

# SAMCRA: Um sistema para avaliação experimental de Redes Ad Hoc

Rodrigo S. Granja<sup>1</sup>, Renato C. Dutra<sup>1</sup>, Hébert F. de Moraes<sup>1</sup>, Cláudio L. de Amorim<sup>1</sup>

<sup>1</sup>Coppe - Programa de Engenharia de Sistemas e Computação  
Universidade Federal do Rio de Janeiro (UFRJ)  
Caixa Postal - 68.511 - Rio de Janeiro - RJ - Brasil

{rodrigoss, rcdutra, hmoraes, amorim}@lcp.coppe.ufrj.br

**Abstract.** SAMCRA (Ad Hoc Network Monitoring, and Configuration System) is a system to support the development of routing algorithms for Ad Hoc networks. SAMCRA is implemented in Java and offers modules to monitor, configure, collect data, trace messages, and perform automatic network executions, for experimental evaluation of routing algorithms. We present a detailed description of SAMCRA and results for a routing algorithm running on a 20-node sensor network.

**Resumo.** SAMCRA (Sistema de Automação, Monitoração e Configuração de Redes Ad Hoc) é um sistema de apoio ao desenvolvimento de algoritmos de roteamento para redes Ad Hoc. SAMCRA está implementado em Java e oferece módulos de monitoração, configuração da rede, coleta de dados, rastreamento de mensagens e execuções automatizadas para avaliação experimental de algoritmos de roteamento. Apresentamos uma descrição detalhada do SAMCRA e resultados de um algoritmo de roteamento executado em uma rede de sensores sem fio com vinte dispositivos.

## 1. Introdução

O roteamento de mensagens em redes Ad Hoc sem fio [Corson and Macker 1999] tem sido objeto de várias pesquisas [Kini et al. 2006, Novatnack et al. 2005] e os algoritmos de roteamento propostos são adequados para classes específicas de aplicações. A avaliação desses algoritmos em uma rede real é importante, mas a construção de experimentos em redes Ad Hoc apresenta dificuldades práticas, o que leva a maioria dos trabalhos a recorrer a simulações, cujos resultados deveriam ser validados em ambientes reais [Cavilla et al. 2004].

Para avaliar este tipo de rede, especialmente algoritmos de roteamento, é preciso que se conheça o comportamento de variáveis, tais como a quantidade de mensagens que foi trocada entre os nós, a quantidade de mensagens descartadas, quais foram os grupos formados na rede, a eficiência dos mecanismos de controle de propagação de mensagens, o custo de envio das mensagens, a capacidade de transmissão da rede sem perdas, atraso de entrega, entre outras [Novatnack et al. 2005].

Neste trabalho, apresentamos uma ferramenta chamada SAMCRA (Sistema de Automação, Monitoração e Configuração de Redes Ad Hoc), criada para apoiar o processo de avaliação de algoritmos de roteamento para redes Ad Hoc reais. Ela permite configurar a rede, monitorar a execução de aplicações, coletar dados sobre os nós da rede

e informações sobre a interação entre eles, que podem ser usadas para gerar grafos do comportamento da rede, rastrear mensagens (*trace*), controlar execuções automatizadas e gerar planilhas eletrônicas com dados consolidados de múltiplas execuções. SAMCRA é usado atualmente para realizar experimentos de um novo protocolo para redes Ad Hoc, tanto para avaliar o seu funcionamento quanto para a sua comparação com os algoritmos de inundação e *Gossip*.

O restante deste texto está organizado da seguinte forma: na Seção 2, é apresentada a motivação para o desenvolvimento do trabalho. Na Seção 3, o sistema SAMCRA é descrita em detalhes. Na Seção 4, o estado atual do sistema é apresentado. Na Seção 5 são delineadas oportunidades de trabalhos futuros e, finalmente, na Seção 6, é detalhada a demonstração que será feita para apresentar as funcionalidades da ferramenta.

## 2. Motivação

Uma etapa importante no desenvolvimento de algoritmos de roteamento para redes Ad Hoc é a avaliação do seu funcionamento e desempenho em cenários realistas. Dentre as dificuldades encontradas nessa avaliação, se destacam: a grande quantidade de variáveis que precisam ser monitoradas e que devem ser analisadas tanto de forma global quanto individual em cada nó; a dificuldade para a realização de execuções repetidas, para aumentar a confiabilidade estatística dos resultados; e a necessidade de configurar a rede de forma homogênea com várias configurações diferentes, para avaliar os impactos causados pela variação dos parâmetros do algoritmo.

