

## ListenU uma Ferramenta para Monitoramento Ambiental Usando Redes de Sensores Sem Fio

Ilan Sousa<sup>1,2</sup>, Lauro Américo<sup>1,2</sup>, Lilian Freitas<sup>1,2</sup>, Aldebaro Klautau<sup>1</sup>, João Costa<sup>2</sup>.

<sup>1</sup>Lab. de Sensores e Sistemas Embarcados – Universidade Federal do Pará (UFPA)  
Caixa Postal 479 – Belém – Pa – Brasil

<sup>2</sup>Lab. de Eletromagnetismo Aplicado – Universidade Federal do Pará (UFPA)  
Caixa Postal 479 – Belém – Pa – Brasil

{ilan, americano, liliancf, aldebaro, jweyl}@ufpa.br

**Abstract.** *This work presents ListenU, a tool for environmental monitoring using wireless sensor networks that allows collecting physical information. Based on the TinyOS operating system and presented with MicaZ sensors, a base station and an easy graphical interface, ListenU can be used to collect temperature, light, acceleration and ambient sounds.*

**Resumo.** *Este trabalho apresenta o ListenU, uma ferramenta de monitoramento ambiental que permite a coleta de informações, tais como temperatura, luminosidade, aceleração e sons do ambiente. Tendo como base o sistema operacional TinyOS, a mesma foi testada nos nós sensores MicaZ. Apresenta uma interface gráfica de fácil visualização, caracterizando-se pela utilização dos recursos de hardware de forma otimizada.*

### 1. Introdução

Redes de sensores sem fio (RSSF) são compostas por inúmeros nós sensores, comunicando-se diretamente entre si, sem a necessidade de infra-estrutura, como as redes sem fio tradicionais [Bulusu et al, 2005]. Os recursos básicos de uma rede de sensores são: capacidade de auto-organização em caso de falha de comunicação e mudança de topologia; comunicação em *broadcast* em curto alcance e roteamento por múltiplos saltos para comunicação de sensores fora de alcance [TEP-118, 2011], [TEP-119, 2011].

As maiores limitações de uma RSSF são a quantidade de recursos (processamento, memória e alcance de rádio) baixos ou escassos [Callaway, 2003]. Além disso, as métricas de desempenho, em geral, são abaixo das observadas em redes sem fio tradicionais, considerando-se eficiência energética ou vida útil do sistema, latência para coleta ou transmissão de informações e tolerância a falhas. Entretanto, apesar das limitações impostas por restrições de recursos, a natureza distribuída e móvel das RSSF torna-as interessantes para diversas áreas, como no controle de ambientes, de tráfego, na medicina, na segurança de prédios, monitoramento de ambiental, dentre outros.

Em aplicações de monitoramento ambiental, a utilização das RSSF se torna vantajosa por permitir o monitoramento de áreas de difícil acesso de forma autônoma,

evitando situações que colocariam em risco a vida de pessoas, ou a situações adversas, além de evitar a intrusão da ação humana no ambiente. Por exemplo, monitoramento de áreas florestais com RSSF permite analisar focos de incêndio causados pela ação humana. Além disso, o monitoramento de sons nas florestas poderia ser utilizado para monitorar desmatamentos por motosserras e similares, enviando um alerta aos órgãos responsáveis, antes mesmo que pudessem ser notados por satélites.

A maioria das soluções de RSSF, tanto as comerciais quanto as acadêmicas tem como base o sistema operacional TinyOS, o qual traz consigo inúmeras vantagens, por exemplo, o fato de ser livre e de código aberto, permitir programação por componentes, ser projetado para ter um uso eficiente de energia, entre outros. A popularidade do TinyOS é comprovada pelo fato dele ter uma média de 35 mil *downloads* por ano [TinyOS, 2011].

