

Uma proposta de arquitetura de middleware para redes de sensores e atuadores sem fio utilizando web services

Gustavo Arthur da Costa e João Henrique Kleinschmidt

Resumo— Neste artigo é proposto um middleware orientado a serviços para redes de sensores e atuadores sem fio (RSASFs). O middleware é uma ferramenta capaz de abstrair detalhes da infraestrutura e dos serviços da rede das aplicações clientes. Uma padronização para a comunicação da rede com as aplicações clientes é fornecida usando tecnologias de web services como o protocolo SOAP e a linguagem XML. A arquitetura de middleware proposta se adequa aos requisitos de uma RSASF, com o diferencial de proporcionar interoperabilidade entre diferentes componentes de hardware e software da rede.

Palavras-Chave—Redes de Sensores e Atuadores sem fio, Middleware, Web Services, Java.

Abstract— In this paper is proposed a service oriented middleware for wireless sensor and actor networks (WSANs). The middleware is a tool capable of hiding details of the infrastructure and providing services of the network from client applications. Standardization of the network's communications is provided using web services technologies such as SOAP protocol and XML language. The proposal middleware architecture is suitable for WSANs characteristics, with the differential of providing interoperability between different hardware and software components of the network.

Keywords—Wireless sensor and actors network, middleware, web services, java.

I. INTRODUÇÃO

As redes de sensores sem fio (RSSFs) são redes compostas por vários sensores interligados, os quais são dispositivos de baixo custo, baixa capacidade de processamento, sensoriamento e comunicação, e também baixa autonomia de energia [1,2, 3]. Já as redes de sensores e atuadores sem fio (RSASFs) possuem atuadores, que são de um porte maior, com maior capacidade de processamento e comunicação, com baterias de maior duração e executam alguma ação na rede [2]. As RSASFs diferem-se das RSSFs em algumas características como requisitos de comunicação em tempo real e coordenação.

Os desafios de se implementar uma RSASF são maiores que uma rede convencional, pois a maioria dos protocolos convencionais geralmente não é aplicável nessas redes. A energia é um ponto crucial devido ao fato dos sensores possuírem uma bateria limitada e que muitas vezes o custo para manutenção é inviável. Assim a arquitetura de rede e protocolos devem sempre visar economia de energia [2].

O middleware em uma RSASF é uma camada de software entre a aplicação e a infraestrutura provendo através de interfaces um reuso de serviços que podem ser compostos e configurados para desenvolvimento facilitado de aplicações.

Permite também abstrair detalhes de baixo nível quanto ao acesso a serviços distribuídos como protocolos de comunicação e gerenciamento de dados. Alguns modelos de middleware e frameworks para RSASFs foram propostas com diversas abordagens, como o TC-WSANS [4] e USEME [5].

Nesse artigo é proposto um middleware orientado a serviços onde a RSASF é vista como uma fornecedora de serviços para os clientes e os atuadores da rede são fornecedores de serviços para os sorvedouros.

Essa abordagem foi introduzida em [3], porém foi usada para RSSFs. Utiliza web services com tecnologias padrão como o protocolo SOAP (Simple Object Access Protocol) e a linguagem WSDL (Web Service Description Language), ambas baseadas na linguagem XML (Extensible Markup Language), que não foram usados nos trabalhos propostos em [4,5]. O objetivo de usar essa abordagem a nível externo a rede é facilitar a integração com as aplicações clientes e fornecer um modelo abstrato de programação para aplicações de RSASFs. Internamente fornece interoperabilidade entre os módulos internos da rede, como a comunicação entre sensores, atuadores e sorvedouros, possibilitando a interação entre dispositivos de diversos fabricantes e sistemas operacionais.

II. ARQUITETURA

Para o middleware proposto foi definida primeiramente uma topologia de rede em que seria aplicado. Foram definidas então quais as comunicações seriam feitas, em que sentido essas comunicações se dariam e qual o papel do middleware em cada componente da rede.