Durante o desenvolvimento de um protocolo para redes Ad hoc, chamado REPI [Dutra and Amorim 2010], buscamos ferramentas para nos auxiliar neste tipo de avaliação. Os principais requisitos de tal ferramenta eram: oferecer as capacidades de avaliação listadas anteriormente, atender ao projeto da arquitetura computacional proposta, oferecer flexibilidade para reorganização física dos nós, além de suporte e disponibilidade integral para a realização de experimentos.

Uma das opções que avaliamos foi a utilização de um ambiente para testes (*test-bed*) como o proposto em [Werner-Allen et al. 2005]. Porém, este tipo de ambiente não cumpria todos os nossos requisitos, pelos seguintes motivos: o ambiente é compartilhado e os usuários possuem cotas de uso limitadas; a posição dos nós não pode ser alterada; não há garantia de suporte em caso de falhas dos dispositivos e do serviço em si, apenas um relatório de quais deles estão funcionando; a coleta de dados oferecida é simples, se resumindo a um registro das mensagens trocadas entre os dispositivos.

Em [Viterbo et al. 2006] é proposta uma arquitetura de software chamada de MoCA, cujo objetivo é apoiar o desenvolvimento e execução de aplicações em redes formadas por dispositivos móveis. MoCA foi projetada para o desenvolvimento de aplicações sensíveis ao contexto e por isso coleta informações específicas para este fim, como qualidade da conexão sem-fio, carga da bateria, uso da CPU e memória livre. Ou seja, é uma ferramenta para que as aplicações possam se beneficiar do contexto da rede e não para avaliar o seu desempenho. Por esta razão ela também não atende aos nossos requisitos.

Em função da flexibilidade que buscávamos para atender os requisitos propostos, decidimos desenvolver um sistema, chamado SAMCRA, para a realização de experimentos em uma rede de sensores sem fio com vinte dispositivos Tmote Sky [Moteiv 2006].

Embora os ambientes de testes de redes [Werner-Allen et al. 2005] não tenham sido utilizados neste trabalho, SAMCRA também pode ser utilizado para monitorar experimentos realizados com eles. Para isso, seria necessário o desenvolvimento de uma aplicação Java capaz de se comunicar com SAMCRA, como mostrado na Seção 4, e de coletar os dados dos dispositivos disponibilizados por esses ambientes.

Uma vantagem da utilização do ambiente de testes é a maior quantidade de dispositivos, com até setenta e quatro no mesmo ambiente. Além disso, ele permite reprogramar todos os nós de uma única vez, funcionalidade ainda não incluída em SAMCRA.

### 3. SAMCRA

A Figura 1 apresenta a arquitetura de uma rede Ad Hoc monitorada por SAMCRA. Nesta rede existem três elementos computacionais principais: (1) um servidor, onde o SAMCRA é executado e cujo endereço é conhecido por todos os nós da rede; (2) um ou mais nós computacionais que executam uma ou mais instâncias da aplicação Java que coleta os dados dos dispositivos de comunicação conectados a essas máquinas; (3) os dispositivos de comunicação sem fio capazes de formar a rede Ad Hoc.

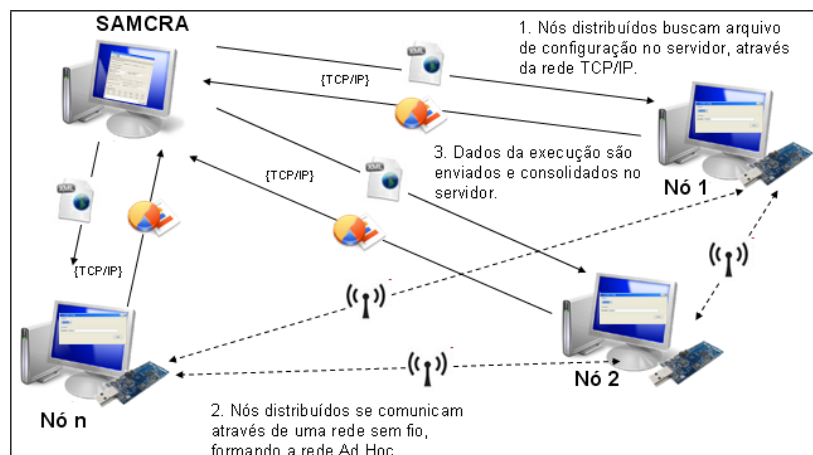


Figura 1. Esquemático de conexões entre SAMCRA, a rede móvel e a rede fixa.

