 9). Os nós sensores, ao receberem essa consulta, estimam

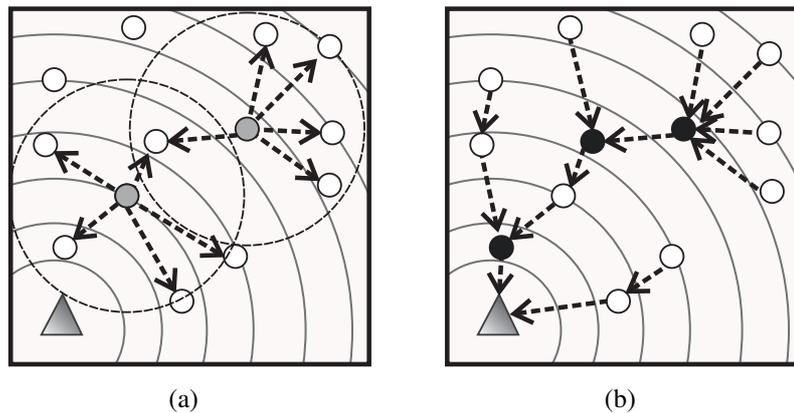


Figura 3. Exemplo do funcionamento do algoritmo RADR.

suas distâncias ao nó *sink* usando a técnica de RSSI (linhas 10 e 11). Em seguida, cada nó irá enviar a seus vizinhos um pacote de anúncio, que contém a sua distância estimada para o *sink* (linha 12). 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 18 e 19, ilustrado na Figura 3(a)). Em seguida, 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 13-15). 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 16 e 17). 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*.

Quando o temporizador de um nó expira (linha 20), ele prepara um pacote com todos os dados agregados e envia para o vizinho mais próximo do nó *sink* (linhas 20-23). Este pacote será recebido pelo vizinho escolhido (linha 24) 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 e continuar esperando que seu temporizador expire para enviar esses dados adiante (linha 26). Finalmente, quando o *sink* recebe algum pacote de resposta, ele envia os dados recebidos de volta à central de monitoramento (linhas 27-29).

Como este procedimento é executado em toda a rede, em vários casos nós vizinhos poderão escolher o mesmo nó de encaminhamento, ou seja, o vizinho eleito como mais próximo do *sink*. Desta forma, o nó de encaminhamento irá receber pacotes de vizinhos distintos, agregar as informações recebidas e repassar apenas um único pacote agregado ao próximo nó de encaminhamento. Este procedimento é repetido até que o pacote atinja o *sink*, conforme ilustrado na Figura 3(b). Desta forma, toda a rede pode responder a mensagem enviada pelo *sink* usando o esquema de comunicação multi-saltos e com agregação de dados.

Com o objetivo de visualizar e comparar as rotas geradas pelos algoritmos, a Figura 4(a) apresenta um gráfico de rotas gerado pela árvore inversa, gerada pelo algoritmo *Flooding+Agregação*, enquanto a Figura 4(b), ilustra o gráfico de rotas gerado pelo algoritmo *RADR*. Os gráficos foram gerados a partir de uma simulação realizada e

---

**Algoritmo 1** Algoritmo do RADR.
 

---

**Variáveis:**

- |                                     |  |
|-------------------------------------|--|
| 1: $n_i$ ;                          | {Id do nó atual. Para o nó <i>sink</i> , $n_i = 0$ } |
| 2: $tabelaVizinhos_i = \emptyset$ ; | {Distâncias dos vizinhos ao <i>sink</i> }            |
| 3: $dadosAgregados_i = \emptyset$ ; | {Dados agregados para serem enviados}                |
| 4: $dadosConsulta_i$ ;              | {Dados obtidos a partir da consulta do <i>sink</i> } |
| 5: $tempoEnvio_i$ ;                 | {Tempo de espera para o envio da resposta}           |
| 6: $temporizador_i$ ;               | {Temporizador de espera para envio da resposta}      |
| 7: $proximoSalto_i$ ;               | {Próximo salto para enviar os dados agregados}       |

**Evento:**

- 8: *Sink* recebe da central de monitoramento uma requisição:  $requisicao(consultaId_k, consulta_k)$ ;

**Ação:**

- 9: Envia  $consulta(consultaId_k, consulta_k)$ ; {*Sink* envia a consulta para todos os nós}

