



# Middleware Baseado em Serviços para Redes de Sensores sem Fio

Flávia Delicato<sup>3</sup>; Orientadores: Luci Pirmez<sup>1</sup> e José Ferreira de Rezende<sup>2</sup>

NCE<sup>1</sup>, GTA<sup>2</sup> – Universidade Federal do Rio de Janeiro – DIMAp – Universidade Federal do Rio Grande do Norte<sup>3</sup>

{ fdelicato, luci } @ nce.ufrj.br, rezende@gta.ufrj.br

**Abstract.** *We propose a service-oriented middleware for Wireless Sensor Networks (WSNs) which supplies an abstraction layer between applications and the network infrastructure and provides services specifically tailored to meet the requirements of WSN applications and developers. The middleware design was leveraged by the Web Services area, and it adopts correlated technologies, as SOAP protocol and XML language. The middleware takes low-level decisions on behalf the application and it provides adaptation and context awareness capabilities, to deal with the high dynamics inherent to WSNs.*

**Resumo.** *Nós propomos um sistema de middleware baseado em serviços para redes de sensores sem fio (RSSFs). O middleware fornece uma camada de abstração entre aplicações e a infra-estrutura de rede subjacente e serviços especialmente designados para atender às necessidades das aplicações e dos desenvolvedores de RSSFs. O middleware provê uma interface padrão para o acesso aos serviços, baseada em tecnologias da área de Serviços Web, como o protocolo SOAP e a linguagem XML. O middleware toma decisões de baixo nível para a aplicação e fornece capacidades de adaptação e ciência de contexto, a fim de lidar com o alto dinamismo das RSSFs.*

## 1. Introdução

Redes de Sensores sem Fio (RSSFs) são compostas por dezenas a milhares de dispositivos de baixo custo e tamanho reduzido, capazes de realizar sensoriamento, processamento e transmissão de informação através de enlaces sem fio. Os sensores atuam de forma colaborativa, extraindo dados ambientais e transmitindo-os para um ou mais pontos de saída da rede, chamados sorvedouros, para serem analisados e processados. Sensores são em geral movidos a baterias não recarregáveis e, devido à sua enorme quantidade e sua instalação em locais de difícil acesso, devem operar sem assistência humana por longos períodos. Portanto, é importante manter os sensores “vivos” o maior tempo possível. Economia de energia torna-se, então, uma questão crucial em RSSFs. Todas as etapas de projeto e funcionamento dessas redes devem levar em conta o consumo de energia e procurar otimizá-lo.

Em geral, vários sensores monitoram o mesmo fenômeno, gerando dados redundantes.. Como a maior fonte de consumo de energia nos sensores é a transmissão de dados, muitos trabalhos propõem soluções para obter e rotear os dados de forma eficiente em energia, a fim de estender o tempo de vida da rede. Por um lado, há propostas cujo enfoque é minimizar o número e o tamanho das mensagens, realizando o roteamento eficiente em energia [Intanagonwiwat, Govindan and Estrin 2000]. Por outro lado, pesquisas recentes [Perillo and Heinzelman 2003] mostram que a alta densidade de nós pode ser aproveitada para obter economia de energia, através (i) do ajuste da taxa de dados; (ii) da alteração do ciclo de operação dos sensores; (iii) do desligamento de sensores redundantes por determinados períodos de tempo; ou (iv) do emprego de técnicas de agregação de dados. Obviamente, há uma negociação entre o consumo de energia na rede e a qualidade global do dado entregue para a aplicação, quando da escolha do método a adotar. Qualquer abordagem adotada deve ser balanceada com os requisitos de qualidade requeridos pela aplicação.

