

Localização 3D em Redes de Sensores Sem Fio Utilizando Veículo Aéreo não Tripulado

Leandro A. Villas^{1,5}, Daniel L. Guidoni², Azzedine Boukerche³,
Antonio A. F. Loureiro⁴ e Jo Ueyama⁵

¹Instituto de Computação – UNICAMP

²Departamento de Ciência da Computação – UFSJ

³PARADISE Research Laboratory – University of Ottawa

⁴Departamento de Ciência da Computação – UFMG

⁵Instituto de Ciências Matemáticas e de Computação – USP

(leandro, joueyama)@icmc.usp.br, guidoni@ufs.j.edu.br

boukerch@site.uottawa.ca e loureiro@dcc.ufmg.br

Abstract. *A wireless sensor network (WSN) is designed to perform event detection, data collection, and reporting of such data to a monitoring station. In many cases, it is necessary to know the location of sensor nodes to relate the detection of the event at a specific location. However, the geographical location of the sensor nodes in most applications can only be set after their deposition in the area of interest. Therefore, for the sensor nodes to know their location, it is necessary to use specific algorithms to solve the problem of discovering the geographical position of sensor nodes. This work addresses the problem of 3D localization in WSNs using an unmanned aerial vehicle (UAV). The UAV is equipped with GPS and it flies over the monitoring area broadcasting its geographical position. Thus, the sensor nodes are able to estimate their geographical position without being equipped with GPS device. Simulation results show that using an UAV leads to a smaller error in the calculation of geographic location as compared to solutions presented in the literature.*

Resumo. *Uma rede de sensores sem fio (RSSF) é projetada para executar detecção de eventos, coleta de dados e o relato de desses dados para uma estação de monitoramento. Em muitos casos, é necessário saber a localização dos nós sensores para relacionar a detecção do evento à uma localização específica. Porém, a localização geográfica dos nós na maioria das aplicações só pode ser definida após a sua deposição na área de interesse. Logo, para os nós saberem a sua localização, é necessário a utilização de algoritmos específicos para resolver o problema de descoberta da posição geográfica dos nós sensores. Este trabalho aborda o problema de localização 3D em RSSFs utilizando um veículo aéreo não tripulado (VANT). O VANT é equipado com GPS e sobrevoa a área de monitoramento disseminando a sua posição. Dessa forma, os nós sensores são capazes de encontrarem a sua posição geográfica sem serem equipados com GPS. Resultados de simulação mostraram que a abordagem utilizando o VANT produz um erro menor no cálculo da posição geográfica em comparação a soluções propostas na literatura.*

1. Introdução

Uma rede de sensores sem fio (RSSF) [Akyildiz et al. 2002, Boukerche 2008, Romer and Mattern 2004, Villas et al. 2010] é projetada para detectar eventos de interesse de uma aplicação, coletar dados referentes a esses eventos e enviar um relato desses dados para uma estação de monitoramento. Tipicamente, a informação coletada e os nós sensores devem ser localizados no espaço para identificar a localização do evento de interesse. Este posicionamento é feito por um sistema de localização, que é uma parte fundamental das RSSFs. Esse sistema é usado não apenas na localização de eventos, mas também na identificação e correlação de dados recolhidos [Villas et al. 2011b], endereçamento dos nós [Heidemann et al. 2001], gestão e consulta de nós localizados em uma determinada região [Navas and Imielinski 1997], construção de mapas de energia [Mini et al. 2004], roteamento geográfico [Villas et al. 2011a], rastreamento de objetos [Kumar et al. 2000], e muitos outros algoritmos que de alguma forma fazem uso de informações geográficas.

A definição de um sistema de localização para nós sensores [Boukerche et al. 2008a, Boukerche et al. 2008b, Bulusu et al. 2004, Doherty et al. 2001, de Oliveira et al. 2009a, de Oliveira et al. 2009b, Rudafshani and Datta 2007, Savvides et al. 2001] é um pré-requisito necessário para que muitas aplicações de RSSFs se tornem viáveis. O problema de localização em uma rede de sensores consiste em identificar a localização física (por exemplo, latitude, longitude e altitude) dos nós sensores. A importância de um sistema de localização em RSSFs surge a partir da necessidade de nomear os dados coletados [Heidemann et al. 2001] e eventos associados com o seu local de ocorrência [Intanagonwiwat et al. 2000]. Além disso, alguns algoritmos de roteamento usam informações de localização para melhorar seu desempenho como, por exemplo, o *Dynamic and Scalable Tree* [Villas et al. 2011a], que otimiza o processo de encontrar pontos de agregação de dados criando as rotas considerando as posições geográficas dos nós. No entanto, dependendo da precisão da informação de localização, tais rotas podem não alcançar os nós corretos para encontrar pontos de agregação.

Tipicamente, os sistemas de localização propostos para RSSFs usam uma abordagem recursiva [de Oliveira et al. 2009a], onde um nó estima a sua localização com base em informações de posição de três ou mais nós de referência (vizinhos que conhecem as suas posições). Uma vez que a sua posição é calculada, o nó torna-se uma referência (âncora) e transmite a informação com sua própria localização para auxiliar os outros nós na estimativa de suas posições. Essa abordagem apresenta alguns inconvenientes, como a propagação do erro de localização. Isto significa que o erro na estimativa de posição de um nó pode ser utilizado por outros nós para estimar as suas próprias posições, aumentando ainda mais o erro da posição estimada. Além disso, o nó deve ter pelo menos três vizinhos de referência para poder calcular a sua posição.