**Evento:**

- 10:  $msg_i = consulta(consultaId_k, consulta_k)$ ; tal que {Nó recebe consulta do *sink*}
- 11:  $dist_i = RSSI(msg_i)$ ; {Calcula distância ao *sink* com base no RSSI}

**Ação:**

- 12: Envia  $anuncio(n_i, dist_i) \forall n_j \in vizinhos_i$ . {Nó envia pacote de anúncio com sua distância}
- 13: **Se**  $dadosConsulta_i = executaConsulta(consulta_k)$  **Então** {Tenho dados para responder ao *sink*?}
- 14:  $dadosAgregados_i := dadosAgregados_i \cup dadosConsulta_i$ ; {Atualiza dados agregados}
- 15: **Fim Se**
- 16:  $tempoEnvio_i = estimaTempoEnvio(dist_i)$ ; {Configura o tempo de espera para agregação}
- 17:  $temporizador_i.inicia(tempoEnvio_i)$ ; {Inicia o temporizador de espera}

**Evento:**

- 18:  $msg_i = anuncio(vizinho_k, dist_k)$ ; {Recebido pacote de anúncio}

**Ação:**

- 19:  $tabelaVizinhos_i = tabelaVizinhos_i \cup (vizinho_k, dist_k)$ ; {Adiciona vizinho na lista}

**Evento:**

- 20:  $temporizador_i$  expirado; {Acabou o tempo de agregação. Vamos enviar a resposta}

**Ação:**

- 21:  $proximoSalto_i = vizinhoMaisPertoDoSink()$ ; {Próximo salto será o vizinho mais perto do *sink*}
- 22: Envia  $resposta(dadosAgregados_i)$  para  $proximoSalto_i$ ; {Envia dados agregados}
- 23:  $dadosAgregados_i = \emptyset$ ; {Limpa a lista de dados agregados}

**Evento:**

- 24:  $msg_i = resposta(dadosAgregados_k)$ ; {Recebi um pacote com dados agregados de um vizinho}

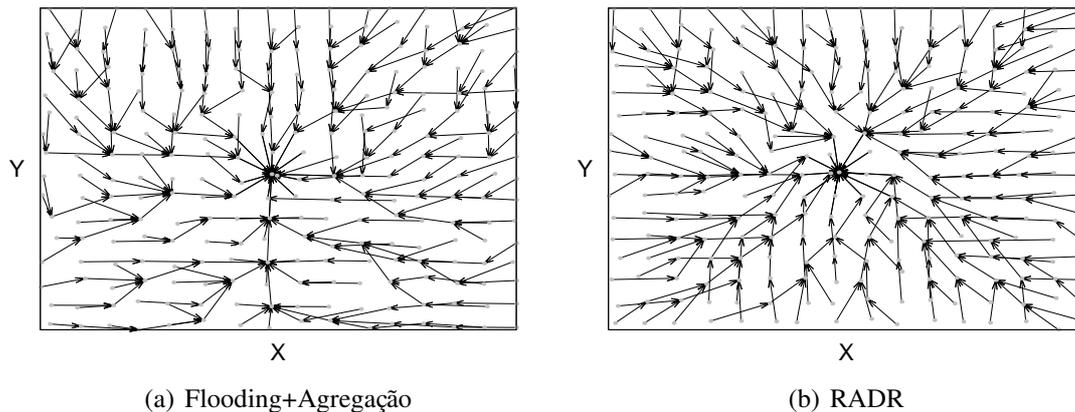
**Ação:**

- 25: **Se**  $n_i \neq 0$  **Então** {Sou um nó normal?}
- 26:  $dadosAgregados_i := dadosAgregados_i \cup dadosAgregados_k$ ; {Atualiza meus dados agregados}
- 27: **Senão** {Sou o nó *sink*}
- 28: Envia os dados recebidos para a central de monitoramento.
- 29: **Fim Se**
- 

é possível ver claramente os caminhos de roteamento guloso formados pela abordagem proposta usando apenas as informações de RSSI.

#### 4. Avaliação de Performance

Na presente seção, avaliamos o algoritmo RADR proposto e comparamos sua performance com dois outros algoritmos: o *RSSR Selection* [Boukerche et al. 2008] e o *Flooding+Agregação*. O *RSSR Selection* serviu de base para a construção da nossa abordagem, mas não implementa agregação de dados. Já o *Flooding+Agregação* é um algo-



**Figura 4. Fluxo de pacotes encaminhados ao sink**

ritmo básico de agregação de dados que usa o *flooding* clássico para divulgar a consulta do *sink* em múltiplos saltos e o retorno dos nós é feito usando a árvore inversa do *flooding* e realizando agregação de dados nos nós que possuem filhos.