Além dos recursos escassos, outra característica das RSSFs é que sua estrutura topológica é altamente dinâmica. Sensores podem se deslocar, ser retirados da rede ao terem sua bateria esgotada, novos sensores podem ser inseridos ou podem ser temporariamente desligados para economizar energia, alterando assim a topologia ativa da rede. Adicionalmente, tanto as condições do ambiente no qual está instalada a rede quanto os interesses da aplicação podem mudar. Para prover suporte a tal grau de dinamismo, RSSFs devem possuir alguma capacidade de adaptação. Qualquer solução que busque otimizar o uso dos recursos da rede deve adotar mecanismos que permitam a adaptação dinâmica às possíveis variações de contexto. Quanto às aplicações para RSSFs, uma ampla gama é mencionada na literatura como aplicações militares, médicas, monitoramento ambiental, e muitas outras. Embora se estendam por um amplo espectro, a maioria das aplicações caracterizam-se por requerer alguma forma de cooperação para atingir sua meta e demandarem um nível mínimo de qualidade de serviço (QoS). Entretanto, apesar de características comuns, os requisitos específicos de QoS e de outros aspectos, como o modelo de entrega de dados solicitado, são altamente dependentes de cada aplicação. Trabalhos destacam a importância da participação da aplicação no processo de comunicação em RSSFs [Tilak, Abu-Ghazaleh and Heinzelman 2002]. Otimizações específicas da aplicação podem reduzir o número de transmissões e, por conseguinte, o consumo de energia da rede.

Vários protocolos específicos para roteamento de dados em RSSFs foram propostos nos últimos anos [Intanagonwiwat, Govindan and Estrin 2000]. Como há uma extensa gama de aplicações, é improvável que um único protocolo atenda a todas elas. Uma questão importante, então, passa a ser a seleção do protocolo mais adequado a cada aplicação. Uma vez decidido o protocolo mais adequado, são necessários mecanismos para seletivamente ativar os nós sensores mais apropriados para atender a essa aplicação. Em computação distribuída, o uso de sistemas de middleware facilita o trabalho de desenvolvedores de aplicações, liberando-os de lidar com a complexidade gerada pela distribuição. RSSFs também poderiam se beneficiar do uso de uma camada de middleware que fornecesse um ambiente de execução genérico para as aplicações, provendo uma abstração das funcionalidades de infraestrutura da rede. Entre outras funções, o middleware pode decidir o melhor protocolo a ser usado de acordo com os requisitos da aplicação, coordenar a operação dos sensores na realização do objetivo dessa aplicação e gerenciar o uso dos recursos da rede.

Tecnologias de middleware convencionais não são adequadas para RSSFs pois exigem muito recurso computacional e são construídas aderindo à metáfora da “caixa preta”, onde a complexidade da distribuição é escondida dos usuários e desenvolvedores. Para obter um melhor desempenho em RSSFs, deve-se permitir que a aplicação afete o comportamento dos protocolos usados e dos sensores. Entretanto, é indesejável que desenvolvedores de aplicações tenham conhecimento detalhado da infraestrutura da rede, ou que uma aplicação fique presa em tempo de projeto a uma infra-estrutura específica. Portanto, o middleware deve atuar como intermediário, traduzindo requisitos das aplicações em parâmetros de configuração da RSSF. Ao mesmo tempo, deve fornecer mecanismos que permitam à aplicação monitorar o contexto de execução atual através de uma interface de alto nível e, a partir do monitoramento ser capaz de adaptar dinamicamente algum comportamento da rede, se necessário. Portanto, sistemas de middleware para RSSFs devem fornecer ciência de contexto. Princípios de middleware reflexivo [Capra, Emmerich and Mascolo 2003] podem ser adotados no projeto da RSSF para atender as necessidades de suas aplicações e para lidar com o dinamismo do seu ambiente.

Nós propomos um middleware para redes de sensores sem fio que adota uma abordagem de serviços, onde a RSSF é vista como uma fornecedora de serviços para aplicações clientes. A idéia do middleware foi introduzida em [Delicato et al. 2005d]. A abordagem foi inspirada na área de Serviços Web e o projeto do sistema faz uso de tecnologias que fazem parte da especificação de Serviços Web, como o protocolo SOAP e a linguagem XML. Usando tal abordagem, o sistema proposto fornece uma camada de abstração entre aplicações e a infra-estrutura de rede, facilitando a integração de diferentes aplicações com a RSSF. Além de fornecer um modelo de programação abstrato para aplicações de

RSSFs, o middleware procura balancear os requisitos de QoS das aplicações e o tempo de vida da rede, que é diretamente relacionado ao consumo de energia. O middleware toma decisões quanto aos protocolos de comunicação a serem adotados, os parâmetros de configuração de tais protocolos, a organização topológica da rede, o escalonamento e os modos de operação dos sensores para a execução das tarefas de sensoriamento, entre outras. O middleware monitora os estados de execução da rede e das aplicações, realizando procedimentos de adaptação quando necessário, com ou sem interferência da aplicação. Tais funcionalidades são fornecidas como serviços. Como o middleware implementa e, conseqüentemente, esconde dos desenvolvedores de aplicações a maior parte dos detalhes das decisões de baixo nível, ele minimiza os esforços de desenvolvimento de aplicações para RSSFs.

