

Um Sistema para Localização em Tempo Real de Terminais Wi-Fi em Ambientes *Indoor*

Bitter A. G. Cruz², Lisandro Lovisolo¹, Rafael S. Campos¹, Alexandre Sztajnberg^{1,2}

Universidade do Estado do Rio de Janeiro – UERJ

¹Faculdade de Engenharia – FEN / Programa de Pós-Graduação em Eletrônica – PEL

²Instituto de Matemática e Estatística – IME / Bacharelado em Ciência da Computação

bitter.cruz@gmail.com, lisandro@uerj.br, rafael_saraiva@ig.com.br,
alexsz@uerj.br

Resumo. *O aumento da utilização de redes sem fio, abre perspectivas no campo de pesquisas e aplicações baseadas nessas redes. Uma classe de aplicações desse tipo é composta por sistemas de localização que se propõem a identificar a posição geográfica do usuário, em tempo real, através de dispositivos sem fio. Neste artigo é apresentada a proposta de um sistema de localização em tempo real baseado em redes Wi-Fi. O sistema é baseado no cadastro de cômodos, que servirão como referências para a coleta prévia de sinais RF de pontos de acesso nesses cômodos e a posterior localização do terminal. O algoritmo de localização implementado emprega como padrão de referência as médias dos sinais coletados. Essas médias são comparadas aos sinais de RF recebidos pelo equipamento de forma a estimar a localização do dispositivo sem fio, em tempo real. O sistema foi implementado sobre o sistema operacional Linux, utilizando a linguagem C e a biblioteca gráfica GTK.*

Abstract. *The increase in the use of wireless networks opens perspectives, both for research and applications based on these networks. One class of such applications consists of positioning systems which aim at estimating the user's geographic position in real time via wireless devices. In this paper, we propose a system for real-time location based on Wi-Fi networks. The system is based on the registration of rooms, which are used as references in the process of collecting RF signals levels received from neighboring access points by a Wi-Fi terminal in each room. The gathered data is then used for estimating the location of the terminal among the known references. The location algorithm implemented uses as a benchmark the averages of signal levels collected from the different access points that are available. These averages are compared to the RF signal levels received by the equipment in order to estimate its location in real time. The system was implemented on the Linux operating system, using the C++ language and the GTK graphics library.*

1. Introdução

Com o advento das redes sem fio Wi-Fi (“Wireless Fidelity”) [Sta05], observa-se uma grande quantidade de equipamentos interagindo entre si sem a necessidade de cabos. Em redes Wi-Fi, normalmente há a figura do ponto de acesso (ou Access Point - AP) como centralizador das conexões, possibilitando diversas conexões Wi-Fi e fazendo muitas vezes o papel de gateway entre a infraestrutura cabeada e os dispositivos sem fio. O alcance de um ponto de acesso depende, entre outras características, do tipo de antena, da potência de transmissão, do ambiente em que opera, dos obstáculos entre o AP e o terminal, etc. [Cam10].

Esse cenário permite implantar sistemas capazes de localizar dispositivos móveis baseado nos sinais RF (característicos da comunicação sem fio), aproveitando principalmente a grande profusão de APs [Vil07]. Cada AP cobre uma área distinta e o sinal recebido no terminal depende da posição do AP, do próprio terminal e dos obstáculos entre eles [Yac93]. De fato, existem trabalhos relacionados a sistemas de localização [Cam10, Gez08, Pah02, Bah00, Spi09] baseados nessas redes. Essas propostas diferem entre si na forma como atuam e em suas arquiteturas, buscando aprimorar a precisão e a qualidade da estimativa de posição fornecida.

1.1 Sistemas de Localização em Tempo Real

Um Sistema de Localização em Tempo Real (ou *Real Time Locating System* - RTLS) [SC31] é um sistema capaz de fornecer a localização de objetos em tempo real através de hardware e/ou software específicos, utilizando sinais de redes sem fio. Em [SC31] considera-se que um RTLS é a combinação de um hardware sem fio e um software que são utilizados para continuamente e em tempo real determinar e prover a localização de dispositivos equipados com a interface necessária para interagir com o sistema.

Existem várias maneiras de se implementar um RTLS. Talvez a mais conhecida delas seja o *Global Positioning System* (GPS) que utiliza uma combinação de sinais de satélites para estimar a localização de dispositivos. O GPS tem sido incorporado a aparelhos celulares, e costuma prover uma grande acurácia, sendo inclusive muito utilizado em veículos automotivos, permitindo ao motorista trafegar em territórios desconhecidos sem a necessidade de mapas. Porém, sua eficácia é quase nula em locais fechados ou cobertos, isto é em ambientes ditos *indoor*, além de ser uma tecnologia que demanda hardware específico. Há propostas e implementações de sistemas de localização utilizando tecnologias de RF diferentes [ZB11]. Neste trabalho empregam-se sinais de WLAN (*Wireless Local Area Network*). A grande vantagem da utilização desses sinais é a de poder aproveitar-se de infra-estruturas já existentes, sem a necessidade de fazer adaptações ou alterações significativas no ambiente.

Dentre as implementações de sistemas de localização têm-se aquelas que baseiam a localização nos sinais recebidos dos pontos de acesso próximos ao equipamento, que é o caso do sistema proposto, e tem-se a utilização de equipamentos para medir sinais provenientes do equipamento a ser localizado. Assim, a estimativa de posição do terminal móvel pode ser calculada a partir de medidas realizadas pelo próprio terminal ou pela rede de acesso via rádio. No primeiro caso, o terminal mede os valores de nível de sinal recebido (RSS – *received signal strength*) de pontos de acesso

próximos. No segundo caso, os APs medem os valores de RSS dos sinais transmitidos pelo terminal a ser localizado.

No presente trabalho, o software que retorna a posição é embarcado no equipamento a ser localizado, e a informação usada para estimar a localização é medida pelo equipamento. O sistema desenvolvido, assim como grande parte dos sistemas localizadores [Eka10, Cam10], funciona em duas fases: i) a fase de treino (ou calibração) e ii) a fase de operação, isto é, de determinação da localização. A primeira fase é composta pela coleta dos dados de sinais de pontos de acesso, feitos em várias posições de referência a serem localizadas, de forma a fornecer um banco de padrões de referências [Cam10], que será empregado para servir de base ao algoritmo de localização utilizado para retornar a posição do terminal. Os padrões medidos nas fases de treinamento e teste são denominados assinaturas de RF ou impressões digitais de RF (RF *Fingerprint*). Pois, espera-se que em cada posição/local o conjunto de níveis de sinais recebidos pelo terminal de diversos APs seja diferente, assim como as impressões digitais de diferentes indivíduos.

Existem várias implementações de RTLS baseadas em redes Wi-Fi. Podem-se citar o sistema desenvolvido pela Cisco, o *Cisco Wireless Location Services* [Cis08] que além de operar com sinais Wi-Fi também funciona utilizando RFID (*Radio Frequency Identification*). Ele requer que a infraestrutura sem fio seja composta por equipamentos Cisco e uma configuração prévia dos pontos de acesso. Outro exemplo é o sistema produzido pela Ekahau [Eka10], cujo processamento é realizado em um dispositivo central permitindo a localização de dispositivos móveis com pouco poder computacional. Este sistema utiliza um método de correlação, ou mapeamento, de assinaturas de RF e exige uma fase inicial de calibração muito rigorosa, para que os resultados estejam dentro da precisão esperada.