A ideia de se utilizar esses dois algoritmos como base de comparação é que o primeiro permitirá avaliar o avanço do nosso novo algoritmo ao se adicionar a nova técnica de agregação de dados, enquanto que o segundo permitirá avaliar o ganho do algoritmo proposto em relação a outro que já faz agregação. O foco de nossa avaliação está na aplicação da nova técnica de encaminhamento guloso bem como na técnica de agregação de dados, que é baseada na métrica para o cálculo do tempo de espera para agregação. Espera-se também entender o funcionamento do algoritmo proposto nos mais diversos cenários.

#### 4.1. Metodologia

A avaliação de performance foi realizada através de simulações usando o simulador Sinalgo [ETH-Zurich 2012], que provê um ambiente completo para simulação de algoritmos distribuídos. As simulações foram realizadas em um campo de sensoriamento de  $130\text{ m} \times 130\text{ m}$ . O posicionamento dos nós foi feito com base em uma grade perturbada de forma a evitar buracos na rede e mantendo a densidade de  $0.03\text{ nós/m}^2$ . Os parâmetros de simulação foram baseados no sensor MicaZ e os valores utilizados são mostrados na Tabela 2.

**Tabela 2. Cenário padrão de simulação.**

Parâmetro	Valor padrão
Área monitorada	$130\text{ m} \times 130\text{ m}^2$
Quantidade de nós	512 nós (grade perturbada)
Densidade	$0.03\text{ nós/m}^2$
Alcance da comunicação	40 m
Imprecisão do RSSI a curta distância	5 % da dist. real
Posição do Sink	Centro da área monitorada

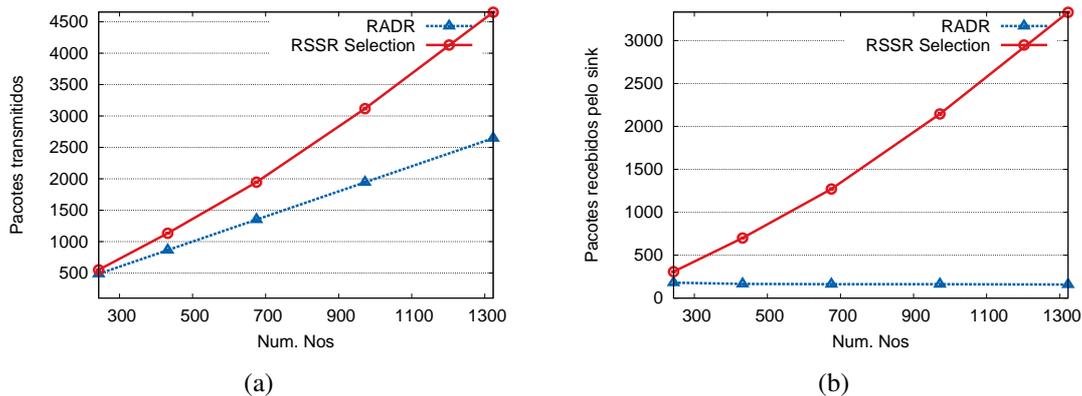


Figura 5. Resultados obtidos ao se avaliar o impacto da escalabilidade da rede.

## 4.2. Impacto da Escalabilidade da Rede

A escalabilidade de rede foi avaliada através do aumento do número de nós na rede de 243 para 1343 nós, aumentando-se o tamanho da área monitorada de modo a manter a densidade constante. Em todos os casos, a taxa de entrega dos dados ao *sink* ficou acima de 95% e, portanto, não mostraremos o gráfico. Entretanto, pudemos confirmar que a solução proposta mantém a boa confiabilidade obtida pelos outros algoritmos estudados. Já o gráfico da Figura 5(a) compara a quantidade de pacotes transmitidos na rede. Pode-se perceber que o número de transmissões do algoritmo proposto cresce muito mais lentamente do que o *RSSR* ao se aumentar o número de nós na rede, o que mostra que a nova técnica de agregação de dados proposta pelo RADR é capaz de gerar grandes reduções no consumo de energia.

