

CeuNaTerra: testbed para espaços inteligentes

Silvana Rossetto¹, Bruno Silvestre², Noemi Rodriguez³,
Adriano Branco³, Leonardo Kaplan³, Ian Lopes³, Miguel Boing³
Douglas Santana², Vinicius Lima²,
Raul Gabrich¹ e Denis Aoki¹

¹Departamento de Ciência da Computação (DCC), Instituto de Matemática (IM)
Universidade Federal do Rio de Janeiro (UFRJ)

²Instituto de Informática
Universidade Federal de Goiás (UFG)

³Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)

silvana@dcc.ufrj.br

Resumo. Neste trabalho, desenvolvemos um ambiente de experimentação físico (testbed) para aplicações de Redes de Sensores sem Fio (RSSF) e Internet of Things (IoT) o qual permite experimentar aplicações de sensoriamento nas quais parte da aplicação pode executar em plataformas de sensores e outra parte pode executar em nós remotos gerenciados pelo usuário em outras sub-redes conectadas via Internet. A arquitetura funcional do testbed divide-se em duas partes: (i) a interface com o usuário via portal Web, para configurar e monitorar os experimentos; e (ii) a gerência de execução dos experimentos, para disparar as ações necessárias para a rede de sensores física. Além de permitir a realização de experimentos nas áreas de RSSF e IoT, esse ambiente também poderá ser utilizado como laboratório para atividades práticas no ensino de conteúdos das áreas de Sistemas Distribuídos e Redes de Computadores.

Abstract. In this work, we developed a testbed for Wireless Sensor Network (WSN) and Internet of Things (IoT) applications to evaluate sensing applications in which part of the application run on sensor platforms and other party run on remote nodes managed by the user on other subnets connected via Internet. The functional architecture of the testbed is divided into two parts: (i) the user interface (by using a web portal) to configure and monitor the experiments; and (ii) the execution manager to trigger actions necessary for the physical sensors. Besides allowing to carry out experiments in the areas of WSN and IoT, this environment may also be used for educational activities as a laboratory for practical activities in the disciplines of Distributed Systems and Computer Networks.

1. Introdução

A crescente demanda pela construção dos chamados “espaços inteligentes” tem fomentado a investigação e o uso combinado de tecnologias de sensoriamento e comunicação sem fio acopladas a dispositivos de tamanho reduzido, com capacidade de processamento e armazenamento de dados, e fonte de energia independente. A minimização desses

dispositivos, dotados de poder computacional e sensorial, tem como finalidade principal possibilitar diferentes ângulos de visão sobre um mesmo evento (ex., a variação de temperatura em uma floresta, ou a frequência de vibração em pontos distintos de uma estrutura civil), permitindo observar ambientes físicos ou monitorar equipamentos com elevado grau de precisão.

O desenvolvimento de aplicações baseadas nessas tecnologias requerem a construção de redes específicas, chamadas Redes de Sensores sem Fio (RSSF), formadas por dezenas, centenas ou até milhares desses pequenos dispositivos, e a definição de soluções de software particulares para lidar com suas principais características e restrições, entre elas, interação direta com o mundo físico, ambiente de computação inerentemente distribuído, fonte de energia e recursos computacionais limitados [Yick et al. 2008]. Em cenários mais complexos, é necessário ainda o desenvolvimento de mecanismos de colaboração para permitir que os dados coletados por diferentes sensores possam ser associados a outras informações de contexto, permitindo atuar sobre o ambiente em tempo real (ex., dado o nível de umidade do solo e a previsão meteorológica atualizada, controlar dinamicamente a vazão do equipamento de irrigação). Para isso, torna-se fundamental a integração das RSSFs isoladas com outras redes, em particular a Internet, possibilitando o uso dos dados coletados no mundo físico por aplicações remotas com as mais diversas finalidades, por exemplo, o controle automatizado de ambientes de produção ou o monitoramento remoto de reservas ambientais.