1.2 Resultados

Este trabalho apresenta um sistema de localização baseado na correlação de assinaturas de RF em redes Wi-Fi. Nele a estimativa da localização de um nó sem fio é calculada com base nos sinais RF emitidos por pontos de acesso fisicamente próximos ao dispositivo a ser localizado. O sistema implementado considera um dispositivo móvel, com interface de rede WiFi, capaz de executar o programa de localização. Este programa determina onde o dispositivo está localizado. O sistema pressupõe que existe ao menos um AP próximo ao dispositivo. Por último, o sistema considera que a localização será realizada em uma área limitada, por exemplo, uma residência, e que pode ser particionada ou dividida em regiões, por exemplo, cômodos. Durante a fase de treinamento, esses cômodos precisam ser previamente cadastrados no aplicativo de localização e assim, nessa fase, realiza-se a calibração adequada (construção dos padrões de RF associados as cômodos) no programa de localização. Desta forma, o sistema consegue identificar em tempo real o cômodo no qual o dispositivo se encontra.

1.3 Organização do Texto

A Seção 2 apresenta as características básicas de redes Wi-Fi e os aspectos relevantes das mesmas, empregados para obter as estimativas de localização dos terminais. A Seção 3 apresenta a estratégia de localização empregada neste trabalho e o algoritmo usado para obter a localização do terminal Wi-Fi. Na Seção 4 são discutidos os aspectos

de implementação do sistema desenvolvido. A Seção 5 traz exemplos de uso do sistema e alguns resultados obtidos. Conclusões e trabalhos futuros são discutidos na Seção 6.

2. Redes Wi-Fi

Wi-Fi se refere a *Wireless Fidelity*. Esse termo é normalmente empregado para referir-se a redes WLAN (*Wireless Local Area Network*) que operem empregando alguma das especificações IEEE 802.11 (802.11a, 802.11b, 802.11g, 802.11n). Neste trabalho os sinais provenientes de pontos acesso Wi-Fi são empregados num sistema para prover a localização de terminais sem fio.

2.1 Funcionamento Básico

Segundo as especificações, redes 802.11 podem ser usadas para duas finalidades. A primeira é o provimento de um canal de comunicação entre quaisquer dois terminais. Nesse caso, tem-se uma rede *ad hoc* para troca de informações entre dois equipamentos, similar ao emprego objetivado para o Bluetooth. Entretanto, o emprego mais comum dos padrões IEEE 802.11 é na construção de redes locais sem fio. Neste caso, um ponto de acesso 802.11 é empregado para prover a comunicação entre diferentes terminais em uma pequena área. Quando esse ponto de acesso está ligado à Internet ele se torna capaz de conectar os terminais a ele ligados à Internet.

Há diversas versões do padrão IEEE 802.11. Cada uma delas possui especificidades e emprega tecnologias um pouco distintas. Entretanto, alguns princípios básicos são sempre comuns. As redes IEEE 802.11 são especificadas para operar nas bandas de ISM (*Industrial, Scientific and Medical*) nas quais não é necessário obter uma licença de operação. Com isso qualquer um pode montar uma rede IEEE 802.11 sem enfrentar problemas regulatórios. Além disso, os padrões 802.11 provêm um canal a ser compartilhado no tempo pelos diferentes terminais ligados à rede implantada.

2.2 Versões

A versão original do padrão IEEE 802.11 foi lançada em 1997. O padrão IEEE 802.11 especificava três tecnologias para camada física: infravermelho; espalhamento espectral por salto em frequência (FHSS – *Frequency Hopping Spread Spectrum*) e por sequência direta (DSSS – *Direct Sequence Spread Spectrum*). O FHSS e DSSS operavam a taxas de 1 e 2 Mbps na banda não licenciada de 2,4 GHz, enquanto o infravermelho suportava apenas a taxa básica de 1 Mbps [IEEE802.11a].

O padrão IEEE 802.11a utiliza a mesma estrutura de quadros (blocos de mensagem que trafegam no canal) do padrão IEEE 802.11, porém opera na banda de 5 GHz e utiliza multiplexação por divisão em frequências ortogonais (OFDM – *Orthogonal Frequency Division Multiplexing*), o que possibilita taxas de até 54 Mbps [IEEE802.11a]. Por operar em uma faixa frequências mais alta, os sinais do IEEE 802.11a sofrem maior atenuação que os padrões IEEE 802.11b/g/n na propagação, o que resulta em menor cobertura dos APs.

O padrão IEEE 802.11b suporta uma taxa máxima de 11 Mbps e opera na banda de 2.4 GHz [IEEE802.11b]. O padrão IEEE 802.11g [IEEE802.11g] opera na banda de 2.4 GHz, assim como o IEEE 802.11b, e utiliza OFDM, assim como o IEEE 802.11a. Assim, o IEEE 802.11g suporta a mesma taxa máxima de 54 Mbps do IEEE 802.11a,

sem o inconveniente da menor cobertura do IEEE 802.11a. Além disso, é compatível com dispositivos IEEE 802.11b. Na fase de treinamento do sistema de localização proposto neste trabalho, foram realizadas medidas em redes Wi-Fi IEEE 802.11b/g.

O padrão IEEE 802.11n eleva a taxa máxima através do uso de antenas MIMO (*multiple-input multiple-output*). O IEEE 802.11n opera nas bandas de 2.4 GHz e 5 GHz e foi publicado em 2009 [IEEE802.11n].

2.3 Canais de Anúncio - *Broadcast*

O canal de *broadcast* é um canal lógico através do qual cada AP anuncia sua presença, informando seu *timestamp* e SSID (*Service Set Identifier*), dentre outras informações. Para tal, cada AP transmite periodicamente quadros chamados de *beacon* (*beacon frames*) contendo tais informações. Os terminais móveis WiFi monitoram estes quadros no canal de *broadcast* para selecionar com qual AP irão associar-se [Sta05].

Os parâmetros selecionados para compor a assinatura de RF – ou pelo menos suas médias temporais – devem ter uma baixa variabilidade no tempo em uma dada posição [Cam10]. Do contrário, a associação entre cada assinatura de referência e uma posição seria inviável. Como as transmissões nos canais de *broadcast* são feitas com potência constante pelos APs, o nível de sinal recebido (RSS – *Received Signal Strength*) dos canais de *broadcast* é razoavelmente constante no tempo, tornando-se então uma opção para a composição das assinaturas de RF.

2.4 Características do Canal

Os sinais no canal rádio sofrem efeitos de desvanecimentos lentos e rápidos. Essas flutuações em torno do nível médio de RSS do sinal de cada AP, sobretudo os desvanecimentos lentos, ajudam a estabelecer uma relação unívoca entre as assinaturas de RF e posições/localizações. Isso permite aumentar a precisão do sistema de localização proposto.

Os desvanecimentos rápidos devem-se à recepção em múltiplos percursos do sinal. Esses múltiplos percursos adicionam-se fasorialmente, provocando flutuações rápidas no RSS. Tipicamente, estas flutuações em torno do RSS médio são aproximadas segundo uma variável aleatória com distribuição Rayleigh [Yac93].

Os desvanecimentos lentos devem-se, sobretudo ao sombreamento, ou seja, à interposição de obstáculos entre o transmissor e o receptor. Tipicamente, as flutuações em torno do RSS médio produzidas pelo sombreamento são aproximadas segundo uma variável aleatória com distribuição lognormal [Yac93].