O gráfico da Figura 5(b) mostra a quantidade de pacotes recebidos pelo nó *sink*. Neste resultado, fica evidente que o número de pacotes recebidos pelo *sink* permanece praticamente constante mesmo ao aumentar bastante o número de nós, mostrando novamente a viabilidade da técnica proposta de agregação de dados. Neste contexto, o número de pacotes entregues se limita ao número de vizinhos do nó *sink* que possuem o mesmo raio de comunicação. Nestes e nos resultados seguintes, até que seja especificado o contrário, o algoritmo de agregação de dados *Flooding+Agregação* obteve os mesmos resultados do RADR, confirmando o bom funcionamento deste último e, portanto, tais resultados não foram colocados nos gráficos. Mais para o final da avaliação, mostraremos em quais aspectos o nosso algoritmo se destaca em relação ao *Flooding+Agregação*.

## 4.3. Impacto do Raio de Comunicação

Para avaliarmos o impacto do raio de comunicação dos nós em nosso algoritmo, começamos com um raio relativamente pequeno, de 20 m, e o aumentamos para 40 m. Raios de comunicação pequenos reduzem o número de nós vizinhos, o que consequentemente reduz o número de possibilidades para o próximo salto [Boukerche et al. 2008]. Na Figura 6(a), é possível observar que o RADR consegue manter o número de pacotes transmitidos na rede bem baixo, mesmo quando há baixa densidade de nós, o que poderia dificultar a agregação de dados. Podemos observar ainda a grande economia no número de transmissões em relação ao *RSSR* que, ao se aumentar o raio de comunicação, transmite menos pacotes, uma vez que os pacotes serão entregues diretamente ao *sink*.

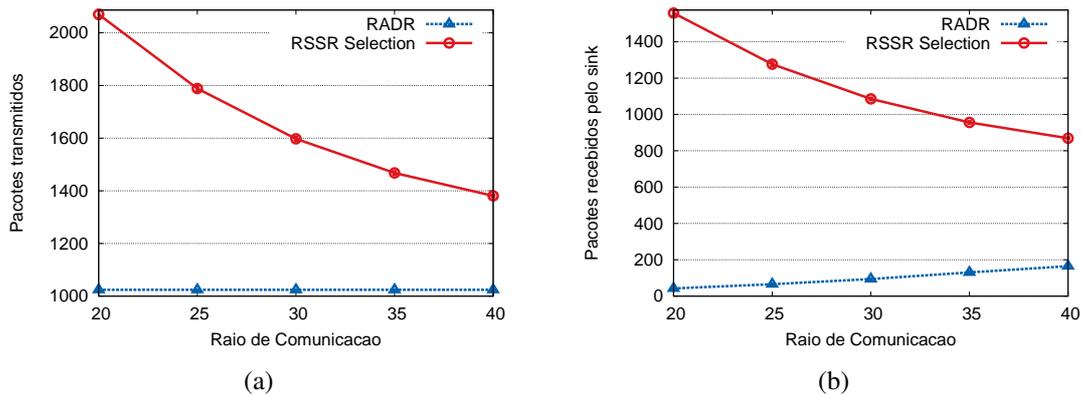


Figura 6. Resultados obtidos ao se avaliar o impacto do raio de comunicação dos nós.

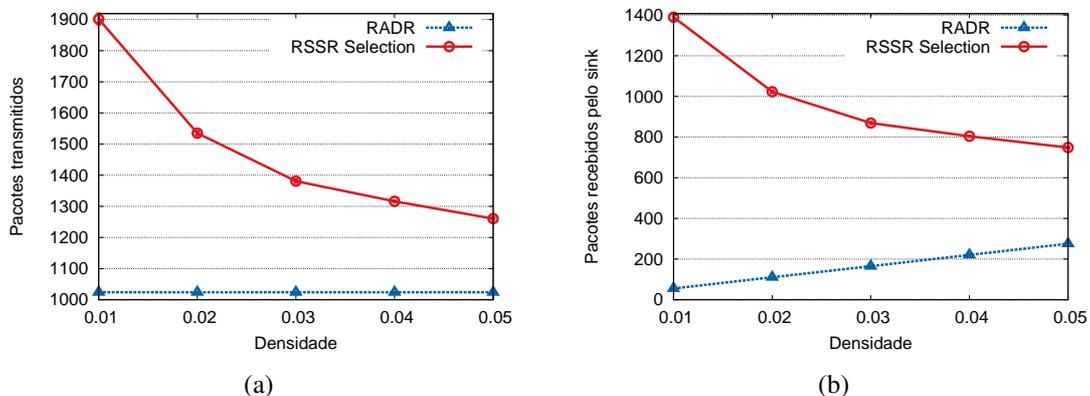


Figura 7. Resultados obtidos ao se avaliar o impacto da densidade da rede.

