

INTEGRAÇÃO DE UMA REDE DE SENSORES SEM FIO COM A WEB UTILIZANDO UMA ARQUITETURA ORIENTADA A SERVIÇO

André S. PFAFFENBACH¹; Thiago C. TAVARES²

RESUMO

Este artigo tem como principal objetivo realizar a integração de uma rede de sensores sem fio com a Web através de uma arquitetura orientada a serviço (SOA). Essa integração é provida através de uma infraestrutura de *Web services* com a utilização de protocolos de comunicação, linguagens de programação e descrição de serviços tais como WSDL, UDDI, SOAP ou REST. Além disso, este trabalho discute uma avaliação de desempenho comparando outros paradigmas de acesso a redes de sensores tais como tecnologias de acesso binárias (sockets e RMI) em relação à *Web services*.

INTRODUÇÃO

Uma rede de sensores sem fio (RSSF) é composta por pequenos sensores que monitoram um ou um conjunto de fenômenos que serão enviados para um usuário final. As redes de sensores podem ser utilizadas em uma gama de aplicações de monitoramento e rastreamento (Yick et al., 2008). Um grande desafio na utilização de redes de sensores está na viabilidade de se gerenciar uma rede de sensores e de prover as informações necessárias para utilização em diferentes aplicações. Por um lado tem-se a infraestrutura composta pelos sensores e pelas estratégias de utilização desses sensores e das informações obtidas por eles. Por outro lado, têm-se aplicações ou observadores que devem receber as informações e processá-las.

Para viabilizar a utilização de redes de sensores pode-se desenvolver um Middleware que oferece as ferramentas necessárias. A literatura apresenta diversas propostas e implementações de Middlewares que podem ser utilizados para facilitar o desenvolvimento dessas redes em relação à instalação, manutenção e execução

¹ Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Inconfidentes. Inconfidentes/MG, email: 2300@ifs.ifsuldeminas.edu.br;

² Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais – Campus Inconfidentes. Inconfidentes/MG, email: thiago.tavares@ifsuldeminas.edu.br;

de aplicações (Wang et al., 2008). Uma abordagem que vem sendo avaliada considera a rede de sensores como um *Web Service* que oferece uma interface padronizada para acessar as informações. Um Middleware que utiliza os conceitos da arquitetura orientada a serviço (SOA - Service Oriented Architecture) é apresentado em (Delicato et al., 2005). Assim, este artigo mostra uma implementação de uma rede de sensores sem fio exposta através de uma abordagem SOA. Além disso, é discutido uma avaliação de desempenho da aquisição dos dados providos pela rede por meio de um planejamento de experimento considerando diferentes implementações de acesso.

MATERIAL E MÉTODOS

Como comentado anteriormente, três tipos de componentes de software foram utilizados para a realização deste trabalho:

- **Sockets:** a utilização de sockets considerou a implementação de um servidor que realiza a coleta das informações da rede de sensores. Assim, um cliente através de uma ligação (*bind*) por socket pode acessar esse servidor e obter as informações da RSSF.
- **RMI:** O *Remote Method Invocation* (RMI) é um mecanismo que permite ao programador criar uma aplicação distribuída baseada na tecnologia Java. Aplicações que utilizam RMI compreendem dois programas distintos, um servidor e um cliente (assim como em Socket). Um servidor cria alguns objetos remotos e os torna acessíveis por meio de um registro de objetos chamado de RMIRegistry. Após esse registro, clientes podem requisitar (invocar) métodos desses objetos. Um programa cliente típico pode obter diversas referências remota a um ou mais objetos em um servidor.
- **Web service:** Os *Web Services* podem ser definidos como um sistema de software projetado para suportar interoperabilidade entre máquinas sobre uma rede. *Web services* caracterizam-se por ser uma tecnologia para computação distribuída na Web. Suas principais características são a utilização de padrões abertos e XML. Os softwares utilizados para a execução dos testes foram o toolkit de processamento SOAP Apache Axis2 1.7.

A Figura 1 apresenta o ambiente de testes considerando as implementações de aplicações clientes e servidoras para cada um dos componentes de software considerados (Socket, RMI e *Web Services*).

