



MAURO LEONARDO MAIMONI

**ANÁLISE E IMPLEMENTAÇÃO DE TÉCNICAS DE LOCALIZAÇÃO
PARA RSSF**

INCONFIDENTES-MG

2013

MAURO LEONARDO MAIMONI

**ANÁLISE E IMPLEMENTAÇÃO DE TÉCNICAS DE LOCALIZAÇÃO
PARA RSSF.**

Trabalho de Conclusão de Curso apresentado como pré-requisito de conclusão do curso de Graduação Tecnológica em Redes de Computadores no Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Câmpus Inconfidentes, para obtenção do título de Tecnólogo em Redes de Computadores.

Orientador (a): LUIZ CARLOS BRANQUINHO CAIXETA FERREIRA

INCONFIDENTES-MG

2013

MAURO LEONARDO MAIMONI

**ANÁLISE E IMPLEMENTAÇÃO DE TÉCNICAS DE LOCALIZAÇÃO
PARA RSSF.**

Data de aprovação: ____ de _____ 2013

**Luiz Carlos Branquinho Caixeta Ferreira
(IFSULDEMINAS - Câmpus Inconfidentes)**

**Ivan Paulino Pereira
(IFSULDEMINAS - Câmpus Inconfidentes)**

**Bruno Amarante Couto Rezende
(IFSULDEMINAS - Câmpus Inconfidentes)**

DEDICATÓRIA

Dedico este trabalho a minha avó Nelcy de Melo Costa, ao meu pai Mauro Maimoni Filho, a minha mãe Maria Marta da Costa Maimoni, a minha tia Maria Luisa Costa Lima Ferreira, ao meu tio Misael Lima Ferreira, ao meu tio Aparecido Leonardo, a minha namorada Micaely B. Cavalcante, a minha irmã Marília Costa Maimoni Medeiros, ao meu irmão Marcos Leonardo Maimoni e toda a minha família que sempre esteve ao meu lado e me incentivou a buscar um futuro melhor.

AGRADECIMENTOS

Primeiramente agradeço a Deus por tudo que tem realizado em minha vida e por permitir vivenciar a experiência de uma faculdade. A minha mãe Maria Marta da Costa Maimoni por sempre estar ao meu lado, apoiando com toda a sua dedicação e nunca mediu esforços para me ajudar. Ao meu pai Mauro Maimoni Filho, ao meu irmão Marcos Leonardo Maimoni, a minha irmã Marília Costa Maimoni, e a minha namorada Micaelly B. Cavalcante por sempre me apoiar em todos os momentos.

Agradeço a minha tia Maria Luisa Costa Lima Ferreira por me ajudar e incentivar a buscar o melhor para minha carreira profissional. Aos familiares que sempre estiveram comigo em momentos de dificuldade e alegria e não mediram esforços para me ajudar durante este período. Aos meus amigos que se tornaram meus irmãos Douglas Nascimento Sousa, Rafael Fernandes, Elson Junior, Adeilson Silverio de Almeida, Evelyn Vitor, e aos meus colegas de classe que estiveram ao meu lado e participaram junto comigo durante este período.

Aos professores que estiveram presentes durante a minha formação e ao professor Luiz Carlos Branquinho Caixeta Ferreira por incentivar durante está pesquisas na área de tecnologia em rede de sensores sem fio sobre o tema de localização. Ao instituto por proporcionar um ambiente agradável de convivência por investir cada vez mais em melhorias e na capacitação dos seus alunos.

RESUMO

As redes de sensores sem fio (*RSSF*) são projetadas com o intuito de monitorar aplicações em diferentes contextos. É importante destacar que aplicações de localização podem ser importantes tanto para aplicações de localização da origem de dados produzidos pelo ambiente até mesmo para aplicações na robótica ao permitir identificar a posição de um robô e determinar que ele se desloque para outra posição. Devido a isto, um sistema de coordenadas relativas poderá auxiliar na localização da posição do sensor. Este trabalho faz uma revisão bibliográfica dos principais algoritmos de localização, métodos de estimativa de distância, e técnicas para o cálculo da posição o que permitirá implementar um algoritmo de localização com nós âncoras. Após isto, os cenários serão definidos para realizar os testes necessários com o algoritmo de localização e verificar os resultados produzidos, por fim, a conclusão sobre o algoritmo de localização resultando sua importância.

ABSTRACT

The wireless sensor networks (WSN) are designed in order to monitor applications in different contexts. Importantly location applications may be important both for localization applications data source produced by the environment even for applications in robotics to permit identifies the position of a robot and that he intended to move to another position. Because of this, a coordinate system relating may assist in locating the position of the sensor. This paper reviews literature on the main localization algorithms, methods of distance estimation, and techniques for calculating the position which will implement a location algorithm with node anchors. After that, the settings are set to perform the necessary tests on the algorithm of locating and verifying the results produced, finally, the completion of the algorithm resulting location of its importance.

SUMÁRIO

RESUMO	VI
ABSTRACT	VII
1. INTRODUÇÃO	13
1.1. MOTIVAÇÕES E OBJETIVOS	15
1.2. CONTEXTO	17
1.3. PRINCIPAIS CONTRIBUIÇÕES	17
1.4. ORGANIZAÇÃO DA MONOGRAFIA	17
2. REVISÃO DE LITERATURA	19
2.1. PROPRIEDADES DE LOCALIZAÇÃO	19
2.2. SISTEMA DE LOCALIZAÇÃO	21
2.3. ESTIMATIVA DE DISTÂNCIA/ÂNGULO.....	23
2.3.1. Intensidade do Sinal Recebido (<i>RSSI</i>).....	23
2.3.2. [Diferença do] tempo de chegada (<i>TDoA/ToA</i>)	24
2.3.3. Ângulo/direção de chegada (<i>AoA/DoA</i>).....	25
2.4. CÁLCULO DA POSIÇÃO	25
2.4.1. Trilateração	25
2.4.2. Multilateração	27
2.4.3. Triangulação.....	28
2.5. ALGORITMO DE LOCALIZAÇÃO	29
2.5.1. <i>Ad hoc Positioning System (APS)</i>	30
2.5.2. <i>Recursive position estimation (RPE)</i>	31
2.5.3. <i>The global positioning system (GPS)</i>	32
3. MATERIAL E MÉTODOS	33
3.1. METODOLOGIA	33
3.2. PLATAFORMA RADIUINO.....	36
3.3. BIBLIOTECA SCIPY.....	38
3.4. BIBLIOTECA NUMPY	39
4. RESULTADOS E DISCUSSÃO	41
4.1. CÁLCULO DA DISTÂNCIA DOS SENSORES AOS NÓS ÂNCORAS	45
4.2. LATERAÇÃO	46

4.2.1. Cálculo da Trilateração	46
4.2.2. Cálculo da Multilateração	48
4.3. IMPLEMENTAÇÃO DO ALGORITMO DE LOCALIZAÇÃO	49
4.4. TESTES COM O ALGORITMO DE LOCALIZAÇÃO.....	49
5. CONCLUSÃO	61
REFERÊNCIAS BIBLIOGRÁFICAS	63
APÊNDICE A	66

Índice de Figuras

Figura 1: Rede de sensores sem fio com seus principais componentes.....	14
Figura 2: Componentes do sistema de localização.....	22
Figura 3: Propagação do sinal no <i>RSSI</i>	23
Figura 4: Intersecção dos círculos na técnica da trilateração.	26
Figura 5: Modelo realista da trilateração: (a) as inconsistências das posições e distâncias das referências geram um sistema com infinitas posições, e (b) o valor residual, como sendo a diferença entre as distâncias e as calculadas.	27
Figura 6: Exemplo da técnica da multilateração.	28
Figura 7: Triangulação (a) com duas referências para trocar informações entre si e calcularem a posição do nó sensor, e (b) usa três referências, onde o próprio nó livre é capaz de calcular sua posição.....	29
Figura 8: Exemplo das fases do <i>Recursive position estimation</i> : (a) inicialmente o nó escolhe as referências, (b) estima sua distância a cada uma das referências, (c) calcula sua posição utilizando trilateração, (d) divulga sua posição para auxiliar outros nós no cálculo de suas posições.	31
Figura 9: Cenário 1 mostra o posicionamento dos nós no ambiente.	34
Figura 10: Cenário 2 montado para testes com nós sensores.	34
Figura 11: Algumas plataformas de sensores.	36
Figura 12: <i>RFBee</i> usado para realizar comunicação entre os sensores.	37
Figura 13: Mostra uma plataforma <i>Arduino</i> em funcionamento.	38
Figura 14: Posição simbólica dos sensores em relação às distâncias d_0 e d	42
Figura 15: Nós âncoras e nós comuns.	43
Figura 16: Etapas do algoritmo de localização.....	44
Figura 17: Exemplo da aplicação da trilateração.....	47
Figura 18: Multilateração.	48
Figura 19: <i>RSSF</i> montada em laboratório para testes com o algoritmo de localização no cenário 1.	50
Figura 20: Testes realizados com o sensor 2 na posição (3, 3.5).	52
Figura 21: Rede de sensores montada com destaque para o sensor 5.	53
Figura 22: <i>RSSF</i> montada com os sensores 6, 8, 9, e 10 adicionados.	55

Figura 23: Cenário 2 montado com a maior parte dos sensores inseridos na *RSSF*.....56

Índice de Tabelas

Tabela 1: Comparação entre os métodos utilizados nas estimativas de posição entre dois nós.	35
Tabela 2: Comparação entre as técnicas do cálculo de posição.	36
Tabela 3: Resultados do sensor 2 coordenadas (2, 3) no cenário 1.	51
Tabela 4: Resultados do sensor 2 cenário 1 coordenadas (3, 3.5).	53
Tabela 5: Resultados obtidos no sensor 5 na posição (5.5, 6) no cenário 2.	54
Tabela 6: Resultado sensor 6 posição (4, 7.5) no cenário 2.	56
Tabela 7: Resultados do sensor 7 no cenário 2 com coordenadas (1, 6).	57
Tabela 8: Resultados obtidos no sensor 8 na posição (3, 6) no cenário 2.	58
Tabela 9: Resultados do sensor 9 no cenário 2 com coordenadas (3, 3).	59
Tabela 10: Resultados do sensor 10 no cenário 2 com coordenadas (4, 3).	60

1. INTRODUÇÃO

Os avanços tecnológicos provocaram modificações na maneira como a computação busca de dados e os disponibiliza para análise. Neste contexto, algumas tecnologias têm sido aplicadas para análise de vários ambientes o que promete desenvolver diversas áreas de conhecimento (como agrícola, doméstica, hospitalar, militar, entre outras). A todo instante o ambiente esta em constante mudança, por isso, alguns processos podem contribuir para melhorar a produção e as tomadas de decisões que ocorrem cada vez mais rapidamente com o surgimento de novas tecnologias. Neste contexto, uma das tecnologias que promete suprir essa necessidade monitoramento e rastreamento é conhecida por redes de sensores sem fio (*RSSF*) que são capazes de captar dados do ambiente e disponibiliza-los através da *internet* em tempo real (Loureiro et al., 2003).

As redes de sensores sem fio podem ser consideradas uma classe especial de *Ad hoc* de múltiplos saltos (*MANETs – Multihop Ad Hoc Networks*) por apresentar características comuns como (Loureiro et al., 2003):

- São alimentados com bateria com restrições no limite de energia;
- Possuem baixa capacidade de processamento;
- Pouca memória para armazenamento de dados;
- E o sistema de comunicação tem baixa largura de banda.

Além disto, os projetos de protocolos de comunicação produzidos para as *RSSF* devem considerar a otimização do consumo de energia. Por isto, definir os ciclos de operação em um sensor é importante para compreender como as fontes de consumo relacionam com as

atividades executadas referentes ao processamento, ao sensoriamento e a comunicação. Os ciclos de operação são empregados para economizar energia em implantações de redes de sensores para permitir que os nós sensores alternem em períodos de atividades, inatividade (*idle*) e o modo de baixo consumo (*sleep*) (Margi et al., 2009).

Com o desenvolvimento da tecnologia de redes de sensores sem fio será possível várias aplicações com inúmeros mecanismos de *hardware* e *software* em muitos ambientes, e isto contribuirá para tornar realidade as técnicas que hoje são vistas como o futuro. Pode-se citar como exemplo, o manejo de animais no campo onde é importante conhecer a posição aproximada de um animal, o controle dos movimentos de robôs, a identificação dos dados produzidos em uma plantação, entre outras aplicações. A figura 1 mostra um exemplo de rede de sensores sem fio com seus principais componentes e a desafios (Oliveira, 2005).

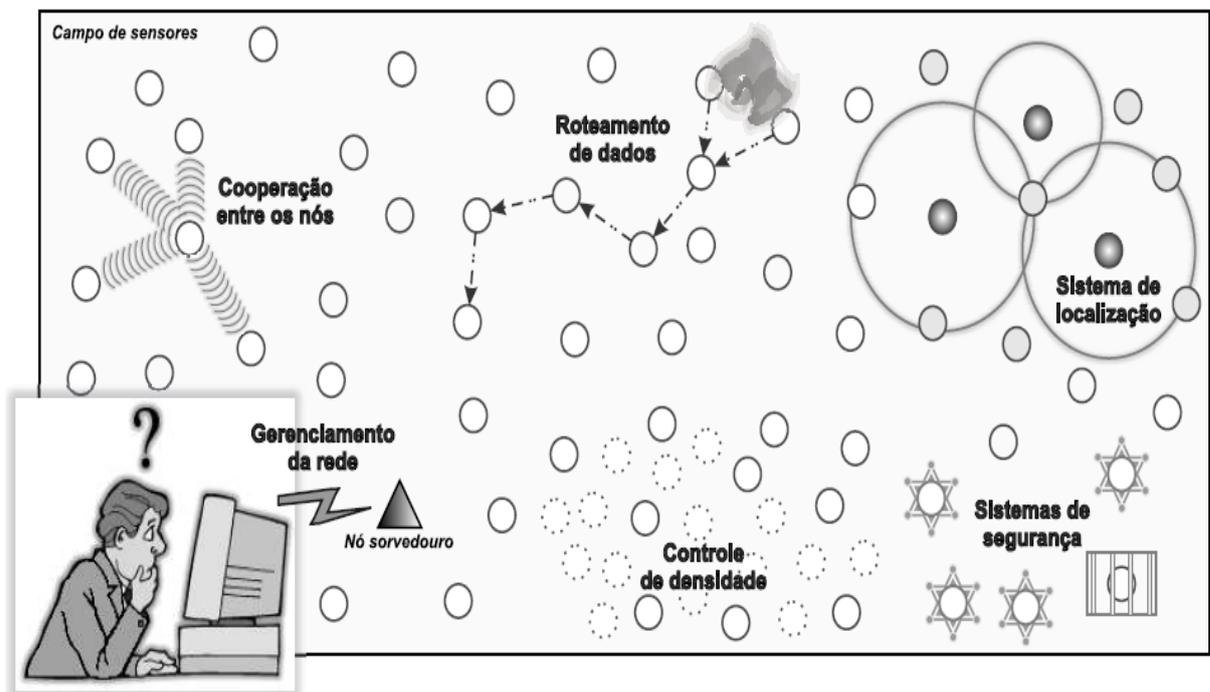


Figura 1: Rede de sensores sem fio com seus principais componentes.

Fonte: (Oliveira, 2005).

No entanto, o objetivo principal das *RSSF* é realizar o monitoramento ou rastreamento de uma área específica. Entretanto, será preciso ultrapassar alguns pré-requisitos necessários como prover mecanismos de operação entre os nós, permitir o roteamento de dados coletados, prover o gerenciamento da rede, definir um sistema de localização, realizar um controle de densidade, e fornecer um serviço de comunicação seguro (Margi et al., 2009).

Por isto, ao definir o sistema de localização poderão ser desenvolvidos muitas aplicações às quais permitirá identificar a origem das informações no ambiente, elaborar gráficos, localizar e evitar problemas, otimizar recursos, entre outras. Desta forma, o problema de localização consiste em identificar o lugar onde os nós sensores estão (e. g., longitude, latitude e altitude). Apesar de algumas tecnologias poderem solucionar o problema de localização como, por exemplo, o *Global Position System (GPS)* há uma serie de limitações físicas impostas as *RSSF* quando aplicados com visibilidade aos satélites como custos elevado na produção dos nós, imprecisão, aumento no tamanho dos sensores e consumo de energia. Devido a estas limitações, alguns algoritmos de localização precisam adequar os melhores métodos de estimativa de distância e as melhores técnicas para o calculo da posição para alcançar a precisão adequada para o sistema de localização tanto ambientes internos como externos (Oliveira, 2005).