A tarefa de desenvolvimento de aplicações para o sensoreamento de grandezas físicas e o processamento desses dados considerando as possibilidades de cooperação com outras aplicações é, portanto, intrinsecamente complexa. Envolve o uso de plataformas de hardware heterogêneas; o projeto, implementação e teste de algoritmos distribuídos com características específicas para cada aplicação; a integração entre diferentes meios físicos e protocolos de comunicação em rede; a necessidade de minimizar o consumo de energia dos nós sensores e desenvolver soluções de software eficientes para executar em plataformas com baixa capacidade de processamento, armazenamento e de transmissão de dados; e a capacidade de avaliar cenários em larga escala e sob condições ambientais dinâmicas.

A estratégia mais comum adotada pelos grupos de pesquisa nessa área consiste na adoção de ferramentas de simulação ou emulação de RSSF — em particular para os experimentos de larga escala — e o uso de *testbeds* físicos para avaliar situações mais próximas dos cenários reais de execução, permitindo, por exemplo, aferir a taxa real de perdas de mensagens e consumo de energia dos dispositivos. Há diferentes ferramentas de simulação e emulação de RSSF disponíveis [Polley et al. 2004, Titzer et al. 2005, Levis et al. 2003], assim como *testbeds* físicos de domínio público [Crepaldi et al. 2007, Chatzigiannakis et al. 2009, Handziski et al. 2006]. Há ainda propostas mais recentes de construção de ambientes de experimentação virtualizados que integram nós físicos com nós simulados e emulados, com o objetivo de facilitar a realização de experimentos considerando diferentes cenários de execução [Coulson et al. 2012]. Entretanto, todos esses ambientes têm como foco a experimentação de aplicações voltadas especificamente para a execução em uma RSSF, i.e., não visam aplicações que requerem explicitamente a integração de uma (ou mais) RSSF isolada com outras redes conectadas à Internet, por exemplo, para permitir o acesso remoto aos dados coletados por diferentes sensores, por

aplicações com diferentes finalidades.

Neste trabalho, propomos um ambiente de experimentação alternativo — denominado *CeuNaTerra*. Trata-se de um testbed para educação e pesquisa em aplicações híbridas distribuídas, isto é, aplicações que combinem máquinas convencionais com dispositivos embutidos e de monitoramento, como nós de uma RSSF. Além de permitir a realização de experimentos nas áreas de RSSF e IoT, o testbed *CeuNaTerra* também poderá ser utilizado como laboratório para atividades práticas no ensino de conteúdos das áreas de Sistemas Distribuídos e Redes de Computadores.

O testbed *CeuNaTerra* está sendo construído pelo GT-TeI (*Grupo de Trabalho - Testbed para Espaços Inteligentes*), financiado pela RNP. O testbed dispõe hoje de 14 nós da plataforma MicaZ e 4 nós da plataforma TelosB, e em breve irá incorporar, entre outros, dispositivos Arduino e RaspberryPi. O portal do testbed permite o agendamento de experimentos e a carga remota de programas arbitrários nos diferentes nós. O programador pode incorporar relatórios (logs) de execução dentro das suas aplicações e monitorar a execução do experimento através do portal do testbed, ou conectando-se diretamente às portas de comunicação serial ligadas a cada nó da rede. O site do projeto encontra-se neste endereço: <http://ceunaterra.voip.ufrj.br/Site/>. A partir desse site é possível acessar o testbed, ou diretamente através do seguinte endereço: <http://ceunaterra.voip.ufrj.br/TBportal/>. Uma conta de usuário foi criada para permitir o acesso experimental ao testbed: *user1@testbed.com.br*. A senha é *user*.

O restante deste texto está organizado da seguinte forma. Na seção 2, apresentaremos uma visão geral da arquitetura do testbed proposto e na seção 3 descreveremos a demonstração planejada para o salão de ferramentas.