Em suma, os principais benefícios do middleware proposto são: (i) oferecer uma camada de interoperabilidade entre diferentes aplicações e a rede de sensores; (ii) criar a abstração de uma rede genérica e otimizável, que fornece serviços acessados através de uma interface padrão e de alto nível; (iii) fornecer um mecanismo para selecionar o protocolo de comunicação mais adequado e a configuração dos seus parâmetros, baseado nos requisitos da aplicação; (iv) realizar a seleção dos nós a serem ativados para a execução de uma tarefa de sensoriamento; (v) gerenciar o uso dos recursos da rede durante a execução da tarefa e adaptar dinamicamente o funcionamento da rede de acordo com as variações do contexto de execução; e (vi) permitir à aplicação inspecionar o estado da rede e participar do processo de adaptação. Os benefícios do middleware, bem como a viabilidade de sua implantação em redes de sensores reais foram avaliados através da realização de simulações e da implementação de um protótipo com várias das funcionalidades apresentadas nesta tese. Os resultados das avaliações foram promissores, demonstrando as vantagens de adotar o middleware na construção de RSSFs.

## 2. Trabalhos Relacionados

Nos últimos anos, surgiram várias propostas de middleware especialmente projetados para RSSFs, com diferentes objetivos e abordagens. Nossa proposta possui características que a distinguem de outras. Primeiro, adotamos uma abordagem baseada em serviços para o projeto de RSSFs, onde todas as interações entre aplicações e a rede são baseadas em uma relação cliente-fornecedor. Em contraste com a abordagem baseada em serviços, há trabalhos que propõem abordagens de bancos de dados [Bonnet, Gehrke and Seshadri 2001] ou modelos de programação dirigidos a eventos [Shen, Srisathapornphat and Jaikao 2001]. A principal vantagem de nossa abordagem é oferecer um modelo de programação genérico e flexível, com grande capacidade de interoperabilidade entre diferentes aplicações e a rede. A segunda característica particular do trabalho é a proposta de uma interface de alto nível para o acesso à rede. Essa interface provê um mecanismo padrão para a representação de dados e a formatação de mensagens trocadas entre aplicações e a rede, em vez de basear-se em formatos ou linguagens proprietárias. O alto grau de extensibilidade da linguagem XML permite que ela seja usada para consultas e para a submissão de tarefas sendo, portanto, mais adequada para uso em RSSFs do que linguagens baseadas em SQL. A terceira característica é que o middleware proposto fornece componentes de serviços configuráveis que permitem adaptar dinamicamente, de forma transparente para a aplicação, a configuração da rede e dos protocolos utilizados aos requisitos específicos de cada aplicação. Apesar de haver outros trabalhos que compartilham esse objetivo, o problema de como receber e interpretar os requisitos da aplicação não foi tratado. Outra característica relevante é o uso de mecanismos para adaptação do comportamento da rede e fornecimento de ciência de contexto. Tal característica também é contemplada por CARISMA [Capra, Emmerich and Mascolo 2003] que, no entanto, foi projetado para redes sem fio genéricas, não abordando requisitos específicos de RSSFs.

## 3. Descrição do Sistema

O middleware fornece para a aplicação uma abstração do serviço de entrega de dados, de modo que o mesmo possa ser configurado de acordo com as suas necessidades. Essa abstração é provida pelo **serviço de comunicação** que, entre outras funções, fornece para a aplicação uma interface de alto nível para

acesso à rede. A descoberta das capacidades de sensoriamento fornecidas pela RSSF é realizada através do **serviço de descoberta**. Para lidar com questões de baixo nível, referentes à infra-estrutura da rede, é fornecido um **serviço de configuração** [Delicato et al. 2004], responsável pela escolha e parametrização dos protocolos a serem usados, e um **serviço de seleção de nós ativos** [Delicato et al. 2005b][Delicato et al.][Delicato et al. 2005c], responsável pela escolha dos sensores que devem ser ativados para realizar uma tarefa de sensoriamento. Uma vez tomadas as decisões de infra-estrutura necessárias, o middleware deve gerenciar a execução das tarefas de sensoriamento e a utilização dos recursos da rede. Para isso, são fornecidos serviços de **gerência de recursos** e **controle de admissão**. A fim de dar suporte ao contexto de execução dinâmico das RSSFs, são providos **serviços de inspeção e adaptação**. Adicionalmente, são providos componentes de **serviços genéricos**, úteis para todas as RSSFs, como segurança e agregação.