O objetivo deste trabalho é usar uma estratégia diferente, que não seja baseada em nós *beacons*¹ e que não gere a propagação do erro de sistemas de localização existentes. Assim, propomos um sistema de localização 3D diferente que apresenta três contribuições para a área de sistemas de localização em RSSFs. A principal contribuição é que todos os nós da rede são aptos a calcular a sua posição geográfica com alta precisão. Além disso, o algoritmo proposto é eficiente tanto para redes esparsas quanto densas, diferente da mai-

¹Nós que conhecem a sua posição geográfica utilizando, por exemplo, um receptor GPS.

oria das soluções da literatura que apresenta alguns inconvenientes, como baixa precisão na estimativa de posição em cenários de rede esparsa. Por fim, o sistema de localização proposto reduz o custo da rede drasticamente, uma vez que necessita de apenas um nó (VANT) munido com receptor GPS.

Este trabalho está organizado da seguinte forma. A próxima seção apresenta as definições de sistemas de localização em RSSFs. Os trabalhos relacionados são apresentados na seção 3. A seção 4 descreve o algoritmo proposto para o problema de localização 3D em RSSFs. A seção 5 apresenta os resultados de simulação. Finalmente, a seção 6 apresenta as conclusões e trabalhos futuros.

2. Sistemas de Localização em Redes de Sensores sem Fio

Uma RSSF é composta por n nós sensores, com raio de comunicação r_c e os mesmos são distribuídos em um campo tridimensional. A rede pode ser representada por um Grafo Euclidiano $G = (V, E)$, com as seguintes propriedades:

- $V = \{v_1, v_2, \dots, v_n\}$ é o conjunto de nós sensores;
- $\langle i, j \rangle \in E$ se e somente se v_i alcança v_j , isto é, a distância entre v_i e v_j é menor do que r_c ;
- $w(e) \leq r_c$ é o peso da aresta $e = \langle i, j \rangle$, isto é, a distância entre v_i e v_j .

A seguir, estão definidos alguns termos que podem ser utilizados para designar o estado de um nó sensor na rede.

Definição 1 (Nós Desconhecidos – D) *este termo refere-se aos nós da rede que não sabem a sua localização ainda. O principal objetivo de um sistema de localização é permitir que esses nós possam estimar suas posições.*

Definição 2 (Nós com Posições Estimadas – P) *esses nós eram inicialmente nós desconhecidos que conseguiram estimar suas posições usando o sistema de localização. O número de nós com posições estimadas e o erro de posição estimada destes nós são os parâmetros principais para determinar a qualidade de um sistema de localização.*

Definição 3 (Nós de Referência – R) *esses nós não precisam de um sistema de localização para estimar suas posições físicas. A localização desses nós é obtida por colocação manual ou por meios externos, como o GPS. Esses nós formam a base da maioria dos sistemas de localização para RSSFs.*

O problema da localização pode ser definido como segue:

Definição 4 (Problema de Localização) *dada uma rede $G = (V, E)$ e um conjunto de nós R e suas posições (x_r, y_r, z_r) , para todo $r \in R$, queremos encontrar a posição (x_u, y_u, z_u) do maior número possível de $u \in D$, assim transformando os nós desconhecidos D em nós com posições estimadas P .*

A maneira mais simples para resolver o problema da localização em uma RSSF é munir cada nó sensor da rede com um receptor GPS. Esta solução apresenta algumas vantagens, como um erro de localização relativamente pequeno (2–15 m, dependendo do receptor GPS) e bastante preciso, uma vez que o erro seria semelhante para todos os nós sensores da rede. Porém, esta solução apresenta uma série de inconvenientes como aumento do tamanho dos sensores, falta de visada dos satélites e, por fim, aumento no

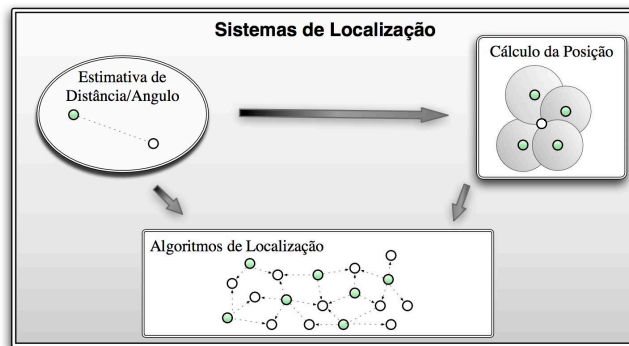


Figura 1. Sistemas de Localização divididos em três componentes distintos.

consumo de energia e do custo dos nós sensores. Esta solução torna-se impraticável para redes com centenas ou milhares de nós sensores, o que nos leva à necessidade de projetar sistemas de localização.

Os sistemas de localização podem ser dividido em três componentes distintos (veja a figura 1):

1. *Estimativa de Distância/Ângulo*: este componente é responsável por estimar as informações a respeito da distância e/ou ângulo entre dois nós. Técnicas utilizadas neste componente incluem indicador de intensidade do sinal recebido (RSSI), o tempo de chegada (ToA), o tempo diferença de chegada (TDoA), número de saltos, ou o ângulo de chegada (AoA).
2. *Cálculo da Posição*: este componente é responsável por calcular a posição de um nó com base na informação disponível sobre as distâncias/ângulos e posições dos nós de referência. Algumas técnicas usadas para calcular uma posição incluem trilateração, multilateração ou triangulação.
3. *Algoritmo de Localização*: este é o principal componente de um sistema de localização. Ele determina como a informação disponível será manipulada para permitir que a maioria ou todos os nós da rede de sensores sem fio possam estimar as suas posições.

O desempenho de um sistema de localização depende diretamente de cada um desses componentes. Além disso, cada componente tem seu próprio objetivo e métodos de solução.

3. Trabalhos Relacionados

Esta seção discute algumas abordagens da literatura para o problema de localização em RSSFs.