Na Figura 1, duas redes distintas são apresentadas. A rede sem fio, usada para a execução da Rede Ad Hoc e que é controlada pelo algoritmo que está sendo avaliado; e uma rede TCP/IP, usada para a coleta de dados e configuração da rede Ad Hoc. O objetivo dessa configuração é isolar a instrumentação da rede Ad Hoc, evitando que possíveis *overheads* de medição gerem distorções na avaliação da rede.

SAMCRA atua na rede TCP/IP, configurando os nós através da distribuição de um arquivo XML com as configurações da rede e coletando as informações enviadas pelas aplicações Java através de invocação remota de métodos - RMI [Number et al. 2002].

SAMCRA realiza as seguintes tarefas:

#### Configurar a rede

Durante a avaliação de um protocolo de roteamento é muito importante que seus parâmetros possam ser variados e avaliados. Por isso, SAMCRA oferece um módulo de configuração, onde todos os nós podem ser configurados em uma única interface;

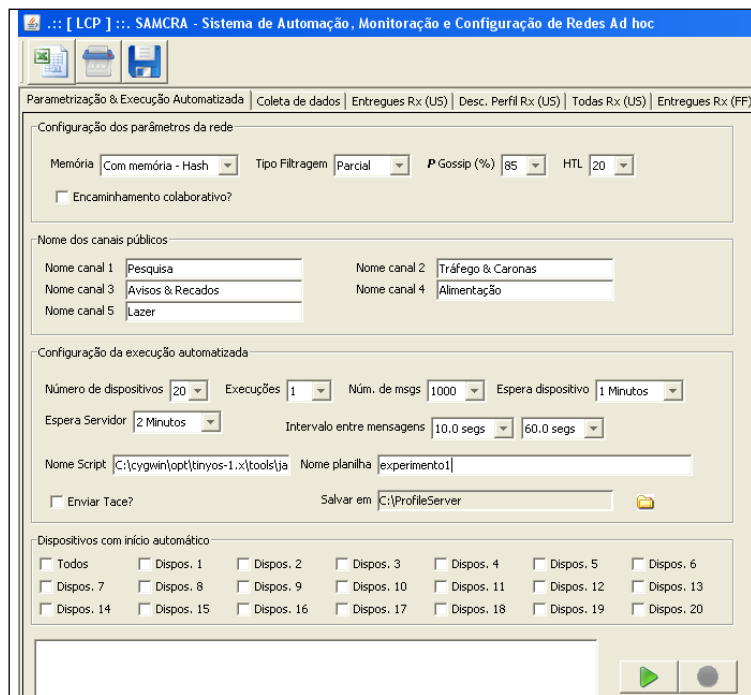
### Coletar dados sobre a execução

Outra etapa importante é a avaliação do comportamento da rede. Por isso, o sistema coleta trinta e cinco variáveis que descrevem de forma detalhada o comportamento de cada nó, além de obter dados detalhados sobre a interação entre eles, tanto em relação aos nós com troca direta de mensagens quanto fim-a-fim;

### Realizar execuções automatizadas

Para validar um algoritmo de roteamento, ele precisa ser testado em vários cenários, repetidas vezes. Por isso, o sistema oferece uma ferramenta que executa experimentos na rede de forma automatizada, repetindo as execuções automaticamente e consolidando os dados gerados em planilhas eletrônicas.

A Figura 2 apresenta a tela inicial do SAMCRA. Como é possível notar, o sistema é organizado em abas. Na tela da primeira aba, mostrada na Figura, estão os quatro painéis onde é possível configurar a rede e realizar execuções automatizadas. Nas próximas subseções são descritos os três módulos do sistema: Módulo de Coleta de dados; Módulo de Configuração; e Módulo de Execução Automatizada.



**Figura 2.** Tela inicial do SAMCRA contendo o painel de configuração da rede e o painel de execução automatizada.

### 3.1. Módulo de Coleta de dados

Este módulo é responsável pela coleta dos dados dos nós da rede. Cada nó envia as informações que coletou para o servidor a cada dez segundos e elas são consolidadas em nove tabelas diferentes, apresentadas a seguir <sup>1</sup>.