Assim, os métodos e técnicas empregadas no algoritmo de localização para realizar o monitoramento do ambiente visa melhorar a precisão das aplicações para referenciar um ou mais nós sensores adicionados à rede. Para realizar esta tarefa será preciso conhecer os desafios computacionais envolvidos para escolher um método para estimativa de distância, uma técnica para o cálculo da posição e implementa-los em um algoritmo de localização para que seja possível chegar a resultados próximos à realidade da posição real do sensor com as coordenadas x e y (Oliveira, 2005). Na próxima seção tratará das motivações e objetivos deste trabalho.

1.1. MOTIVAÇÕES E OBJETIVOS

Nas redes de sensores sem fio foram desenvolvidas para fazer o monitoramento do ambiente e o rastreamento de objetos. Entretanto, para aumentar a escalabilidade e as possibilidade novas aplicações para as *RSSF* em ambientes internos será preciso implementar um algoritmo de localização capaz de identificar a posição de um nó sensor adicionado na rede. Devido a isto, surge um problema de localização da necessidade de identificar os dados coletados no ambiente e de se mapear geograficamente os eventos de uma determinada área. Para solucionar este problema de localização, será preciso conhecer e aplicar os métodos de estimativa de distância e as técnicas para o cálculo da posição e implementa-los em um algoritmo de localização capaz de realizar a tarefa de identificação dos nós sensores inseridos na *RSSF*.

Desta forma, o objetivo deste trabalho é obter a posição mais próxima à realidade para as coordenadas x e y de um nó sensor inserido em uma *RSSF*. Para isto, foi feita uma revisão bibliográfica capaz de auxiliar a determinar qual dentre os métodos de estimativa de localização e quais as técnicas usadas para o cálculo da posição poderão ser empregados no algoritmo de localização. Com isto, o algoritmo de localização poderá ser implementado em uma plataforma de sensor que melhor atende as expectativas deste trabalho e realizar os testes necessários para consolidação do mesmo.

Dentre os métodos de estimativa de localização existentes será empregado neste trabalho o *Received Signal Strength Indicator (RSSI)* que estima a posição através da perda de potência do sinal. As técnicas usadas para o cálculo da posição empregado no algoritmo de localização foram a trilateração e a multilateração na qual três ou mais nós sensores são empregado para determinar a posição de um nó sensor inserido na *RSSF*.

Entre os algoritmos de localização existentes para solucionar o problema de localização o implementado neste trabalho obedecem a vários princípios, como:

- Utilizam poucos recursos de memória uma vez que os nós armazenaram somente os dados da estimativa de distância dos nós âncoras em relação aos sensores inseridos;
- Precisam de poucos nós âncoras que ajudam os outros nós a estimarem suas posições por conhecerem suas coordenadas;
- Pode-se utilizar em redes pouco densas desde que possua no mínimo três nós âncoras com conhecimento de suas coordenadas relativas no sistema de localização;
- Não haverá propagação do erro de localização, pois cada nó sensor inserido calculará a sua posição em relação nos âncoras;
- A escolha dos nós referência: serão feitas para determinação dos nós âncoras.
- Permite trabalhar com coordenadas locais: por serem organizados em um lugar específico no qual os nós âncoras possuem as coordenadas relativas em coerência com as dimensões de espaço que deseja se monitorar.

Assim, após considerar os aspectos gerais que motivaram o desenvolvimento deste trabalho, o próximo passo será implementar o algoritmo de localização baseado em nós âncoras com o método de estimativa de posição *RSSI* com as técnicas de cálculo de posição trilateração e multilateração definidos na seção metodologia. O intuito é implementar um algoritmo de localização capaz de realizar o cálculo da posição o mais próximo dos valores reais nos cenários propostos. A próxima seção tratará sobre o contexto envolvido.

1.2. CONTEXTO

O problema de localização pode ser encontrado na literatura em alguns trabalhos que estudam e propõem soluções para o problema de localização, e com isto, será possível chegar ao estado da arte para a área de *RSSF*. Desta maneira, a revisão bibliográfica relaciona os principais métodos de estimativa de localização, técnicas para o cálculo da posição, e algoritmos de localização existentes serão abordados para nortear quais serão os principais métodos de estimativa de distância e técnicas para o cálculo da posição que poderiam ser aplicados e implementados no algoritmo de localização baseado em nós âncoras com coordenadas relativas. Na próxima seção as principais contribuições deste trabalho alcançadas pelo algoritmo de localização.

1.3. PRINCIPAIS CONTRIBUIÇÕES

Com este trabalho pretende-se analisar e implementar um algoritmo de localização baseado em nós âncoras para *RSSF* com algumas características:

- Oferecer escalabilidade para *RSSF* inserir outros nós sensores;
- Funcionamento em ambientes de coordenadas relativas.

Além disto, algumas contribuições secundárias deste trabalho são:

- Através do estudo detalhado do tema é discutido, relatado e mostrado adaptações às soluções atualmente existentes;
- Pesquisas e análise de técnicas e algoritmos de localização com objetivos similares ao proposto neste trabalho;
- Avaliação do algoritmo de localização proposto, por meio de testes em laboratório, e análise dos resultados.

Com tudo isso, a próxima seção descreverá resumidamente a organização deste trabalho.

1.4. ORGANIZAÇÃO DA MONOGRAFIA

Esta monografia está organizada em cinco seções. Na seção 2 uma revisão de literatura sobre o assunto principal deste trabalho para abordar os conceitos mais importante pertinente ao algoritmo de localização. A seção 3, materiais e métodos, destaca quais as ferramentas, os

métodos de estimativa de localização, e as técnicas para o cálculo da localização utilizadas na implementação do algoritmo de localização. Os resultados e discussão serão apresentados durante a seção 4. A seção 5 finalizará este trabalho com a conclusão e abordará também uma perspectiva para novos trabalhos relacionada ao tema principal.

2. REVISÃO DE LITERATURA

Nesta seção será realizada uma revisão bibliográfica das propriedades envolvidas no sistema de localização, dos principais métodos de estimativa de distância, das técnicas para o cálculo da posição e dos algoritmos de localização existente que compõem o sistema de localização.

2.1. PROPRIEDADES DE LOCALIZAÇÃO

A simples intuição de providenciar informações sobre a localização de um nó sensor tem algumas propriedades que devem ser classificadas para tornar o processo de localização claro. Sendo assim, as propriedades mais importantes dentro do sistema de localização são (Karl & Willig, 2005):

- Posição física ou localização simbólica: o sistema fornece a posição física de um nó (em alguns casos, coordenadas numéricas) ou faz o nó aprender sobre sua localização simbólica (por exemplo, “uma sala” ou “um quarto em hotel”).
- Coordenadas absolutas ou relativas: um sistema de coordenadas absoluto é válido para todos os objetos e é incorporado em alguns quadros de referência geral. Por exemplo, as posições no Universal Transverso de Mercator (*UTM*) as coordenadas formam um sistema de coordenadas absolutas para qualquer lugar na Terra. As coordenadas relativas, por outro lado, podem ser diferentes para qualquer objeto localizado ou conjunto de objetos. No presente trabalho, as *RSSF*, os nós sensores usaram o sistema de coordenadas relativas para localizarem suas posições em relação ao sistema de localização montado em um laboratório.

- Computação localizada ou centralizada: a computação localizada busca realizar todos os cálculos nos próprios nós sensores, enquanto, a computação centralizada as informações de distância são enviadas para a base (ou gerência) que realiza os cálculos e as distribui de volta para os nós sensores. Uma questão importante de privacidade pode acontecer, por exemplo, quando um nó que não deseja revelar sua posição para a entidade central, mas quer conhecer sua posição, por isto, usa-se a computação localizada.
- Exatidão e precisão: Os dois mais importantes fatores de mérito para um sistema de localização são a exatidão e a precisão de seus resultados. Precisão é a distância mais próxima medida que um algoritmo de localização, por exemplo, poderá chegar da posição real do nó sensor adicionado na *RSSF* (alta precisão indica um pequeno desencontro). No entanto, a exatidão é a posição real do sensor dentro do sistema de coordenadas relativa, ou seja, é a medida perfeita. Por exemplo, um sistema pode pretender dar uma exatidão de 20 cm, com pelo menos 95% de precisão. Evidentemente, os valores de exatidão e precisão só fazem sentido quando consideradas em conjunto, formando a exatidão/precisão característica de um sistema.
- Escala: Um sistema pode ser destinado a diferentes escalas, por exemplo, na implantação interior - o tamanho de uma sala ou um edifício - ou em implantação ao ar livre - um parque, um estacionamento ou até mesmo no mundo. Duas métricas importantes aqui são a área na qual o sistema pode cobrir por unidade de infraestrutura e o número de objetos localizáveis por unidade de infraestrutura por intervalo de tempo.
- Limitações: Para algumas técnicas de posicionamento, existem limitações de implantação inerentes. *GPS*, por exemplo, não funciona dentro de casa; outros sistemas têm apenas faixas limitadas sobre o qual atuam.
- Custos: Sistemas de posicionamento possuem custos de tempo (instalação de infraestrutura, administração), espaço (o tamanho do dispositivo e o espaço para infraestrutura), energia (durante a operação) e capital (o preço de um nó e a instalação de infraestrutura).

As propriedades descritas acima auxiliaram no entendimento do sistema de localização ao permitir compreender os principais aspectos envolvidos no processo para determinar a posição dos nós sensores na *RSSF* (Karl & Willig, 2005).

2.2. SISTEMA DE LOCALIZAÇÃO

O sistema de localização surge da necessidade de mapear geograficamente um espaço para que seja possível identificar a origem das informações produzidas pelo ambiente. Por isto, o problema de localização de sensores é muito abrangente e inclui áreas como robótica, redes *Ad hoc*, telefonia sem fio, militar, aviação, astronomia e *RSSF* (Oliveira, 2005).

Entretanto, o problema de localização nesse trabalho será tratado sob a perspectiva de redes de sensores sem fio. Alguns fatores importantes para o sistema de localização restrito as *RSSF* são (Oliveira, 2005):

- Identificação dos dados coletados pelos nós: um dos principais objetivos de uma *RSSF* no qual realiza o monitoramento do ambiente através da coleta dos dados, e assim nasce a necessidade de identificar a origem dos dados coletados dentro da área de referência;
- Endereçamento dos nós: aproveitar a posição física do nó sensor como identificação única de um nó na *RSSF*; e
- Gerenciamento da rede: permite consultar e gerenciar os nós sensores localizados em áreas definidas, avaliar a cobertura e a geração de mapas de energia.

Com as características e aplicações mencionadas pelas *RSSF* o sistema de localização considera alguns requisitos que incluem:

- Auto-organização: não depende de qualquer infraestrutura;
- Escalabilidade: pode ser aplicada a redes de larga e densa escala;
- Robustez: consiste na tolerância a eventuais falhas de comunicação e de estimativas imprecisas de distâncias;
- Eficiência no uso de recursos: o sistema de localização não é o objetivo principal das *RSSF* sendo assim deve custar o mínimo possível.

Agora, para melhor compreensão do sistema de localização e de seu funcionamento, esse será dividido em partes como exemplificado na figura 2 (Oliveira, 2005):

- Estimativa da distância: este componente é responsável por estimar distâncias entre dois nós. Estas informações são utilizadas por outros componentes do sistema de localização. Alguns dos métodos mais usados são o *RSSI* (*Received Signal Strength Indicator*), o *ToA* (*Time of Arrival*), o *TDoA* (*Time Difference of Arrival*) e o *AoA* (*angle of arrival*).
- Cálculo da posição: este componente é responsável pelo cálculo da posição do nó sensor na base com as informações disponíveis. As técnicas mais usadas são a trilateração, a multilateração, e a triangulação.

➤ Algoritmo de localização: é o principal componente do sistema. Ele realiza o papel de ordenação das informações com a finalidade de estimar a posição dos nós sensores.

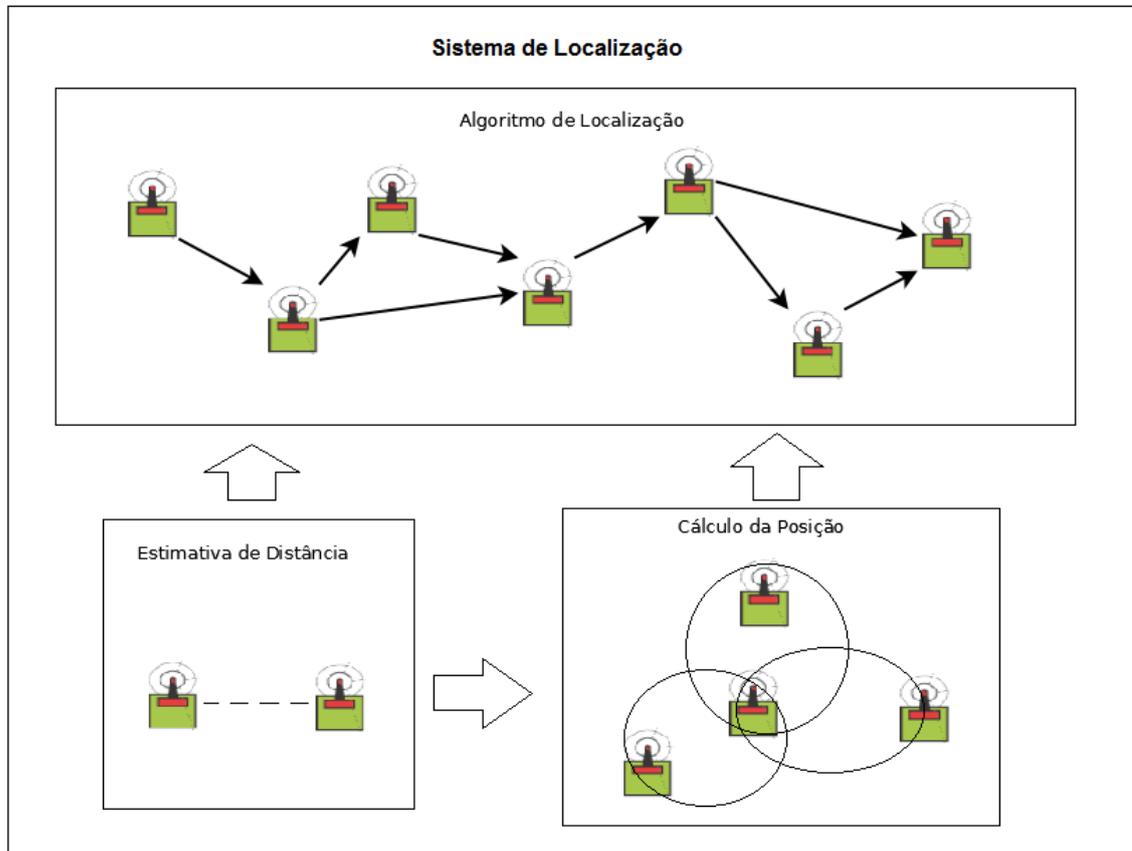


Figura 2: Componentes do sistema de localização.

Fonte: (Oliveira, 2005).

A escolha destes componentes pode afetar o sistema de localização e provocar erros, em alguns casos significativos para a determinação da posição. Por exemplo, os resultados obtidos com o método *ToA* são melhores do que *RSSI* de acordo com Oliveira (2005). Entretanto, alguns fatores devem ser considerados para escolher quais dentre os métodos de estimativa de distância e quais dentre as técnicas para o cálculo da posição serão selecionados para serem implementados no algoritmo de localização. Nas próximas seções serão abordados os principais métodos de localização, em seguida as técnicas para o cálculo da posição e os algoritmos de localização existentes.

2.3. ESTIMATIVA DE DISTÂNCIA/ÂNGULO

A estimativa da distância/ângulo é uma etapa importante para sistemas de localização através de estudos pode se mensurar a distâncias entre os nós de *RSSF* com a potência do sinal transmitido, o tempo de propagação do sinal, ou até mesmo com ângulo. Nas seções a seguir a revisão bibliográfica dos principais métodos de estimativa de distância para determinar qual será implementado no algoritmo de localização. Dentre os métodos existentes serão abordados o *RSSI* (*Received Signal Strength Indicator*), o *ToA* (*Time of Arrival*), o *TDoA* (*Time Difference of Arrival*) e o *AoA* (*angle of arrival*) (Oliveira, 2005). A próxima seção iniciará com o método *RSSI*.

2.3.1. Intensidade do Sinal Recebido (*RSSI*)

O modelo de propagação de sinal *RSSI* é capaz de estimar a posição de um nó sensor vizinho através da potência do sinal transmitido e dos cálculos das perdas de propagação. A figura 3 ilustra um nó envia um sinal com uma determinada potência que reduz conforme o sinal se propaga, isto é, quanto maior a distância menor a potência do sinal ao atingir este nó sensor (Oliveira, 2005; Savvides et al., 2001).