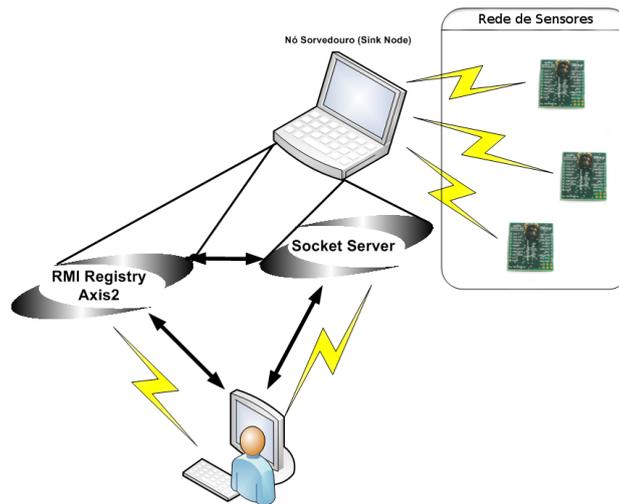


Figura 1: Ambiente de Testes

O ambiente ainda define uma rede de sensores construída por três sensores Radiuino (Radiuino, 2014) que são pode ser definido com uma placa Arduino composta de um micro-controlador Atmega328 e um transceptor CC1101 da Texas Instruments que é utilizado para realizar a transmissão dos dados sem fio dos nós sensores para o nó sorvedouro. Cada sensor também possui um transdutor de temperatura que faz a aferição da temperatura ambiente. A programação deste tipo de sensor é feita utilizando-se a linguagem C++. Uma aplicação foi implementada e instaladas nos sensores. Essa aplicação faz o monitoramento de temperatura a cada cinco segundos e os envia ao nó sorvedouro que é responsável por concentrar as medições da rede. Então, cada valor é armazenado em uma base de dados que será acessada pelas aplicações servidoras (Socket, RMI e *Web services*) quando estas receberem requisições de clientes solicitando informações sobre a temperatura ambiente.

Com as implementações dos servidores foi possível planejar um conjunto de experimentos de forma a comparar as diferenças de desempenho entres os paradigmas considerados. Esse planejamento está representado na Tabela 1. O planejamento leva em conta as três formas de acesso já discutidas (Socket, RMI e *Web service*), dois meios de comunicação (Cabeada e Wireless) e quantidade de clientes/threads (50/100). O fator meio de comunicação determina a forma de acessar os dados aferidos pela rede variando-se os meios de comunicação do cliente. Ou seja, o cliente pode acessar o servidor através de uma rede sem fio (Wireless) ou através de uma rede cabeada. O objetivo de se variar o meio de comunicação é encontrar diferenças de desempenho em diferentes meios de

comunicação. Por fim, a quantidade de threads simula diferentes clientes acessando concorrentemente os serviços.

Tabela 1: Fatores

Fatores		
Formas de Acesso	Meios de Comunicação	Threads (Clientes)
Socket	Cabeada	50
RMI	Wireless	100
<i>Web Service</i>		

Uma vez definido o planejamento de experimentos, foi efetuada a execução dos testes. Essa execução foi realizada de forma a permitir uma análise estatística dos resultados aferidos. É importante destacar que as experimentações consideraram que cada thread submeteu 10 requisições às aplicações servidoras. Além disso, cada experimento foi replicado 30 vezes. Essa quantidade de replicações foi definida de forma a obter um menor intervalo de confiança possível e, dessa forma, facilitar a análise dos resultados. A próxima seção apresenta e discute os resultados obtidos na execução do planejamento de experimentos aqui discutidos.

RESULTADOS E DISCUSSÃO

Nesta seção são apresentadas algumas análises sob a ótica das formas de acesso avaliadas. Contudo, essa diferença é elevada para 437% no meio Cabeado. A diferença de tempos de acesso entre essas duas tecnologias é de aproximadamente 1 e 4 vezes para os meios Wireless e Cabeado, respectivamente. apresenta os gráficos comparativos entre as formas de acesso com uma carga de 50 threads (clientes), enquanto a Figura 3 apresenta um gráfico comparativo para uma carga de 100 threads. Os gráficos representados na Contudo, essa diferença é elevada para 437% no meio Cabeado. A diferença de tempos de acesso entre essas duas tecnologias é de aproximadamente 1 e 4 vezes para os meios Wireless e Cabeado, respectivamente. mostram que em comparação ao Socket, o *Web service* é de 24 e 49 vezes mais ineficiente no acesso aos dados da rede de sensores nos meios Wireless e Cabeado, respectivamente. Os resultados para o acesso aos dados da rede de sensores através de RMI também se apresentaram mais eficientes. Pode-se notar que, em relação a esses resultados, o *Web service* é aproximadamente 96% pior que o RMI no meio Wireless. Adicionalmente, no meio Cabeado esse valor é acrescido para 359%. Os resultados obtidos para a carga de 100 threads são apresentados na Figura 3. Os resultados mostraram que os Sockets

são 40 e 100 vezes mais rápidos que os *Web services*, para os meios Wireless e Cabeado, respectivamente. No meio Wireless a diferença entre o acesso aos dados da rede de sensores através de RMI e *Web service* é de 110%.