No Sistema de Posicionamento Ad Hoc – APS [Niculescu and Nath 2001], um reduzido número de nós *beacons* (por exemplo, três ou mais) é depositado junto com os nós que não conhecem suas posições geográficas. Em seguida, cada nó calcula a distância para os nós *beacons*, através de *multihops*. Uma vez que estas distâncias são estimadas, os nós podem calcular suas posições usando a triangulação. Três métodos de propagação de distância *hop by hop* são propostos: Dv-Hop, Dv-Distância, e a Euclidiana. No APS

Dv-Hop, os nós *beacons* iniciam a propagação de suas informações de posição. Todos os nós da rede recebem as informações de posição de todos os nós *beacons*, bem como o número de saltos para eles (nós *beacons*). Quando um nó *beacon* recebe a informação sobre a posição dos outros nós *beacon*, pode-se considerar que ele tem informações suficientes para calcular o tamanho médio de um salto com base na sua própria posição (em relação aos outros *beacons*), e também no número de saltos entre eles. Este último valor é então inundado de maneira controlada para toda a rede, como um mecanismo (fator) de correção. Quando um nó desconhecido recebe tal fator de correção, então ele é capaz de converter a sua distância entre os nós *beacons* do número de saltos (*hops*) para metros. Uma vantagem do APS é a de que o seu algoritmo de localização requer apenas um pequeno número de nós *beacon* para que possa funcionar. No entanto, a maneira como as distâncias são propagadas, em particular no Dv-Hop e no Dv Distância, assim como a forma como estas distâncias são convertidas do número de *hops* para metros (no Dv-Hop), resultam em cálculos de posicionamentos imprecisos, o que aumenta o erro de localização final do sistema.

No algoritmo de Estimativa de Posicionamento Recursivo – RPE [Albrowicz et al. 2001], os nós estimam as suas posições com base em um conjunto de nós *beacons* iniciais (por exemplo, 5 a 10% dos nós). Nós *beacons* são cientes de suas posições (por exemplo, nós equipados com GPS). O algoritmo RPE é dividido em quatro fases. Na primeira fase, os nós *beacons* começam a transmitir a sua informação de posição, de modo que esses nós possam ser usados como nós de referência. Na segunda fase, um nó calcula a distância para os nós de referência, utilizando, por exemplo, o RSSI. Na terceira fase, o nó calcula a sua posição através da utilização, por exemplo de triangulação. Na fase final, o nó se torna uma referência e transmite a sua posição estimada para ajudar os seus vizinhos com as suas estimativas de posição. Uma vantagem deste algoritmo é que o número de nós de referência aumenta rapidamente, de tal maneira que a maioria dos nós possam calcular as suas posições. No entanto, esta técnica tem a desvantagem de propagar o erro de localização. Isto significa que a estimativa de posição imprecisa de um nó pode ser utilizada por outros nós para estimar as suas próprias posições, aumentando ainda mais a imprecisão. Além disso, o nó deve ter pelo menos três vizinhos de referência para poder calcular a sua posição.

No algoritmo de Localização com um Beacon Móvel – MBL [Sichitiu and Ramadurai 2004], o *beacon* móvel locomove-se no campo onde os nós sensores localizam-se e o nó *beacon* móvel transmite periodicamente mensagens com informações sobre a sua posição geográfica. Quando um nó sensor recebe três ou mais mensagens com as coordenadas do *beacon* móvel, então ele calcula a sua posição usando uma abordagem probabilística, com base nas coordenadas recebidas e nas estimativas de distância do RSSI. Uma vantagem deste algoritmo é que as estimativas de posição são calculadas com base no mesmo nó (*beacon* móvel), fazendo com que a média do erro de localização seja baixa e impedindo a sua propagação. Por outro lado, os autores consideram como nó *beacon* um robô, mas tipicamente os robôs terão dificuldade para se locomover em ambientes de difícil acesso. Além disso, o trabalho não considera o problema de localização em ambientes 3D.

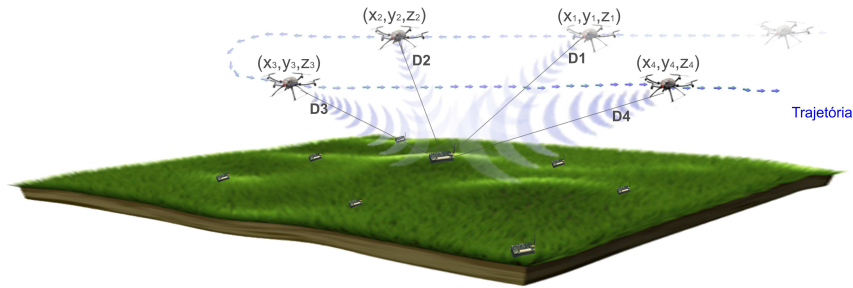


Figura 2. Sistema de localização proposto.

4. Sistema de Localização 3D Proposto

Esta seção apresenta a solução proposta para o problema de localização 3D utilizando um Veículo Aéreo Não Tripulado (VANT). Inicialmente, todos os nós sensores pertencem ao conjunto D e o VANT pertence ao conjunto R .

A figura 2 ilustra o sistema de localização 3D proposto que faz a utilização de um VANT. O VANT sobrevoa a área de monitoramento onde os nós foram depositados. Durante o voo, o VANT faz um *broadcast* periódico da sua posição geográfica. Quando um nó recebe quatro ou mais mensagens contendo a posição do VANT, o nó pode calcular a sua posição. É importante ressaltar que para calcular a posição em um sistema de três dimensões, são necessários 4 pontos de referência, ao invés de 3 pontos de referência em um sistema de duas dimensões. A seguir, apresentaremos em detalhes todos os componentes do sistema de localização (ilustrados na figura 1) para o funcionamento do sistema de localização 3D utilizando VANT proposto, que são: Estimativa da Distância, Cálculo da Posição e Algoritmo de Localização. Após, apresentaremos a aplicabilidade do sistema de localização proposto.