No ambiente de propagação *indoor*, o sombreamento é produzido, sobretudo pelas paredes, introduzindo grandes atenuações nos sinais no canal de rádio. Esta característica, embora limite a cobertura dos APs, pode ser utilizada para melhor discriminação dos cômodos. A severa atenuação imposta pelas paredes ajuda a diferenciar as assinaturas de RF de cada cômodo, melhorando a precisão do sistema de localização proposto neste trabalho. Contudo, esta característica só será vantajosa, do ponto de vista da localização *indoor*, se houver uma quantidade mínima de APs detectados em cada cômodo. Logo, a precisão da estimativa de localização depende da densidade de APs [Cam10, ZB11].

2.5 Wireless Tools

Wireless Tools é um conjunto de ferramentas que fazem parte de um projeto patrocinado pela Hewlett Packard desde 1996 e é desenvolvido com a contribuição de usuários Linux ao redor do mundo [Tou08]. Essas ferramentas objetivam o uso das *Wireless Extensions*, APIs (*Application Programming Interface*) genéricas que fazem a interface com o *driver* de rede sem fio, permitindo a coleta de dados de configurações e estatísticas de LANs sem fio. As principais ferramentas e funcionalidades do *Wireless Tools* são:

- `iwconfig` - Permite a manipulação dos parâmetros básicos das configurações wireless;
- `iwlist` - Realiza a varredura e lista as frequências, os sinais, as chaves criptografadas, entre outros.

A ferramenta `iwlist` é fundamental no desenvolvimento do sistema apresentado. O `iwlist` exibe uma série de informações como os nomes das redes sem fio percebidas pelo terminal, os canais e as frequências das redes, a taxa, o tipo de encriptação empregado, a potência de transmissão das mesmas, o tipo de autenticação, entre outros parâmetros [Tou08]. No modo *scanning* o `iwlist` exibe as informações necessárias para o funcionamento do algoritmo do programa apresentado. Ele fornece uma lista com os APs visíveis, seus ESSIDs (*Extended Service Set ID*), a qualidade do sinal, a intensidade do sinal, entre outros. Um trecho de tal lista é exemplificado a seguir:

```
wlan0
Scan completed :
Cell 01 - Address: 00:00:00:00:00:00
ESSID:"default"
Mode:Master
Channel:6
Frequency:2.437 GHz (Channel 6)
Quality=47/100 Signal level=-75 dBm
Encryption key:on
IE: WPA Version 1
    Group Cipher : TKIP
    Pairwise Ciphers (1) : TKIP
    Authentication Suites (1) : PSK
Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s;
           11 Mb/s; 22 Mb/s; 6 Mb/s;
           9 Mb/s; 12 Mb/s; 18 Mb/s;
           24 Mb/s; 36 Mb/s; 48 Mb/s;
           54 Mb/s

Extra:tsf=000001787ef9b923
```

Atualmente, a maioria das distribuições Linux tem integrado em suas rotinas de inicialização de rede o *Wireless Extensions* e também inclui as *Wireless Tools* como parte dos seus pacotes padrão de instalação. O uso da ferramenta é restrito aos superusuários (`root`).

3. Estratégia de Localização

Para localizar o terminal, ou melhor, o cômodo ou região em que o terminal se encontra, primeiramente, define-se para cada cômodo ou região de interesse um padrão, referência ou identidade. Essa identidade é obtida em função dos APs (redes sem fio) que estão disponíveis (ou são percebidas) no cômodo e dos respectivos níveis de potência. Após a definição dessa identidade, na fase de treinamento ou calibração, pode-se realizar a localização do terminal em função desses padrões de referência. Para isso, as redes sem fio captadas pelo terminal são comparadas a esses padrões de forma a estimar a sua localização [Cam10, ZB11].

3.1 Identidades dos Cômodos

Assume-se que os cômodos ou regiões onde se deseja estimar a posição do terminal são finitos e podem ser indexados. Assim, utiliza-se $i \in \{0, \dots, I-1\}$ para indicar um cômodo. Em cada cômodo faz-se uma varredura dos APs captados durante um intervalo de tempo T . Nessa varredura captam-se os canais de *broadcast* dos APs e suas respectivas potências em diferentes momentos $t \in [0, \dots, T]$. Suponha que sejam realizadas N coletas dessas informações em cada cômodo, isto é $t \in \{0, \dots, N-1\}$. Assim, em cada cômodo i , para cada AP k , obtém-se um vetor

$$x_{i,k} = [x_{i,k,0} \cdots x_{i,k,N-1}],$$

que registra essas informações. Para cada AP k no cômodo i pode-se totalizar essas medidas de forma a obter

$$xt_{i,k} = \sum_{n=0}^{N-1} x_{i,k,n} \quad (1)$$

Para gerar a identidade do cômodo, identifica-se em cada um deles quais os L APs que resultam nos maiores $xt_{i,k}$. Observa-se que os L APs não precisam ser necessariamente os mesmos nem serem de um mesmo fabricante. Nos resultados apresentados neste trabalho utiliza-se $L=4$ pois assim obteve-se uma acurácia razoável na identificação do cômodo, entretanto, valores maiores poderiam ser empregados. A utilização de um valor de L menor pode provocar ambigüidades na posição estimada, já que as medidas são realizadas no espaço tridimensional. Assim, para cada cômodo tem-se uma lista K_i que indica os 4 APs percebidos com maiores potências. Note-se que os APs $k \in K_i$ podem ser diferentes para os diferentes cômodos i . Para cada cômodo i calcula-se a média desses 4 APs nas N coletas realizadas, definindo-se

$$x_{i,k} = \frac{xt_{i,k}}{N}, \forall k \in K_i \quad (2)$$

Neste trabalho, utiliza-se $N=50$, pois observou-se que esse valor leva a uma estimação confiável da potência média recebida de um AP por um terminal WiFi. (cabe salientar que de fato a cada dois segundos o sistema coleta uma medida correspondente a médias temporais da potência recebida de um terminal por um AP). Assim, as médias obtidas usando $N=50$ provêm uma estimativa da potência média recebida ao longo de 100 segundos. Realizando testes estatísticos, subdividindo as séries temporais de medidas de potência recebidas durante dois minutos na mesma taxa de amostragem, percebeu-se que com 100 segundos (50 amostras) obtinham-se médias muito similares àquelas obtidas durante um total de quatro minutos e com uma variância insignificante das médias assim computadas.

Assim, a identidade de um cômodo é definida pelo par K_i (APs considerados) e

$$\overline{X}_i = \left[\overline{x_{i,k_{i1}}}, \overline{x_{i,k_{i2}}}, \overline{x_{i,k_{i3}}}, \overline{x_{i,k_{i4}}} \right],$$

na qual, $k_{i,n}$ se refere ao índice do n -ésimo AP de maior potência no cômodo i , as potências médias dos sinais emitidos por cada um dos APs no cômodo.

3.2 Estimativa de Posição

A partir dessa definição, retorna-se como cômodo candidato à posição/localização do terminal aquele cuja assinatura de RF for mais similar às informações de RF coletadas pelo terminal. Na presente aplicação utiliza-se a norma l_1 , também conhecida como SAD (*Sum of Absolute Differences*) como medida de similaridade, isto é:

$$\tilde{i} = \arg \min_i \sum_{k \in K_i} |x_{i,k} - y_k| \quad (3)$$

na qual, \tilde{i} é o índice para o cômodo estimado, o conjunto K_i indica os APs considerados na comparação, i representa um cômodo qualquer e $\overline{x_{i,k}}$ e y_k indicam os valores pré-armazenados como referências dos cômodos e o valor medido respectivamente. Assim, retorna-se como estimativa de cômodo aquele que possui uma assinatura que proporciona um erro, relativamente à assinatura de RF coletado, de menor norma l_1 ou soma dos valores absolutos.