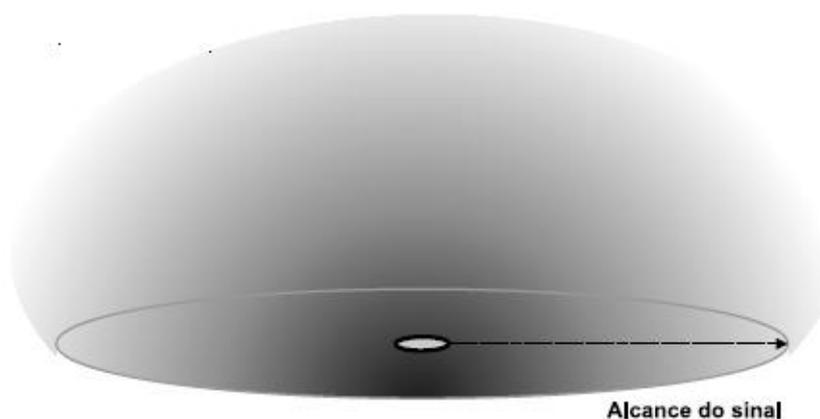


Figura 3: Propagação do sinal no *RSSI*.

Fonte: (Oliveira, 2005).

A principal vantagem deste método é seu baixo custo e facilidade para ser implementado, pois qualquer receptor é capaz de obter informações de potência do sinal, isso

permitirá uma relação custo/desempenho eficaz para determinar o posicionamento dos nós em um futuro próximo. Entretanto, esta técnica algumas desvantagens, um exemplo é a precisão que pode resultar em erros de alguns metros nas estimativas causados fatores físicos (Oliveira, 2005; Savvides et al., 2001). Além disso, a potência diminui com o quadrado da distância o que na prática este indicativo pode ser influenciado por ruídos, obstáculos e o tipo da antena (Oliveira, 2005). Na seção seguinte tratará do método *TDoA/ToA*.

2.3.2. [Diferença do] tempo de chegada (*TDoA/ToA*)

Os métodos baseados em tempo de chegada (*Time of arrival*, ToA) ou diferença do tempo de chegada (*Time difference of arrival*, TDoA) usam o tempo de propagação para determinar a distância, baseado no conhecimento da velocidade de propagação do sinal. Estes métodos podem ser aplicados com vários sinais diferentes, tais como rádio frequência, acústico, ultrassom de acordo com a infraestrutura (Savvides et. al., 2001).

Segundo Oliveira (2005) o *ToA* é um método simples e iterativo, a distância entre os nós é diretamente proporcional ao tempo que o sinal leva para se propagar de um ponto ao outro. Desta maneira, se um sinal for enviado à distância entre o nó sensor transmissor e o receptor pode ser encontrada pela velocidade de propagação do sinal de rádio representada por s_r (que dada pela velocidade da luz), e t_1 e t_2 são os tempos de saída e de chegada do sinal como mostra a equação 1 (Oliveira, 2005).

$$d = s_r * (t_2 - t_1) \quad (1)$$

O *TDoA* tem por base a diferença de tempo de que um sinal chega a múltiplos nós ou a diferença que múltiplos sinais chegam a um nó. Para isto acontecer, os nós receptores devem estar sincronizados. Além disto, precisão de um *hardware* capaz de enviar/receber os dois tipos de sinais simultaneamente. E a distância pode ser determinada com a diferença entre velocidade de propagação de rádio e do ultrassom, representadas por s_r e s_s respectivamente com o produto da diferença entre tempo de saída e o de chegada dos sinais representados respectivamente por t_1 e t_2 , exemplificado com a equação 2 (Oliveira, 2005).

$$d = (s_r - s_s) * (t_1 - t_2) \quad (2)$$

Na pratica o *TDoA* pode apresentar erros de estimativa na faixa de centímetros. Savvides et. al. (2001), realizou testes com ultrassom e obteve erros na faixa de 2 a 3 centímetros em cenários onde os nós estavam a uma distância de 3m. E conforme Whitehouse (2002) através de som audível (acústico) alcançou erros um pouco maiores, 23 cm, em um cenário onde os nós estavam localizados a 2m de distância.

2.3.3. Ângulo/direção de chegada (*AoA/DoA*)

Este método utiliza ângulo para estimativa da distância para isto são enviados sinais de propagação que formam ângulos com uma referência de reta normal no sistema de localização e permite calcular as posições dos nós através da geometria simples. O ângulo pode ser em relação com o próprio nó, com a bússola eletrônica ou um segundo sinal recebido. Geralmente, com o auxílio de três ou mais nós referencia separados uniformemente podem ser utilizados para fazer a estimativa do ângulo de chegada do sinal em relação ao nó sensor inserido na *RSSF* (Oliveira, 2005; Savvides et al., 2001).

Priyantha et. al. (2001), alcançou com esses métodos resultados com erros de estimativa na faixa de cinco graus. Entretanto, estes métodos possuem algumas desvantagens como a necessidade de *hardware* extra o que causa um aumento nos custos e no tamanho dos sensores. Além disto, na pratica são poucos sistemas de localização que necessitam de informações de ângulo de chegada do sinal.

2.4. CÁLCULO DA POSIÇÃO

Nesta etapa será realizado o cálculo com os dados das estimativas das distância/ângulo e posição dos nós sensores disponíveis. Há vários métodos que podem ser usados para o calculo da posição, dentre eles abordaremos os seguintes: a trilateração, a multilateração, e a triangulação. Nas seções seguintes será realizada a revisão bibliográfica das técnicas mais empregadas para o cálculo da posição dos nós sensores.

2.4.1. Trilateração

A trilateração calcula a posição de um nó por meio da intersecção de três círculos como ilustrado na figura 4. Com os métodos anteriores pode-se estimar a distância que será aplicada

a um nó sensor para calcular sua posição através da trilateração, será necessário conhecer a posição de três nós referências e sua distância com relação de cada um dos nós (Oliveira, 2005; Karl & Wllig, 2005).

O cálculo de cada círculo pode ser feito com equação abaixo:

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2 \quad (3)$$

Onde (x_i, y_i) é a posição de referência i , (x, y) é a posição que se deseja calcular e d_i a distância do nó à referência i . Desta forma, obtêm-se três equações quadráticas com duas variáveis desconhecidas, o que pode ser resolvido, teoricamente, em uma solução. Utilizando o método de eliminação Gauss é possível resolver este sistema (Oliveira, 2005; Karl & Wllig, 2005).

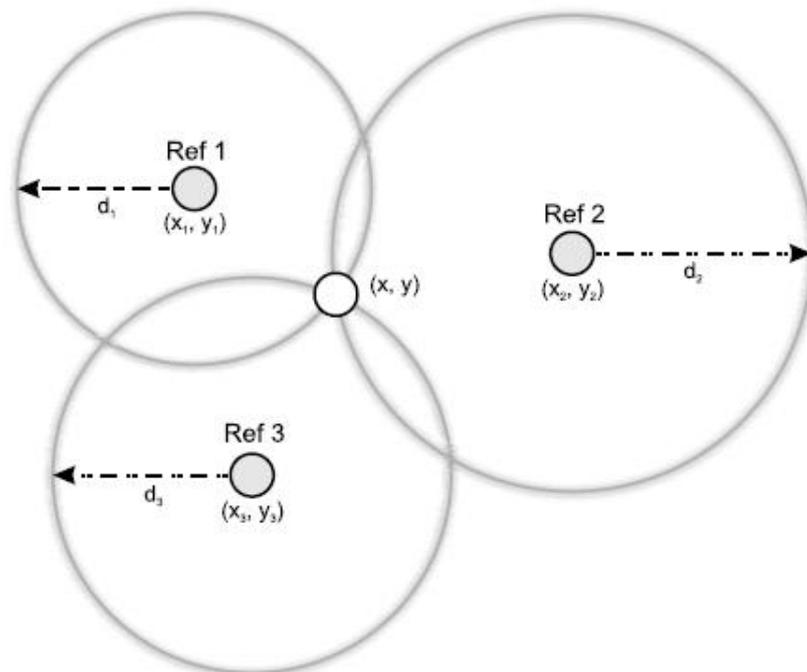


Figura 4: Intersecção dos círculos na técnica da trilateração.

Fonte: (Oliveira, 2005).

A imprecisão das estimativas de distância, e das informações de localização das referências pode dificultar o cálculo da posição na prática. Na figura 5(a) mostra os círculos um conjunto de possíveis soluções, e não um ponto singular de solução (Oliveira, 2005, Karl & Wllig, 2005).

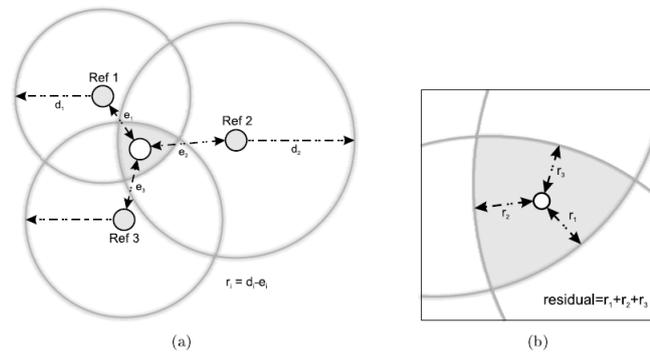


Figura 5: Modelo realista da trilateração: (a) as inconsistências das posições e distâncias das referências geram um sistema com infinitas posições, e (b) o valor residual, como sendo a diferença entre as distâncias e as calculadas.

Fonte: (Oliveira, 2005).

Assim, outros métodos mais complexos foram utilizados para resolver o sistema de equações. O método dos quadrados mínimos procura minimizar o quadrado da diferença entre as distâncias através da estimativa da posição e os valores obtidos das distâncias. O valor residual é conhecido pela diferença entre as distâncias calculadas e as obtidas conforme exemplificado na figura 5(b). E representado formalmente pela equação 4 (Oliveira, 2005):

$$pos = \min(\sum_{i=0}^m (\|pos - ref_i\| - d_i)^2) \quad (4)$$

Onde pos é posição que se quer calcular, d_i é a distância de referencia i , ref_i posição de referencia i , $\|pos - ref_i\|$ simboliza a distância entre a posição calculada e a posição da referencia i . Outros métodos podem ser utilizados como a linearização de *Taylor* e o método de *Gauss* pode ser encontrado na literatura. É importante levar em consideração o custo computacional e a precisão que se quer obter (Oliveira, 2005).

2.4.2. Multilateração

A multilateração possui os mesmos princípios da trilateração, porém com mais de três referências devem ser utilizadas para o cálculo da posição, o que gera um sistema linear sobre determinado, onde existem mais equações do que variáveis. A figura 6 exemplifica isso. (Oliveira, 2005; Karl & Wllig, 2005).

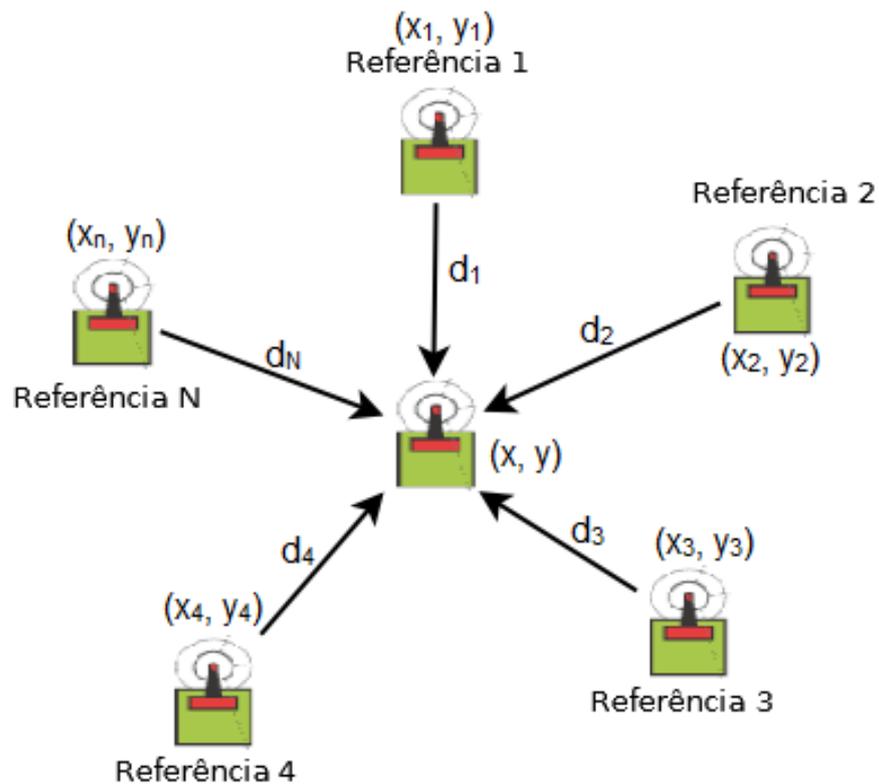


Figura 6: Exemplo da técnica da multilateração.

Fonte: (Oliveira, 2005).

Tipicamente, sistemas sobre determinados possuem várias soluções. Neste caso, a solução deste sistema será realizada, geralmente, minimizando o valor residual, como explicado na seção anterior. O número de equações e método de solução empregado no sistema determinará o custo computacional (Oliveira, 2005).

2.4.3. Triangulação

Este método foi desenvolvido para direcionar os nós a instanciarem as distâncias estimadas nos sistemas *AoA*. Os nós sensores realizam os cálculos, neste caso, usando as leis do seno e cosseno em trigonometria segundo Savvides et. al. (2001). A figura 7(a) ilustra o funcionamento da triangulação, neste caso, dois nós referências estimam o ângulo e a posição de um terceiro nó. Em *RSSF* é mais interessante o modelo exemplificado na figura 7(b) onde o próprio nó estima sua posição (Oliveira, 2005).

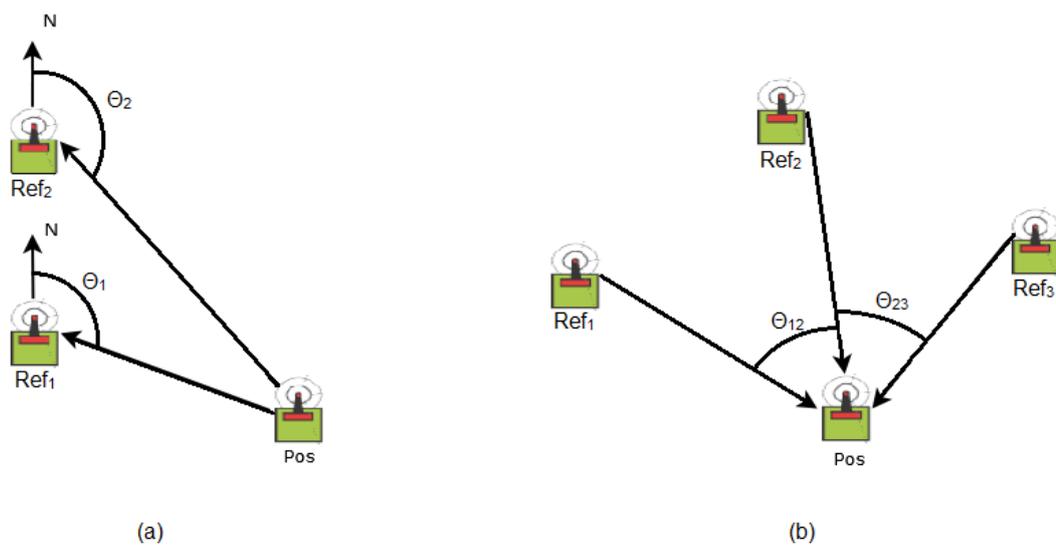


Figura 7: Triangulação (a) com duas referências para trocar informações entre si e calcularem a posição do nó, e (b) usa três referências, onde o próprio nó calcula sua posição.

Fonte: (Oliveira, 2005).

Assim, relações trigonométricas são úteis para realizar os cálculos da posição através da estimativa de ângulos de um nó livre e os três outros pontos conhecidos (Oliveira, 2005).

2.5. ALGORITMO DE LOCALIZAÇÃO

Dentro do sistema de localização o algoritmo de localização é seu principal componente. Além disto, ele é quem determina como serão manipuladas as informações da estimativa da distância e do cálculo da posição para permitir que grande parte dos nós de uma *RSSF* possa ter uma posição no sistema de coordenadas relativas (Oliveira, 2005).

Classificação dos algoritmos de localização (Oliveira, 2005):

- Centralizados ou distribuídos: o posicionamento será calculado de forma centralizada em um único nó sensor ou de forma distribuída pelos nós sensores da rede;
- Com ou sem infraestrutura: se existe a necessidade de criar alguma infraestrutura para localização (por exemplo, posicionamento manual de nós âncoras);
- Posicionamento absoluto ou relativo: as posições poderão estar em relação a um único nó sensor (ponto da rede) ou em coordenadas (por exemplo, posicionamento manual de nós âncoras); e

➤ Cenários internos ou externos: o tipo de sistema é apropriado para ambientes internos (indoors) ou externos (outdoors);

Nas próximas seções a revisão bibliográfica sobre os principais algoritmos de localização como o *Ad hoc positioning system*, o *Recursive position estimation*, e o *The global positioning system*.