2. Visão geral da arquitetura do testbed *CeuNaTerra*

O testbed *CeuNaTerra* é voltado para ambientes fechados, sendo constituído por nós sensores independentes e de tipos diferentes, dispostos em uma topologia que pode requerer comunicação multi-saltos, acoplados a um board para carga de código e transferência de dados de execução, com fonte de energia constante. O acesso ao testbed *CeuNaTerra* é feito de forma remota, via portal Web, sem a necessidade de instalação de nenhum software ou hardware especial. A Figura 1 mostra uma visão geral do testbed.

Vários usuários podem acessar o portal Web ao mesmo tempo para agendar seus experimentos, mas apenas um usuário pode usar a rede de sensores para executar o seu experimento em um determinado intervalo de tempo (respeitando a agenda programada via portal). Essa restrição se faz necessária para evitar interferências entre os experimentos uma vez que a comunicação entre as plataformas de sensores é sem fio. A aplicação do usuário poderá ser desenvolvida usando ambientes de programação de sua preferência. Essas aplicações poderão ser híbridas, i.e., aplicações que executarão em parte no testbed e em parte em outros dispositivos conectados via protocolos IP.

A Figura 2 apresenta uma visão geral da arquitetura do testbed, destacando seus elementos principais denominados por *Kit Nó* e *Kit Servidor*. O *Kit Nó* contém os componentes necessários para o funcionamento completo de um nó da rede do testbed: interfaces externas de rádio, sensores físicos, núcleo de processamento, interface de controle e fonte

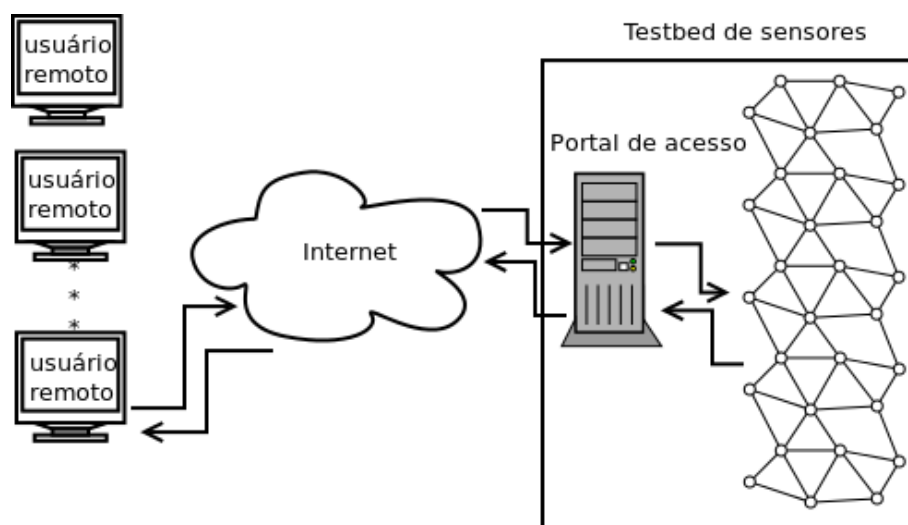


Figure 1. Visão geral do testbed CeuNaTerra.

de alimentação de energia. Cada Kit Nó está ligado, via cabo de rede, a um equipamento de rede (switch) conectado ao Kit Servidor. Essa ligação é usada para carregar o código executável das aplicações nos nós sensores (direção Kit Servidor para Kit Nó) e para transferir as informações de log de execução das aplicações para os usuários remotos (direção Kit Nó para Kit Servidor). O *Kit Servidor* representa o conjunto de equipamentos (servidores e/ou desktops) utilizados no controle central e acesso remoto ao testbed. Este kit é responsável pelas funcionalidades de gerência do testbed.