#### Tabela Dados de participação por Tmote

Esta tabela é onde estão as principais informações para a análise da rede. Ela

<sup>1</sup>Uma tabela foi omitida da apresentação, pois apresenta dados específicos do algoritmo REPI

consolida os valores de trinta e cinco variáveis que apresentam as seguintes informações sobre cada nó da rede:

- quantidade de mensagens recebidas, encaminhadas e enviadas, de acordo com o tipo;
- quantidade de mensagens descartadas, de acordo com o motivo do descarte;
- descrição de erros ocorridos nos dispositivos de comunicação sem fio, classificados em oito tipos;
- número médio de saltos de todas as mensagens da rede e apenas das que foram entregues aos seus destinatários;
- atraso médio de entrega, em milissegundos, das mensagens da rede com e sem o *overhead* da aplicação Java de controle;

#### **Tabelas com detalhamento de mensagens recebidas por nó - último salto**

Nessas três tabelas são detalhadas as mensagens recebidas pelo nó de acordo com o último salto. Esta distinção é fundamental em uma rede Ad Hoc, pois neste tipo de rede as mensagens podem ser encaminhadas por vários nós antes de serem entregues aos destinatários finais. A primeira delas contabiliza as mensagens que foram entregues aos destinatários finais; a segunda, as que foram descartadas; e a terceira, o somatório de todas as mensagens, incluindo as que foram descartadas;

#### **Tabelas com detalhamento de mensagens recebidas por nó - fim-a-fim**

Essas três tabelas são similares às descritas no item anterior, com a diferença que elas contabilizam as mensagens de acordo com o endereçamento fim-a-fim, ou seja, em relação ao nó que inseriu a mensagem inicialmente na rede;

#### **Tabela com rastreamento de mensagens**

Esta tabela apresenta o rastreamento de uma execução na rede, armazenando dez variáveis com metadados de todas as mensagens recebidas por todos os nós na rede. Essas variáveis são: hora de recebimento, endereço do nó que recebeu a mensagem, endereço do nó de origem da mensagem (fim-a-fim), endereço do último nó por onde a mensagem passou (último salto), valor do HTL (*Hops-to-live*) ao chegar no destino, número sequencial, indicador se a mensagem foi descartada, o motivo do descarte e duas variáveis particulares do protocolo REPI.

### **3.2. Módulo de Configuração**

Nos dois painéis superiores, chamados “Configuração dos parâmetros da rede” e “Nome dos canais públicos”, o usuário determina os parâmetros que definirão o comportamento da rede. Estas informações são transmitidas para os nós através da rede TCP/IP durante a inicialização das aplicações Java distribuídas na rede. Esta configuração é feita através de um arquivo XML, disponibilizado em um servidor Web e acessado pelas aplicações em sua inicialização. Os parâmetros que podem ser configurados são os seguintes<sup>2</sup>:

#### **Memória**

Indica como será a “memória” da rede, ou seja, como ela controlará a duplicação de mensagens, evitando que um mesmo nó encaminhe a mesma mensagem mais de uma vez. Esta configuração tem três opções: controle com número sequencial, controle com tabela hash e sem memória;

---

<sup>2</sup>As configurações Encaminhamento Colaborativo e Nomes dos Canais Públicos não serão detalhadas por serem específicas do algoritmo REPI

**Tipo de Filtragem na rede**

Esta configuração define o critério que determinará se as mensagens da rede serão encaminhadas ou não. Ela pode ter os seguintes valores: algoritmo de inundação; parcial e completo, que são duas configurações do protocolo REPI; *Gossip*, na qual os pacotes são encaminhados seguindo um algoritmo probabilístico em que cada mensagem tem um probabilidade  $P$  de ser encaminhada;

**HTL**

Esta configuração indica qual será o HTL (*Hops-to-live*) da mensagem, ou seja, por quantos nós a mensagem pode passar antes de ser descartada;

**3.3. Configuração da execução automática**

Os dois painéis inferiores, chamados “Configuração da execução automatizada” e “Dispositivos com início automático”, fazem parte da ferramenta de execução automatizada, que permite que mensagens sejam enviadas na rede por até vinte nós simultaneamente sem a intervenção do usuário. Além disso, a ferramenta também permite que essas execuções sejam repetidas várias vezes, para aumentar a confiabilidade dos dados coletados.