2.5.1. *Ad hoc Positioning System (APS)*

De acordo com Niculescu & Nath (2003) o *Ad hoc positioning system (APS)* é um algoritmo distribuído, que utiliza múltiplos saltos e que trabalha como uma extensão dos protocolos de roteamento *distance vector* e do *GPS*. Desta forma, o *APS* é um sistema de localização onde apenas uma limitada fração dos nós possui capacidade de autolocalização (dos nós âncoras). Nele as estimativas das distâncias podem ser realizadas pelos seguintes métodos *Dv-hop*, *Dv-distance* e Euclidiano, os nós podem calcular suas posições através da multilateração (Oliveira, 2005).

Este algoritmo de localização necessita de poucos nós âncoras para funcionar, porém com menos de três nós as distâncias e as posições calculadas resultam em erros maiores de localização. No *APS*, cada nó âncora possui um fator de correção para ajudar os nós sensores a lidarem com os erros de localização (Oliveira, 2005).

O *Dv-hop* e o *Dv-distance* são semelhantes sua principal diferença consiste na forma de propagação na qual um utiliza o número de saltos e o outro usa as estimativas de distâncias através do *RSSI*. Nesses casos, não são necessários fatores de correção, uma vez que os dados de distâncias estarão em metros. No método Euclidiano propaga-se a distância euclidiana real dos nós em relação aos nós âncoras. No entanto, um nó precisa no mínimo de outros dois nós que conheçam sua distância a um determinado nó âncora, da distância a estes dois últimos nós e também entre os nós envolvidos neste caso. Usa o teorema de Pitágoras nos triângulos gerados entre as retas formadas para obter o resultado das distâncias do nó sensor ao nó âncora (Oliveira, 2005).

O algoritmo *APS* precisa de poucos nós âncoras para funcionar, o modo como às distâncias são propagadas geram dados sem precisão para calcular as posições o que ocasiona imprecisão nos resultados levando a altos erros de localização. Para concluir, este algoritmo distribuído não precisa de infraestrutura, por proporcionar posicionamento absoluto.

Entretanto, é apropriado para ambientes externos e operam através de múltiplos saltos para aumentar os dados para localização dos nós sensores (Oliveira, 2005).

2.5.2. Recursive position estimation (RPE)

O algoritmo *Recursive position estimation* realiza a tarefa de localização com base em um grupo de pelo menos três nós (e/ou geralmente 5% dos nós) com apenas informações locais. As informações de localização vão aumentando à medida que os nós calculam suas posições. Este algoritmo pode ser subdividido em quatro fases como mostrado na figura 8. Na primeira fase, um nó determina seus pontos de referência, em seguida a estimativa de distância dos nós de referências, conforme Oliveira (2005), por exemplo, usando *RSSI*. Depois, os nós estimam sua posição com a trilateração ou a multilateração. Na fase final o nó torna-se referência e divulga sua posição aos demais (Oliveira, 2005).

Uma das vantagens deste método é o rápido crescimento do número de referências desta forma em pouco tempo a maioria dos nós calculam suas posições. Entretanto, o erro de localização será propagado pelos nós e quanto maior à distância do nó do conjunto de referência maior será o seu erro de localização. Além disso, esse algoritmo se caracteriza por ser distribuído, não precisa de infraestrutura densa, mas se adequada a ambientes externos e opera em múltiplos saltos para expandir as informações de localização (Oliveira, 2005).

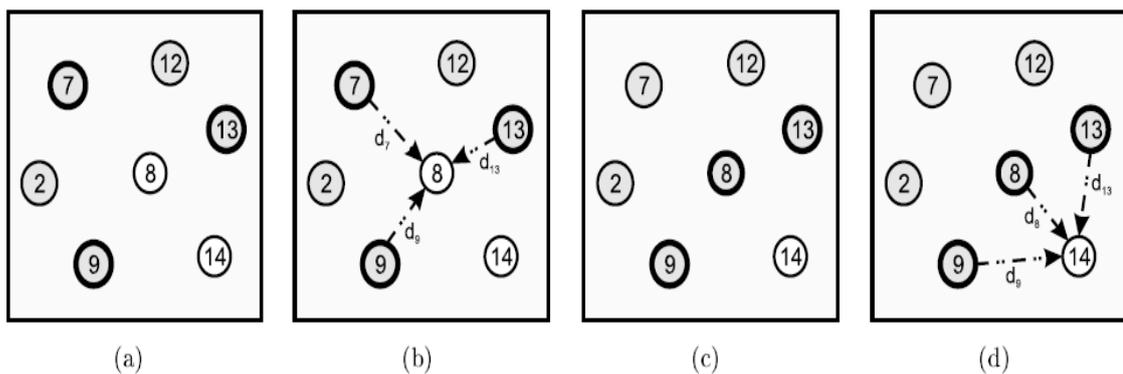


Figura 8: Exemplo das fases do *Recursive position estimation*: (a) inicialmente o nó escolhe as referências, (b) estima sua distância a cada uma das referências, (c) calcula sua posição utilizando trilateração, (d) divulga sua posição para auxiliar outros nós no cálculo de suas posições.

Fonte: (Oliveira, 2005).

2.5.3. *The global positioning system (GPS)*

The global positioning system (GPS) conforme Hofmann-Wellenho et. al. (1997) é um sistema composto por 24 satélites em órbita sobre a terra. Cada satélite faz duas rotações completas por dia a uma altura de aproximadamente 19.300 km ao redor da terra de forma que haja sempre em qualquer lugar da terra, pelo menos quatro satélites “visíveis” no céu. Os receptores *GPS* utilizam informações constantes enviadas pelos satélites, estimar sua distância utilizando *TDoA*, e por fim, calcular sua posição com a trilateração (Oliveira, 2005).

A principal vantagem deste sistema seria um erro de localização relativamente pequeno aproximadamente (2-15 metros, dependendo do receptor) o que não afetaria o sistema, porque todos os outros nós também teriam um erro similar. Entretanto, Niculescu & Nath (2001) e Savvides et. al. (2001) retratam as suas principais desvantagens como o aumento do custo, falta de visada entre os satélites, imprecisão, aumento dos sensores, e o consumo de energia quando aplicados a *RSSF*. Assim, na próxima seção identificaram quais serão os materiais e métodos empregados para implantação e implementação do algoritmo de localização proposto neste trabalho em consequência da revisão bibliográfica.

3. MATERIAL E MÉTODOS

Neste capítulo será definido um dos métodos para realizar a estimativa da distância, quais as técnicas utilizadas no cálculo da posição, e as ferramentas usadas para implementação do algoritmo localização. Entretanto, para isto deve-se primeiramente definir a metodologia usada, e avaliar o algoritmo de localização durante os testes com sistema de localização.

3.1. METODOLOGIA

A rede de sensores sem fio deverá passar por uma avaliação para verificar se o algoritmo de localização conseguirá determinar a posição real do nó sensor inserida na rede. A *RSSF* é composta por quatro nós âncoras, um sensor e a base (ou gerência) onde os dados são armazenados. Por isto, uma das tarefas mais importantes é determinar o cenário em que o algoritmo poderá ser aplicado para realizar os testes necessários para a sua aplicação (Oliveira, 2005).

Desta maneira, alguns fatores são considerados relevantes para a determinação dos cenários como o ambiente em que os testes podem ser aplicados *indoor* ou *outdoor*. Além disto, poderá ser considerada a distância entre os nós sensores, a altura dos sensores também poderá ser modificada para criar obstáculos durante os testes e comparar a consistência dos dados com os resultados obtidos com visada direta, e os resultados obtidos com a variação do coeficiente de perda de percurso.

No primeiro cenário os testes serão realizados em um ambiente *indoor* (interno) com coordenadas relativas definidas para os quatros nós âncoras e o nó sensor inserido obterá a posição da sua localização através do algoritmo de localização. Os nós sensores estarão na mesma linha de visada e serão posicionados de acordo com a figura 9.

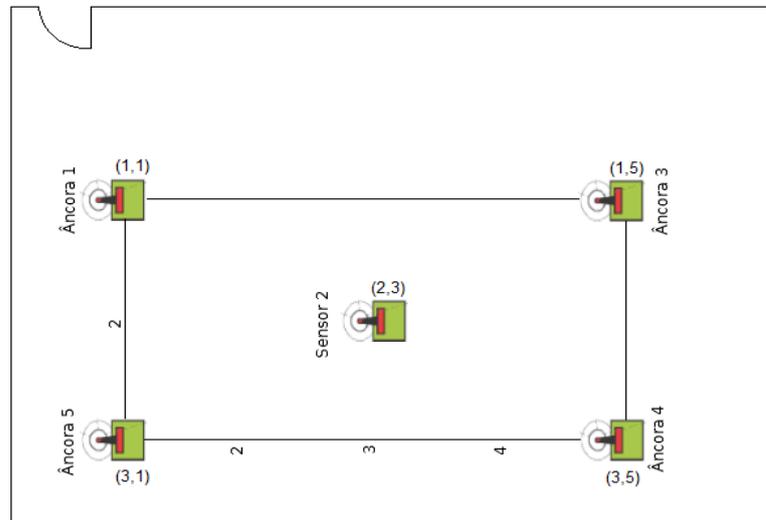


Figura 9: Cenário 1 mostra o posicionamento dos nós no ambiente.

Fonte: (Autor, 2013).

O segundo cenário manterá o mesmo ambiente, a linha de visada e altura, entretanto as dimensões dos cenários serão aumentadas e também a quantidade de sensores será dobrada de cinco para dez sensores agora para realizar os testes, conforme a figura 10.

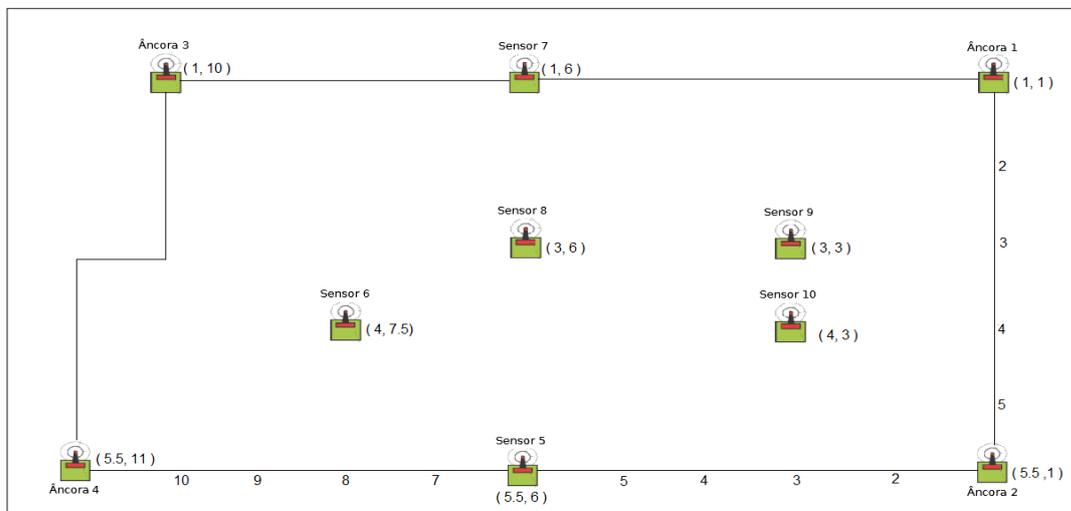


Figura 10: Cenário 2 montado para testes com nós sensores.

Fonte: (Autor, 2013).

A escolha do método da estimativa de distância utilizada pelo algoritmo de localização é extremamente importante e influenciará diretamente no desempenho final do sistema. Além disso, alguns fatores devem ser considerados como o tamanho do sensor e também os custos envolvidos (com relação ao *hardware*, ao processamento, a manutenção e a energia). Deste modo, o *RSSI* apresenta as melhores características para ser implantado em relação ao custo e a facilidade no desenvolvimento. Na tabela 1 uma apresentação simplificada das principais características das técnicas de estimativa de posição (Oliveira, 2005).

MÉTODO	PRECISÃO	DIST. DE FUNCIONAMENTO	HARDWARE	DESAFIOS
RSSI	Metro (2-4m)	Alcance do sinal	Nenhum	Grande variedade do RSSI, interferências
ToA	Centímetros (2-3cm)	Alcance do sinal	Nenhum	Sincronização dos nós
TDoA	Centímetros (2-3cm)	Alguns metros (2,3-10 m)	Envio/recepção de ultrassom	Aumento da distância de funcionamento
AoA	Alguns graus (5°)	Alcance do sinal	Conjunto de receptores	Funcionar em pequenos sensores

Tabela 1: Comparação entre os métodos utilizados nas estimativas de posição entre dois nós.
Fonte: (Oliveira, 2005).

A grande vantagem do *RSSI* para Karl (2005) nenhum *hardware* adicional é necessário, além disso, segundo Savvides et al. (2001) a distância de poucos metros este método apresenta nível de precisão aceitável para as estimativas de posição. Assim, conclui-se que o *RSSI* será empregado como método de estimativa da posição com a revisão bibliográfica realizada neste trabalho.

Diante do cenário proposto as técnicas utilizadas para o cálculo da posição utilizadas serão a trilateração e a multilateração. Estas técnicas podem apresentar uma taxa de erros considerada irrelevante. Além disso, à quantidade de nós sensores disponíveis permitiram a aplicação dessas técnicas com as ferramentas empregadas no sistema de localização. Outro fator relevante que contribuiu para a escolha destas técnicas foi à relação custo versus precisão que poderá apresentar resultados animadores com a implementação destas técnicas. A tabela 2 mostra o como às técnicas para o cálculo da posição (Oliveira, 2005).

MÉTODO	DISTÂNCIA	ÂNGULO	DESAFIOS
TRILATERAÇÃO	Sim	Não	Susceptível a imprecisão da distância
MULTILATERAÇÃO	Sim	Não	Complexidade de cálculo
TRIANGULAÇÃO	Não	Sim	Necessita de <i>hardware</i> extra

Tabela 2: Comparação entre as técnicas do cálculo de posição.

Fonte: (Oliveira, 2005).

Na seção seguinte será descreve as principais plataformas e define a plataforma *Radiuino* para implementação do algoritmo de localização e realização dos testes.

3.2. PLATAFORMA RADIUINO

Embora haja no mercado várias plataformas de sensores como *Sum Spot*, *Crossbow*, *Texas Instruments*, *Freescale*, *Telos*, e *BEAN*. A plataforma *Radiuino* foi escolhida devido à facilidade de encontrar a mesma, o baixo custo do *hardware*, por possuir um firmware estruturado com uma identificação clara de onde inserir as instruções, e pela possibilidade desenvolver o *hardware e software* para aplicações específicas, além disto, os testes poderão ser realizados em uma plataforma real não sendo necessário recorrer a simuladores nem emuladores. A figura 11 ilustra algumas das principais plataformas de sensores existentes atualmente (Maimoni & Tavares, 2012).



Figura 11: Algumas plataformas de sensores.

Fonte: (Radiuino, 2013).

Desta forma, a plataforma *Radiuino* possui as seguintes características de transmissão microcontroladores *AVR Atmel*, transmissões sem fio na banda de 915MHz, transceptor CC1101 da *Texas Instruments*, transdutores, atuadores, e uma bateria ou uma fonte de 12V de tensão. Além disto, o *firmware* é organizado em camadas semelhante ao modelo *OSI* com cinco camadas a física, enlace, rede, transporte e aplicação (Maimoni & Tavares, 2012; Radiuino, 2013).

Atualmente para esta plataforma existem dois módulos presentes no mercado conhecidos como o *BE900* fabricado pela *RadioIt* no Brasil com homologação da Anatel para operação no território nacional e o *RFBee* fabricado pela *SeedStudio* e revendido pelo próprio site. A política de comunicação desta plataforma é realizada por um *host* (computador) que envia pacotes de 52 bytes para o nó sensor, através da base. O computador monta os pacotes de acordo com o *firmware* que os prepara para executar as funções desejadas pelo nó sensor. O nó sensor recebe e processa este pacote executando o que foi determinado e responde com outro pacote contendo as grandezas medidas (Maimoni & Tavares, 2012; Radiuino, 2013).

Esta plataforma usa um transdutor de temperatura conhecido por *MCP9700* da empresa *Microchip* para coletar a informações do ambiente em grau Celsius e um transdutor de luminosidade chamado *LDR* realiza as medições em *LUX*. A base é composta por um *BE900* (que possui um microcontrolador *Atmega328* e um transceptor *CC1101* da *Texas*) e a programação do microcontrolador é realizada pelo *Usart*. A figura 12 mostra uma placa *RFBee* usada na comunicação de sensores da plataforma *Radiuino* (Radiuino, 2013).