A Figura 1 representa a arquitetura proposta, visando demonstrar em que sentido as comunicações acontecem na rede e com as aplicações clientes e quais os componentes presentes na mesma.

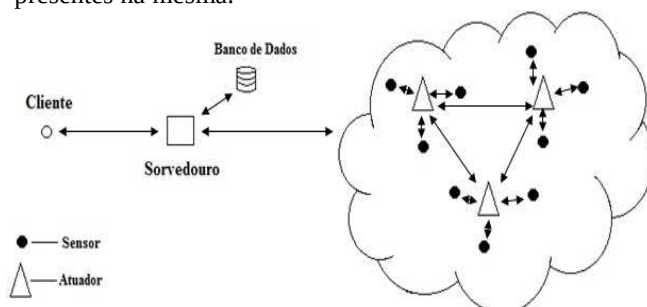


Fig. 1. Arquitetura da RSASF proposta

O cliente é uma aplicação qualquer que deseja acessar os dados da RSASF. Neste trabalho foi representado por uma página web. Esse cliente se comunica com o sorvedouro e solicita quais são os serviços disponíveis na rede. O sorvedouro então responde mostrando os serviços disponíveis

para esse cliente. Assim o cliente pode requisitar o serviço desejado e então o sorvedouro responde com o resultado.

Logo, o cliente sempre é quem inicia a comunicação. A comunicação é bidirecional, ou seja, tanto envia como recebe dados da rede. Uma definição de nível de acesso à rede é definida para os clientes: um cliente com nível de acesso alto é como um administrador da rede, que além de ter acesso aos dados também pode interferir no funcionamento da mesma. Com o nível baixo de acesso o cliente só pode ter acesso aos dados já coletados pela rede.

O sorvedouro por sua vez ao receber a requisição para informar os serviços disponíveis na rede possui uma lista com todos esses serviços. Porém, quando recebe a requisição de algum serviço específico, dependendo do nível de acesso do cliente, vai buscar os dados de maneira diferente.

Para um cliente com um nível de acesso baixo o sorvedouro vai simplesmente buscar os últimos valores coletados em um banco de dados e retornar ao cliente. Já para um cliente com um nível de acesso maior, o sorvedouro vai acessar a rede para buscar o resultado do serviço solicitado em tempo real. O sorvedouro é visto então como um provedor de serviços. O atuador é responsável pela comunicação com o sorvedouro, uma vez que por definição da topologia da rede cada sensor se comunica somente com um atuador. Deve então passar as informações de suas ações e dos sensores para o sorvedouro. Com isso o atuador pode tanto iniciar uma conexão com o sorvedouro quando responder uma requisição de conexão do mesmo.

Quando o atuador inicia a conexão, envia para o sorvedouro algum dado da rede, como um dado de algum sensor ou de alguma ação tomada por ele. O envio da informação pelo atuador sem uma requisição pode acontecer em dois casos: em caso de algum evento “diferente” acontecer, como ultrapassar uma temperatura pré-estabelecida; ou uma coleta periódica realizada pela rede. Essa comunicação não depende que um cliente esteja acessando a rede.

Já quando recebe a requisição de algum de seus serviços do sorvedouro o atuador requisita para os sensores que estão na sua área de abrangência os dados coletados para retornar o resultado novamente para o sorvedouro. Assim o atuador também representa um provedor de serviços, mas seus serviços são acessados pelo sorvedouro e posteriormente repassados para uma aplicação cliente.

Por último os sensores da rede se comunicam com um atuador pré-definido e essa comunicação também é bidirecional, tanto o sensor pode enviar um dado para o atuador sem ter sido requisitado quanto o atuador pode requisitar que o sensor envie um dado. Na topologia proposta os sensores não se comunicam diretamente com os sorvedouros com o intuito de economizar energia, por estarem geralmente a distâncias menores dos atuadores e pelo fato dos atuadores serem dispositivos com maiores recursos de energia.