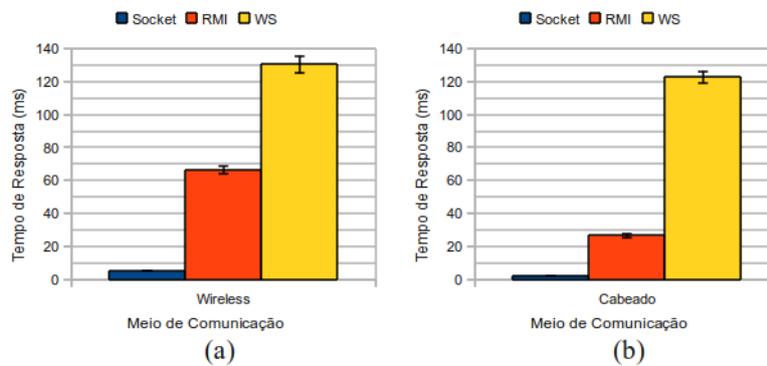


Figura 2: Formas de Acesso com 100 threads

Contudo, essa diferença é elevada para 437% no meio Cabeado. A diferença de tempos de acesso entre essas duas tecnologias é de aproximadamente 1 e 4 vezes para os meios Wireless e Cabeado, respectivamente.

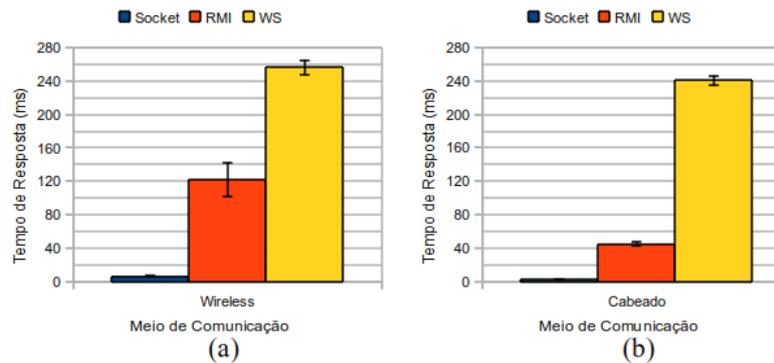


Figura 3: Formas de Acesso com 100 Threads

A análise para todas as formas de acesso apresentou o que se esperava de valores para os tempos de resposta. O Socket por ser um mecanismo de comunicação primitivo e com pouca abstração para o desenvolvedor apresentou os melhores resultados, assim como apresentados em outros trabalhos como em (Jagannadham et al., 2007). De fato, a opção de programação ao acesso das redes de sensores através de Socket é, do ponto de vista de desempenho, a melhor opção. Todavia, os Sockets apresentam restrições com relação a abertura de portas no processo de conexão. Outros requisitos como segurança, representação de dados e questões de interoperabilidade são fatores desvantajosos para esse tipo de mecanismo de comunicação. Por outro lado, o RMI apresenta resultados intermediários (com tempos de resposta maiores que os encontrados para os

Sockets e menores que os encontrados para *Web services*). Embora ele apresente uma boa abstração e suporte no momento de se implementar uma aplicação distribuída (no caso dos testes desse trabalho o acesso aos dados de um RSSF), ele ainda é deficiente em termos de interoperabilidade e na comunicação através de firewalls. Por fim, os *Web services* apresentaram, em muitos dos casos analisados, valores altos em relação às demais formas de acesso. Por exemplo, em um meio Cabeado, o *Web service* obteve valores de tempo de resposta de aproximadamente 100 vezes maiores que os tempos obtidos com Sockets. Entretanto, diferentemente das demais formas de acesso os *Web services* apresentam vantagens com relação à interoperabilidade, representação de dados (*XML Schema*), independência de linguagens programação, facilidade na implementação de grandes sistemas (modularização), etc.

CONCLUSÕES

Os resultados obtidos neste trabalho mostraram que os *Web services* possuem desempenho bastante inferior em comparação a outros métodos como Sockets e RMI. Contudo, os *Web services* apresentam benefícios em relação à padronização, interoperabilidade e facilidade de implementação. Além disso, devem-se considerar os valores absolutos dos tempos de resposta obtidos na execução dos experimentos. Mesmo em grande quantidade de acessos concorrentes, o tempo de resposta não ultrapassou o valor de 250ms. Esse valor pode ser tolerado por diferentes aplicações que utilizam dados de redes de sensores, além de trazer benefícios à interoperabilidade entre sistemas.

REFERÊNCIAS BIBLIOGRÁFICAS

- DELICATO, F. C. Middleware baseado em serviços para redes de sensores sem fio. Tese de doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ - Brasil, 2005.
- JAGANNADHAM , D.; R AMACHANDRAN , V.; K UMAR , H. Java2 distributed application development (socket, rmi, servlet, corba) approaches, xml-rpc and *web services* functional analysis and performance comparison. In: Communications and Information Technologies, 2007. ISCIT'07. International Symposium on, 2007, p. 1337 -1342.
- RADIOIT. Produtos. Disponível em: <http://www.radioit.com.br/produtos>. Último acesso: 21/06/2014, 2014.

WANG, M.; CAO, J.; LI, J.; DAS, S. K. Middleware for wireless sensor networks: A survey. *J. Comput. Sci. Technol.*, v. 23, n. 3, p. 305–326, 2008.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. *Comput. Netw.*, v. 52, n. 12, p. 2292–2330, 2008.