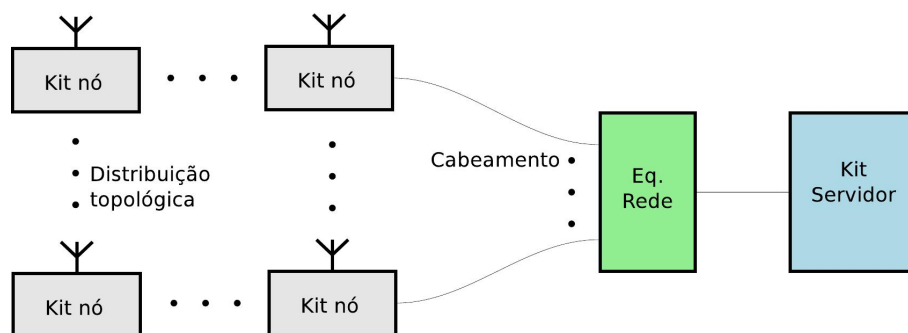


Figure 2. Arquitetura geral do testbed CeuNaTerra.

A arquitetura funcional do testbed divide-se em duas partes: (i) a parte de interface com o usuário, para configuração e monitoração dos testes, é atendida por uma solução de portal Web; (ii) a parte responsável pela execução dos testes é atendida por um processo contínuo que monitora quando é necessário executar um teste configurado e dispara as ações necessárias para a rede de sensores física do testbed. Essas duas partes são integradas via um banco de dados que armazena as configurações dos testes dos usuários. A Figura 3 apresenta a arquitetura funcional do testbed.

O testbed é constituído por três elementos principais: (i) a rede de sensores sem fio, (ii) o controle geral, e (iii) a interface de operação. A *rede de sensores sem fio* é formada pelo conjunto de dispositivos (kits Nó) distribuídos fisicamente por uma área que forma o testbed. A comunicação entre os nós é feita via rádio e o alcance entre cada par

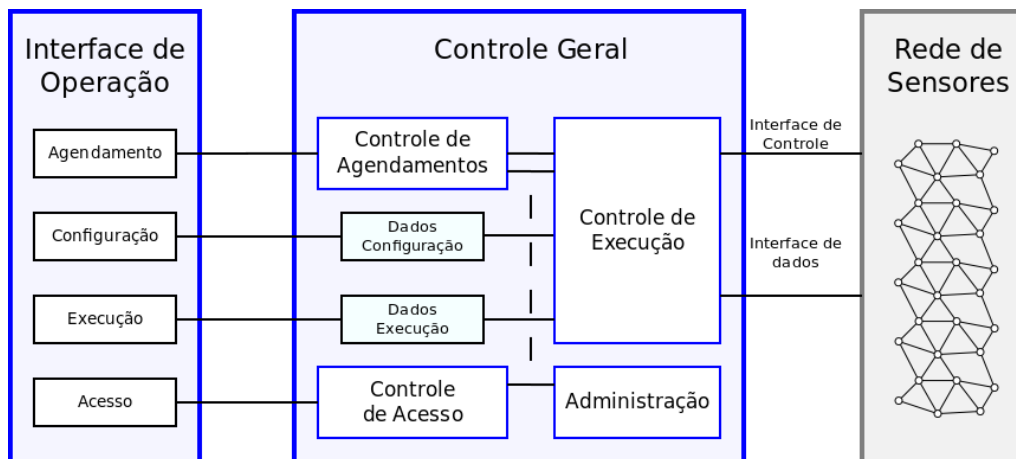


Figure 3. Arquitetura funcional do testbed CeuNaTerra.

de nós da rede depende da distância entre eles e da potência do sinal de rádio. O *controle geral* é composto por módulos básicos que controlam a execução dos experimentos no testbed. A *interface de operação* é composta por quatro funcionalidades: Agendamento, Configuração, Execução e Controle de Acesso. A Interface de Agendamento permite ao usuário consultar, reservar e cancelar as janelas de execução no testbed. Na Interface de Configuração, o usuário pode configurar a topologia da rede com a ativação e desativação de determinados nós, armazenar o seu código executável para posterior execução na rede de sensores, e definir condições de execução específicas para o seu experimento. A Interface de Execução permite ao usuário controlar e monitorar o andamento dos testes. Finalmente, a Interface de Acesso controla o acesso do usuário aos recursos do testbed.