As funcionalidades dos serviços do middleware são fornecidas por componentes. O principal componente é o serviço de comunicação e os demais serviços são diretamente conectados a este. Serviços adicionais podem ser incluídos no sistema, desde que se conectem com o serviço de comunicação através das suas interfaces. Tecnologias da área de Serviços Web foram empregadas na especificação e na implementação dos componentes do middleware. Dessa forma, os serviços oferecidos pela rede são expostos como Serviços Web. As interfaces lógicas entre os componentes e com o mundo externo são descritas através de documentos WSDL e de Esquemas XML. As mensagens trocadas entre os componentes externos e internos do sistema são implementadas como mensagens SOAP ou XML.

Os nós sorvedouros, que são dispositivos mais robustos e representam o ponto de acesso à rede, foram projetados com base na especificação da arquitetura de Serviços Web. Portanto, a implementação do serviço de comunicação nesses nós baseia-se no protocolo SOAP. Para evitar o *overhead* imposto pelo SOAP, o serviço de comunicação dos sensores é baseado no uso de mensagens XML formatadas segundo esquemas compactos. Os componentes de serviços do middleware foram implementados como módulos de software. O **Módulo de Comunicação** implementa o serviço de comunicação e inclui um *proxy* SOAP para a comunicação com a aplicação e *drivers* XML para a comunicação com os protocolos de rede e dispositivos. O módulo de descoberta externa **do serviço de descoberta** permite às aplicações descobrirem a localização e o formato de acesso de uma RSSF. O padrão UDDI foi escolhido como protocolo de descoberta de serviços. Aplicações clientes, após localizarem o sorvedouro utilizando o protocolo UDDI, obtêm o documento WSDL que descreve o formato das mensagens para acessar a rede. O **módulo de descoberta externa** é composto pelos documentos WSDL e pelos registros necessários para a publicação do Serviço Web da rede no UDDI. O **serviço de configuração** é implementado pelo **módulo de decisão**. O **serviço de inspeção e adaptação** é implementado como: (i) um **módulo de inspeção**, que permite à aplicação inspecionar o comportamento da rede, fornecendo uma representação do seu estado corrente; e (ii) um **módulo de monitoramento e adaptação**, responsável pelo monitoramento dos estados da rede e da aplicação e pela ativação de políticas de adaptação quando necessário. O **serviço de gerência de recursos e tarefas** é implementado por um **módulo de controle de admissão** e um **módulo de seleção de nós ativos**. Cada **serviço genérico** fornecido é implementado como um módulo separado, que pode ser fornecido por terceiros. Diferentes visões da arquitetura do sistema proposto são apresentadas em [Delicato et al. 2003][Delicato et al.2005a] [Delicato et al.2005e][Delicato et al.2006b].

#### 4. Os Serviços de Decisão e de Seleção de Nós Ativos

O middleware proposto oferece um serviço de decisão para a configuração da RSSF, abrangendo a escolha da topologia lógica da rede e do protocolo de roteamento a serem adotados, e um serviço de gerência de recursos, que consiste em um módulo responsável pela seleção dos sensores que permanecerão ativos para executar uma tarefa solicitada. O núcleo do serviço de decisão consiste em um algoritmo de decisão, cujas entradas são obtidas das aplicações e das bases de dados de sensores. Suas saídas consistem no modelo de entrega de dados, no protocolo de roteamento e na topologia lógica da rede a serem adotados. A versão inicial do algoritmo de decisão, bem como resultados preliminares da

avaliação de seu desempenho, são descritos em [Delicato et al. 2004]. Alguns parâmetros utilizados pelo algoritmo são precisamente definidos pelo usuário ao submeter seus interesses. Outros parâmetros são extremamente subjetivos, ou descritos de forma vaga. Portanto, um modelo baseado em lógica nebulosa foi empregado para aprimorar os resultados do processo de decisão [Delicato et al. 2006a].