Apesar disso, realizar a coleta de informações em uma RSSF com base no TinyOS ainda não é uma tarefa trivial, dada a indisponibilidade de uma ferramenta que realize essa tarefa de forma automática, e tendo em vista a incompatibilidade de pacotes de certos protocolos de comunicação que utilizam diferentes pacotes de mensagens. Nesse contexto, este artigo implementa uma ferramenta chamada ListenU, a qual é uma ferramenta pra monitoramento ambiental genérica, podendo ser usada para coletar dados de uma RSSF de forma automática (conforme configuração do operador da rede), tais como temperatura, luminosidade, aceleração e sons. Além disso, a ferramenta ListenU também realiza a compressão dos pacotes de dados, objetivando aumentar a taxa de transmissão da rede.

O restante deste artigo está organizado da seguinte forma: a Seção 2 descreve as plataformas de hardware e software utilizados. A Seção 3 descreve os resultados obtidos através da implementação da ferramenta ListenU. A Seção 4 mostra as conclusões e trabalhos futuros.

## **2. Background**

### **2.1. Plataforma de Software: TinyOS**

O TinyOS é um sistema operacional livre, especialmente desenvolvido para dispositivos de baixa potência, tais como RSSF. Baseia-se em uma arquitetura que minimiza o código escrito através do uso de componentes, reduzindo a quantidade de memória utilizada. Foi desenvolvido de forma a maximizar a energia disponível nos nós sensores, tanto na execução do código do programa quanto na transmissão de pacotes por meio sem fio.

Por ser software livre, muitos desenvolvedores fornecem novos protocolos a serem suportados pelo TinyOS que podem ser utilizados por outros, além do suporte a um número cada vez maior de nós sensores, microcontroladores e rádios.

Apesar de inúmeros protocolos de comunicação estarem disponíveis atualmente para as RSSF utilizando TinyOS, os protocolos de comunicação considerados na implementação do ListenU são dois. Um deles é o de *Disseminação* [TEP-118, 2011] de informação. É um protocolo *multihop* básico utilizado para enviar com os pacotes de dados através da RSSF. Outro é o de *Coleta* [TEP-119, 2011] de informação, o qual envia pacotes de dados dos sensores a um nó raiz [Philip et al, 2008].

O protocolo de *Disseminação* permite que administradores da rede atualizem valores de variáveis, façam requisições a rede ou até mesmo reprogramem uma rede inteira. A segurança na entrega das informações é importante para tornar a operação robusta o suficiente contra desconexões temporárias ou um grande número de pacotes perdidos. Neste protocolo, quando um nó, emite um evento de atualização com uma informação, todos os outros o recebem. Na ferramenta ListenU implementada, esse método foi utilizado para pedir aos sensores que obtivessem informações de um ou mais sensores, ou mesmo, realizar uma requisição de som. Os outros nós sensores, por outro lado, respondem com o que foi pedido e transmitem a informação ao nó principal. Para isso, usou-se o protocolo de *Coleta*, enviando as informações para um nó raiz, neste caso, a estação-base.

## **2.2. Plataforma de Hardware**

### **2.2.1. Nós Sensores**

Para a construção da ferramenta ListenU, foram utilizados nós sensores MicaZ [Crossbow, 2008], desenvolvidos na Universidade de Berkeley, Califórnia, EUA. Essa plataforma de nó sensor é uma das mais versáteis para a construção de protótipos. Eles são compostos por um microcontrolador de alto desempenho e baixa potência, o ATmega128L, operando a 8MHz. Possui 128 kB para memória flash de programa, 4 kB de memória de dados SRAM e 512 kB de memória de armazenamento, flash serial, externa ao microcontrolador. O MicaZ mede 5.7 cm x 3.1 cm x 1.8 cm e é alimentado por duas pilhas AA, podendo ser recarregáveis. As grandezas físicas são lidas pelo MicaZ através de conversores analógico/digitais e canais digitais, os quais são conectados às placas sensoras.