4.1. Estimativa de Distância

Existem vários métodos para estimar a distância entre dois nós. O método mais utilizado é o *RSSI*, já que ele não necessita de nenhum *hardware* extra além do rádio transmissor/receptor embutido no nó sensor. Além disso, a técnica de *RSSI* não necessita de nenhuma mensagem de controle para que a estimativa da distância em relação ao nó que enviou seja feita, o que é observado nas outras técnicas de estimativa da distância descritas anteriormente. A figura 3 ilustra a força do sinal quando um nó envia uma mensagem considerando três dimensões. O nó envia um sinal com uma determinada força, que reduz a medida que o sinal é propagado. Neste caso, se o nó que recebeu o sinal está distante do nó que o enviou, a força do sinal recebido será baixa.

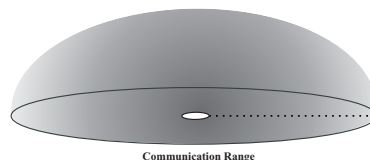


Figura 3. Decremento na força do sinal.

A tabela 1 exemplifica dois modelos de propagação de sinais conhecidos na literatura [Boukerche 2008]. Utilizando um modelo de propagação, a força do sinal pode ser

Modelo	Descrição	Fórmula
<i>Free Space</i>	Considera uma condição de propagação ideal, sem interferências ou obstáculos	$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$
<i>Two-Ray Ground</i>	Similar ao <i>Free Space</i> , mas considera a possibilidade de reflexão do sinal no solo	$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}$

Tabela 1. Dois modelos de propagação.

convertida em distância. Dessa forma, quando um nó recebe uma mensagem, ele pode medir a força do sinal recebido e utilizar um modelo de propagação de sinal para encontrar a distância em relação ao nó que enviou a mensagem. Para ser mais realista com um sistema de localização 3D, neste trabalho, nós utilizamos o modelo de propagação de sinal *Two-Ray Ground*, que considera a possibilidade de reflexão do sinal no solo.

4.2. Cálculo da Posição

Para o nó calcular sua posição em um sistema com três dimensões, são necessários quatro pontos de referências. Esses pontos de referências são obtidos durante o vôo do VANT sobre a área de monitoramento. Além dos quatro pontos de referências, é necessário saber a distância para cada um desses pontos. Como descrito na seção anterior, isso pode ser feito medindo a força do sinal (RSSI) quando o VANT envia sua posição para os nós sensores. Quando o nó sensor possui quatro ou mais posições e suas respectivas distâncias, ele pode estimar a sua posição. O método mais utilizado para estimar a distância quando se tem quatro ou mais pontos de referência é a multilateração.

Utilizando a multilateração para estimar a posição de um nó, temos um sistema de equações (com no mínimo quatro equações) e três variáveis (x, y, z) . O sistema de equações é construído a partir das posições recebidas e suas respectivas distâncias. O sistema de equações pode ser ilustrado da seguinte maneira, onde (x_i, y_i, z_i) e d_i , são respectivamente uma posição referência e a estimativa da distância para esse ponto de referência obtido através da técnica RSSI.

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= d_2^2 \\ &\vdots \\ (x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2 &= d_n^2 \end{aligned}$$

O sistema de equações acima pode ser facilmente linearizado subtraindo a última equação, que é $(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2 = d_n^2$ das demais. Uma vez linearizado, o sistema de equações pode ser facilmente solucionado utilizando métodos de soluções de sistemas lineares. O método utilizado neste trabalho é o *least squares* [Golub and Loan 1996], devido a sua simplicidade e desempenho, sendo estes fatores importantes no projeto de soluções para redes de sensores sem fio. Como o sistema linear possui três variáveis, a solução do sistema linear representa a posição (x, y, z) do nó que executou o procedimento descrito.

4.3. Algoritmo de Localização

O algoritmo proposto é dividido em duas partes. A primeira se refere ao VANT e a segunda se refere ao cálculo da posição dos nós sensores. Como descrito anteriormente,

a figura 2 ilustra o funcionamento do algoritmo proposto. O VANT sobrevoa a área onde os nós sensores foram depositados. Durante o voo, o VANT faz *broadcasts* periódicos informando para os nós sensores a sua posição geográfica. Quando um nó sensor recebe uma mensagem do VANT, ele calcula a sua distância em relação ao VANT utilizando a técnica de *RSSI*. Quando o nó sensor possui quatro ou mais posições recebidas do VANT (pontos de referência), o nó está apto a calcular a sua posição. Os algoritmos 1 e 2 descrevem essas etapas.

Neste trabalho, utilizamos o plano de voo ilustrado na figura 4. O algoritmo 1 descreve o plano de voo do VANT. As linhas 1 a 4 definem as variáveis utilizadas no algoritmo. A linha 4 escalona pela primeira vez o broadcast periódico da posição geográfica pelo VANT. Após *broadcastInterval*, a função *Timeout* (linhas 12 a 17) será invocada. A função cria uma mensagem contendo a posição do VANT e faz o *broadcast* dessa mensagem. Após, o temporizador para um novo *broadcast* é escalonado. As linhas 6 a 11 executam o plano de voo do VANT. Enquanto o VANT não chegou ao final da rota, ele recupera do plano de voo o próximo ponto (coordenadas geográficas) no qual o VANT deve se deslocar. Após, o VANT se desloca para o ponto com uma determinada velocidade. É importante ressaltar que o *broadcast* periódico da posição é executado em paralelo com o deslocamento do VANT sobre a área monitorada.