Já no gráfico da Figura 6(b), é possível perceber que mesmo quando aumentamos o raio de comunicação entre os nós, o número de pacotes recebidos pelo *sink* ainda é expressivamente menor em nosso algoritmo que, neste caso, tem o número de vizinhos diretos do *sink* aumentado e, portanto, maior possibilidade de envio direto dos pacotes (mas com menos pacotes 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<sup>2</sup> e aumentando tal densidade até 0.05 nós/m<sup>2</sup>. Em densidades baixas, um nó terá poucos vizinhos, dificultando a agregação, enquanto que ao se aumentar a densidade, o número de vizinhos de cada nó aumentará. O gráfico da Figura 7(a) mostra a quantidade de pacotes transmitidos na rede ao se aumentar a densidade. Neste gráfico, é possível observar que mesmo a densidades baixas o nosso algoritmo continua funcionando e que ele é capaz de funcionar de forma bem mais eficiente que o *RSSR Selection*. Neste contexto, o número de pacotes recebidos pelo *sink* também foi avaliado. Na Figura 7(b), é possível observar o aumento do número de pacotes chegando no *sink* com uma agregação menor, uma vez que o *sink* terá mais vizinhos que enviarão os pacotes diretamente para ele.

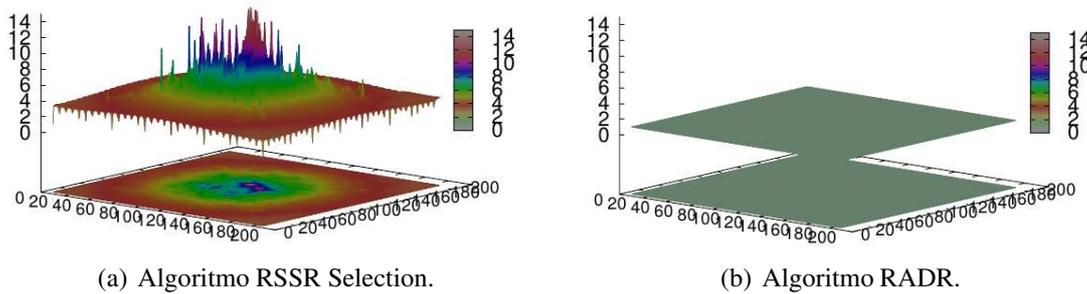


Figura 8. Quantidade de pacotes transmitidos em relação à posição física dos nós.

#### 4.5. Tempo de Vida da Rede e o Problema de Esgotamento de Energia dos Nós Próximos ao Sink

O gráfico da Figura 8(a) mostra a quantidade de pacotes transmitidos por um nó em relação à sua posição física após executar o *RSSR Selection*. Como pode-se observar, os nós mais próximos do *sink* tendem a enviar muitos pacotes, uma vez que estes devem enviar não só os seus dados para o *sink* mas também retransmitir os pacotes dos nós mais distantes. Tal comportamento resulta no esgotamento mais rápido da energia dos nós mais próximos do *sink*, chegando a um momento em que o *sink* não terá mais nenhum vizinho. Por outro lado, os nós mais distantes ainda estarão com uma grande quantidade de energia mas não poderão enviar seus dados, uma vez que não haverá nós para retransmiti-los para o *sink*.

Já o gráfico da Figura 8(b) mostra o comportamento do RADR, proposto neste trabalho. Como pode-se observar, devido à agregação de dados, os nós da rede como um todo trocam quase a mesma quantidade de pacotes, de modo que quando a energia de um nó estiver acabando em virtude das trocas de pacotes, é provável que toda a rede deva estar no mesmo nível de energia. Tal característica resolve o problema do esgotamento de energia dos nós próximos ao *sink* e ainda resulta na otimização do tempo de vida da rede.