Figura 12: *RFBee* usado para realizar comunicação entre os sensores.

Fonte: (Radiuino, 2013).

No mapa dos pacotes os *bytes* são alterados e obtêm-se o controle de quais dispositivos enviam as suas informações e quais estão somente retransmitindo estas informações para a base, estes *bytes* também são alterados para atender solicitações que medem grandezas existentes, por exemplo, temperatura, luminosidade, umidade, entre outras. Na figura 13 mostra os principais componentes da plataforma *Radiuino* (Radiuino, 2013).

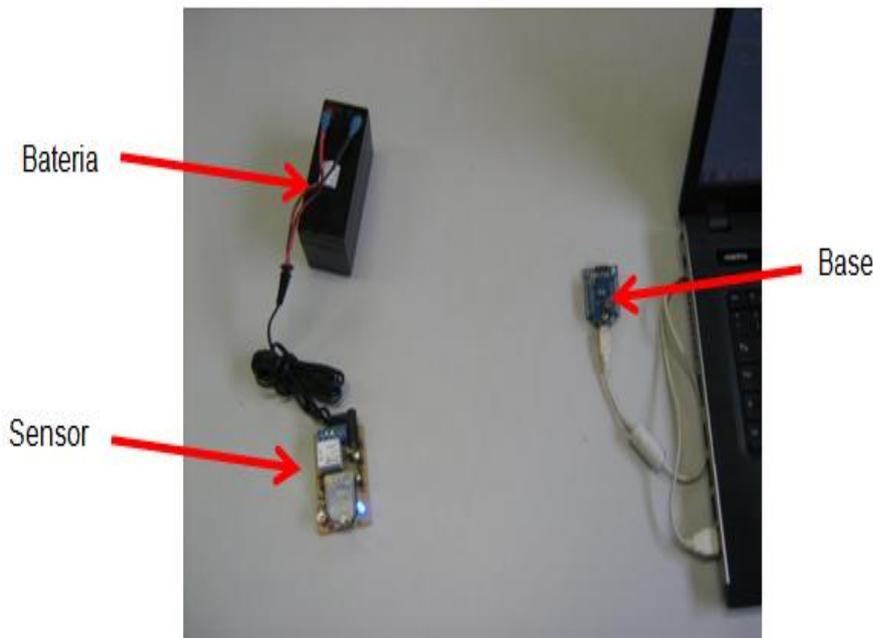


Figura 13: Mostra uma plataforma *Arduino* em funcionamento.

Fonte: (Radiuino, 2013).

A próxima seção tratará da biblioteca *Scipy* que será empregada na implementação do algoritmo de localização.

3.3. BIBLIOTECA *SCIPY*

Eric Jones, Travis Oliphant, Pearu Peterson e outros estudiosos desenvolveram o *Scipy* um *software open source* usado para realizar os cálculos em áreas como matemática, ciência e engenharia. Esta biblioteca é preparada para trabalhar junto com o módulo *Numpy*, além disso, provê inúmeras rotinas numéricas eficientes e fáceis de compreender como rotinas para integração numérica e otimização (Pereira, 2009).

O *Scipy* abrange uma variedade de algoritmos matemáticos e funções, embasada na extensão numérica para *Python*. Desta maneira, o módulo *Scipy* consegue tornar uma sessão

interativa do *Python* significativamente interessante, fornecendo ao usuário comandos de alto nível, classes para manipulação, e visualização de dados. Isto permite uma sessão interativa *Scipy* concorrer com ambientes de dados e prototipação de sistemas como o *Multlab*, *IDL*, *Octave*, *R-Lab* e *Scilab*. Entretanto, com a eficiência e com toda a expressividade da linguagem *Python* foi preciso buscar a biblioteca *Scipy* da linguagem *python* com tais requisitos (Pereira, 2009).

A funcionalidade do *Scipy* está organizada em subpacotes e engloba diferentes domínios da computação científica ao prover uma infinidade de algoritmos para o cálculo científico sendo possível aplicar cálculos avançados. Para exemplificar isto, alguns destes módulos serão destacados a seguir (Pereira, 2009):

- Funções de interpolação,
- Rotinas de álgebra linear,
- Resolução de sistemas de equações lineares,
- Rotinas de otimização,
- Rotinas de processamento de sinais,
- Funções estatísticas, distribuições discretas e contínuas,
- Entrada/saída de dados,
- Definição de muitas funções matemáticas,
- Integração em C/C++.

Tudo isto possibilita a criação de inúmeros algoritmos aplicados a diversas áreas do conhecimento. As funções do *Scipy* permitem a elaboração e implementação das funcionalidades do sistema de localização em questão neste trabalho.

3.4. BIBLIOTECA *NUMPY*

A palavra *Numpy* significa número *python* (*Numeric Python*) é modulo projetado para trabalhar com arranjos, vetores, matrizes de N dimensões. Esta biblioteca provê muitas funções sofisticadas algumas delas em destaque como objeto *array* para a implementação de arranjos multidimensionais, cálculo com matrizes, ferramentas para álgebra linear, transformadas de *Fourier* básicas, ferramentas sofisticadas para geração de números aleatórios (Pereira, 2009).

Além disso, as classes criadas podem ser facilmente herdadas, permitindo a customização do comportamento (por exemplo, dos operadores típicos de adição, subtração,

multiplicação, e entre outras.) sendo ideal para o cálculo da posição dos sensores. A principal vantagem do *Numpy* é entidade *arrays* são semelhantes às listas do *Python*, porém são considerados vetores homogêneos porque cada elemento do *array* deve ser do mesmo tipo, tipicamente um valor numérico como *int* ou *float* e geralmente podem ser muito mais eficiente que listas. Também é possível aplicar matriz com o conceito de *array* bidimensional e efetuar várias operações entre elas (Pereira, 2009).

Assim, aplicação desta biblioteca ao sistema de localização permitirá realizar os cálculos de posição para as técnicas da trilateração e multilateração e determinar as funções aplicadas aos nós sensores. A próxima sessão tratará dos resultados e discussão sobre o algoritmo de localização.

4. RESULTADOS E DISCUSSÃO

Atualmente existem diversas maneiras de implementar um algoritmo de localização como abordado na seção revisão de literatura, neste trabalho será implementado um algoritmo de localização com coordenadas relativas para localizar os nós sensores inseridos na rede conforme descrito no Apêndice A. Existem vários algoritmos de localização com a finalidade de definir a posição dos nós em ambiente externo (*outdoor*), por exemplo, o *GPS*. Entretanto, este trabalho pretende desenvolver um algoritmo de localização com coordenadas relativas voltado para ambiente interno (*indoor*) que consiga determinar a posição dos nós sensores adicionados na *RSSF* em coordenadas (Patwari et al., 2005).

O algoritmo de localização com coordenadas relativas usará os nós sensores com posições conhecidas denominadas nós âncoras para determinar a posição de um nó sensor adicionado na *RSSF*. Para isto, é importantíssimo realizar uma análise do cenário para obter o valor do coeficiente de perda de percurso (η) do ambiente, tendo em vista que este valor poderá ser alterado a cada requisição que o sistema fizer para determinar a posição do nó sensor inserido na *RSSF*. Após a análise e determinação dos cenários realizada na seção metodologia que será implantada a *RSSF* os valores para o coeficiente de perda de percurso será realizado através dos seguintes passos (Patwari et al., 2005):

- Escolher uma distância de referência d_0 ;
- Colocar os sensores a uma distância d_0 um do outro e medir a potência recebida;
- Escolher uma distância de referência para d (que pode ou não ser diferente de d_0);
- Colocar os sensores a uma distância d um do outro e medir a potência recebida.

De acordo com Patwari et al. (2005) são usados no mínimo dois sensores para realizar o processo de determinação do coeficiente de perda de percurso. O algoritmo de localização poderá também utilizar as informações das coordenadas dos nós âncoras para o cálculo da potência recebida o que permite o uso do próprio cenário para o cálculo da potência de recepção entre os nós âncoras. Com as informações obtidas no processo descrito acima, poderá encontrar o coeficiente da perda de percurso η através da equação (5) (Karl & Wllig, 2005; Patwari et al., 2005).

$$\eta = \frac{(L_{(d_0)} - L_{dB})}{10 \log\left(\frac{d}{d_0}\right)} \quad (5)$$

Onde L_{dB} é potência de recepção (dBm) em relação a distância d , e $L_{(d_0)}$ é potência de recepção entre a distância d_0 . O coeficiente de perda de percurso é importante para o cálculo do *RSSI* por estipular a perdas de propagação do sinal causas pelo ambiente, além disso, permite um ajuste para o cálculo da posição entre os nós âncoras e os nós sensores comuns inseridos na *RSSF*. A figura 14 ilustra o posicionamento dos sensores e as respectivas distâncias d_0 e d (Karl & Wllig, 2005; Patwari et al., 2005).

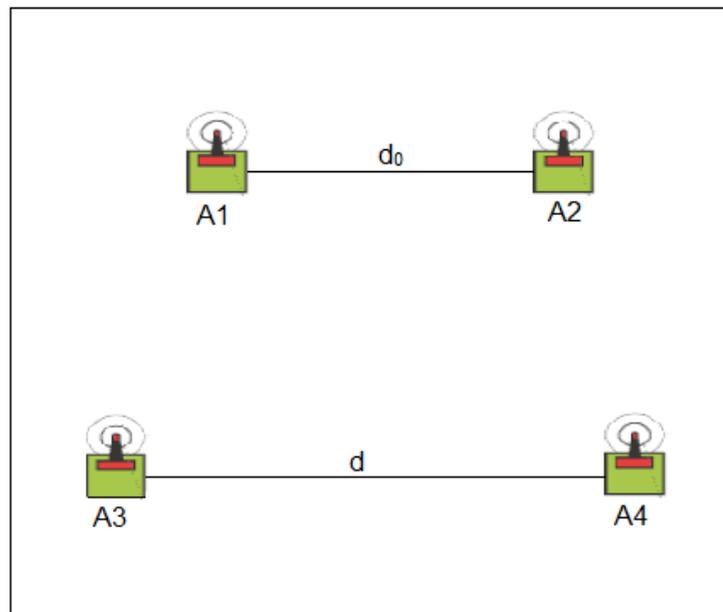


Figura 14: Posição simbólica dos sensores em relação às distâncias d_0 e d .

Fonte: (Autor, 2013).

Após definir o ambiente para aplicar o algoritmo de localização os nós âncoras foram posicionados no ambiente e foram representados por *Âncora 1*, *Âncora 2*, *Âncora 3* e *Âncora 4* é importante frisar que estes nós conhecem suas respectivas posições e auxiliaram os nós sensores comuns para solucionar o problema de localização e encontrar a sua posição no sistema de coordenadas relativas do algoritmo de localização. Em seguida são adicionados os nós comuns representados por *N* para serem localizados pelo sistema e definir a posição deste na *RSSF* como exemplificado na figura 15 (Akyildiz & Vuran, 2011; Patwari et al., 2005).

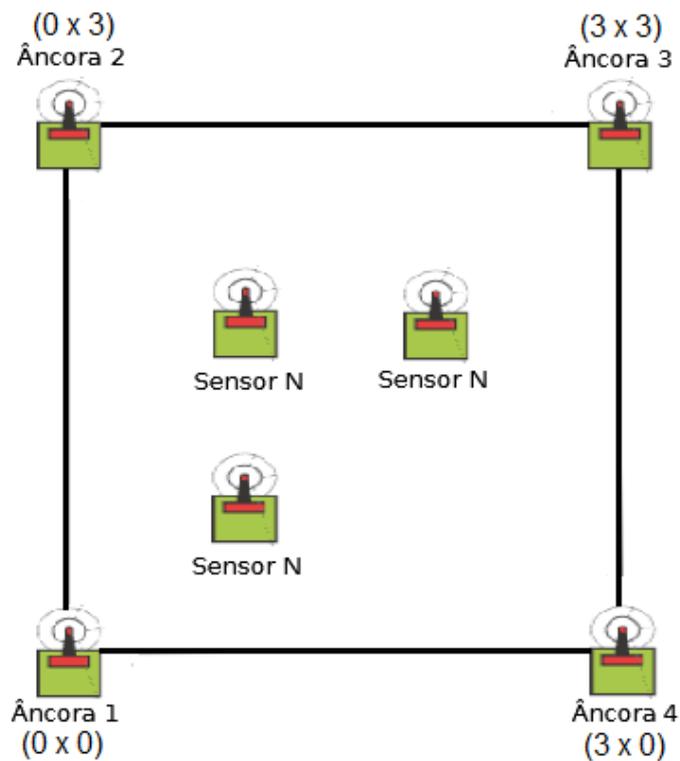


Figura 15: Nós âncoras e nós comuns.

Fonte: (Autor, 2013).

Após isto, para determinar as coordenadas relativas (x, y) de um nó sensor adicionado na *RSSF* à base envia um pacote para o e mesmo. O pacote enviado pela base para o sensor adicionado contém os endereços dos nós âncoras que em seguida o nó sensor deverá enviar um pedido dos dados do *RSSI* dos respectivos nós âncoras para que seja possível o cálculo de sua posição para os nós âncoras (Patwari et al., 2005). O nó recebe outro pacote com os dados dos nós âncoras com as informações da sua posição (x, y) , e o *RSSI*. O sensor armazena tais informações, até terminar de receber os dados de todos nós âncoras, e enviar um único pacote

como exemplificado na matriz (6) para a base realizar os cálculos da posição (Akyildiz & Vuran, 2011; Patwari et al., 2005).

$$\begin{bmatrix} x1 & y1 & RSSI1 \\ x2 & y2 & RSSI2 \\ x3 & y3 & RSSI3 \\ x4 & y4 & RSSI4 \end{bmatrix} \quad (6)$$

O fluxograma da figura 16 ajudará a compreender melhor como funciona o algoritmo de localização para realizar o processo de requisição do *RSSI* aos nós âncoras descrito acima até chegar ao cálculo da posição do sensor inserido na *RSSF*.

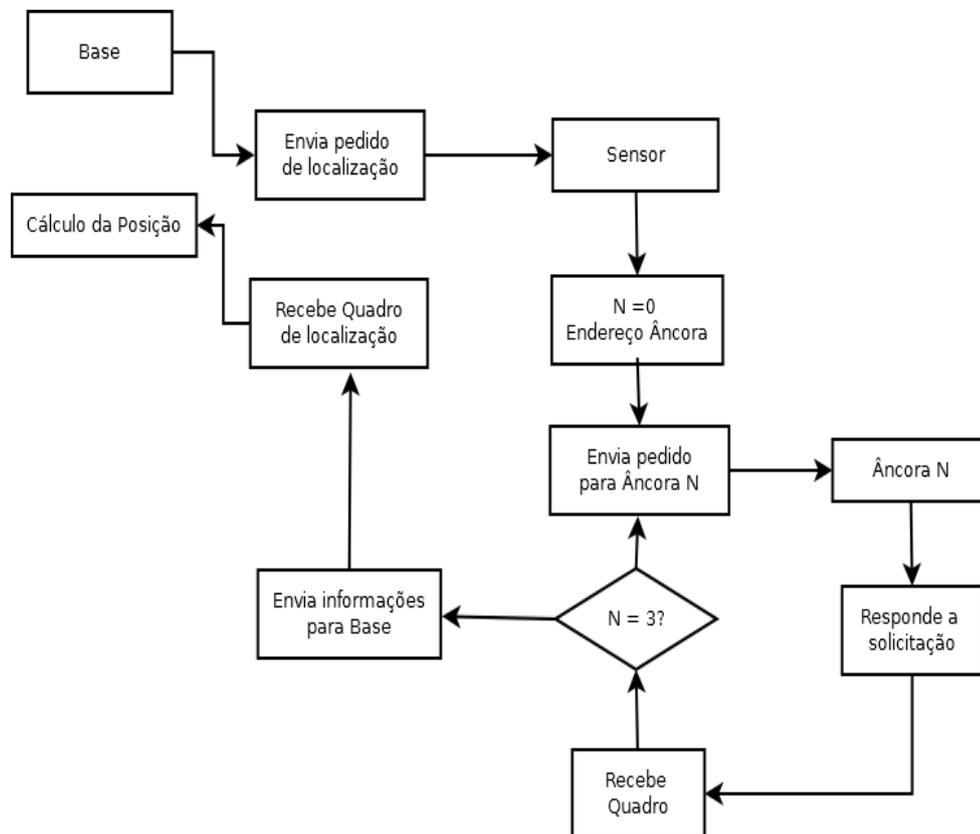


Figura 16: Etapas do algoritmo de localização.

Fonte: (Autor, 2013).

Depois de realizar a estimativa das distâncias entre o nó sensor adicionado na rede e os nós âncoras deve-se realizar o cálculo da distância como descrito na seção seguinte que tratará sobre o cálculo da distância dos sensores aos nós âncoras.