4. Implementação do Sistema

A estratégia de localização proposta, discutida na Seção 3 foi utilizada na implementação de um protótipo. Neste protótipo também foi incluída uma interface gráfica para facilitar a interação com o usuário.

O protótipo foi desenvolvido utilizando-se a linguagem C, fazendo-se uso da biblioteca *Wireless Tools* para acesso às informações das redes sem fio, conforme discutido na Subseção 2.5, e a biblioteca GTK para o desenvolvimento da interface gráfica.

A Figura 1 apresenta a estrutura de módulos utilizada no protótipo desenvolvido. Os módulos foram agrupados de acordo com a função que executam e estruturados em camadas, onde um módulo de uma camada superior faz uma chamada ou utiliza os dados fornecidos por módulos da camada inferior. Os módulos à esquerda são responsáveis pela coleta e obtenção de dados, os mais a direita pelo processamento e transformação dos dados, e por último, o módulo **Localizar Cômodo**, que recebe os dados processados, finaliza a execução e exhibe o resultado para o usuário.

Nas subsecções seguintes discutimos como o algoritmo de localização foi implementado através de rotinas encapsuladas nos módulos do protótipo.

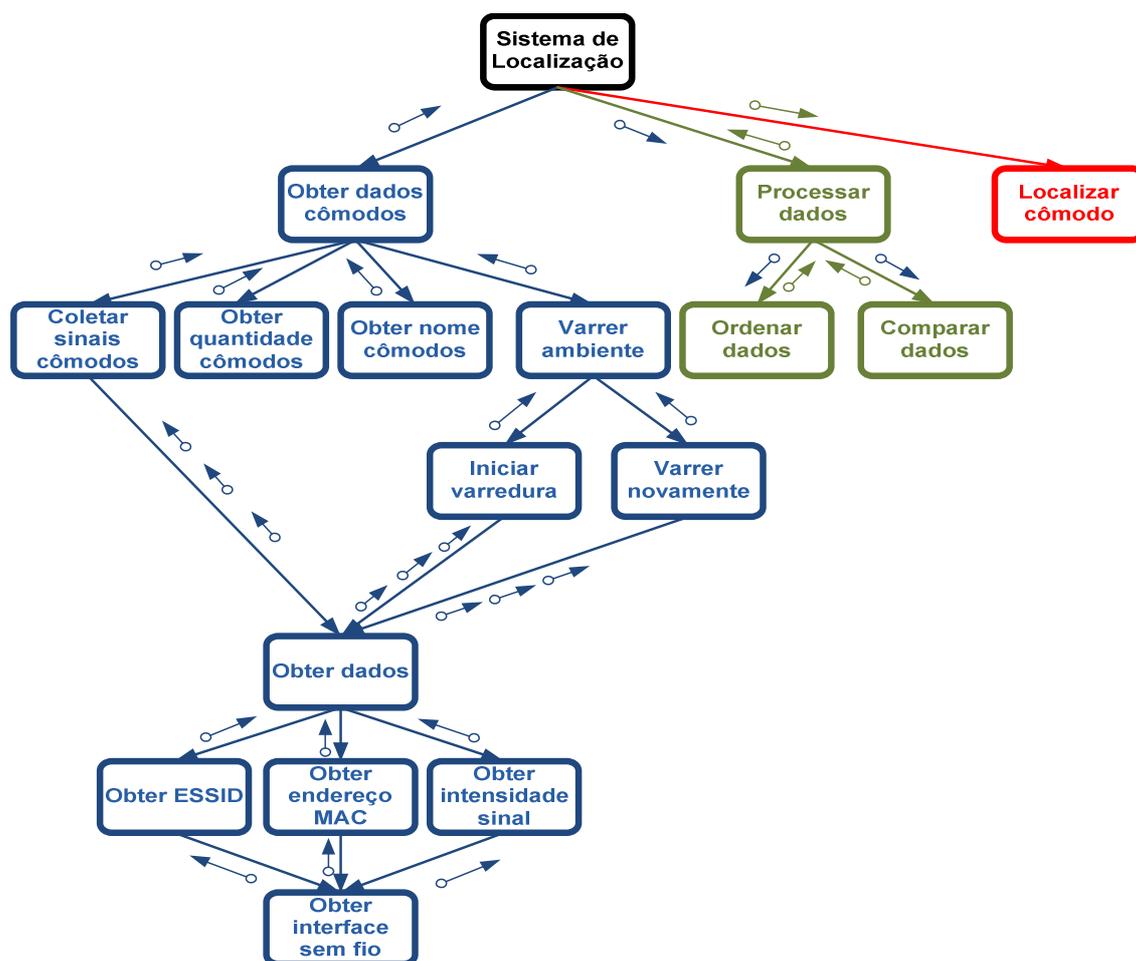


Figura 1. Diagrama de módulos do protótipo

4.1 Algoritmo de localização

Inicialmente, como já foi discutido, temos a obtenção da quantidade de cômodos que servirão de parâmetro para a localização, e temos também a designação dos respectivos nomes. A designação dos nomes é arbitrária.

A seguir, é necessário obter os sinais dos pontos de acesso que conseguem ser detectados no ambiente. O protótipo requer que sejam rastreados pelo menos quatro pontos de acesso para que o programa continue com o seu funcionamento. O usuário

tem a liberdade de realizar várias operações de varredura até obter pontos de acesso com intensidade de sinal forte o suficiente para se prosseguir para a próxima etapa. Uma vez concluído o procedimento de varredura, o sistema irá acatar os quatro pontos de acesso selecionados como sendo os *pontos de acesso base*.

Após a definição dos pontos de acesso base, inicia-se a fase da coleta de dados em cada cômodo para treinar o sistema, isto é, criar os padrões correspondentes aos cômodos, conforme discutido na Subseção 3.1.

São coletadas, uma a cada dois segundos, cinquenta medidas de intensidade do sinal sobre os pontos de acesso base naquele cômodo de acordo com a ordem que foi estabelecida para os pontos de acesso base. Como exemplo, temos conjunto de dados na seguinte forma, cada linha representando uma medida em cada um dos quatro pontos de acesso base:

```
...  
30677787  
31677888  
30667889  
...
```

Com os dados coletados nas 50 medidas, a média aritmética de cada ponto de acesso é calculada para ser utilizada no algoritmo. Uma vez concluída a coleta de dados em cada cômodo tem-se o registro das respectivas *identidades* com o que se considera que o sistema está treinado.

Para realizar um procedimento de localização, posiciona-se o dispositivo que está executando o sistema no cômodo-alvo. Em seguida, são coletados mais uma vez os valores de intensidade de sinal dos pontos de acesso base. Então, toma-se a média aritmética obtida anteriormente. Por exemplo, para a média dos pontos de acesso na cozinha seria:

```
30.466.9278.3888.8
```

Continuando, diminui-se de cada média obtida em cada cômodo, o valor do ponto de acesso medido naquele momento, e para o valor obtido retiramos o valor absoluto, e armazenamos, para ser somado aos valores de diferença dos quatro cômodos. Esse valor, que é a soma da diferença entre os valores naquele momento e os valores da *identidade* do cômodo, chamaremos de *Dif(cômodo)*. Supondo que os valores das intensidades dos sinais obtidos no momento da localização sejam

```
31657782
```

Assim, a diferença entre o sinal atual e as médias é:

```
abs(31-30.4)=0.6  
abs(65-66.92)=1.92  
abs(77-78.38)=1.38  
abs(82-88.8)=6.8
```

levando a $Dif(cozinha)=0.6+1.92+1.38+6.8=10.7$ Os valores de *Dif()* de cada cômodo são armazenados em um vetor.