Ao realizar uma execução com repetições, o sistema salva os resultados de todas as execuções em planilhas eletrônicas e, ao final, cria uma planilha adicional com a média aritmética e desvio padrão dos resultados. Para realizar uma execução automatizada no SAMCRA, é preciso configurar onze parâmetros, a saber:

**Número de dispositivos; Execuções e Núm de Msgs**

Determinam, respectivamente, as quantidades de nós usados na execução automatizada, de execuções e de mensagens que os nós com início automático enviarão;

**Espera dispositivo e Espera Servidor**

Determinam, respectivamente, quanto tempo cada nó irá esperar para notificar ao servidor que sua execução terminou e quanto tempo o servidor irá esperar para iniciar uma nova execução depois que receber todas notificações de término;

**Intervalo entre mensagens**

Determina o intervalo de tempo entre o envio de duas mensagens consecutivas;

**Nome Script; Nome planilha e Salvar em**

Determinam, respectivamente, o nome do *script* que será executado para iniciar as aplicações Java nos nós remotos, o prefixo dos nomes das planilhas eletrônicas que serão gravadas com os dados coletados e o diretório onde elas serão salvas;

**Enviar Trace?**

Determina se os nós remotos enviarão o rastreamento da mensagens recebidas;

**Dispositivos com início automático**

Determina quais nós enviarão mensagens durante a execução automatizada.

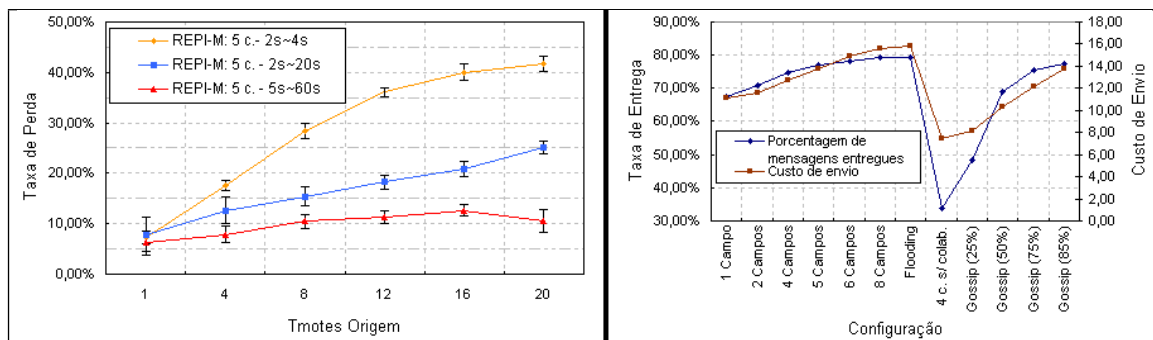
A execução automatizada é realizada em um ciclo, no qual o *script* especificado inicia as aplicações Java nos nós remotos e estas buscam no servidor as configurações da rede. Após a execução do *script*, o SAMCRA entra em modo de espera, aguardando que os nós notifiquem o término de suas execuções. Ao concluir a execução, os nós aguardam o período especificado e notificam o seu término. Após receber todas as notificações, o sistema aguarda o tempo especificado para encerrar a rodada de experimentos.

Ao concluir uma rodada, todos os dados coletados são salvos em nove planilhas eletrônicas e uma nova rodada é iniciada. Ao final da última rodada, o servidor cria um nova planilha com os dados consolidados e encerra a execução.

#### 4. Estado atual do sistema

SAMCRA é usado atualmente para realizar experimentos do protocolo REPI, tanto para avaliar o seu funcionamento quanto para a comparação com os algoritmos de inundação e *Gossip*. Na Figura 3 são apresentados gráficos construídos com dados coletados em experimentos realizados com SAMCRA. Em ambos os casos, os experimentos foram feitos com vinte nós e, para cada uma das configurações, representadas pelos pontos no eixo das abscissas dos gráficos, foram realizadas vinte execuções de forma automática.

O sistema é usado há cerca de quatro meses e já foram executados cinco tipos de experimentos com mais de oitenta tipos de configurações. Até o momento, o uso do sistema têm se mostrado promissor, apresentando grande estabilidade em testes de longa duração, que demoram de algumas horas até alguns dias para sua conclusão.