Simulações foram realizadas com uma parte do algoritmo de decisão utilizando-se o NS-2, com o intuito de mostrar que a configuração da RSSF segundo os requisitos das aplicações pode aumentar o seu desempenho, em termos de consumo de energia. Adotou-se como métrica de desempenho a **energia média dissipada**, uma medida diretamente relacionada ao tempo de vida da rede. Adotou-se o protocolo de disseminação de dados Difusão Direcionada [Intanagonwiwat, Govindan and Estrin 2000] e foram simulados seus modos de operação *pull* e *push*. O modo *pull* é mais adequado para aplicações onde há um pequeno número de sorvedouros, enquanto o modo *push* é mais adequado para quando há grande número de fontes de dados [Heideman, Silva and Estrin 2003]. Foi, portanto, analisado o comportamento dos protocolos em função das variações do número de sorvedouros e do número de fontes. A variação do número de sorvedouros não mostrou diferença significativa entre os modos *pull* e *push*. Esse resultado pode ser atribuído ao fato de que o *overhead* de mensagens de controle gerado pelo modo *pull* não é significativo para o número máximo de sorvedouros simulados. Como em cenários típicos de instalação em geral há apenas um pequeno número de sorvedouros, não se considerou necessário simular um número maior desses nós. Já com relação a fontes, para números variando de 1 a 7, o modo *push* dissipou menos energia. A partir de 7 fontes, o *pull* passou a apresentar os menores valores de energia dissipada. Esses resultados comprovam o melhor desempenho do modo *pull* para cenários com muitos nós gerando dados e indicam que o número de fontes é um fator importante de decisão. Como resultado das avaliações do algoritmo, ganhos de até 25% da energia média dissipada foram obtidos quando o algoritmo selecionava o modo de operação do DD mais apropriado segundo a aplicação, de uma forma transparente para os desenvolvedores.

Quanto a seleção dos nós ativos, o esquema utilizado pelo middleware visa maximizar o tempo de vida da rede e garantir a QoS solicitada pela aplicação. Duas estratégias foram usadas no algoritmo de seleção: (i) minimizar o consumo de energia da rede escolhendo o menor número possível de nós capazes de fornecer a QoS desejada; e (ii) maximizar a energia residual dos nós escolhidos, consumindo energia de modo uniforme entre eles ao longo de tempo e evitando a morte prematura de nós excessivamente utilizados. Além disso, o algoritmo leva em conta a relevância potencial dos sensores, do ponto de vista da aplicação. O esquema para seleção de nós foi modelado como um Problema da Mochila [Cormen et al 2001]. O objetivo do problema consiste em encontrar uma coleção de objetos, a mais valiosa possível, que respeite a capacidade da mochila. Para a formalização do problema de seleção dos nós ativos como um Problema da Mochila, os objetos considerados são os nós sensores, com seus respectivos pesos e valores. A capacidade da Mochila é dada pelo orçamento de energia, definido como a quantidade máxima de energia que se aceita consumir na rede para a execução da tarefa de sensoriamento solicitada. O peso do nó também é definido em termos de energia, sendo dado pelo custo de energia do nó, caso seja escolhido para participar da tarefa. Para o valor do nó adotou-se uma abordagem que prioriza nós com valores maiores de energia residual e relevância. A relevância potencial de um nó para uma tarefa determina o quanto ele pode contribuir para fornecer informação relevante para a aplicação. Assim, a relevância do nó depende de características físicas e topológicas, a saber: precisão nominal; ruído ambiental da medição; número de nós vizinhos; proximidade da área alvo. De posse dessas variáveis, foi construída uma função objetivo para o problema [Delicato et al. 2005c].

Os benefícios de utilizar o serviço de seleção de nós ativos foram verificados por simulações. Adotou-se uma heurística gulosa para resolver o Problema da Mochila, implementada para executar no sorvedouro, com os dados necessários para sua execução transmitidos pelos nós sensores. Uma aplicação de monitoramento ambiental foi simulada. A tarefa solicitada consistia em monitorar valores de temperatura de uma região alvo durante um período de tempo, dividido em ciclos (o algoritmo de

seleção executava antes do 1º ciclo). A aplicação desejava os dados brutos de temperatura, com os requisitos: (i) resolução espacial mínima de 40m<sup>2</sup> com um grau de cobertura igual a 1; (ii) taxa de aquisição de 1 dado a cada 10s, e (iii) acurácia de dados acima de um limiar definido. A acurácia foi dada pelo valor do erro médio quadrático (MSE), calculado como a diferença entre um conjunto de valores considerados como “reais” e o conjunto de valores gerados pelos sensores. O tempo de vida da rede devia ser longo o suficiente para garantir a coleta de dados durante o período de tempo solicitado pela aplicação (10 ciclos), respeitando os requisitos de QoS.