Para realizar a verificação da localização, os $Dif()$ são ordenados. O $Dif()$ de menor valor indica que o cômodo a ele associado tem a maior probabilidade de ser a localização atual do dispositivo. No exemplo dado, os valores de $Dif()$ de cada cômodo seriam:

```
Dif(cozinha)=10.7
Dif(sala)=28.6
Dif(quarto)=17.6
Dif(banheiro)=37.6
```

Logo, como o $Dif(cozinha)$ é o menor, o sistema o indicará a cozinha como a localização do dispositivo no momento.

Para se ter uma noção da acurácia da estimativa do programa, convencionou-se determinar de “precisão” o resultado da seguinte fórmula:

$$\text{Precisão} = (100 - \text{Dif}(\text{cômodo}))$$

O resultado não pode ser expresso em porcentagem, pois a fórmula não leva em consideração os outros cômodos, mas serve de base para uma barra de status na tela do programa, dando ao usuário uma idéia de quão próximo do correto o resultado estimado pelo programa está. Quando o $Dif(\text{cômodo})$ é zero, o que indica a localização exata do cômodo, a barra de status fica cheia. Conforme o $Dif(\text{cômodo})$ aumenta, a barra decresce, dando uma noção da imprecisão da localização.

4.2 Captura de dados

Para o processo de captura de dados é necessária a interação com a interface de rede WiFi do computador onde o sistema está executando. Primeiramente deve-se obter o nome da interface. Nos sistemas Linux, alvo do protótipo, esta informação pode ser encontrada no diretório `/proc/net/wireless` e o módulo `get_interface` retorna um ponteiro para o nome da interface.

Uma vez obtido o nome da interface os dados dos pontos de acesso podem ser coletados. O módulo responsável pela obtenção dos dados (`pega_dados`) inicialmente gera arquivos temporários contendo os dados necessários para o programa.

```
22  sprintf(comando,"iwlist %s scan > ~temp",interface);
23  system(comando);
...
41  system("cat ~temp | grep Cell | cut -c 30- > ~MAC");
42  system("cat ~temp | grep ESSID | cut -c 27- > ~ESSID");
43  system("cat ~temp | grep Quality | cut -d- -f2 |
      cut -c -2 > ~SIGNAL");
```

Nas linhas 22, 41, 42 e 43 temos o código para a geração dos arquivos temporários. Na linha 22, temos a utilização direta da *Wireless Tool iwlist*, utilizando a variável `interface`, que será inicializada pela função de captura de interface, discutida anteriormente.

Nas outras linhas foram utilizados comandos nativos do sistema operacional para criar três arquivos auxiliares (`~MAC`, `~ESSID`, `~SIGNAL`), colocando neles somente as informações úteis ao programa.

4.3 Processamento dos dados

O processamento dos dados é realizado por dois módulos: o ordena e o compara. O primeiro módulo recebe como parâmetro um vetor de uma estrutura (A_P), que é definida no trecho de código seguinte.

```
44  typedef struct
45  {
46  gchar  MAC[18];
47  gchar  ESSID[NAME];
48  gchar  Signal[3];
49  } A_P;
```

O índice do vetor recebido é justamente o número de pontos de acesso identificados, referenciado pela constante AP. Após receber o vetor, o módulo ordena os sinais dos pontos de acesso e os devolve ordenados.

Já o módulo *compara* recebe como parâmetro dois vetores de estrutura A_P, retornando outra estrutura, composta de quatro campos de caracteres, que irão corresponder às intensidades dos sinais dos pontos de acessos. Em resumo, o módulo verifica se os pontos de acesso base (que se encontram no primeiro parâmetro passado) também foram encontrados no segundo parâmetro, que é composto por pontos de acesso encontrados em outro local. Caso sejam encontrados, seus sinais são dispostos na ordem imposta pelos pontos de acesso base. Se não for encontrado algum ponto de acesso base, é atribuído a este um sinal de -99, que significaria um sinal extremamente fraco. Segue um trecho da função referida:

```
71  for(j=0;j<4;j++)
72  {
73      for(i=0;i<n-1;i++)
74          if(strcmp(EndPoint_Base[j].MAC,EndPoint[i].MAC) == 0)
75              strcpy(Signal[j].Signal,EndPoint[i].Signal);
76          if (strcmp(Signal[j].Signal,"0") == 0)
77              strcpy(Signal[j].Signal,"99");
78  }
```

Na linha 74 tem-se a comparação entre os pontos de acesso base com os encontrados localmente. Um ponto importante é que a comparação é feita entre o endereço MAC dos pontos de acesso, uma vez que são únicos.

4.4 Rotinas de coleta de dados

Entre as rotinas de coleta de dados temos as que coletam os dados iniciais (nº de cômodos e nome dos cômodos), bem como as que coletam os sinais de pontos de acesso pelos cômodos. Iniciaremos pelas primeiras, pois são mais simples.

A rotina *evento_botao_ok* e *evento_botao_avancar*, como os nomes indicam, repondem, respectivamente, aos eventos dos botões ok e avançar. A primeira rotina verifica a quantidade de cômodos que entrarão no sistema, e depois habilita o botão avançar. A rotina *evento_botao_avancar* obtém o nome dos cômodos inseridos pelo usuário, e os armazena, para serem utilizados posteriormente na janela principal. Já a rotina *coleta_dados* merece uma atenção especial, por ser mais complexa e abrigar

detalhes essenciais para o correto funcionamento do programa. A rotina `coleta_dados` é chamada ao clicar no botão referente a qualquer cômodo na tela principal.

Essa rotina, essencialmente, cria a janela de coleta de dados, mostrada na Figura 4, instancia uma barra de progresso, uma etiqueta, onde inicializa com o texto “Coletando dados de [cômodo]...”, e cria o botão ok, deixando-o desabilitado.

```

109  gtk_widget_set_sensitive (Coleta->botao, FALSE);
110  Signal_comodo[indice].signal_1 = 0.0;
111  Signal_comodo[indice].signal_2 = 0.0;
112  Signal_comodo[indice].signal_3 = 0.0;
113  Signal_comodo[indice].signal_4 = 0.0;
114  progress_value= 0.0;
115  progress_step= 0;
116  Coleta->indice = indice;
117  handle_timeout =
      gtk_timeout_add(TIME,(GtkFunction)evento_timeadd_coleta_dados,
                    (gpointer) Coleta);

```

Na linha 109 o botão ok é desabilitado e nas linhas 110-113, são inicializadas as variáveis que vão conter os sinais coletados no cômodo de índice [indice]. Ainda nesse trecho de código também temos a importante chamada ao evento `timeadd_coleta_dados`, na linha 117, que é o evento que será executado repetidamente, a cada `TIME` vezes, onde `TIME` é uma constante inicializada como 2000 milisegundos.

```

118  c_Signal = compara(AccessPoint_Base,AP_local);
119  sprintf(endereco,"%s.info",
      gtk_label_get_text(GTK_LABEL(lb_botao_comodo[Coleta->indice])));
120  arquivo = fopen (endereco,"a");
121  if(arquivo != NULL)
122  {
123      sprintf(saida,"%s %s %s %s\n", c_Signal.c_signal_1,
          c_Signal.c_signal_2, c_Signal.c_signal_3,
          c_Signal.c_signal_4);
124      fputs(saida, arquivo);
125      fclose(arquivo);
126  }