3. Demonstração para o salão de ferramentas

Para demonstrar a utilização do testbed CeuNaTerra, apresentaremos duas aplicações que poderão ser executadas durante o evento. A primeira aplicação será uma aplicação convencional para redes de sensores sem fio, com a coleta e roteamento de dados até uma estação base. Nesse caso, exploraremos as várias facilidades oferecidas pelo testbed CeuNaTerra, como por exemplo, configurar experimentos dinâmicos, onde nós da rede entram e saem da rede em tempo de execução. A segunda aplicação simulará um cenário simples de “casa inteligente” onde um usuário pode visualizar e alterar remotamente o estado dos equipamentos de uma casa. A finalidade dessa aplicação será mostrar como o testbed permite avaliar a execução de aplicações que requerem a integração, via protocolos IP, entre dispositivos de sensoriamento e outros dispositivos conectados à Internet.

3.1. Aplicação 1: coleta e disseminação de dados dos sensores

Uma aplicação convencional em redes de sensores é a coleta e disseminação dos dados sensorizados até uma estação base, como ilustrado na Figura 4. Usando uma aplicação com essa finalidade, demonstraremos as principais facilidades que o testbed CeuNaTerra pode oferecer para avaliar o comportamento da aplicação durante a sua execução.

Na implementação da aplicação, mostraremos as alternativas que o programador pode utilizar para gerar relatórios (logs) de execução da sua aplicação. Ilustraremos como coletar medidas, como perdas de mensagens, número de mensagens transmitidas e rece-

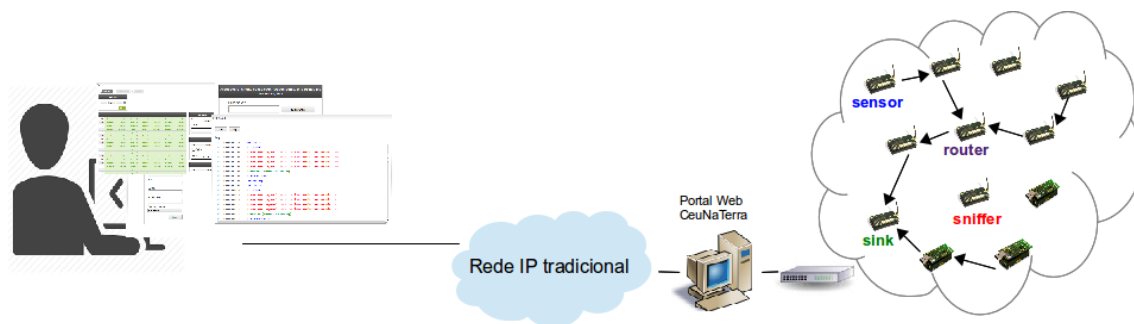


Figure 4. Visão geral da aplicação de coleta e disseminação de dados.

bidas em cada nó, e metadados das mensagens observadas na rede por um nó especial, denominado nó *sniffer*.

Na configuração do experimento, mostraremos como planejar a ativação e desativação de nós da rede durante a execução da aplicação, permitindo simular situações reais em que os nós sofrem algum dano definitivo, perdem contato momentaneamente ou simplesmente entram em modo de economia de energia. Durante a execução do experimento, apresentaremos as diferentes alternativas de monitoramento da aplicação, incluindo a conexão direta com a interface serial de cada nó da rede.

Na máquina do usuário, executaremos aplicações Java que receberão e exibirão os logs das aplicações que executarão nos nós comuns e no nó *sniffer*.

3.2. Aplicação 2: casa inteligente

No conceito de *casas inteligentes*, os equipamentos como lâmpada, TV, fogão, etc., podem se comunicar e com isso podemos remotamente controlar suas funções. Nesta aplicação consideramos cada um dos motes (nós da rede) como sendo o controlador de um desses equipamentos. De acordo com o tipo de dispositivo que o mote esteja controlando, ele será associado a um conjunto de operações que poderá realizar (e.g., acender uma lâmpada) e um conjunto de informações que poderá prover (e.g., volume atual da TV).