Na primeira fase de simulações, analisou-se o comportamento em termos de energia residual da rede e acurácia de dados ao se selecionarem diferentes porcentagens de nós. Como resultado, um ganho de mais de 1000% de energia ao final do 10º ciclo de simulação foi obtido quando apenas 30% dos nós são ativados, em contraste com a ativação de 100% dos nós. Observou-se que o MSE para orçamentos de 40 a 100% comportou-se de forma bastante similar até o 8º ciclo. Para o orçamento de 30%, o MSE foi maior desde o 1º ciclo. A partir do 8º ciclo, o MSE começou a aumentar para todos os orçamentos. Tal resultado deveu-se ao fato de, a partir desse ciclo, um número maior de sensores ter sua energia esgotada. A expiração do tempo de vida dos nós fontes ou de nós localizados no caminho entre fontes e sorvedouros impede a entrega de dados, aumentando o erro. Embora o MSE aumente, até o 9º ciclo ele permanece abaixo do ponto tolerado pela aplicação (0.3), para todos os orçamentos, exceto 30%. No 10º ciclo, o MSE está abaixo do limiar tolerado apenas para orçamentos de 80 a 100%, significando que a QoS solicitada não está mais sendo atendida pelos demais orçamentos. Como o tempo de monitoramento solicitado foi de 10 ciclos, os resultados mostram que com 80% de nós a QoS da aplicação pôde ser satisfeita, com uma economia de energia de mais de 200%. Economia de energia adicional foi obtida, porém às custas de não satisfazer os requisitos de acurácia da aplicação. Evidências de que, para orçamentos inferiores a 80%, o erro obtido ultrapassava o limiar estipulado pela aplicação, apesar de haver uma considerável quantidade de nós vivos na rede, levantaram as considerações: (i) executar o algoritmo de seleção apenas no 1º ciclo, onde a energia residual dos nós é muito semelhante (considerando a rede recém instalada), pode não ser uma estratégia eficiente; e (ii) adotar um orçamento fixo em todos os ciclos de uma tarefa pode levar ao uso ineficiente da RSSF ou ao não atendimento dos requisitos da aplicação, apesar de haver recursos disponíveis. Se um valor de erro acima do limiar desejado é detectado, um aumento do orçamento inicial poderia levar a rede a se recuperar da quebra de QoS e atender a aplicação por ciclos adicionais. Por outro lado, quando o erro apresenta valores abaixo do limiar logo nos primeiros ciclos, economia de energia poderia ser obtida diminuindo-se o orçamento inicial. Foram, então, simuladas estratégias de adaptação do middleware a fim de avaliar seu impacto sobre o desempenho da rede. Tais estratégias consistiam em executar novamente o algoritmo de seleção, alterando-se o orçamento da rede, de acordo com o seu estado atual. Nas simulações observou-se que, diminuindo-se o orçamento quando o erro estava abaixo do limiar, economias de até 200% na energia final da rede foram obtidas para orçamentos de 100% a 70%, em comparação com a estratégia de manter fixo o orçamento. Para orçamentos menores do que 70% não houve variação significativa pois, com orçamentos baixos, o erro já fica próximo ao limiar logo nos primeiros ciclos, impedindo que o orçamento seja significativamente diminuído em ciclos posteriores.

Em outra etapa de simulação, explorou-se o fato de que os papéis de fonte e roteador requerem diferentes características dos nós. Nós com potencial para desempenhar o papel de roteador devem possuir maior número de vizinhos (considerando alcance rádio) e uma alta energia residual. Nós com potencial para atuar como fontes devem possuir um menor número de vizinhos (considerando raio de sensoriamento) e uma alta acurácia de dados. Portanto, o algoritmo de seleção foi dividido em duas etapas, onde foram usadas duas funções objetivo diferentes. Na primeira, são selecionados apenas nós para atuar como fontes e na segunda, nós para atuar como roteadores. A função objetivo foi alterada para considerar apenas as características mais relevantes para cada papel. Foram executadas simulações utilizando essa variação do algoritmo em conjunto com o mecanismo adaptativo. Como resultado, o tempo de vida da rede foi estendido para mais de 30 ciclos, respeitados os requisitos de QoS.