**Figura 3. Imagem que mostra dois gráficos criados com os dados coletados pelo sistema SAMCRA para a avaliação do protocolo REPI .**

A utilização de SAMCRA com outras aplicações é simples, bastando que o desenvolvedor importe uma classe em seu código e utilize os três métodos que ela oferece, um para a inserção das variáveis e tabelas com os dados do nó, outro para inserção de entradas de rastreamento de mensagens e um para notificar o servidor o fim da execução em um nó. Este procedimento e os métodos são descritos em detalhes na documentação de SAMCRA, inclusive com uma aplicação de exemplo.

A documentação e o sistema podem ser obtidos no seguinte URL:

<http://www.lcp.coppe.ufrj.br/samcra/>

#### 5. Trabalhos Futuros

Apesar de já estar sendo utilizado em experimentos, o SAMCRA ainda é um trabalho em desenvolvimento. Durante a realização de experimentos e a discussão dos seus resultados é comum que surjam sugestões de novas funcionalidades e, durante os quatro meses de uso, ela já sofreu algumas alterações, para aumentar sua flexibilidade e desempenho.

As principais melhorias que identificamos como oportunidades de trabalhos futuros para o sistema SAMCRA são:

- Criação de um módulo para execução de experimentos em lote, para que diversas configurações possam ser testadas várias vezes com apenas um comando;
- Criar mecanismos para adição de novos protocolos de forma dinâmica;
- Aprimorar a configuração dos experimentos, permitindo que a configuração do intervalo e quantidade de mensagens enviadas seja definida para cada nó;
- Criar um módulo para gerar os grafos da rede automaticamente.

## 6. Descrição da Demonstração

A demonstração do sistema será feita com a construção de uma Rede Ad Hoc sem fio formada por dez dispositivos de comunicação Tmote Sky, ligados a pelo menos três máquinas. A rede executará experimentos com pelo menos duas configurações diferentes, demonstrando como o sistema controla os nós remotamente, iniciando e finalizando as aplicações, e como é feita a coleta e consolidação dos dados.

Além disso, também será ilustrada a situação em que usuários reais utilizam a rede e esta é monitorada pelo SAMCRA, gerando o rastreamento dos metadados de todas as mensagens trocadas.

## Agradecimentos

Agradecemos o apoio da Capes, CNPq e FINEP. Rodrigo S. Granja agradece à Petrobras pela liberação parcial de suas atividades profissionais para desenvolver o seu mestrado.

## Referências

- Cavilla, A., Baron, G., Hart, T., Litty, L., and de Lara, E. (2004). Simplified simulation models for indoor manet evaluation are not robust. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 610 – 620.
- Corson, S. and Macker, J. (1999). Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. RFC 2501.
- Dutra, R. C. and Amorim, C. L. (2010). Modelo de comunicação endereçada por interesses. Technical report, Relatório Técnico ES 733 / PESC - COPPE - UFRJ.
- Kini, A. V., Veeraraghavan, V., Singhal, N., and Weber, S. (2006). Smartgossip: an improved randomized broadcast protocol for sensor networks. In *IPSN '06: Proceedings of the 5th international conference on Information processing in sensor networks*, pages 210–217, New York, NY, USA. ACM.
- Moteiv (2006). *Tmote Sky Datasheet* <http://sentilla.com/files/pdf/eol/tmote-sky-datasheet.pdf>. Acessado em março de 2010.
- Novatnack, J., Greenwald, L., and Arora, H. (2005). Evaluating ad hoc routing protocols with respect to quality of service. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, volume 3, pages 205 – 212 Vol. 3.
- Number, M., Goulding, A., Spacey, R., and England, L. (2002). Java remote method invocation. In *68th IFLA Council and General Conference*, page 10.
- Viterbo, J., Sacramento, V., Rocha, R., and Endler, M. (2006). Moca: Uma arquitetura para o desenvolvimento de aplicações sensíveis ao contexto para dispositivos móveis. In *Salão de Ferramentas, Brazilian Symposium on Computer Networks and Distributed Systems (SBRC 2006)*, Curitiba-PR, Brazil.
- Werner-Allen, G., Swieskowski, P., and Welsh, M. (2005). Motelab: a wireless sensor network testbed. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 68, Piscataway, NJ, USA. IEEE Press.