Algorithm 1 Plano de Voo do VANT

```

1: broadcastInterval ← valor;                                \\ Intervalo entre cada broadcast da posicao
2: direcao ← PosicaoInicial();                                \\ Posicao inicial do VANT
3: velocidade ← velocidadeVANT;                              \\ Velocidade do VANT
4: timer.schedule(broadcastInterval);                        \\ Timer para escalonar o broadcast periodico
5: planoVoo();

6: procedimento PLANOVoo()
7:   enquanto nao chegou ao final da rota faça
8:     direcao ← getNextPoint()                                \\ Proximo ponto no plano de voo
9:     moveTo(direcao, velocidade)                            \\ Deslocamento do VANT
10:   fim enquanto
11: fim procedimento

12: procedimento TIMEOUT()
13:   (xvant, yvant, zvant) ← getPosicaoVANT()
14:   mensagem ← (xvant, yvant, zvant)
15:   broadcast(mensagem)                                     \\ Enviando a posicao para os nos sensores
16:   timer.schedule(broadcastInterval)                       \\ Escalonando um novo broadcast
17: fim procedimento

```

O algoritmo 2 descreve o algoritmo executado nos nós sensores para o cálculo de suas posições. A linha 1 define o conjunto referência, que são as posições geográficas do VANT juntamente com a distância relacionada com cada posição. A linha 2 define a variável que representará a posição calculada do nó sensor. Quando o nó sensor recebe uma mensagem do VANT, os seguintes passos são executados (linhas 3 a 9). O *RSSI* relacionado a mensagem recebida é convertido em distância utilizando a propagação de sinal *Two-Ray Ground* descrita anteriormente. Em seguida, o nó atualiza o conjunto referência com a mensagem contendo a posição do VANT juntamente com a distância em relação ao VANT. Se o conjunto referência for maior/igual a 4, o nó está apto a calcular a sua posição geográfica em três dimensões. Para isso, o conjunto referência é passado para o método *leastSquares* (descrito na seção anterior) e, como retorno, é calculada a posição do nó sensor.

Algorithm 2 PositionComputation

```

1: conjuntoReferencia ← ∅                                \\ Conjunto contendo as posições de referências
2: myPosition ← ∅                                       \\ Posição (x, y, z) do nó

3: procedimento HANDLEMESSAGE(mensagem, rssi)
4:   distancia ← convertRssiToDistance(rssi)
5:   conjuntoReferencia.add(mensagem, distancia)
6:   se conjuntoReferencia.size() ≥ 4 então
7:     myPosition ← leastSquares(conjuntoReferencia)
8:   fim se
9: fim procedimento

```

4.4. Aplicabilidade do Sistema de Localização Proposto

O sistema de localização proposto neste trabalho utiliza um VANT. Dessa forma, o VANT deverá percorrer toda a área onde os nós foram depositados para que todos os nós obtenham a quantidade de informação suficiente para o cálculo de sua posição. Uma consideração importante sobre esse esquema é o tempo para percorrer toda a área de monitoramento. A solução proposta neste trabalho irá gerar um atraso considerável para que todos os nós sejam capazes de calcular as suas posições. Entretanto, o algoritmo de localização será executado apenas uma vez durante todo o tempo de vida da rede², fazendo com que essa tarefa seja executada durante o seu *startup*. Dessa forma, não será gerado nenhum atraso desnecessário durante o seu funcionamento. Em contrapartida, como o VANT irá percorrer toda a área de monitoramento, todos os nós sensores serão capazes de calcular sua posição geográfica.

5. Avaliação de Desempenho

Esta seção avalia o desempenho do algoritmo proposto para o problema de localização 3D utilizando um veículo aéreo não tripulado em RSSF.

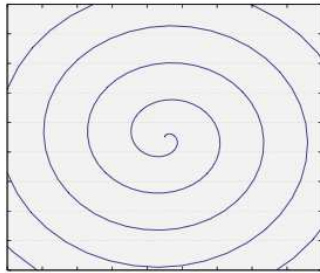
5.1. Metodologia

O algoritmo proposto nesse trabalho é comparado com o algoritmo Estimativa de Posicionamento Recursivo (RPE). A escolha do RPE deve-se ao seu bom desempenho em cenários 3D comparados com as outras soluções da literatura para o mesmo cenário, já que o algoritmo que utiliza um *beacon* móvel foi proposto para um sistema de localização 2D. A principal proposta da comparação é a avaliação do algoritmo proposto considerando as seguintes métricas: (i) Erro em metros no cálculo da posição geográfica; (ii) Quantidade de nós que não calcularam a sua posição geográfica. Para realizarmos as avaliações, variamos três parâmetros de rede, que são: (i) Quantidade de nós; (ii) Densidade e (iii) Erro no cálculo do RSSI. O erro no cálculo da posição foi computado utilizando a seguinte fórmula: $Erro = \frac{1}{n} \sum_{i=1}^n | PosicaoCalculada_i - PosicaoReal_i |$, onde n é a quantidade de nós sensores, $PosicaoCalculada_i$ é a posição calculada do i -ésimo nó e $PosicaoReal_i$ é a posição real do i -ésimo nó. O plano de voo utilizado é ilustrado na figura 4, enquanto os parâmetros de simulação são apresentados na Tabela 2. É importante observar que o raio de comunicação dos nós sensores e do VANT é de 50m. Isso foi feito para termos uma comparação justa com o algoritmo da literatura. Para termos resultados mais próximos de um ambiente real, utilizamos o modelo de propagação de sinal *Two-Ray Ground* e para estimar a distância em relação a força do sinal obtido, introduzimos um erro na técnica

²Neste trabalho consideramos uma RSSF estática.

de RSSI, que varia de 0 a 20% da potência do sinal recebido. Para calcularmos a área de sensoriamento (x, y) , utilizamos a quantidade de nós (n) e o raio de comunicação r_c dos nós sensores. O relevo (terceira dimensão) para cada nó sensor é obtido através de um valor aleatório entre 0 e 10m. A altura do VANT é um valor aleatório entre 20 e 50m, que muda a cada nova direção durante o vôo sobre a área de monitoramento.