Para interagir com cada um desses equipamentos, construímos uma interface em Java, mostrada na Figura 5. Temos à direita da interface um mapa da casa, onde podemos encontrar a localização de cada um dos equipamentos. Ao clicar em um deles, o painel será modificado, apresentando uma interface de acordo com as operações que o equipamento selecionado poderá realizar.

A comunicação entre a interface Java e os equipamentos será realizada por meio de pacotes UDP. Cada um dos equipamentos foi programado com TinyOS e possui um endereço IPv6 conhecido pela interface. O modelo de interação empregado será *Request/Response*, ou seja, os equipamentos sempre responderão aos comandos da interface, mas não iniciarão nenhuma comunicação.

Quando desenvolvemos uma aplicação IoT (aplicação que usa a pilha de protocolos IPv6) com TinyOS, temos um nó particular da RSSF, denominado *sorvedouro*, ligado a um computador por meio de um cabo USB. Juntos, o computador e o nó *sorvedouro*, são responsáveis por fazer a conexão entre a rede cabeada e a rede de sensores sem fio.

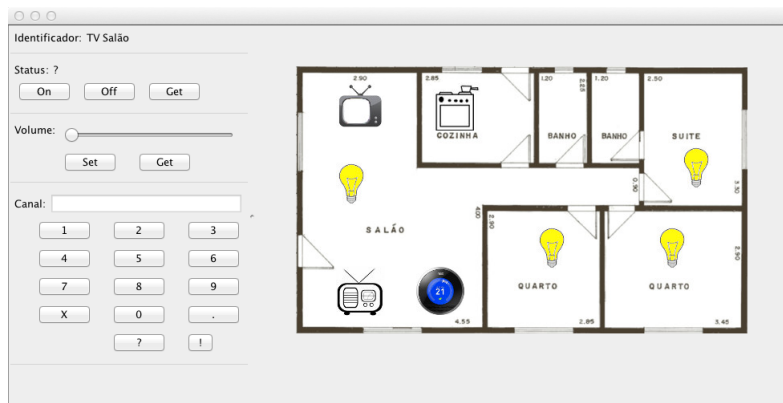


Figure 5. Interface Java para a interação com os equipamentos.

Para realizar a transmissão de dados entre o sorvedouro e o computador, utilizamos o protocolo PPP (*Point-to-Point Protocol*). No nó sorvedouro instalamos uma aplicação chamada PPPRouter (distribuída com o TinyOS). No computador, utilizamos o sistema Linux que já possui uma ferramenta de linha comando chamada `ppp`. Essa ferramenta cria uma interface de rede virtual, com isso, aplicações no computador conseguem se comunicar via IPv6 com os motes.

Para que essa mesma ideia funcione com o testbed, permitindo a interação com aplicações IPv6 que executam na máquina do usuário, projetamos a solução ilustrada na Figura 6. Nessa solução, a ligação do nó sorvedouro com o computador não se dá mais pelo cabo USB. Neste caso, utilizamos a capacidade oferecida pelo testbed CeuNaTerra de exportar a porta serial dos nós para uma porta TCP. Dessa forma, qualquer informação enviada para uma determinada porta TCP no servidor do testbed (aquela que está ligada à entrada serial do sorvedouro) será redirecionada para a entrada serial do sorvedouro, e qualquer dado enviado pelo nó sorvedouro para a entrada serial será redirecionado para a porta TCP.