```

O trecho compreendido entre as linhas 118-126, já dentro do evento `timeadd_coleta_dados`, copia os dados dos sinais do cômodo em questão para um arquivo texto da forma [nome do cômodo].info. Esse arquivo é disponibilizado para ser lido por outro módulo qualquer que venha a ser desenvolvido, ou serve de *log* para avaliação dos sinais obtidos em cada cômodo.

```

127  Signal_comodo[Coleta->indice].signal_1 =
      Signal_comodo[Coleta->indice].signal_1 +
      (gfloat)(atoi(c_Signal.c_signal_1));
128  Signal_comodo[Coleta->indice].signal_2 =
      Signal_comodo[Coleta->indice].signal_2 +
      (gfloat)(atoi(c_Signal.c_signal_2));
129  Signal_comodo[Coleta->indice].signal_3 =
      Signal_comodo[Coleta->indice].signal_3 +
      (gfloat)(atoi(c_Signal.c_signal_3));
130  Signal_comodo[Coleta->indice].signal_4 =
      Signal_comodo[Coleta->indice].signal_4 +
      (gfloat)(atoi(c_Signal.c_signal_4));
131  progress_value = progress_value+(VALUE*0.1);
132  progress_step = progress_step+(VALUE*10);

```

Acima, da linha 127 à 131, tem-se a acumulação dos sinais obtidos. Esses sinais são somados em uma variável global (Signal_comodo[índice]), para que no próximo módulo seja possível calcular a média dos sinais do cômodo, a partir do total acumulado.

```
133  if (progress_step == 100)
134      gtk_progress_bar_set_fraction
          (GTK_PROGRESS_BAR(Coleta->progress),1.0);
135  gtk_label_set_text(GTK_LABEL(Coleta->lb),
          "Dados coletados com sucesso...");
136  gtk_widget_set_sensitive (Coleta->botao, TRUE);
137  g_signal_handler_disconnect( botao_comodo[Coleta->indice],
          handle_botao_comodo[Coleta->indice]);
138  relatorio_gerado = relatorio_gerado+1;
```

Na linha 132, observa-se o incremento de progress_step, que é uma variável global que tem como função acumular quantas vezes foram coletados os dados, para que a coleta seja finalizada, na linha 133, quando progress_step for igual a 100. VALUE é uma constante com valor de 0.2, indicando que a coleta vai ocorrer 50 vezes.

```
139      if(relatorio_gerado == comodos)
140          gtk_widget_set_sensitive (botao_rodar, TRUE);
141          gtk_widget_set_sensitive (botao_parar, TRUE);
```

Quando a coleta é finalizada, a variável relatorio_gerado é incrementada (linha 138) e na linha 139 verifica-se se todos os cômodos já foram percorridos. Caso a resposta seja afirmativa, os botões Rodar e Parar são habilitados para o início da execução da localização em si.

4.5 Rotinas de localização

As últimas rotinas a serem analisadas são as que, de fato, estimam a localização do terminal. A função evento_timeadd_rodar efetivamente executa a estimação da localização. Abaixo temos alguns trechos importantes desta função.

```
142  c_Signal = compara(AccessPoint_Base,AP_local);
143  valor = (gfloat)(atoi(c_Signal.c_signal_1));
144  valor = ((100.0 - valor)*0.01);
```

Na linha 142 os dados dos pontos de acesso atuais são coletados e armazenados na variável AP_local.

```
148  timer = timer + TIME*0.001;
149  sprintf(time,"%d s",timer);
150  gtk_label_set_text(GTK_LABEL(lb_tempo_2), time);
```

O tempo que é exposto na tela e atualizado a cada iteração é incrementado na linha 148, e colocado em uma etiqueta na linha 150.

```
151  for(i=0;i<comodos;i++)
152  {
153      diferenca[i].signal_1 = abs(Signal_atual.signal_1 -
          (Signal_comodo[i].signal_1*VALUE*0.1));
154      diferenca[i].signal_2 = abs(Signal_atual.signal_2 -
          (Signal_comodo[i].signal_2*VALUE*0.1));
```

```

155     diferenca[i].signal_3 = abs(Signal_atual.signal_3 -
                                (Signal_comodo[i].signal_3*VALUE*0.1));
156     diferenca[i].signal_4 = abs(Signal_atual.signal_4 -
                                (Signal_comodo[i].signal_4*VALUE*0.1));
157     diferenca_total[i] = diferenca[i].signal_1 +
                            diferenca[i].signal_2 + diferenca[i].signal_3 +
                            diferenca[i].signal_4;
158     diferenca_aux[i] = diferenca_total[i];
159 }

```

O trecho de código anterior contém um ponto crítico do programa. Neste ponto é definida a diferença entre o sinal atual de cada ponto de acesso e a média dos sinais coletados pelos pontos de acesso no momento da coleta. A média dos sinais é realizada pegando o valor armazenado na variável global `Signal_comodo`, multiplicando por `VALUE`, que é uma constante que armazena 0.2 e multiplicando em seguida por 0.1. Essa operação equivale à média aritmética de todos os sinais e dividir pelo número de vezes que os sinais foram coletados (cinquenta, no caso).

```

160     j=comodos-1;
161     do
162     {
163         flag = 0;
164         for(i=0; i<j; i++)
165             if( diferenca_aux[i] > diferenca_aux[i+1] )
166             {
167                 aux          = diferenca_aux[i];
168                 diferenca_aux[i] = diferenca_aux[i+1];
169                 diferenca_aux[i+1] = aux;
170                 flag=1;
171             }
172         j--;
173     }

```

Entre as linhas 160 e 173 o método da bolha é utilizado mais uma vez para ordenar as diferenças, que foram devidamente alocadas em um vetor (`diferenca_aux[i]`).

Após a utilização do algoritmo de ordenação, é obtida a menor diferença entre todas estará alocada na posição zero do vetor. E a menor diferença, neste caso, denota justamente o cômodo para o qual existe a maior probabilidade de que seja aquele no qual o dispositivo sem fio se encontra, para dispor ao usuário, refletindo sua atual localização.

```

180     for(i=0;i<comodos;i++)
181     {
182         if(diferenca_total[i] == diferenca_aux[0])
183         {
184             gtk_widget_modify_bg (botao_comodo[i],
                                    GTK_STATE_NORMAL, &azul);
185             gtk_label_set_text (GTK_LABEL(lb_comodo_localizado),
                                   gtk_label_get_text (GTK_LABEL(lb_botao_comodo[i])));
186             precisao = (100 - diferenca_total[i])*0.01;
187             gtk_progress_bar_set_fraction (GTK_PROGRESS_BAR
                                             (progress_precisao), precisao);
188         }
189     }

```

Por fim temos, entre as linhas 180 e 181, a comparação da menor diferença (`diferenca_aux[0]`), com cada uma das diferenças obtidas (`diferenca_total[i]`). Quando o

laço encontra a diferença correta, ele colore o botão do cômodo respectivo de azul (linha 184), coloca na etiqueta lb_comodo_localizado o nome do cômodo em questão (linha 185), e verifica a precisão da localização, para ser colocada na respectiva barra de progresso (linhas 186 e 187). O cálculo da precisão é simples: caso a diferença tenda a zero, a precisão tende a 100%. Conforme a diferença aumenta, temos a degradação da precisão, de acordo com a fórmula descrita na linha 186. O fator precisão não leva em conta a proximidade relativa com os outros cômodos, mas o quão precisa está sendo a informação sobre a localização do cômodo em questão. Não foram estipulados valores limites aceitáveis para a precisão.