A avaliação foi realizada através de simulações utilizando o simulador SinalGo v.0.75.3 [Sinalgo 2008]. Cada simulação foi replicada 33 vezes utilizando diferentes sementes para a geração de números aleatórios. Em todos os resultados, as curvas representam os valores médios, enquanto que as barras de erros representam o intervalo de confiança de 95%.



Área de monitoramento

Figura 4. Plano de voo.

Parâmetros	Valores
Quantidade de nós	250, 500, 750, 1000 e 1250
Densidade (número médio de vizinhos)	20, 30, 40 e 50
Raio de comunicação dos nós sensores	50m
Raio de comunicação do VANT	50m
Velocidade do VANT	10m/s
Modelo de propagação	Two-Ray Ground
Erro do RSSI (%)	0, 5, 10 e 15 e 20
Quantidade de nós Beacons (RPE)	50, 75, 100 e 125
Área de sensoriamento $(x$ e $y)$	$x = y = \frac{n \times \pi \times r_c^2}{Densidade}$
Relevo (z)	Valor aleatório entre 0 e 10m
Altura do VANT	Valor aleatório entre 20 e 50m
Intervalo de broadcast (VANT)	1/segundo

Tabela 2. Parâmetros de simulação.

5.2. Quantidade de Nós

Nós avaliamos o desempenho dos algoritmos para diferentes quantidade de nós na rede. A densidade foi fixada em 30 e o erro no cálculo do RSSI foi fixado em 5%. É importante ressaltar que, uma vez que a densidade da rede é de 30, ao aumentarmos a quantidade de nós também aumentamos a área de sensoriamento. A figura 5 ilustra os resultados obtidos para diferentes quantidade de nós.

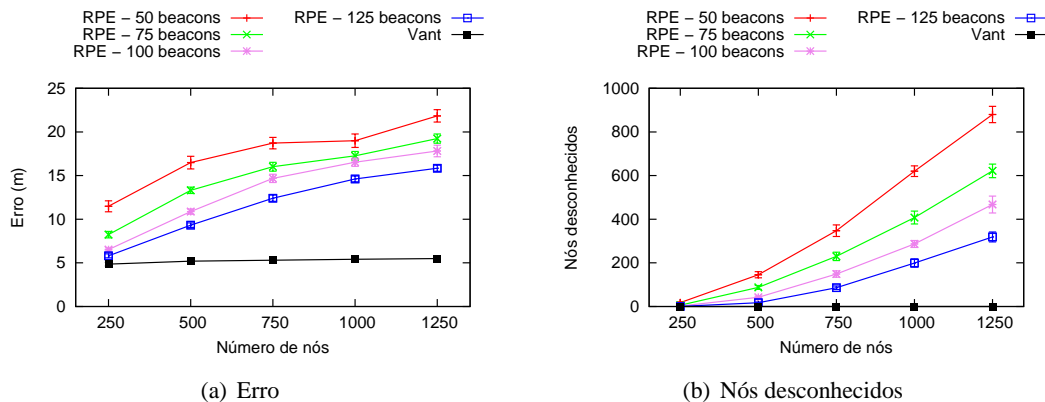


Figura 5. Quantidade de nós.

A figura 5(a) mostra o erro no cálculo da posição geográfica. O algoritmo de localização utilizando VANT proposto apresenta um erro pequeno nas posições estimadas. Observe que o erro não é afetado pela quantidade de nós na rede. Já o erro nas posições

estimadas pelo algoritmo RPE é grande (em média 3x maior que o erro das posições estimadas pelo algoritmo proposto), além disso, o erro no cálculo de posições estimadas pelo algoritmo RPE é afetado pela quantidade de nós na rede. Aumentando a quantidade de nós da rede, podemos observar que o erro na estimativa da posição também aumenta. Esse aumento do erro deve-se ao fato de muitos nós calcularem a sua posição baseado na posição calculada de outros nós. Dessa forma, o erro no cálculo se espalha pelos nós. Podemos observar que quando aumentamos a quantidade de nós *beacons* para o RPE, o erro no cálculo da posição geográfica pelo RPE diminui. Mas o principal problema com o aumento do número de nós *beacons*³ é que eles são mais caros do que o resto dos nós sensores. Isto significa que, mesmo que seja apenas 10% dos nós *beacons*, o preço da rede aumenta aproximadamente em dez vezes. Outra observação é que, após os nós da rede estimarem sua posição geográfica, os nós *beacons* tornam-se inúteis, pois eles já não usam mais seus receptores (caro) de GPS.

A figura 5(b) ilustra a quantidade de nós desconhecidos após a execução dos algoritmos. Um nó é considerado desconhecido em dois casos: (i) quando ele não recebeu informações suficientes para o cálculo da posição ou (ii) quando o erro da posição calculada for maior do que seu raio de comunicação. Podemos observar que, quando a rede possui poucos nós *beacons*, a quantidade de nós desconhecidos aumenta consideravelmente quando aumentamos a quantidade de nós da rede. Entretanto, para uma rede com poucos nós, a quantidade de nós desconhecidos é baixa. É importante observar que o sistema de localização proposto neste trabalho é capaz de fazer com que todos os nós da rede sejam aptos a calcular a sua posição geográfica, já que o VANT irá sobrevoar toda a área de monitoramento.