Para realizar comunicações, o MicaZ utiliza o padrão ZigBee e o rádio ChipCon CC4220, com taxa de transmissão de até 50 Kbps, frequência de operação programável entre 2400 MHz e 2483.5 MHz e potência de transmissão ajustável entre 0 dBm e 25 dBm, cujo alcance pode chegar a dezenas de metros em campo aberto. A frequência de amostragem máxima do A/D de 10 bits é de 13 KHz, bem aquém do exigido por muitas aplicações.

### **2.2.2 Placa sensora**

Para testar a ferramenta ListenU foi utilizada a placa sensora MTS310CA [Crossbow, 2008], desenvolvida pela Crossbow Technology Inc., a qual contém os sensores de temperatura, som e luz, além de um buzzer, acelerômetro e magnetometro.

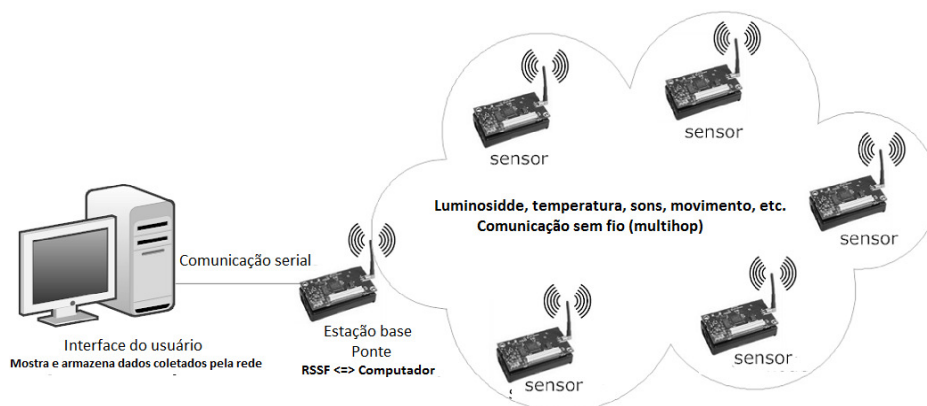
### **2.2.3. Gateway**

A extração de informações da rede é realizada através do gateway MIB510 [Crossbow, 2008], permitindo a comunicação do computador, comunicando a estação-base com a RSSF, via porta serial. Uma alternativa também possível é na utilização do gateway MIB600, realizando a comunicação entre a rede de sensores e a estação-base via Ethernet, descartando a necessidade de conexão imediata com um computador. Essa placa, porém, necessita de conectividade a uma rede local.

## **3. Resultados**

O ListenU é uma ferramenta completa para monitoramento, abrangendo desde o *firmware* presente nos nós sensores e na estação-base, até a aplicação que é executada no computador. A Figura 1 mostra a organização dos componentes da ferramenta, sendo

estes divididos em três módulos: sensor, estação base e interface com o usuário. Cada um destes é descrito nos tópicos a seguir.



**Figura 1: Estrutura da aplicação.**

### 3.1. Módulo Sensor

O módulo sensor agrega várias funções de monitoramento, podendo coletar informações referente à temperatura, luminosidade, aceleração e sons, podendo analisar o nível de bateria, potência de sinal de rádio recebido e a sua localização em relação a outro sensor. O princípio básico do módulo sensor é enviar amostras de dados a cada 20 minutos ou esperar uma requisição do usuário, enquanto fica em modo de economia de energia, aguardando. A requisição chega a todos o nós da rede utilizando o protocolo de disseminação de dados, pedindo quais grandezas devem ser monitoradas, quantas amostras, etc. Ao capturar os dados da requisição o módulo sensor prepara um pacote e os envia à base utilizando o protocolo de coleta de dados.