III. IMPLANTAÇÃO

O middleware proposto foi construído utilizando a linguagem de programação Java, utilizando o ambiente de desenvolvimento NetBeans. Cada módulo do sistema de middleware é um tipo diferente de aplicação Java. O módulo cliente foi desenvolvido utilizando a tecnologia JSP (*Java Server Pages*) e se comunica somente com o módulo sorvedouro, que é um web service.

O sorvedouro foi definido como um web service para prover uma interoperabilidade entre sistemas, no caso as aplicações clientes e a RSASF. Ele possibilita uma abstração de hardware ou software utilizado nas aplicações clientes [4]. Para a comunicação com o web service é utilizado o protocolo SOAP. Um banco de dados MySQL foi utilizado para armazenar os dados coletados pelos sensores e as ações tomadas pelos atuadores. É acessado quando uma aplicação cliente com um nível de acesso baixo requisita um dado, sendo que os dados são buscados diretamente no banco sem precisar acessar a RSASF.

O sorvedouro também se comunica com os atuadores da rede através do protocolo SOAP, já que os atuadores também são web services. Os atuadores por sua vez se comunicam com outros atuadores e com os sensores. A comunicação entre dois atuadores utiliza SOAP, já a comunicação atuador-sensores utiliza o protocolo UDP (*User Datagram Protocol*). O protocolo UDP foi escolhido porque as mensagens UDP são mais leves que as SOAP (XML), economizando energia dos sensores.

Por fim os módulos sensores foram implementados em aplicações J2ME (*Java 2 Mobile Edition*). Para exemplificar a comunicação na rede pode se pensar em uma aplicação cliente, no caso uma página web, que tem acesso ao sorvedouro, um web service. O primeiro passo seria a requisição de todos os serviços que esse sorvedouro tem a oferecer. Após a aplicação cliente tomar conhecimento de todos os serviços, ela pode escolher um serviço da lista e requisitar o mesmo para o sorvedouro.

Caso a aplicação cliente requisitar o serviço que retorna a temperatura em torno de um atuador, o sorvedouro faz uma requisição via o protocolo SOAP para um atuador. O atuador por sua vez passa uma requisição via UDP para cada sensor que estiver em comunicação com ele e assim os sensores retornam o dado coletado, também via UDP. Assim o atuador retorna esse valor para o sorvedouro pelo protocolo SOAP. Por final o sorvedouro retorna o resultado para a aplicação cliente também pelo protocolo SOAP.

IV. CONCLUSÕES

A arquitetura de middleware proposta se adequa aos requisitos de uma RSASF, tendo seu diferencial na interoperabilidade entre os diferentes hardwares e sistemas operacionais utilizados entre os dispositivos da rede. Com a arquitetura proposta o desenvolvimento de aplicações clientes voltadas para RSASFs se torna uma tarefa mais fácil, por utilizar padrões de comunicação muito difundidos e todos multiplataforma.

REFERÊNCIAS

- [1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. e Cayirci, E. “A survey on sensor networks”. *IEEE Communications Magazine*, pp. 102-114, 2002.
- [2] Akyildiz, I. F. e Kasimoglu, I. H. “Wireless sensor and actor networks: research challenges”. *Ad Hoc Networks*, vol. 2, pp. 351-367, 2004.
- [3] Delicato, F., Pires, P., Lages, A., Rezende, J. F. e Pirmez, L. “Middleware orientado a serviços para redes de sensores sem fio”. XXII Simpósio Brasileiro de Redes de Computadores, Gramado-RS, 2004.
- [4] Barbarán, J., Díaz, M., Esteve, I., Garrido, D., Llopis, L., Rubio, B. e Troya, M. “Programming Wireless Sensor and Actor Networks with TC-WSANS”, IEE International Conference on pervasive services, 2007.
- [5] Cañete, E., Chen, J., Díaz, M., Llopis, L. e Rubio, B. “USEME – A Service-oriented framework for Wireless Sensor and Actor Networks”, 8th International workshop on applications and services in wireless networks, 2008.