Finalmente, para fins de comparação, um esquema de seleção aleatória foi simulado onde, para cada orçamento aloca-se de forma aleatória o número de nós correspondente àquela porcentagem. Leva-se em conta apenas a área alvo solicitada, garantindo que a porcentagem correspondente de nós fontes é selecionada, não se considerando critérios de energia residual ou relevância dos nós. Os resultados mostraram que a energia final da rede foi maior, em todos os orçamentos, quando se adotou o esquema proposto. O MSE foi sempre menor com a abordagem proposta do que com a seleção aleatória. A descrição e análise completas dos resultados do trabalho estão detalhadas em [Delicato et al. 2005b] [Delicato et al.] [Delicato et al. 2005c].

## 5. Protótipo do Sistema

Como prova de conceito para o sistema de middleware proposto, foi construído um protótipo em Java com suas principais funcionalidades. Como as características de hardware e, conseqüentemente, os componentes de software a serem instalados, diferem muito entre sorvedouros e sensores, diferentes plataformas de desenvolvimento foram utilizadas para implementar cada tipo de nó. O nó sorvedouro onde foi executado o protótipo consistiu em um Pentium 4, 1.8GHZ, 1.5GB de RAM e 40GB de HD. As classes com as funcionalidades do sorvedouro e o Serviço Web da rede foram implementados utilizando o ambiente Apache Axis para desenvolvimento de Serviços Web e a plataforma J2SE versão 1.4.2\_01. O Serviço Web foi instalado no servidor de aplicações Apache TomCat. Implementou-se uma aplicação cliente em Java e gerou-se um *proxy* Java para o acesso à rede com a ferramenta WSDL2Java. Já a configuração de hardware para os sensores emulada no trabalho é dos sensores Sensoria WINS NG 2.0 [SENSORIA], dotados de processadores SH-4 com 167 MHz, 32 MBytes de RAM e rodando sistema operacional Linux Familiar. O protótipo nos sensores foi implementado utilizando a plataforma J2ME, com a configuração CLDC (*Connected Limited Device Configuration*) e o perfil MIDP (*Mobile Information Device Profile*). Para a construção do protótipo utilizou-se o *Wireless Toolkit* [WTK]. Implementou-se o módulo de comunicação e um módulo de agregação. O tamanho do arquivo “.jar” (formato de instalação Java) com todas as classes necessárias para um sensor é de 90kBytes, incluindo todas as bibliotecas. A máquina virtual Java utilizada (KVM) ocupa em torno de 80kBytes. Esse tamanho é perfeitamente compatível com os recursos de memória disponíveis nos dispositivos sensores considerados como alvo deste trabalho. Para lidar com a verbosidade da linguagem XML, avaliou-se o uso do formato binário WBXML, que permite a transmissão compacta sem perda de informação semântica. Foram comparados os formatos XML e WBXML, em termos de bytes transmitidos na rede e consumo de memória dos dispositivos. Os resultados mostraram que o formato WBXML diminuiu o consumo de memória nos dispositivos, em cerca de 30%, e reduziu em aproximadamente 70% o tráfego. Portanto, ele representa uma melhor escolha para a representação das mensagens no interior da rede.

## 6. Conclusões

Este trabalho propôs um middleware baseado em serviços para o projeto de RSSFs. Sua características e funcionalidades permitem-no atender ao amplo espectro de requisitos das aplicações de RSSFs, fornecendo QoS e ciência de contexto, bem como satisfazendo a necessidade de otimizar os limitados recursos da rede. Com o modelo de programação definido pelo middleware, a rede de sensores pode ser vista como uma entidade que fornece serviços para vários usuários com necessidades diferentes e dinâmicas. O modelo prevê a incorporação de novas funcionalidades sob a forma de serviços que podem ser implementados por terceiros e adicionados ao sistema. As principais contribuições do sistema proposto podem ser sintetizadas sob dois aspectos. Do ponto de vista de projeto, a proposta oferece um novo paradigma para a arquitetura de redes de RSSFs, baseada na área de Serviços Web, e trazendo todos os benefícios de flexibilidade e interoperabilidade inerentes a essa abordagem. Do ponto de vista de um sistema de middleware, a proposta procura incorporar todas as características e funcionalidades necessárias, tanto para as aplicações clientes, como capacidades de inspeção e ciência de contexto, como para os desenvolvedores dessas aplicações, livrando-os da tarefa de lidar com decisões de infra-estrutura.