4.1. CÁLCULO DA DISTÂNCIA DOS SENSORES AOS NÓS ÂNCORAS

A base ou gerência possui as informações necessárias para realizar o cálculo da posição dos nós sensores. Com a expressão abaixo é possível calcular a distância dos nós sensores em relação aos nós âncoras com a expressão (7).

$$d = d_0 * 10^{\frac{P_{TX} + G_{TX} + G_{RX} - 10 \log\left(\frac{4\pi * d_0}{\lambda}\right)^2 - P_{RX}}{10 * \eta}} \quad (7)$$

Na expressão (7) acima d é a distância que se quer encontrar, d_0 é a distância de referência utilizada para o cálculo de η (que é o coeficiente de perda de percurso), P_{TX} é a potência de transmissão, G_{TX} e G_{RX} são os ganhos de transmissão e o de recepção respectivamente, λ é o comprimento de onda do sinal de rádio e P_{RX} a potência de recepção. É importante observar que o *RSSI* medido entre os nós âncoras e os sensores adicionados na equação (6) é substituído na equação (7) pela potencia de recepção.

Após efetuar os cálculos das posições da equação (7) obtêm-se uma matriz com $x1$, $y1$ e $d1$ onde x representa o eixo das abscissas, y representa o eixo da ordenadas, e d representa às distâncias calculadas em relação aos nós âncoras, resultando na matriz M da equação (8) (Karl & Willig, 2005; Patwari et al., 2005).

$$M = \begin{bmatrix} x1 & y1 & d1 \\ x2 & y2 & d2 \\ x3 & y3 & d3 \\ \dots & \dots & \dots \\ xn & yn & dn \end{bmatrix} \quad (8)$$

Até o momento, foi realizado o cálculo das distâncias entre os nós âncoras e o nó sensor adicionado. Agora iniciará o processo de determinação das posições com a técnica da trilateração ou da multilateração conforme a necessidade da aplicação para a *RSSF* os detalhes serão abordados nas próximas seções o sistema de equações apresentado a seguir poderá ser aplicado para ambas às técnicas com pequenas modificações.

4.2. LATERAÇÃO

Desta forma, com o conjunto de referência obtém-se o seguinte sistema de equações (9) (Karl & Wllig, 2005; Langendoen & Reijers, 2003; Patwari et al., 2005):

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \quad (9)$$

Assim, isola-se uma das equações e substituem-se em cada uma das $n-1$ equações restantes para solucionar o sistema com as seguintes expressões encontradas nos trabalhos de Karl & Wllig (2005), Langendoen & Reijers (2003) e Patwari et al. (2005) conforme a expressão 10.

$$\hat{a} = (B^T B)^{-1} B^T c \quad (10)$$

Ao reordenar os termos os seguintes sistemas de equações lineares serão obtidos na forma de $Ax=b$ de acordo com Karl & Wllig (2005) e Patwari et al. (2005), onde:

$$B = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \quad (11)$$

$$c = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + d_n^2 - d_1^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + d_n^2 - d_{n-1}^2 \end{bmatrix} \quad (12)$$

Como resultado, \hat{a} retorna um vetor com as posições x e y .

4.2.1. Cálculo da Trilateração

A técnica da trilateração, neste algoritmo de localização baseia no cálculo das distâncias entre os nós âncoras e os nós sensores. O cálculo da trilateração é realizado a partir das distâncias para determinar a posição (x, y) do nó sensor. Entretanto, a matriz M serve de

entrada para o cálculo da trilateração criado na matriz B (equação 11) com dimensões 2×2 através do algoritmo seguinte:

$$\begin{aligned} &\text{para } i=0 \text{ até } 2 \text{ faça} \\ &B[i,0] = 2*(M[i,0]-M[2,0]) \\ &B[i,1] = 2*(M[i,1]-M[2,1]) \end{aligned}$$

A figura 17 exemplifica um modelo trilateração.

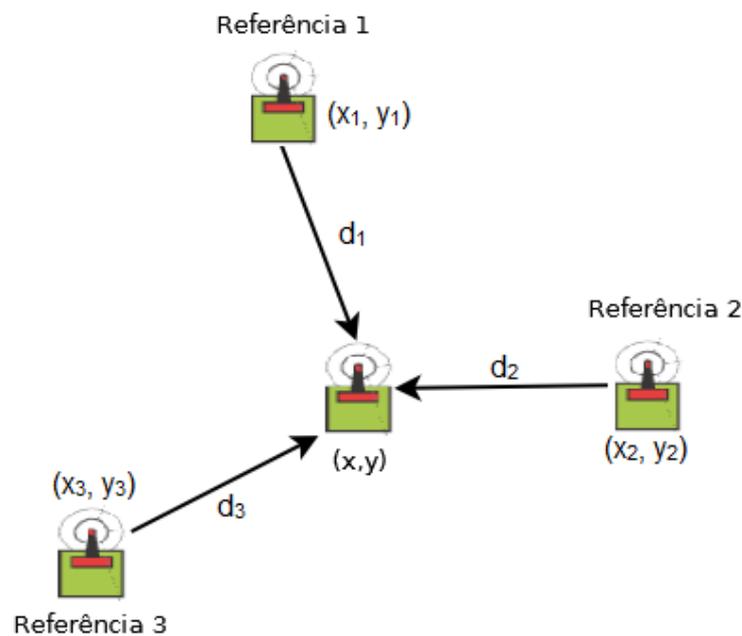


Figura 17: Exemplo da aplicação da trilateração.

Fonte: (Oliveira, 2005).

A matriz c (equação 12) com dimensões 2×1 será criada com o algoritmo:

$$\begin{aligned} &\text{para } i=0 \text{ até } 1 \text{ faça} \\ &c[i] = M[i,0]^2 - M[2,0]^2 + M[i,1]^2 - M[2,1]^2 \end{aligned}$$

Após criar as matrizes B e c usa a equação 10 para determinar a posição do sensor utilizam-se as funções de tratamento de matrizes a seguir (Karl & Wllig, 2005; Patwari et al., 2005):

$$\text{InversaDaTranspostaBporB} = \text{Inversa}(\text{TranspostaB} * \text{B})$$

$$\text{Posicao} = (\text{InversaDaTranspostaBporB} * \text{TranspostaB}) * \text{c}$$

A variável *Posicao* conterá a posição (x, y) do sensor.

4.2.2. Cálculo da Multilateração

A multilateração é uma técnica empregada para o cálculo da posição entre os nós sensores e os nós âncoras visa aumentar a precisão da localização em relação à técnica da trilateração. A multilateração possui como referência mais de três nós sensores para realizar o cálculo da posição, neste trabalho será usado quatro nós âncoras para determinar a posição do nó sensor adicionado. A figura 18 ilustra um exemplo da técnica da multilateração.

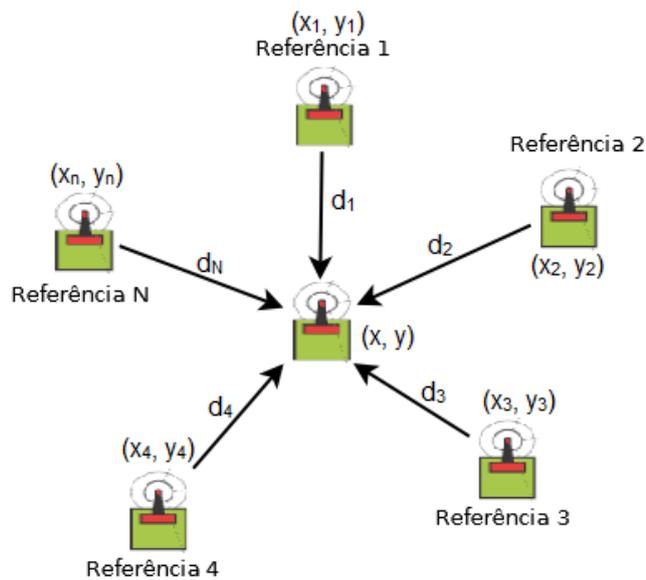


Figura 18: Multilateração.

Fonte: (Oliveira, 2005).

Como entrada para o sistema de equações apresentado anteriormente as distâncias entre o nó sensor e os nós âncoras da matriz *M* também será empregado para o cálculo da posição. Será criada uma matriz *B* (equação 11) de dimensão 3x2, através do seguinte algoritmo:

$$\text{para } i=0 \text{ até } 3 \text{ faça}$$

$$B[i,0] = 2*(M[i,0]-M[3,0])$$

$$B[i,1] = 2*(M[i,1]-M[3,1])$$

A matriz c (equação 12) com dimensões 3x1 será criada com o algoritmo seguinte:

para $i=0$ até 3 faça

$$c[i] = M[i,0]^2 - M[3,0]^2 + M[i,1]^2 - M[3,1]^2 + M[i,2]^2 - M[3,2]^2$$

Após criar as matrizes B e c usa a equação 10 para determinar a posição do sensor utilizam-se as funções de tratamento de matrizes a seguir (Karl & Wllig, 2005; Patwari et al., 2005):

$$\mathbf{InversaDaTranspostaBporB} = \mathbf{Inversa}(\mathbf{TranspostaB} * \mathbf{B})$$

$$\mathbf{Posicao} = (\mathbf{InversaDaTranspostaBporB} * \mathbf{TranspostaB}) * \mathbf{c}$$

A variável $Posicao$ conterà a posição (x, y) do sensor. Na próxima seção a implementação do algoritmo de localização.

4.3. IMPLEMENTAÇÃO DO ALGORITMO DE LOCALIZAÇÃO

Após descrever o passo a passo do algoritmo o algoritmo de localização o código foi implementado com as ferramentas *Numpy* e *Scipy* da linguagem *Python* poderá ser encontrado no Apêndice A em caso de alguma dúvida.

4.4. TESTES COM O ALGORITMO DE LOCALIZAÇÃO

Para consolidar o algoritmo de localização e avaliar quantitativamente o algoritmo de localização foram realizados vários teste para averiguar as funcionalidades do sistema de localização. Nos primeiros testes os nós âncoras foram organizados com 2 metros para a coordenada x e 4 metros para a ordenada y entre os nós. E em seguida adicionar um nó sensor na rede para obter a coordena x e y do mesmo em alguns pontos conhecidos e verificar os resultados obtidos com o algoritmo de localização condiz com as coordenadas reais. A figura 19 mostra a *RSSF* em funcionamento no laboratório com as coordenadas especificadas no cenário 1 descrito na seção metodologia e ilustrado pela figura 9.



Figura 19: RSSF montada em laboratório para testes com o algoritmo de localização no cenário 1.

Fonte: (Autor, 2013).

Na figura 9 descrito na seção metodologia onde o posicionamento real do sensor 2 está coordenada (2, 3). Alguns valores foram atribuídos para o coeficiente de perda de percurso obter os resultados de posicionamento mesmo que alguns valores são hipotéticos e mostram resultados com maior precisão, como por exemplo, o coeficiente de perda de percurso igual 10. Nos testes realizados neste ambiente com o sensor 2 o coeficiente de perda de percurso igual a 10 os experimentos chegaram quase a exatidão da posição com os resultados muito próximos a realidade. Na tabela 3 os resultados obtidos com o cenário exemplificado.

Posição (2, 3)	Trilateração		Multilateração	
	Sensor 2	X	Y	X
$\mu = 2$	111.800912176	52.7699463892	75.5834642378	70.8786703582
$\mu = 2$	104.432971488	64.5993248803	69.1056000343	46.9356391532
$\mu = 2$	28.1424400699	38.1037962897	83.9147548072	65.9899536584
$\mu = 2$	111.800912176	79.9330323427	56.0285974386	52.046874974
$\mu = 5$	3.1302344845	3.33631432007	2.72063701494	3.54111305485
$\mu = 5$	3.1302344845	3.64351242224	2.6007195646	3.37875496229
$\mu = 5$	3.1302344845	3.72558047409	2.6112129458	3.46606970474
$\mu = 5$	3.1302344845	3.72558047409	2.49129549546	3.40611097957
$\mu = 6$	2.64228880513	3.13417215542	2.21409626838	3.3482684238
$\mu = 6$	2.55948989352	3.17557161123	2.24843123572	3.33110094012
$\mu = 6$	2.18230611505	3.17557161123	2.43376196736	3.30129953738
$\mu = 6$	2.18943835027	3.40886560457	2.49133865351	3.25791545295
$\mu = 10$	2.15504424558	3.04445650391	2.09664574816	3.07365575262
$\mu = 10$	2.15504424558	3.11047773764	2.05168141519	3.05879632244
$\mu = 10$	2.17651075395	3.11047773764	2.05883691798	3.05164081965
$\mu = 10$	2.19395938703	3.11920205418	2.06465312901	3.05454892517

Tabela 3: Resultados do sensor 2 coordenadas (2, 3) no cenário 1.

Fonte: (Autor, 2013).

Com o mesmo cenário, entretanto, com a alteração da posição do sensor 2 para a posição (3, 3.5). E com os coeficientes de perda de percurso 5, 6 e 10 atribuídos para os resultados do algoritmo de localização mostraram inconsistência na maioria dos experimentos com a utilização da técnica da multilateração para determinar a posição dos nós sensor sem fio. A trilateração apresentou em alguns momentos valores muito próximo a realidade com os coeficientes de perda de percurso 6 e 10 na maioria dos experimentos para as coordenadas y somente, isto pode ser verificado tabela 4. A figura 20 exemplifica o posicionamento dos sensores na rede e a alteração da posição do sensor 2.

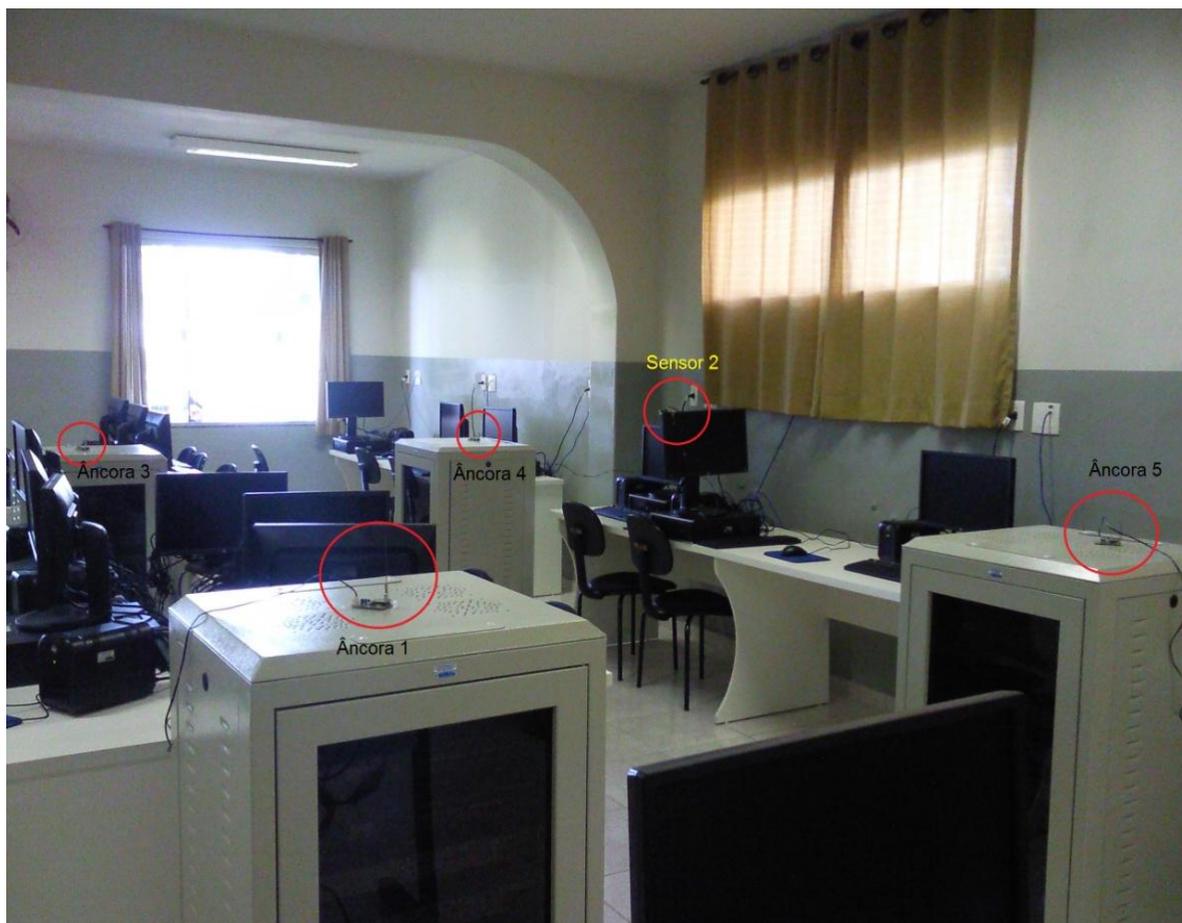


Figura 20: Testes realizados com o sensor 2 na posição (3, 3.5).