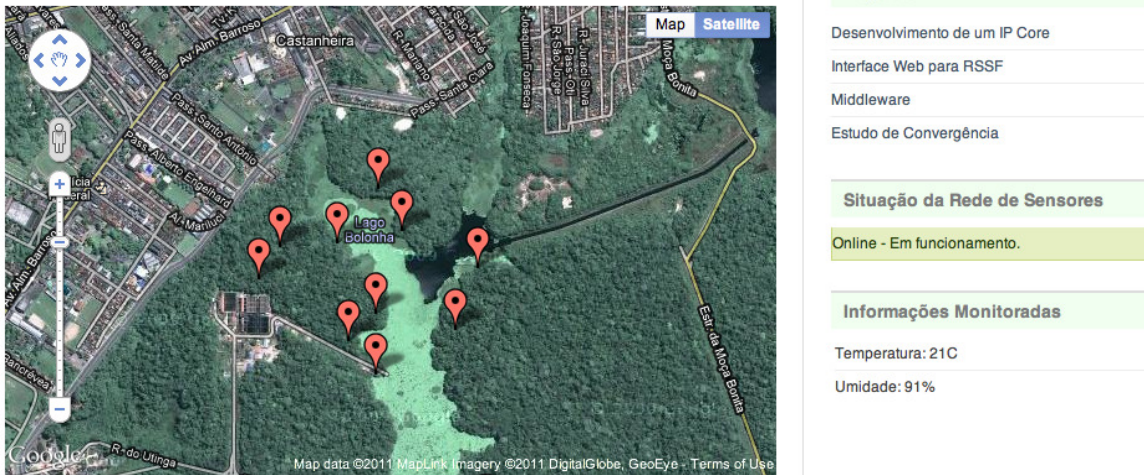
### 3.2. Módulo Estação Base

O módulo que roda na estação base é a ponte de comunicação entre a interface do usuário (computador) e a rede de sensores. Ela é composta por um nó sensor (com *firmware* diferenciado) acoplada a um *gateway* para poder comunicar-se com o computador, a API do TinyOS oferece ferramentas que possibilitam utilizar comunicação com a estação-base via RS232 e USB, pela placa MIB510, e Ethernet, pela placa MIB600. Basicamente o módulo que executado na estação base atua repassando dados do computador para a RSSF e dela para o computador. Uma requisição recebida do computador pela estação base é tratada e repassa a todos os nós da rede (disseminação) e dados de requisição são recebidos pela base, tratados e enviados ao computador. O papel do módulo da estação base, apesar de parecer simples, requer atenção, pois dependendo da quantidade de nós na rede e da quantidade de dados de uma requisição pode haver tráfego intenso, e, portanto, ele precisa conter um código otimizado.

### 3.3. Módulo Interface com Usuário

No computador é executada uma interface gráfica onde o usuário pode fazer uma requisição de leitura e ler as informações recebidas pela rede. A aplicação utiliza uma software presente no TinyOS chamada SerialForwarder que abstrai os dados oriundos da RSSF e repassa ao cliente.

Ainda no módulo de interface com o usuário, está em fase de desenvolvimento uma interface web que permite que os dados da RSSF sejam disponibilizados na Internet. Para esse cenário foi desenvolvida uma aplicação usando-se a API do Google Maps, a qual permite ao usuário acessar as informações da rede em tempo real, inclusive ouvir os sons da floresta à medida que navega o mapa, como ilustrado na Figura 2.

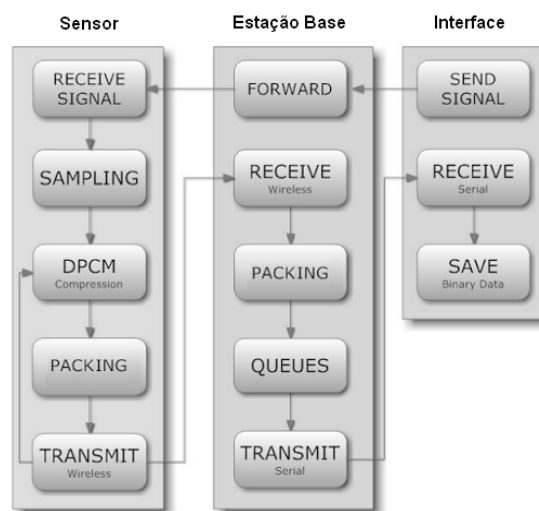


**Figura 2: Interface web para coleta de informações em tempo real de uma RSSF. Pode ser vista em: <http://www.lea.ufpa.br/~sensores/projetos/webrssf.html>**

### 3.4. Fluxo de dados

O ListenU foi projetado para capturar uma amostra toda vez que houver uma requisição do usuário. Essa requisição é originada na interface gráfica e é enviada para a rede. Para permitir que a requisição chegue à rede são usados dois tipos de comunicações: a comunicação entre o computador e a estação base (que pode ser serial via RS232, USB ou ethernet) e a comunicação entre a estação base e os nós sensores.