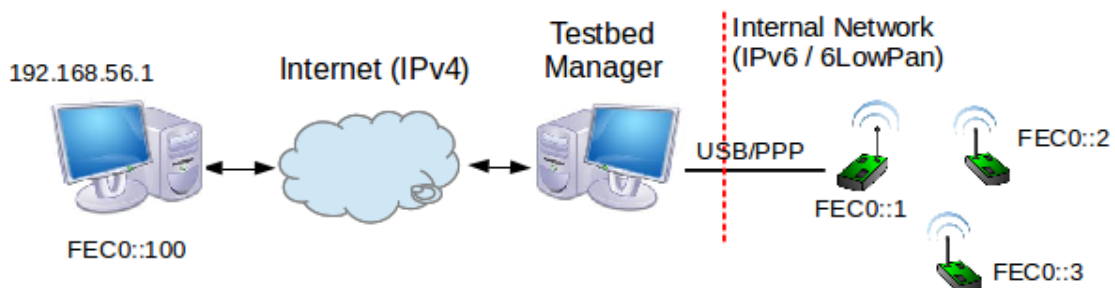


Figure 6. Arquitetura de uma aplicação IPv6 para execução no testbed CeuNaTerra.

A partir de uma máquina remota Linux, o usuário do testbed pode usar a ferramenta `ppp` para estabelecer um canal como descrito anteriormente, mas utilizando comunicação TCP. Dessa forma, a interface de rede virtual será criada, e os dados serão redirecionados para a porta TCP do portal do testbed, que por sua vez a redirecionará para o nó sorvedouro. Através do tunel PPP, embora a máquina do usuário esteja em uma localização remota, é como se ela estivesse conectada diretamente ao nó sorvedouro da rede de sensores. Sendo assim, podemos configurar um endereço IPv6 para a interface

de rede com o mesmo prefixo da rede de sensores, e a comunicação entre o computador remoto e o nó sorvedouro fica transparente, como ilustrado na Figura 6.

Para testar a aplicação Java descrita, criamos um experimento no portal que instalará nos motes do testbed algumas aplicações feitas em TinyOS que simulam os equipamentos da casa inteligente. Assim que o experimento estiver rodando, configuraremos a interface de rede virtual no computador remoto. Por fim, executaremos a aplicação Java. Toda a infraestrutura de comunicação ficará transparente para a aplicação Java e para os motes.

References

- Chatzigiannakis, I., Fischer, S., Koninis, C., Mylonas, G., and Pfisterer, D. (2009). Wisebed: an open large-scale wireless sensor network testbed. In *1st International Conference on Sensor Networks Applications, Experimentation and Logistics (SEN-SAPPEAL 2009)*, volume 29 of *Lecture Notes of the Institute for Computer Sciences, Social-Inf*, pages 68–87, ICST. Springer-Verlag.
- Coulson, G., Porter, B., Chatzigiannakis, I., Koninis, C., Fischer, S., Pfisterer, D., Bimschas, D., Braun, T., Hurni, P., Anwander, M., Wagenknecht, G., Fekete, S. P., Kröller, A., and Baumgartner, T. (2012). Flexible experimentation in wireless sensor networks. *Commun. ACM*, 55(1):82–90.
- Crepaldi, R., Friso, S., Harris, A., Mastrogiovanni, M., Petrioli, C., Rossi, M., Zanella, A., and Zorzi, M. (2007). The design, deployment, and analysis of signetlab: A sensor network testbed and interactive management tool. In *Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on*, pages 1–10.
- Handziski, V., Köpke, A., Willig, A., and Wolisz, A. (2006). Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, REALMAN '06, pages 63–70, New York, NY, USA. ACM.
- Levis, P., Lee, N., Welsh, M., and Culler, D. (2003). TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems*, pages 126–137. ACM Press.
- Polley, J., Blazakis, D., McGee, J., Rusk, D., and Baras, J. (2004). ATEMU: a fine-grained sensor network simulator. In *Sensor and Ad Hoc Communications and Networks*, pages 145–152. <http://www.isr.umd.edu/CSHCN/research/atemu/>.
- Titizer, B., Lee, D., and Palsberg, J. (2005). Avrora: Scalable sensor network simulation with precise timing. In *Fourth International Conference on Information Processing in Sensor Networks*, pages 477–482. <http://compilers.cs.ucla.edu/avrora/>.
- Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52:2292–2330.