#### 4.6. Acertos na Escolha do Próximo Salto

Até o momento, nos resultados apresentados, o algoritmo proposto obteve o mesmo desempenho que o algoritmo básico de agregação de dados *Flooding+Agregação*, de forma que este último não foi mostrado nos gráficos. Tal característica é um ponto positivo para a nossa abordagem, uma vez que o *flooding* é um dos algoritmos mais confiáveis em RSSFs. Entretanto, apesar de sua confiabilidade, o *flooding* não garante os menores nem os melhores caminhos na rede e muito menos a criação de árvores de escoamento de qualidade. O RADR, por outro lado, é capaz de gerar árvores de escoamento muito mais próximas da ótima, uma vez que este se baseia em distâncias ao *sink*.

Para avaliarmos esse aspecto, os gráficos das Figuras 9(a) e 9(b) mostram a quantidade de acertos na escolha do próximo salto de um pacote. Um acerto se dá quando um pacote sai de um nó em direção ao seu vizinho que está fisicamente mais próximo do nó *sink*, enquanto que um erro se dá quando este pacote foi enviado para um vizinho que não é o mais próximo do *sink*, ou seja, ele poderia ter sido enviado para um vizinho melhor. Quanto maior a taxa de acerto, melhor será a agregação e o caminho dos pacotes em direção ao *sink*.

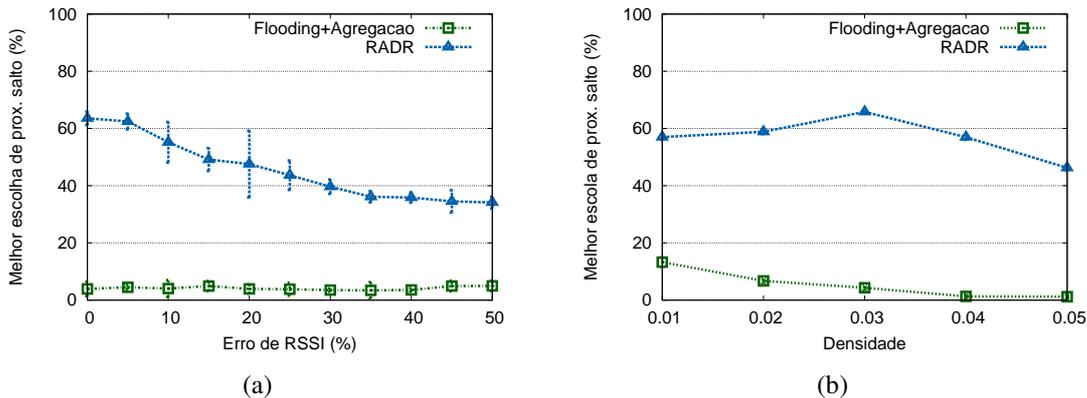


Figura 9. Resultados da taxa de acertos na escolha dos próximos saltos dos pacotes.

No gráfico da Figura 9(a) é possível observar o comportamento da escolha do próximo vizinho ao se aumentar o erro de RSSI, uma vez que este é um dos fatores mais diretamente relacionados à escolha do vizinho em nossa abordagem. Como pode-se observar, a taxa de acerto obtida pelo RADR é muito superior à do *Flooding+Agregação*, mostrando que nossa abordagem possui todas as vantagens deste último, mas obtendo uma árvore de agregação muito mais eficiente. É possível observar ainda a baixa qualidade da árvore obtida pelo *Flooding+Agregação*, o que já era esperado. Finalmente, vê-se que a qualidade do RADR cai ao se aumentar o erro de RSSI mas, em todos os casos, ficando bem acima do *Flooding+Agregação*.

Já o gráfico da Figura 9(b) mostra o comportamento da mesma taxa ao se aumentar a densidade da rede. É possível observar que nos dois algoritmos, a taxa de acerto é maior a baixas densidades, uma vez que há uma quantidade menor de vizinhos a serem escolhidos.

## 5. Aplicabilidade da Solução Proposta

Neste trabalho consideramos um nó *sink* equipado com um dispositivo de comunicação potente, de forma que em um único salto seja possível alcançar todos os nós regulares na rede. Nossa abordagem é aplicável em diversos cenários, dentre eles, o projeto ATTO (*Amazonian Tall Tower Observatory*), para o qual foi proposto. Tal projeto consiste em uma torre no meio da floresta amazônica, com uma estrutura física de 320 metros de altura, somente comparável à torre *Eiffel*, cujo foco é mapear eventos da região e fornecer medidas confiáveis de fontes e sumidouros de gases de efeito estufa como  $CO_2$ ,  $CH_4$  e  $N_2O$  [Tollefson 2010]. A torre, a primeira desta natureza no cenário amazônico, está instalada na Reserva de Desenvolvimento Sustentável do Uatumã, no interior do Amazonas. Neste contexto, o algoritmo RADR é proposto como uma solução de encaminhamento guloso utilizando agregação de dados para os nós sensores em volta da torre, que servirá como *sink*. Ao se aproveitar a comunicação de alto alcance do *sink*, nossa solução evita a dependência de um *flooding* inicial, que tem comportamento ruim em ambientes com baixa qualidade de comunicação, como é o caso na região amazônica.