Fonte: (Autor, 2013).

Na tabela 4 os resultados dos testes feito com o sensor 2 entre os âncoras 4 e 5 na posição (3, 3.5).

Posição (3, 3.5)	Trilateração		Multilateração		
	Sensor 2	X	Y	X	Y
$\mu = 5$		3.7712432451	4.2532606072	-12.311982085	12.294873272
$\mu = 5$		3.7712432451	13.938720142	-9.1427028024	7.4817471190
$\mu = 5$		3.6183401396	16.981544793	-13.250770038	8.5469897045
$\mu = 5$		3.7712432451	16.981544793	-13.199802336	8.4960220026
$\mu = 6$		2.4523155815	3.5775380171	-2.1620462748	5.8847189453
$\mu = 6$		2.4523155815	5.6038100560	-0.3335716708	4.2108664298

$\mu = 6$	2.5180850948	3.5775380171	0.2767407176	4.6982102057
$\mu = 6$	2.4352861831	3.5557940725	0.3205225727	4.6131758778
$\mu = 10$	2.1189182198	3.1061063009	1.5118606998	3.4096350608
$\mu = 10$	2.1371891816	3.6393889603	1.42614563568	3.28386718739
$\mu = 10$	2.13718918163	3.1061063009	1.60362879065	3.37288649639
$\mu = 10$	2.13718918163	3.1061063009	1.63089784961	3.35925196691

Tabela 4: Resultados do sensor 2 cenário 1 coordenadas (3, 3.5).

Fonte: (Autor, 2013).

O segundo cenário a área de cobertura foi ampliada e passou a ser de 10 por 4.5, como descrito na seção metodologia e ilustrado pela figura 10. O número de nós sensores da rede passou a ser de 10 nós sensores nos quais 6 foram adicionados a rede para descobrir suas respectivas posições e 4 foram utilizados como âncoras. Os resultados apresentados pelo sensor 5 estão na tabela 5 pode-se verificar que as coordenadas em relação ao eixo y alguns experimentos foram muito próximos da realidade, já o eixo x não obteve resultados concisos. A figura 21 mostra o sensor 5 inserido na rede de sensores durante os testes realizados.

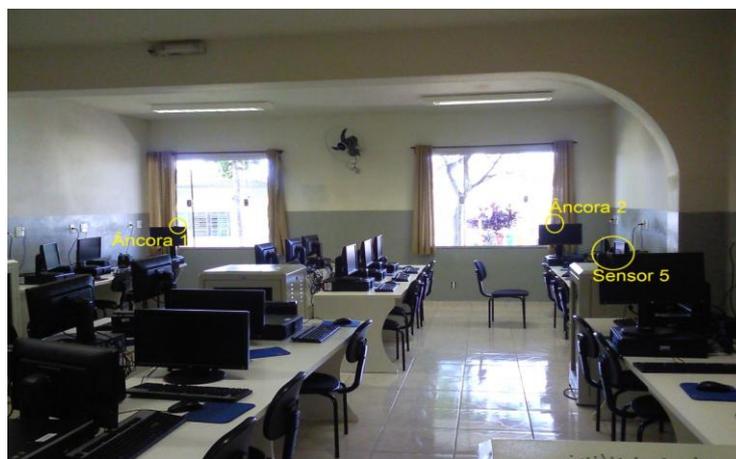


Figura 21: Rede de sensores montada com destaque para o sensor 5.

Fonte: (Autor, 2013).

Os resultados obtidos com o sensor 5 na posição (5.5, 6) aplicado ao cenário 2 foram discrepantes com o coeficiente de perda de percurso 2 os resultados começaram a aproximar da realidade a partir dos valores 5 para a técnica da trilateração. Com o coeficiente de perda

de percurso igual a 6 a multilateração teve resultados mais próximo da realidade sobre as coordenadas y do que a trilateração. Também foi testado um coeficiente de perda de percurso hipotético para analisar o comportamento do algoritmo com valor 10 os resultados foram os que mais se aproximaram da posição real do sensor com exemplificado na tabela 5.

Posição (5.5, 6)	Trilateração		Multilateração	
	Sensor 5	X	Y	X
$\mu = 2$	-10965.519936	61.7136362542	-4486.1721735	3471.89666916
$\mu = 2$	-8037.2629904	63.488325737	-3417.4234893	2494.98280003
$\mu = 2$	-822.23885276	2191.46385591	-3597.9428341	884.300934455
$\mu = 2$	-929.4352946	2612.36484212	-4248.8031014	1049.17418887
$\mu = 5$	-4.5514152012	5.84167955874	-0.5852204537	7.9291504784
$\mu = 5$	-0.9429717326	5.4625492314	-0.1136796008	5.8990187745
$\mu = 5$	-1.9892124783	5.58491255575	0.31036988289	6.79521906165
$\mu = 5$	-2.8666332636	5.60599761102	-0.0317547050	7.09803895764
$\mu = 6$	1.53212478295	5.63119587744	2.81327953533	6.30548785238
$\mu = 6$	3.28612339726	7.08145315318	1.35976241699	6.17427152875
$\mu = 6$	0.692394709803	5.60851743618	2.53204420933	6.57675401488
$\mu = 6$	3.29139843132	6.43272522551	2.24396247589	5.93945596743
$\mu = 10$	2.7639722378	5.51750347035	3.68758738277	6.00361670454
$\mu = 10$	4.11171795761	6.16422089799	3.26323927719	5.76464663569
$\mu = 10$	4.12669823993	6.10160374266	3.38057590596	5.75023217841
$\mu = 10$	2.94754195418	6.11402467365	3.81346828604	5.74486660586

Tabela 5: Resultados obtidos no sensor 5 na posição (5.5, 6) no cenário 2.

Fonte: (Autor, 2013).

O sensor 6 obteve resultados aproximados ao da realidade com coeficiente de perda de percurso 5 e 6 com maior precisão sobre o eixo x. A o eixo y não apresentou resultados próximos com a precisão esperada. A figura 22 ilustra o sensor 6 em destaque na *RSSF*.



Figura 22: RSSF montada com os sensores 6, 8, 9, e 10 adicionados.

Fonte: (Autor, 2013).

A multilateração obteve erros maiores mesmo com a trilateração apresentava resultados que atendiam as expectativas para localização como exemplificado nos experimento com coeficiente de perda de percurso 5 e 6 da tabela 6.

Posição (4, 7.5)	Trilateração		Multilateração		
	Sensor 6	X	Y	X	Y
$\mu = 2$		47.3891931458	24.3966468178	27.6138402039	13.9885663221
$\mu = 2$		50.5592325343	25.464115604	28.7104256803	13.9647435756
$\mu = 2$		47.3891931458	24.3966468178	27.6138402039	13.9885663221
$\mu = 2$		4.62920924739	6.0493545573	4.83726040923	6.15885516879
$\mu = 5$		3.90073511513	5.75985831297	4.3937466646	6.01933807585
$\mu = 5$		3.92567456316	5.74888590131	4.41016848544	6.00388270251
$\mu = 5$		3.8322176235	5.70904705183	4.3656170553	5.98978359488

$\mu = 5$	3.95615918289	5.78004017676	4.40173276624	6.01455258905
$\mu = 6$	3.61615806748	5.63618843937	4.1996521198	5.94329057217
$\mu = 6$	4.47911506065	6.06766693595	3.90488831571	5.79724620142
$\mu = 6$	4.46689723919	6.07451158478	3.91320446654	5.8137609186
$\mu = 6$	3.63869183484	5.64745532305	4.206949532	5.94653832156
$\mu = 10$	3.35072964195	5.53620349994	4.01246591798	5.88448575048
$\mu = 10$	4.44270877884	5.53913277682	3.70445599814	5.88700057924
$\mu = 10$	3.34652701983	5.53515797	4.0085861521	5.88361014488
$\mu = 10$	3.34192933301	5.53180334547	4.00961599587	5.88321737855

Tabela 6: Resultado sensor 6 posição (4, 7.5) no cenário 2.

Fonte: (Autor, 2013).

A figura 23 mostra o cenário 2 com destaque para os sensores 7, 8, 9, e 10 adicionados na *RSSF*.

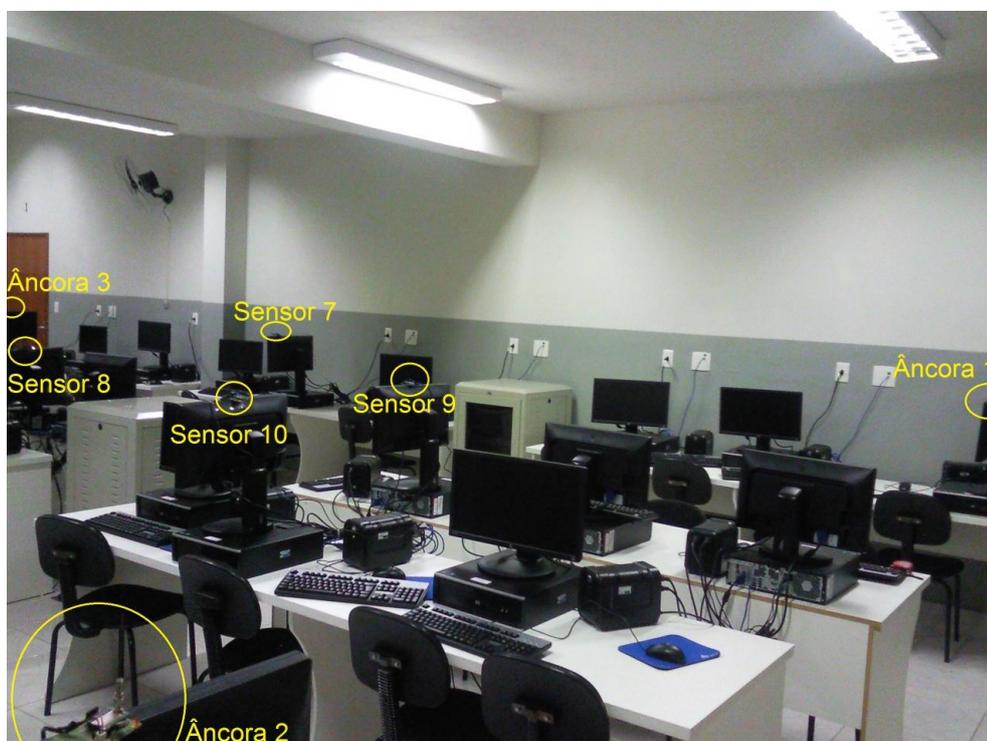


Figura 23: Cenário 2 montado com a maior parte dos sensores inseridos na *RSSF*.

Fonte: (Autor, 2013).

Os resultados do sensor 7 foram mais próximos a realidade com o coeficiente de perda de percurso 5 e 6 e as coordenadas da posição y . Exceto um experimento sobre o eixo das abscissas que obteve como resultado da multilateração o valor bem próximo à realidade como pode ser observado na tabela 7.

Posição (1, 6)	Trilateração		Multilateração	
	Sensor 7	X	Y	X
$\mu = 2$	-62.138159082	-15.506399256	1.2670124741	17.8647436679
$\mu = 2$	-73.451429577	-14.566207836	-3.3858448483	22.3104157053
$\mu = 2$	-59.122479606	-21.308035769	11.326173898	15.770202916
$\mu = 2$	24.0776089161	16.1507587373	24.0776089161	16.1507587373
$\mu = 5$	2.45266565032	5.27988534286	3.83559860596	6.00774479319
$\mu = 5$	2.47676806502	5.26449251092	3.8773745357	6.00165381128
$\mu = 5$	4.46688222357	6.26744862873	3.18324163553	5.66294346809
$\mu = 5$	2.66896250292	5.27163794286	3.93291016998	5.9368735571
$\mu = 6$	2.83329009734	6.15442051995	4.03111589612	5.64376846889
$\mu = 6$	4.44641022635	6.1719560811	3.36772603569	5.66397108433
$\mu = 6$	4.64490246658	5.3493943218	3.42892774261	5.92236670482
$\mu = 6$	2.84540221584	5.3465577152	3.95243046377	5.92920416148
$\mu = 10$	3.12206138039	5.45774547979	3.92136281752	5.8784304467
$\mu = 10$	3.13233302935	5.4577998812	3.92786448164	5.87650064557
$\mu = 10$	4.42575662675	5.46214499744	3.54160465339	5.87876111054
$\mu = 10$	4.42975199295	5.46214499744	3.5321994794	5.88507550121

Tabela 7: Resultados do sensor 7 no cenário 2 com coordenadas (1, 6).

Fonte: (Autor, 2013).

Para o sensor 8 os resultados começaram a aproximar da posição real da localização do nó sensor com o coeficiente perda de percurso igual a 10 para ambas as técnicas da trilateração e da multilateração sendo que a primeira apresentou uma maior consistência dos

dados da posição para o coeficiente de perda de percurso 5 e 6 para a posição x principalmente. A tabela 8 mostra os resultados dos experimentos do sensor 8.

Posição (3, 6)	Trilateração		Multilateração		
	Sensor 8	X	Y	X	Y
$\mu = 2$	52.123992026	-578.66412186	808.918643852	-180.35114721	
$\mu = 2$	48.1795750808	-828.42745011	1148.44283668	-249.34152296	
$\mu = 2$	56.815116052	-697.04987088	977.014408237	-212.73445394	
$\mu = 2$	37.3722175786	-484.41942179	676.142811096	-148.22437257	
$\mu = 5$	3.59740091967	4.09905442223	5.93122083958	5.32738069587	
$\mu = 5$	3.61654614819	3.94453897162	6.17311482948	5.29010143546	
$\mu = 5$	7.48053894649	5.81623748386	4.96547791054	4.6318192053	
$\mu = 5$	7.22837672127	5.80003573297	4.87885729882	4.69357600496	
$\mu = 6$	3.42592392993	4.83374275879	4.86861582367	5.59305428181	
$\mu = 6$	3.43280647937	4.78458502635	4.97439549376	5.59594766551	
$\mu = 6$	3.44876650279	4.82294123841	4.8944898432	5.58384825967	
$\mu = 6$	3.46622909909	4.85389534337	4.88166838735	5.59886338982	
$\mu = 10$	3.29884005667	5.35342638622	4.13918591539	5.79571368028	
$\mu = 10$	3.30196424109	5.35639089268	4.14601594144	5.8006286297	
$\mu = 10$	3.30597316817	5.36286761161	4.13853817902	5.80105972258	
$\mu = 10$	4.52111203149	5.97852752753	3.71796694267	5.60030222407	

Tabela 8: Resultados obtidos no sensor 8 na posição (3, 6) no cenário 2.

Fonte: (Autor, 2013).

Os resultados do sensor 9 foram mais precisos para a posição x para ambas as técnicas usadas com coeficiente de perda de percurso 5 e 6 principalmente. A posição y os resultados foram distorcidos com aproximadamente 2 metros de diferença. A tabela 9 mostra os experimentos coletas com *RSSI* sobre o sensor 9.

Posição (3, 3)	Trilateração		Multilateração	
Sensor 9	X	Y	X	Y
$\mu = 2$	-5.4547321634	-35.215666375	-156.90771150	-114.92776076
$\mu = 2$	-5.4547321634	-35.215666375	-156.90771150	-114.92776076
$\mu = 2$	-5.1977073364	-35.087153961	-146.4943069	-109.45378532
$\mu = 2$	-5.4547321634	-35.215666375	-146.57754271	-109.49082982
$\mu = 5$	2.96203130051	4.92865781021	3.22855105096	5.06893136308
$\mu = 5$	2.96203130051	4.92865781021	3.18523906895	5.04613558308
$\mu = 5$	2.95455879594	4.92492155793	3.21754390548	5.06333477347
$\mu = 5$	3.35619998407	5.12574215199	3.09743275669	5.00388084143
$\mu = 6$	3.06755337951	5.16653712231	3.51989637719	5.40461238425
$\mu = 6$	3.06755337951	5.17646354531	3.51943039539	5.41429355368
$\mu = 6$	3.06755337951	5.16653712231	3.54048014324	5.41544594533
$\mu = 6$	3.7669871721	5.51625401865	3.30781689608	5.30001685374
$\mu = 10$	3.18661396119	5.40078318528	3.83534852158	5.74222242759
$\mu = 10$	3.18661396119	5.40078318528	3.83841979122	5.7438388853
$\mu = 10$	3.18943943842	5.39983864187	3.83965358177	5.74205661206
$\mu = 10$	3.18661396119	5.40078318528	3.83841979122	5.7438388853

Tabela 9: Resultados do sensor 9 no cenário 2 com coordenadas (3, 3).

Fonte: (Autor, 2013).