5. Utilização e Resultados

A primeira etapa para usar o sistema é proceder ao cadastro dos espaços ou cômodos a serem usados como unidade de identificação da localização. Na Figura 2 dois cômodos foram cadastrados: “Quarto” e “Sala”. Essa identificação permitirá associar uma dada configuração de características de sinais dos APs com a localização onde tal configuração foi obtida.



Figura 2. Nomenclatura dos cômodos

O próximo passo é o treinamento do sistema. Inicialmente o sistema é preparado para detectar e identificar os pontos de acesso. Para isso uma varredura por APs deve ser iniciada (Figura 3).

A cada varredura, os APs detectados e algumas das suas características são exibidos. O módulo responsável busca e trata as 10 primeiras ocorrências de pontos de acesso no local, identificando os seus endereços MAC e disponibilizando em ordem crescente de intensidade de sinal. O usuário pode aceitar o conjunto ou iniciar uma nova varredura (Figura 4).

A mesma coleta é feita para cada cômodo cadastrado. Ao pressionar um botão relativo a um cômodo qualquer, inicia-se o processo de coleta relativo ao mesmo. São capturados sinais dos quatro APs selecionados anteriormente em outro cômodo, e, estes, por sua vez, são armazenados em uma base e processados, para fazer parte da referência de localização.



Figura 3. Tela principal

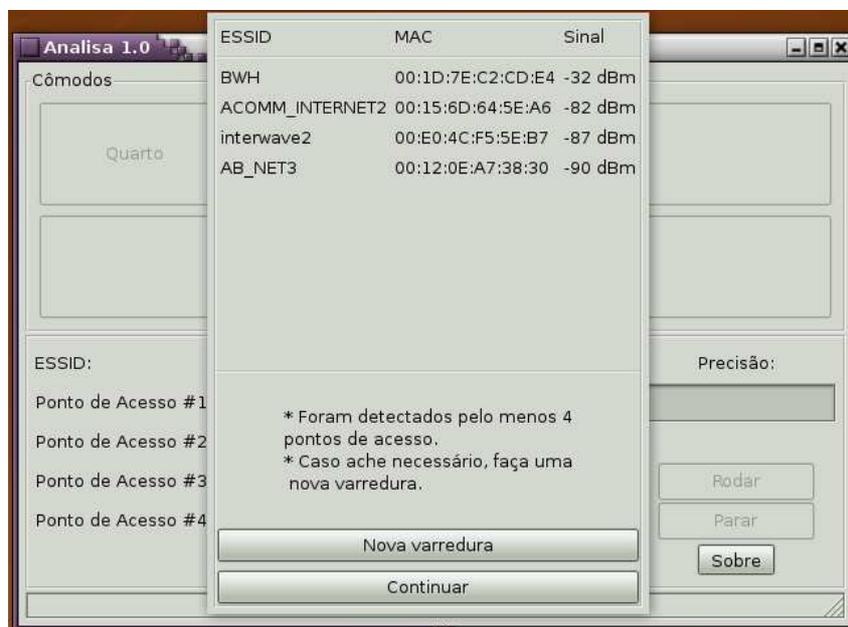


Figura 4. Encontrando pelo menos 4 pontos de acesso

Para que o processo de localização seja feito com qualidade, uma possível estratégia é distribuir a captura dos dados pelo cômodo, passeando lentamente de um canto para o outro, e depois se concentrar no meio do aposento.

Se o nível de potência do sinal RF tivesse um comportamento linear, poderíamos simplesmente concentrar a captura do sinal no centro do aposento, uma vez que a média aritmética de todos os níveis comparados de sinais naquele aposento seria bem similar ao sinal obtido no centro. Porém, como temos características peculiares nos sinais RF, e como estes sofrem atenuações e distorções por objetos (mesas, armários, cadeiras, etc.), recomenda-se que seja feita uma captura percorrendo o aposento. Também é recomendado que o dispositivo realize a captura sempre na mesma altura em relação ao chão. De forma semelhante, para resultados mais proveitosos, é recomendável que, na fase da localização, o dispositivo se encontre a essa mesma altura.

A coleta dos dados em cada cômodo demora cerca de 100 segundos. Ao finalizar a coleta em cada cômodo, o botão que representa o mesmo adquire a cor branca e é desabilitado. Uma vez que a coleta de dados tenha sido realizada em todos os cômodos cadastrados, o treinamento inicial está terminado e o sistema se encontra pronto para uso (Figura 5).



Figura 5. Sistema com todos os cômodos treinados

Ao pressionar o botão *Rodar*, o programa chama o módulo *Localizar Cômodo* de dois em dois segundos. Este por sua vez dispara a rotina de busca de sinais, composta pelo Pegar dados, e faz a comparação entre os sinais obtidos no momento com os de referência armazenados na memória, realizando isso através do módulo Comparar dados. Além disso, a cada busca e comparação, são impressos os dados atuais dos pontos de acesso na tela, é atualizado o tempo e é obtida a localização do cômodo pela comparação dos dados mencionada anteriormente. O cômodo localizado adquire a cor azul na tela, e tem o seu nome impresso no campo Localização, como pode ser visto na Figura 6. Também observamos nesta mesma tela o campo Precisão, que é composto de uma barra de progresso verifica o nível de precisão da localização do respectivo cômodo.



Figura 6. Localizando o primeiro cômodo

Na Figura 7 temos o momento em que o dispositivo foi deslocado para outro cômodo, identificando a sua nova localização, registrando os sinais naquele momento e estimando a precisão da localização gerada.



Figura 7. Localização de outro cômodo

5.1 Avaliação do Sistema

O ambiente ideal para a utilização do sistema deveria dispor de no mínimo quatro pontos de acesso localizados dentro dos limites dos cômodos a serem localizados. Esse fato é evidenciado pela falta de precisão e instabilidade da localização retornada a partir dos sinais medidos dos pontos de acesso, quando estes se encontram muito distantes do

terminal. Cabe ressaltar que a instabilidade dos sinais é inerente à própria tecnologia WiFi.

Testes foram realizados em uma residência onde a maioria dos APs eram roteadores Linksys WRT54G versão 8 [Lin10]. Os demais APs localizados na vizinhança apresentaram sinais consideravelmente baixos, fazendo com que a eficácia do programa fosse reduzida. Ainda assim, os testes proporcionaram resultados satisfatórios, localizando corretamente o terminal nos cômodos em um ambiente residencial.

6. Conclusão

Neste trabalho apresentou-se um sistema para localização de terminais WiFi. O protótipo do sistema permite a possibilidade de cadastro de até oito cômodos, esta restrição pode ser facilmente eliminada. Em ambientes devidamente preparados, com maior densidade de APs, a localização em vários cômodos ou regiões torna-se factível.