5.3. Densidade

Esta seção avalia o desempenho dos algoritmos para diferentes densidades de nós sensores na área de monitoramento. Para essa análise, fixamos a quantidade de nós sensores em 750 e o erro na estimativa da distância utilizando RSSI em 5%. A figura 6(a) mostra o erro no cálculo da posição para diferentes densidades. Pode-se observar que, quanto maior a densidade da rede, melhor é o desempenho do algoritmo RPE. Isso acontece devido ao fato de que quanto maior a densidade para uma mesma quantidade de nós, menor será a área de monitoramento. Como consequência, o erro no cálculo da posição geográfica não irá se espalhar para outros nós. A solução utilizando o VANT também não é afetada pela densidade da rede, uma vez que a troca de mensagem para o cálculo da posição acontece somente entre o VANT e os nós sensores e não entre nós sensores.

O mesmo padrão de resultados é encontrado na figura 6(b), que ilustra a quantidade de nós desconhecidos na rede após a execução dos algoritmos. Quando a densidade da rede aumenta, a quantidade de nós desconhecidos diminui como esperado. Além disso, a solução utilizando o VANT também não possui nenhum nó desconhecido após a execução do algoritmo.

³Um nó beacon pode ser de duas ordens de magnitude mais caro do que um nó desconhecido, devido ao seu receptor GPS.

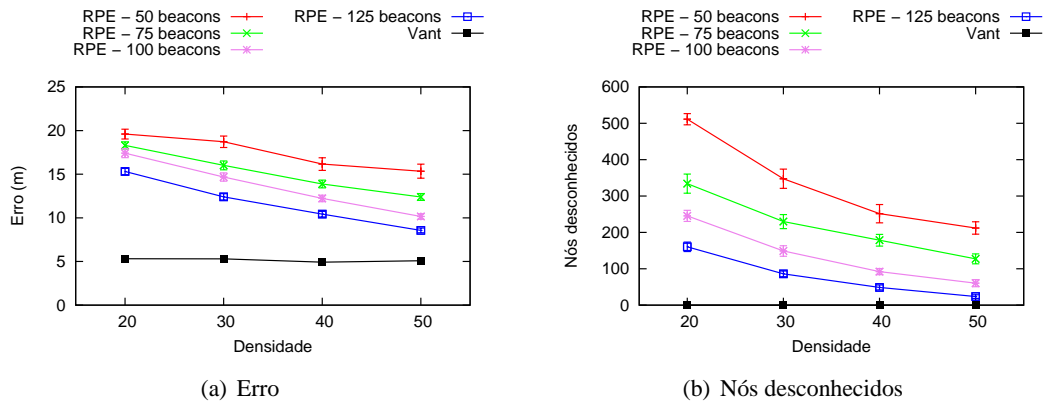


Figura 6. Densidade.

5.4. RSSI

Variamos o erro no cálculo da distância utilizando a técnica RSSI para analisar o desempenho dos algoritmos. Para essa análise, fixamos a quantidade de nós sensores em 750 e a densidade da rede em 30. A figura 7 ilustra os resultados para essa avaliação. A figura 7(a) mostra o erro no cálculo da posição para diferentes erros no RSSI. Podemos observar que quando o erro na estimativa da distância é igual a zero, tanto o RPE quanto a solução proposta neste trabalho conseguem encontrar a posição exata dos nós sensores. A medida que o erro no cálculo do RSSI aumenta, todos os algoritmos aumentam o erro na estimativa da posição. Além disso, para erros no cálculo do RSSI inferiores a 15%, a solução proposta neste trabalho possui melhores resultados considerando qualquer quantidade de nós *beacons* pelo RPE. Entretanto, quando o erro na estimativa da distância é de 20%, a solução proposta neste trabalho possui o mesmo resultado quando a rede possui 125 nós *beacons* pelo RPE.

Considerando a quantidade de nós desconhecidos (figura 7(b)), quando o erro na estimativa da distância é igual a zero, a quantidade de nós desconhecidos após a execução dos algoritmos é zero, já que os nós calculam a sua posição de maneira exata. Entretanto, quando o erro na estimativa da distância aumenta, a quantidade de nós desconhecidos aumenta para todas as soluções avaliadas. É importante observar que mesmo o VANT sobrevoando toda a região de monitoramento, se o erro na estimativa da distância for muito alto, o nó pode calcular a sua posição com um erro superior ao raio de comunicação, fazendo ele ser um nó desconhecido.

6. Conclusões

O problema de localização em RSSFs é fundamental para diversas aplicações neste tipo de rede. Tipicamente, o erro nas posições estimadas pelas abordagens da literatura depende da quantidade de nós *beacons* depositados na rede e esses nós *beacons* aumentam drasticamente o custo da rede. Além disso, a maioria das abordagens propostas na literatura não considera o problema de localização 3D.