## 6. Conclusão

Neste trabalho, propomos um novo protocolo de roteamento geográfico com agregação de dados para RSSFs: o RADR (Roteamento e Agregação de Dados baseado no RSSI). Tal

protocolo parte do princípio de que o nó *sink* possui uma capacidade alta de comunicação de modo a mandar uma única mensagem alcançando todos os nós da rede. Com base apenas no RSSI da mensagem do *sink* chegando em cada um dos nós, propomos um novo e inovador esquema de roteamento e agregação de dados. Nosso algoritmo de agregação de dados utiliza um temporizador para esperar pelos pacotes a serem agregados que também é feito com base na mesma informação de RSSI.

Uma série de experimentos foram realizados para avaliar o protocolo proposto e compará-lo com dois outros algoritmos da literatura: o *RSSR Selection* e o *Flooding+Agregação*. O primeiro utiliza um esquema de encaminhamento baseado no RSSI semelhante ao nosso, mas não faz agregação de dados. Enquanto que o segundo faz agregação de dados mas é baseado na árvore de escoamento obtida a partir de um *flooding*. Os resultados obtidos mostram claramente os benefícios introduzidos pelo esquema de agregação de dados proposto bem como a qualidade superior da árvore de agregação obtida em relação à obtida por um *flooding* clássico.

Apesar de obtermos resultados satisfatórios com vantagens importantes, algumas limitações devem ser exploradas em trabalhos futuros, como, por exemplo, a combinação de nossa solução com algoritmos de reconhecimento de perímetro e desvio de buracos que sejam também baseadas apenas nas informações disponíveis de RSSI, algo ainda não explorado na literatura.

## Referências

- Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3:325–349.
- Al-karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6–28.
- Boukerche, A., Oliveira, H., Nakamura, E., and Loureiro, A. (2007). Localization systems for wireless sensor networks. *Wireless Communications, IEEE*, 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 *Communications, 2008. ICC '08. IEEE International Conference on*.
- ETH-Zurich, D. C. G. a. (2012). 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.
- Jacinto, R. M. P. (2012). Modelação da Propagação numa Rede de Sensores sem Fios. Master's thesis, Universidade Nova de Lisboa.
- Kermarrec, A.-M. and Tan, G. (2010). Greedy geographic routing in large-scale sensor networks: a minimum network decomposition approach. In *Proceedings of the ele-*

- venth *ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '10, pages 161–170, New York, NY, USA. ACM.
- Leong, B., Liskov, B., and Morris, R. (2007). Greedy virtual coordinates for geographic routing. In *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, pages 71–80.
- 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.
- Park, E., Bae, D., and Choo, H. (2010). Energy efficient geographic routing for prolonging network lifetime in wireless sensor networks. In *Proceedings of the 2010 International Conference on Computational Science and Its Applications, ICCSA '10*, pages 285–288, Washington, DC, USA. IEEE Computer Society.
- Pham, N., Youn, J., and Won, C. (2006). A comparison of wireless sensor network routing protocols on an experimental testbed. In *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, volume 2, pages 276–281.
- Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S., and Stoica, I. (2003). Geographic routing without location information. In *MOBICOM'03*, pages 96–108. ACM Press.
- Souza, E. L. d., Nakamura, E. F., and Oliveira, H. A. B. F. d. (2010). Uma abordagem de fusao de dados em redes de sensores para reduzir o impacto de erros de localizacao em algoritmos de rastreamento. In *28o Simposio Brasileiro de Redes de Computadores e Sistemas Distribuidos (SBRC'2010)*, pages 291–304.
- Takagi, H. and Kleinrock, L. (1984). Optimal transmission ranges for randomly distributed packet radio terminals. *Communications, IEEE Transactions on*, 32(3):246 – 257.
- 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 *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–6.
- Zorzi, M. and Rao, R. (2003). Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance. *Mobile Computing, IEEE Transactions on*, 2(4):337 – 348.