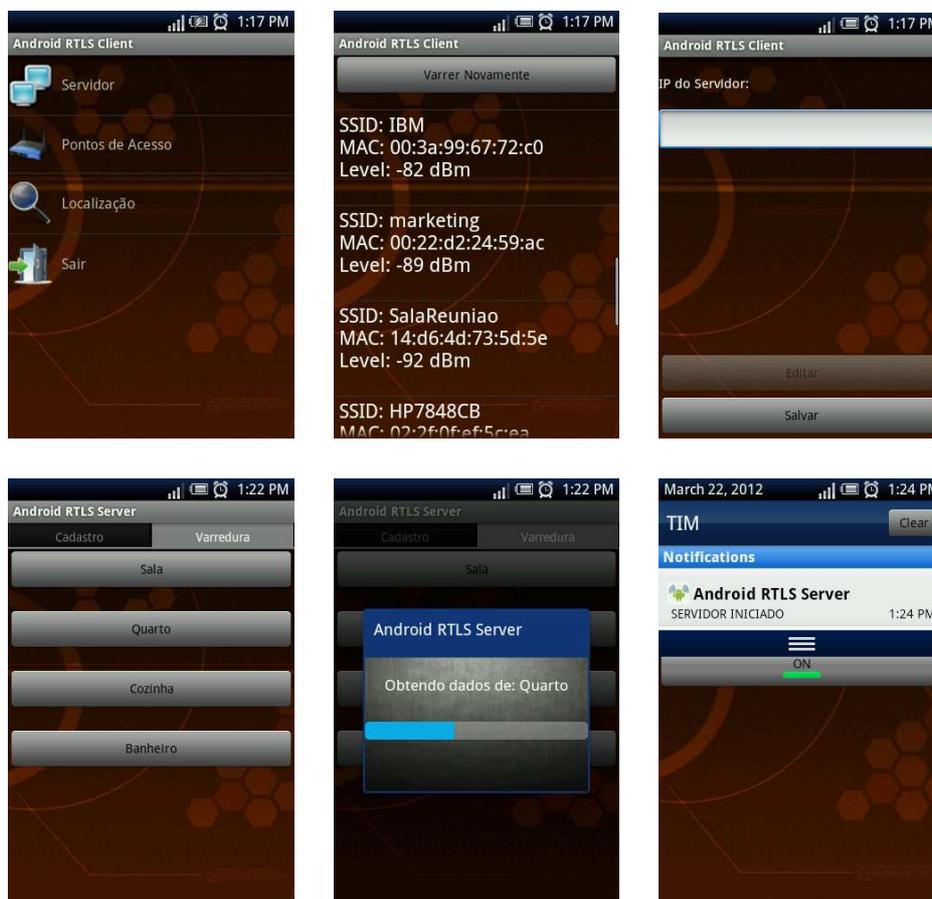


Figura 8. Telas do novo protótipo sobre a plataforma Android

A heurística de identificação apresentada comporta-se bem quando é necessária uma localização de baixa resolução como entre cômodos de uma habitação, ou salas e regiões. Em casos onde a resolução de localização precisa de maior acurácia, outros métodos e meta-heurísticas para reconhecimento de padrões podem ser utilizados. Uma discussão aprofundada é apresentada em [Cam11].

Durante a elaboração deste texto, um novo protótipo foi implementado, com alguns aprimoramentos¹. Os módulos cliente e servidor foram separados e a comunicação entre eles é feita através de *sockets*. Este novo protótipo foi implementado sobre a plataforma Android, também utilizada em outros projetos de pesquisa na área de sistemas cientes de contexto desenvolvidas por nosso grupo. A Figura 8 apresenta algumas telas do módulo cliente e do módulo servidor.

O sistema proposto é o arcabouço para o desenvolvimento de um serviço de localização integrável a sistemas que sejam cientes de contexto e localização. Permitindo assim empregar computação ciente de contexto (*context-aware*) e ciente de localização (*location-aware*).

Referências

- BAHL, P., PADMANABHAN, V. N. RADAR: An In-Building RF-based User Location and Tracking System. In Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 775–784, Tel Aviv, Israel, Março, 2000.
- CAMPOS, R. S. Localização de terminais móveis utilizando correlação de assinaturas de rádio frequência. Dissertação de Mestrado em Engenharia Eletrônica, PEL, UERJ, 2010.
- Campos, R. S., Lovisolo, L. RF Fingerprinting Location Techniques. In: Reza Zekavat; R. Michael Buehrer. (Org.). Handbook of Position Location: Theory, Practice and Advances. 1 ed., John Wiley & Sons / IEEE, pp. 487-522, 2011.
- CHEN, M., SOHN, T., CHMELEV, D., et al. Practical Metropolitan-Scale Positioning for GSMPhones. In Proceedings of 8th International Conference on Ubiquitous Computing, pp. 225–242, Newport Beach, USA, Setembro, 2006.
- Cisco Systems Inc. Wi-Fi Location-Based Services 4.1 Design Guide© 2008. Maio, 2008. http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/lbswifig_external.pdf [Acessado em Março 2010]
- Ekahau Inc. Ekahau RTLS, 2010. <http://www.ekahau.com/products/real-time-location-system/overview.html> [Acessado em Março 2010]
- GEZICI, S. A Survey on Wireless Position Estimation. Wireless Personal Communications: An International Journal, Vol. 44, No. 3 pp. 263–282, Fevereiro, 2008.
- Google.com. Android. 2011. <http://www.android.com/> [Acessado em Março de 2012]
- IEEE 802.11 Comitee, “IEEE 802.11a-1999 High-speed Physical Layer in the 5 GHz band”.
- IEEE 802.11 Comitee “IEEE 802.11b-1999 Higher Speed Physical Layer Extension in the 2.4 GHz band”.

¹ Projeto de Graduação do aluno William César Janoti, DICC/IME/UERJ, 2012.

IEEE 802.11 Comitee "IEEE 802.11g-2003: Further Higher Data Rate Extension in the 2.4 GHz Band".

IEEE 802.11 Comitee "IEEE 802.11n-2009—Amendment 5: Enhancements for Higher Throughput.

Cisco Inc. Cisco Routers - Lynksys® by Cisco, 2010. <http://www.linksysbycisco.com/US/en/products/Routers> [Acessando em Março 2010].

PAHLAVAN, K., XINRONG L., MAKELA, J.P. Indoor geolocation science and technology. Communications Magazine, IEEE , Vol.40, No.2, pp.112-118, Fevereiro, 2002.

International Organization for Standardization. JTC 1/SC 31/WG 5 - Real time locating systems. 2011. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=361518 [Acessado em Março 2012].

SPIRITO, M. A., CACERES, M., SOTTILE, F. WLAN-Based Real Time Vehicle Locating System. In Proceedings of IEEE 69th Vehicular Technology Conference, Barcelona, Espanha, Abril, 2009.

STALLINGS, B. Wireless Communications and Networks. 2nd Edition, Prentice Hall, 2005.

VILELA, U. C., CARDOSO, K. V., REZENDE, J. F. Redes 802.11 em Centros Urbanos: Varredura, Estatísticas e Aplicações. In VI Workshop em Desempenho de Sistemas Computacionais e de Comunicação - WPerformance'2007 (XXVII CSBC), pp. 703–718, Rio de Janeiro, Brasil, Junho, 2007.

TOURRILHES, J. Wireless Tools for Linux. 2008. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html [Acessado em Março 2010]

YACOUB, M. D. Foundations of Mobile Radio Engineering. CRC Press, 1993.

IEEE COMPUTER SOCIETY. IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications 2007.

Handbook of Position Location: Theory, Practice, and Advances, First Edition. Edited by Seyed A. (Reza) Zekavat, Michael Buehrer. © 2011 the Institute of Electrical and Electronics Engineers, John Wiley & Sons.