Design and Implementation of a Web Service for LiteOS-based Sensor Networks

Masaaki Takahashi, Basit Hussain, Bin Tang

Department of Electrical Engineering and Computer Science, College of Engineering

Abstract. This project presents the design and implementation of a web service for LiteOS-based wireless sensor networks (WSNs) to remotely monitor the light, temperature, magnet, and acceleration of the physical world. LiteOS is a newly developed operating system for the sensor motes. Taking advantage of UNIX-like shell commands and C programming language supported by LiteOS, this proposed web service enables the users to remotely query and visualize the sensor readings. Web service system is equipped with secure membership, a visualizer for sensor readings, and accepts parameterized queries.

1. Introduction

Wireless Sensor Networks (WSNs) enables users to interact with the physical world and receive real time information. The applications include agriculture and environmental monitoring, manufacturing and industrial sensing, battlefield, and disaster recovery. Connecting sensors to the Internet gives users more flexibility to manage and use WSNs. Since sensors have very limited memory and power supply, it is difficult to provide users with rich user interface and extensive data processing. Web services, which provide a mechanism for distributed sensor applications to share sensor deployments, are good solutions.

Web services were recently proposed to support interoperable and evolvable sensor networks [1]. However, the study of the architecture of web services for sensor networks is still in its infancy and remains unclear due to the lack of a standard middleware. This work proposes a web service architecture and implements a web service middleware for wireless sensor networks. The layered architecture, shown in Figure 1, is comprised of an application layer, a web service middleware layer, and a WSN layer. The web service middleware interacts with the front-end web application as well as the WSNs such that users can remotely request and view the sensor readings.

LiteOS [2], a new operating system for sensor motes developed by UIUC, is used in the project. A simple front-end web application to query and compare light, temperature, magnet, and acceleration sampled from sensors is developed. The queries can be parameterized in terms of number of the sampling nodes, intervals, and comparison type (e.g., comparing by nodes and networks).

Web services for sensor networks have been developed in both research and commercial fields, such as Tiny Web Service [1, 3] and Arch Rock’s web service [4]. They are both based on TinyOS [5] operating system. TinyOS adopts NesC and the event-based programming model, which introduce a learning curve for most traditional programmers. On the other hand, LiteOS supports C programming and provides UNIX-like abstraction for wireless sensor networks, which greatly improve its compatibility with other development platforms and simplify the sensor network programming. LiteOS includes three subsystems: LiteShell, LiteFS, and LiteOS Kernel. These subsystems provide several desirable features: (1) a hierarchical file system, (2) multi-threading, (3) Unix-like shell interface. These features are handy for the design and implementation of web services for sensor networks.

2. Experiment, Results, Discussion, and Significance

An experimental design and implementation in the project consists of three layers: an application, a web

2 UNIX is a widely used operating system. Providing Unix-like abstraction makes it easier for users to handle LiteOS.
service middleware, and a WSN layer. Application layer is responsible for handling client requests and sending the appropriate requests to middleware. It is implemented using Microsoft ASP.NET, and provides two basic functions: a membership system for login, and a visualizer for sensor results. The visualizer enables users to view the sensor readings by a simple chart and table. Figure 2 shows the light reading comparison of two sensor nodes using the visualizer. The communication between application layer and middleware is done by standard XML using SOAP\(^3\) and WSDL\(^4\) specification. Thus, web service system can be easily accessed by users via standard interface.

Fig 2: Light reading comparison of two sensor nodes.

For middleware, Apache Axis-2, a core engine for web services is used. For WSN Layer, we use Crossbow IRIS motes. Compared with previous generation motes, IRIS demonstrates three times longer radio frequency range, half lower sleep current, and double program memory (8KB), which makes IRIS an ideal platform for web service development. A base station sits between middleware and WSN to facilitate the communication of these two layers.

As a result, several challenges exist in the integration of the middleware with the LiteOS: (1) how to handle multi-threading when multiple sensor responses are expected, (2) how to interact with LiteOS programmatically. For multi-threading, LiteOS's multithreaded kernel is used to run multiple applications concurrently. LiteOS supports Thread class in LiteC++ to perform multi-threading. For programmatic purpose, LiteShell, a subsystem of LiteOS, provides a practical schema such as Unix-like command-line interface to sensor nodes. Java is chosen as the development language for middleware since LiteShell uses Java communication API. As shown in Figure 3, an Event Handler is developed in the middleware to handle the requests of the web services, and to communicate directly with the Command Processor, which interprets user commands into internal forms and communicates with the sensor network.

Fig 3: Interaction between middleware and LiteShell.

After solving these challenges, system has the following features: (1) Middleware accepts parameterized requests through an front-end application, (2) Front-end application visualizes sensor readings, (3) Multi users request sensor readings simultaneously. More significantly, an easy-to-program approach to deploy web service on sensor networks is demonstrated.

3. Conclusions

A web service for LiteOS-based wireless sensor networks to remotely monitor the physical world is designed and implemented, and the feasibility of deploying web services using LiteOS-based sensor networks is demonstrated. Current phase of design and implementation is still in prototype level. The future works includes synchronous radio communication, remote application deployment, and multi-hopness.

4. Acknowledgements

The authors would like to thank Dr. Qing Cao of UIUC for his valuable suggestions.

5. References