## Referências Bibliográficas

- Bonnet, P., Gehrke, J. E. and Seshadri, P. (2001). "Towards Sensor Database Systems". In: Proc. of the 2nd International Conference on Mobile Data Management, Hong Kong, January.
- Capra, L., Emmerich, W. and Mascolo, C. (2003). "CARISMA: Context-Aware Reflective mIddleware System for Mobile Applications". IEEE Transactions on Software Engineering v. 29, n.10, pp. 929-945.
- Cormen, T. H., et al (2001). Introduction to Algorithms, MIT Press.
- Delicato, F. C., et al. (2003). "A Flexible Middleware System for Wireless Sensor Networks". In: Procs. of the ACM/IFIP/USENIX International Middleware Conference, Rio de Janeiro, July.
- Delicato, F. C., et al. (2004). "Middleware Orientado a Serviços para Redes de Sensores sem Fio". In: Anais do 22º Simpósio Brasileiro de Redes de Computadores, RS, Brasil, Maio.
- Delicato, F. C., et al. (2005a). "Reflective Middleware for Wireless Sensor Networks". In: Procs. of the 20th ACM Symposium on Applied Computing (SAC2005), pp. 1155-1159, USA, March.
- Delicato, F. C., et al. (2005b). "Application-Driven Node Management in Multihop Wireless Sensor Networks". In: Procs. of the 4th IEEE Int. Conference on Networking, Reunion Island, April.
- Delicato, F. C., et al. (2005c). "Uma Abordagem Baseada em QoS para Seleção de Nós Ativos em Redes de Sensores sem Fio" - In: Anais do 23º SBRC, Fortaleza, May.
- Delicato, F. C., et al. (2005d). "A Service Approach for Architecting Application Independent Wireless Sensor Networks", Cluster Computing, ISSN: 1386-7857, vol. 8, n. 2-3, pp. 211-221, July.
- Delicato, F. C., et al. (2005e). "Uma Framework baseada em Tecnologias Web para a Construção de Redes de Sensores sem Fio Flexíveis" - In: Anais do XI WebMedia, Poços de Caldas, December.
- Delicato, F. C., et al. (2006a). "Uma abordagem baseada em Lógica Nebulosa para a seleção de Protocolos de Disseminação de Dados em Redes de Sensores sem Fio". In: Anais do 24º SBRC, Curitiba (to appear).
- Delicato, F. C., et al. (2006b). "Exploiting Web Technologies to Build Autonomic Wireless Sensor Networks". In: Procs. of the 8th IFIP/IEEE MWCN 2006, Santiago, Chile (to appear).
- Delicato, F.C., et al. - "An Efficient Heuristic for Selecting Active Nodes in Wireless Sensor Networks" Computer Networks, Elsevier Science, ISSN: 1389-1286 (to appear).
- Heideman, J., Silva, F. and Estrin, D. (2003). "Matching Data Dissemination Algorithms to Application Requirements". In: Procs. of the ACM SenSys, pp. 218-229, USA, November.
- Intanagonwiwat, C., Govindan, R., and Estrin, D. (2000). "Directed diffusion: a scalable and robust communication paradigm for sensor networks". In: Procs. of the ACM/IEEE MobiCom, USA, August.
- Perillo, M. and Heinzelman, W. (2003) "Sensor Management Policies to Provide Application QoS", Elsevier AdHoc Networks, Spec.Issue on Sensor Network App. and Protocols v.1, n.2-3, pp.235-246,.
- SENSORIA. Sensoria WINS 3.0 Spec. Disponível em: <http://www.sensoria.com/products-wins30.htm>.
- Shen, C., Srisathapornphat, C. and Jaikao, C. (2001). "Sensor Information Networking Architecture and Applications", IEEE Personal Communications v. 8, n. 4, pp. 52-59, August.
- Tilak, S., Abu-Ghazaleh, N. B. and Heinzelman, W. (2002). "A taxonomy of wireless micro-sensor network models". ACM SIGMOBILE Review v.6, n.2, pp. 28-36, April.
- WTK. J2ME Wireless Toolkit. Disponível em <http://java.sun.com/products/j2mewtoolkit/>.