Neste trabalho, propomos um algoritmo para o problema de localização 3D em RSSFs que faz a utilização de um VANT, que funciona como um disseminar de posição

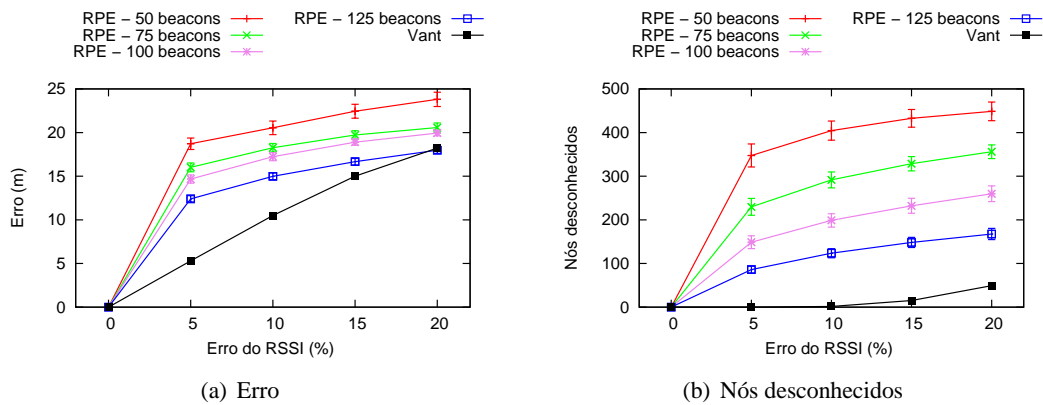


Figura 7. RSSI.

para os nós sensores de alta qualidade em termos de precisão da informação disseminada. Os resultados obtidos mostram que o algoritmo proposto estima a posição dos nós desconhecidos com um erro pequeno. A eficiência do algoritmo proposto é independente do número de nós na rede, o que é um aspecto importante no caso da escalabilidade. Além disso, todos os nós sensores recebem informações suficiente para calcular sua posição.

Como trabalhos futuros pretendemos considerar diferentes planos de vôos, além de fazer experimentos em um ambiente real.

7. Agradecimentos

Os autores gostariam de expressar a sua gratidão pelo apoio concedido pelo CNPq, FAPESP para o INCT-SEC (processos 573963/2008-9 e 2012/12061-1), CAPES e FAPEMIG.

Referências

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.
- Albrowicz, J., Chen, A., and Zhang, L. (2001). Recursive position estimation in sensor networks. In *Network Protocols, 2001. Ninth International Conference on*, pages 35 – 41.
- Boukerche, A. (2008). *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press.
- Boukerche, A., de Oliveira, H. A. B. F., Nakamura, E. F., and Loureiro, A. A. F. (2008a). Localization Systems for Wireless Sensor Networks. In Boukerche, A., editor, *Algorithms and Protocols for Wireless Sensor Networks*, chapter 11, pages 307–340. John Wiley & Sons.
- Boukerche, A., de Oliveira, H. A. B. F., Nakamura, E. F., and Loureiro, A. A. F. (2008b). Using Voronoi Diagrams to Scale a Localization System in Wireless Sensor Networks. *IEEE Wireless Communications Magazine*, 15:1–7.
- Bulusu, N., Heidemann, J., Estrin, D., and Tran, T. (2004). Self-configuring localization systems: Design and experimental evaluation. *ACM Trans. Embed. Comput. Syst.*, 3(1):24–60.
- de Oliveira, H. A. B. F., Boukerche, A., Nakamura, E. F., and Loureiro, A. A. F. (2009a). An Efficient Directed Localization Recursion Protocol for Wireless Sensor Networks. *IEEE Transactions on Computers*, 58(5):677–691.

- de Oliveira, H. A. B. F., Boukerche, A., Nakamura, E. F., and Loureiro, A. A. F. (2009b). Localization in Time and Space for Wireless Sensor Networks: An Efficient and Lightweight Algorithm. *Performance Evaluation*, 66(3–5):209–222.
- Doherty, L., pister, K., and El Ghaoui, L. (2001). Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655 –1663 vol.3.
- Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, third edition.
- Heidemann, J., Silva, F., Intanagonwiwat, C., Govindan, R., Estrin, D., and Ganesan, D. (2001). Building efficient wireless sensor networks with low-level naming. *SIGOPS Oper. Syst. Rev.*, 35(5):146–159.
- Intanagonwiwat, C., Govindan, R., and Estrin, D. (2000). Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00*, pages 56–67, New York, NY, USA. ACM.
- Kumar, S., Alaettinoglu, C., and Estrin, D. (2000). Scalable object-tracking through unattended techniques (scout). In *in: Proceedings of the 8th International Conference on Network Protocols (ICNP)*.
- Mini, R. A., Loureiro, A. A., and Nath, B. (2004). The distinctive design characteristic of a wireless sensor network: the energy map. *Computer Communications*, 27(10):935 – 945. <ce:title>Protocol Engineering for Wired and Wireless Networks</ce:title>.
- Navas, J. C. and Imielinski, T. (1997). Geocast - geographic addressing and routing. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '97*, pages 66–76, New York, NY, USA. ACM.
- Niculescu, D. and Nath, B. (2001). Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 5, pages 2926 –2931 vol.5.
- Romer, K. and Mattern, F. (2004). The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61.
- Rudafshani, M. and Datta, S. (2007). Localization in wireless sensor networks. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 51 –60.
- Savvides, A., Han, C.-C., and Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking, MobiCom '01*, pages 166–179, New York, NY, USA. ACM.
- Sichitiu, M. and Ramadurai, V. (2004). Localization of wireless sensor networks with a mobile beacon. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 174 – 183.
- Sinalgo (2008). Simulator for network algorithms. Distributed Computing Group - ETH-Zurich.
- Villas, L., Guidoni, D., Boukerche, A., Araujo, R., and Loureiro, A. (2011a). Dynamic and scalable routing to perform efficient data aggregation in wsns. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1 –5.
- Villas, L. A., Boukerche, A., de Oliveira, H. A., de Araujo, R. B., and Loureiro, A. A. (2011b). A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks. *Ad Hoc Networks*, (0):–.
- Villas, L. A., Guidoni, D. L., Araújo, R. B., Boukerche, A., and Loureiro, A. A. (2010). A scalable and dynamic data aggregation aware routing protocol for wireless sensor networks. In *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems, MSWIM '10*, pages 110–117, New York, NY, USA. ACM.