A Figura 3 mostra o fluxo de dados no ListenU. Uma requisição é enviada à estação base na forma de um pacote contendo informações sobre o que deve ser lido, por quanto tempo e taxa de amostragem. Ao receber a requisição a estação base repassa a todos os nós sensores da rede.



**Figura 3: Diagrama de fluxo de dados no ListenU.**

Ao receber a requisição os sensores preparam-se para capturar as amostras, no caso de captura de áudio um tratamento especial é necessário, as amostras são comprimidas usando algoritmo DPCM (*Differential Pulse Code Modulation*) [Spanias, 1991]. Após a captura das amostras o sensor envia suas amostras à base utilizando comunicação *multihop*, na forma de um ou mais pacotes. Ao receber pacotes com amostras a estação base repassa ao computador, que por sua vez armazena ou mostra os dados ao usuário.

## 6. Conclusões e Trabalhos Futuros

Monitoramento é uma tarefa árdua, principalmente quando não se dispõe de ferramentas que auxiliem no trabalho. Com base nisso este trabalho apresentou o desenvolvimento de uma ferramenta de sensoriamento bastante flexível, pois permite monitorar condições de um ambiente como luminosidade, temperatura, movimentação e sons. Inicialmente foram apresentadas as ferramentas usadas para o monitoramento, onde foram usados os nós sensores MicaZ, um *gateway* e um computador. Porém não se restringe a esses componentes, podendo ser usado em alguns outros nós sensores que o TinyOS suporta.

Dentre os trabalhos futuros para melhoria da ferramenta ListenU destacam-se: a adição de algoritmos de reconhecimento de padrões no nó sensor, que poderiam ser aplicados ao reconhecimento de sons de animais ou detecção de sons específicos como o som de uma motosserra, para detecção de derrubada de árvores. Outra melhoria, que já está sendo implementada é o módulo de monitoramento da rede em tempo real, através de uma interface web. Assim, não há a necessidade do observador estar próximo da estação-base para o monitoramento. Entretanto, isso adicionaria o requisito de acesso à Internet na estação base, podendo ser via satélite, em situações de difícil acesso.

## Referências

- Bulusu, N., Jha, S., (2005). "Wireless Sensor Networks – A Systems Perspective", London, Artech House.
- Callaway, E. H. (2003). "Wireless Sensor Networks: Architectures and Protocols". Florida, Auerbach Publications; 1 edition (August 26, 2003)
- TinyOS Home Page, (2011). <http://www.tinyos.net/>. Acessado em 26/03/2011.
- Philip Levis, David Gay (2009) "TinyOS Programming", Cambridge, 1ª edition.
- TEP-118 - TinyOS Enhancement Proposal. (2010). "Dissemination of small values", 2010; <http://www.tinyos.net/tinyos-2.1.0/doc/html/tep118.html>
- TEP-119 - TinyOS Enhancement Proposal. (2010). "Collection", 2010; <http://www.tinyos.net/tinyos-2.1.0/doc/html/tep119.html>
- Crossbow Technology, Inc, (2008). "MPR-MIB User Manual". <http://www.cs.ucsb.edu/~nchohan/docs/moteManual.pdf>. Acessado em 26/03/2011.
- Spanias, A.S.; Wu, F.H.; "Speech coding and speech recognition technologies: a review," *Circuits and Systems, 1991., IEEE International Symposium on*, pp.572-577 vol.1, 11-14 Jun 1991.