Os resultados obtidos no sensor 10 na posição (4,3) foram próximos a realidade para alguns dos experimentos com coeficiente de perda de percurso 5 e 6 sobre o eixo x , e principalmente para a técnica da multilateração, como exemplificado na tabela 10.

Posição (4, 3)	Trilateração		Multilateração	
	Sensor 10	X	Y	X
$\mu = 2$	-2.0787965583	-33.659768889	-68.680154476	-68.713115162
$\mu = 2$	-2.0787965583	-37.03540158	-55.319654179	-65.056905596
$\mu = 2$	-2.0787965583	-33.659768889	-68.680154476	-68.713115162
$\mu = 2$	-91.540689568	-79.39426744	-33.61660603	-52.116065310
$\mu = 5$	3.07465254589	4.96556937461	3.5694459538	5.2259869577
$\mu = 5$	3.07465254589	4.92863197072	3.6434218701	5.2279842466
$\mu = 5$	3.06031913877	4.9286319707	3.6378109279	5.2325750176
$\mu = 5$	3.93559217184	5.3591017837	3.3703951654	5.0929334260
$\mu = 6$	3.04959147597	4.41081363558	4.93610785414	5.40371699251
$\mu = 6$	3.1196494636	5.22119232822	3.7817327964	5.56965724023
$\mu = 6$	3.13917497375	5.18992771518	3.74421757672	5.5240894903
$\mu = 6$	4.120589579	5.67174307571	3.46497697531	5.36299527979
$\mu = 10$	3.21172621581	5.40573812159	3.89324838055	5.76443399777
$\mu = 10$	3.20883169672	5.4041645869	3.89156824241	5.76349961096
$\mu = 10$	3.20883169672	5.40416458692	3.88905322088	5.76217591542
$\mu = 10$	3.21172621581	5.40416458692	3.89537629221	5.7639804166

Tabela 10: Resultados do sensor 10 no cenário 2 com coordenadas (4, 3).

Fonte: (Autor, 2013).

Após verificar os resultados da localização dos nós sensores adicionados na *RSSF* na próxima seção a conclusão sobre este trabalho.

5. CONCLUSÃO

Neste trabalho, o método de estimativa de localização *RSSI* e as técnicas utilizadas para calcular a posição dos nós sensores possibilitaram o desenvolvimento deste algoritmo de localização com a finalidade de obter as posições x e y de um nó sensor inserido na *RSSF* aplicada a ambientes internos (*indoor*). Além disso, o algoritmo de localização desenvolvido neste trabalho permitiu obter resultados com uma precisão próxima a realidade para alguns coeficientes de perda de percurso e atingiu resultados próximos a exatidão quando testados em cenários menores como pode ser visto na seção anterior.

Entretanto, existem alguns desafios devem ser considerados neste trabalho, por exemplo, os obstáculos de propagação do sinal o qual causa distorção dos resultados. No laboratório havia alguns objetos presente na sala que alteraram a propagação do sinal e modificaram a maneira como os sensores se comunicam. A potência de transmissão também pode contribuir com melhores resultados e fica uma sugestão para trabalhos futuros aumentar a potência de transmissão e comparar os resultados obtidos para verificar se aumenta ou diminui a precisão da localização dos nós sensores adicionados na *RSSF*.

Através dos resultados é possível concluir que o algoritmo de localização proposto apresenta um desvio de alguns metros da posição real dos sensores localizados na mesma direção que os nós âncoras estabelecidos no sistema de localização. E também em cenários mais amplos os nós sensores inseridos na rede apresentaram resultados inconsistentes algumas vezes para as coordenadas x outras para as coordenadas y . Os resultados apresentados na seção anterior demonstram que o algoritmo de localização precisa de alguns

ajustes para melhorar os aspectos relacionados à propagação do sinal dentro de ambientes com obstáculos.

Estes problemas com a determinação da posição com precisão pode-se levar em consideração aspectos físicos como o tipo da antena do sensor, ruídos, atenuação, ganho e perda da potência de transmissão, perda de caminho, perda por multicaminhos, problemas por interferências (atenuação por obstrução do sinal, condições do tempo, entre outros fatores). Assim, estes fatores podem causar perda de sinal o que ocasiona inconsistência na medição da potência para realizar a estimativa da distância com o método *RSSI*. Além de tudo isto, a maioria dos testes feitos com os algoritmos de localização é baseada em simulações através de programas como o *Silab* devido à dificuldade para comprar os nós sensores. Entretanto nesse trabalho é importante destacar que todos os testes realizados foram sobre a plataforma *Radiuino* como mostra as imagens na seção anterior sendo o mesmo de cunho pratico.

O algoritmo de localização apresentado neste trabalho possibilitará uma variedade de novas aplicações desenvolvidas para *RSSF*, em trabalhos futuros poderá ser melhorado o algoritmo ao aplicar técnicas mais apuradas após compreender melhor os limites que o ambiente impõe a comunicação dos nós sensores. Além disso, podem-se explorar aplicações com intuito de produzir economia de energia no processo de descobertas de rotas durante a troca de informações entre os nós sensores e a base, poderia ser desenvolvido um protocolo de roteamento para realizar os cálculos para determinar uma corrente de sensores percorriam o menor caminho entre os nós até chegar à base com o cálculo da localização empregando o algoritmo de localização, e muitas poderia ser desenvolvidos outros trabalhos com intuito de melhorar a qualidade do sinal recebido.

REFERÊNCIAS BIBLIOGRÁFICAS

AKYILDIZ, I. F.; VURAN, M. C. **Wireless Sensor Networks**. Torquay: Wiley, 2011.

HOFMANN-WELLENHO, B.; LICHTENEGGER, H.; COLLINS, J. **Global positioning System: Theory and Praticce**. 4ª ed. Springer-Verlag, 1997.

KARL, H.; WLLIG, A. **Protocols and architectures for wireless sensor networks**. [s. L.]: John Wiley & Sons, 2005.

LANGENDOEN, K.; REIJERS, N. Distributed localization in wireless sensor networks: a quantitative comparison. **Netherlands: Computer Networks: The International Journal Of Computer And Telecommunications Networking, Special Issue: Wireless Sensor Networks**, p. 20, 2003.

LOUREIRO, A. A. F.; NOGUEIRA, J. M. S.; RUIZ, L. B.; MINI, R. A. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S. Redes de Sensores Sem Fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Belo Horizonte. **Anais...** Belo Horizonte: Sbrc, p.179-226, 2003.

MAIMONI, M. L.; TAVARES, T. C. Estudo da Aplicabilidade das Plataformas de Redes de Sensores Wireless em Ambientes Rurais. In: JORNADA CIENTÍFICA E TECNOLÓGICA, 4., 2012, Muzambinho. **Anais...** . Muzambinho: IFSULDEMINAS- Câmpus Muzambinho, p. 1-6, 2012. (CD-ROM)

MARGI, C. B., JUNIOR, M. S. BARRETO, P. S. M. CARVALHO, T. C. M. B. Segurança em Redes de Sensores Sem Fio. In: SIMPÓSIO BRASILEIRO DE SEGURANÇA, 2009, São Paulo. **Anais...** São Paulo: Minicurso Segurança, 2009.

NICULESCU, D.; NATH, B. Ad hoc positioning system (aps). In: IEEE GLOBAL COMMUNICATIONS CONFERENCE, 1., 2003, San Antonio. **Anais...** San Antonio: GlobeCom '01, p. 1-10, 2003.

OLIVEIRA, H. A. B. F. **Um algoritmo Recursivo de Localização para Redes de Sensores Sem Fio.** 2005. 72 f. Dissertação (Mestrado) - Curso de Pós-graduação em Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2005.

OLIVEIRA, H. A. B. F.; NAKAMURA, E. F.; LOUREIRO A. A.F.; BOUKERCHE, A. **Error Analysis of Localization Systems for Sensor Networks.** Bremem, 2005.

PEREIRA, S. S. L. Computação Científica com Numpy e Scipy. **Departamento de Estatística e Computação da Universidade Estadual do Ceará.** Fortaleza, p.1-7, 2009.

PRIYANTHA, N.B.; CHAKRABORT, A.; BALAKRISHNAN, H. The cricket location-support system. In: MOBILE COMPUTING AND NETWORKING, 2000, Boston. **Anais...** Boston: [s.i.], p. 32-43.

PATWARI, N.; ASH, J. N. ; KYPEROUNTAS, S.; HERO III, A. O.; MOSES, R. L.; CORREAL, N. S. Locating the nodes - cooperative localization in wireless sensor networks. **IEEE Signal Processing Magazine**, [s. L.], p.54-69, 2005.

RADIUINO. Acesso em 19 jun. 2013. Online. Disponível em <www.radiuino.cc>.

SAVVIDES, A.; HAN, C.; STRIVASTAVA, M. B. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In: ACM MOBICOM CONF, 7., 2001, Los Angeles. **Anais...** Los Angeles: Acm Mobicom Conf, p.1-14, 2001.

WHITEHOUSE, C. D. **The design of calamari: an ad-hoc localization system for sensor networks.** 2002. 73 f. Tese (Mestrado) - University Of California At Berkeley, Berkeley, 2002.

APÊNDICE A

```
import serial
import math
import time
import struct
from time import localtime, strftime
from numpy import *
import scipy as Sci
import scipy.linalg

def RSSI(n): #return RSSI from cc1101 in dBm
    if n > 128:
        return(((n-256)/2.0)-74)
    else:
        return((n/2.0)-74)

# Configura a serial
# para COM# o número que se coloca é n-1 no primeiro parâmetro. Ex COM9 valor 8
n_serial = raw_input("Digite o número da serial = ") #seta a serial
n_serial1 = int(n_serial)-1

ser = serial.Serial(n_serial1, 9600, timeout=0.5,parity=serial.PARITY_NONE) # ABRE
A SERIAL E SETA SUAS CONFIGURAÇÕES

array_posicoes = zeros( (10,4) )
array_posicoes[0,0]=0
array_posicoes[0,1]=1
array_posicoes[0,2]=1
array_posicoes[0,3]=0
h=1
```

```

#ID_sensor1 = raw_input('ID_Sensor:')

filename1 = strftime("localização_%Y_%m_%d_%H-%M-%S.txt")
print "Arquivo de log: %s" % filename1
S1 = open(filename1, 'w')

Ptx = 5 # Potência de Transmissão
gt = 1 #ganho das antenas
gr = 1 #
d0 = 1 #distancia de referencia
f = 915e6 #frequencia
c = 3e8 #velocidade da luz

k=0 #algum contador....rs
NR = 1 #numero de repetições
ID_sensor1=1000
cont='s'
rota=0
while cont == 's':
    try:
        ID_sensor1 = raw_input('ID_Sensor:')
        u = float(raw_input('Beta:'))
        anc1 = raw_input('Ancora 1:')
        anc2 = raw_input('Ancora 2:')
        anc3 = raw_input('Ancora 3:')
        anc4 = raw_input('Ancora 4:')
        arrayauxiliar = zeros( (10,7) ) #vetor auxiliar
        Pacote1 = {}

        for i in range(0,52): # faz um array com 52 bytes
            Pacote1[i] = 0

        Pacote1[11] = int(253) # pacote de localização

```

```

Pacote1[8] = int(ID_sensor1) #broadcast (end destino)
Pacote1[10] = int(0) #(end fonte)
Pacote1[12] = int(anc1)
Pacote1[13] = int(anc2)
Pacote1[14] = int(anc3)
Pacote1[15] = int(anc4)
ser.flushInput()
i=0
n=0

#envia pedido para o sensor

for i in range(0,52):
    TXbyte = chr(Pacote1[i]) # Deve converter para caracter em ASCII
para escrever na serial
    ser.write(TXbyte)
    # Tempo de espera para que receba a resposta do sensor (acho que pode
apagar)
    line = ser.read(52) # faz a leitura de 52 bytes do buffer que recebe da
serial pela COM
    if len(line) == 52:
        z = 16

        while z<=35:
            n = n+1
            # x = float(ord(line[z])/10)
            z = z + 1
            y = float(ord(line[z])/10.0)
            z = z + 1
            end = ord(line[z])
            z = z + 1
            rssiu = RSSI(ord(line[z])) # RSSI_UpLink
            z = z + 1

```

```

    rssid = RSSI(ord(line[z])) # RSSI_DownLink
    z = z + 1
        #u = 1.9670 #float( ord(line[14]))+ float((ord(line[15]))/100)
        #u = 4.5
        #u = 5
        #u = 5.5
    #u = 9 #Beta usado

    rssi_aux=10*log10((((10**(rssid/10)))+(10**(rssi_u/10)))/2))
#média das RSSI's, passa pra dBm, soma-se e divide por 2
    Ld0 = 10*log10((4*pi*d0/(c/f))**2) #
    d = d0*(10**((Ptx+gt+gr-Ld0-rssi_aux)/(10*u)))

    arrayauxiliar[n-1,0] = x
    arrayauxiliar[n-1,1] = y
    arrayauxiliar[n-1,2] = d
    arrayauxiliar[n-1,3] = end
    arrayauxiliar[n-1,4] = rssi_u
    arrayauxiliar[n-1,5] = rssid
    arrayauxiliar[n-1,6] = rssi_aux

else:
    print 'sensor ',ID_sensor1,' falhou.'
    #time.sleep(1)

for t in arrayauxiliar :
    print t

#AN = array( [[0,0,2.82],[0,5,3.60],[5,0,3.60],[5,5,4.24]])
print n
if n >= 4:
    j = n #numero de sensores que enviaram, que indica quant. de linhas que
devo
    k=k+1

```

```

ANI = zeros( (3,3) ) #trilateração, cria vetor de 3x3 com zeros
for i in range(0,3):
    ANI[i,0] = arrayauxiliar[i,0]
    ANI[i,1] = arrayauxiliar[i,1]
    ANI[i,2] = arrayauxiliar[i,2]

A1=zeros((2,2))
for i in range(0,2): #coisa da trilateração
    A1[i,0] = 2*(ANI[i,0]-ANI[2,0])
    A1[i,1] = 2*(ANI[i,1]-ANI[2,1])

B1=zeros((2,1)) #coisa da trilateração
for i in range(0,2):
    B1[i] = ANI[i,0]**2 - ANI[2,0]**2 + ANI[i,1]**2 - ANI[2,1]**2
+ ANI[2,2]**2 - ANI[i,2]**2

Atransp1 = transpose(A1) #cria transposta de A1
invaux1 = linalg.inv(dot(Atransp1,A1)) # cria inversa da multiplicação
da (transposta de A1) * A1
resp1 = dot(dot(invaux1 , Atransp1), B1) # coordenadas x e y resultante

AN2 = zeros( (4,3) ) #multilateração
for i in range(0,4): #pega agora 4 linhas do auxiliar, pois se trata de
multilateração
    AN2[i,0] = arrayauxiliar[i,0]
    AN2[i,1] = arrayauxiliar[i,1]
    AN2[i,2] = arrayauxiliar[i,2]

A2=zeros((3,2)) #coisa da multilateração
for i in range(0,3):
    A2[i,0] = 2*(AN2[i,0]-AN2[3,0])
    A2[i,1] = 2*(AN2[i,1]-AN2[3,1])

```

```

B2=zeros((3,1)) #coisa da multilateração
for i in range(0,3):
    B2[i] = AN2[i,0]**2 - AN2[3,0]**2 + AN2[i,1]**2 - AN2[3,1]**2
+ AN2[3,2]**2 - AN2[i,2]**2

Atransp2 = transpose(A2)
invaux2 = linalg.inv(dot(Atransp2,A2))
resp2 = dot(dot(invaux2 , Atransp2), B2) #resultado das coordenadas (x,
y)

print AN2
print 'Posição do sensor:N=3: x = ',resp1[0,0],', y = ',resp1[1,0],
print 'Posição do sensor ',ID_sensor1,'N=4', 'x = ',resp2[0,0],', y =
',resp2[1,0]

print >>S1,time.asctime(),'Matriz contendo posx, posy, dist, id, rssiUp,
rssi_d, rssi_media'

for t in arrayauxiliar :
    print >>S1,time.asctime(),t

    print >>S1,time.asctime(),'Posição do sensor', ID_sensor1, ':N=3: x =
',resp1[0,0],', y = ',resp1[1,0], 'N=4: x = ',resp2[0,0],', y = ',resp2[1,0]
    array_posicoes[h,0]=ID_sensor1
    array_posicoes[h,1]=resp2[0,0]
    array_posicoes[h,2]=resp2[1,0]
    array_posicoes[h,3]=0
    h=h+1
    cont = raw_input('Mais sensores?s/n')
    if cont == 'n':
        array_posicoes[h,0]=255

else:
    print 'n<4'

except KeyboardInterrupt:

```

```
ser.close()  
SI.close()  
print 'fim do programa'  
break
```

```
ser.flushInput()  
ser.close()
```