A Flexible Web Service based Architecture for Wireless Sensor Networks

Flávia Coimbra Delicato[1], Paulo F. Pires[1,2,] Luci Pirmez[1], Luiz Fernando Rust da Costa Carmo[1]
Núcleo de Computação Eletrônica – NCE[1] & Computer Science Departament – DCC[2]
Federal University of Rio de Janeiro
P.O Box 2324, Rio de Janeiro, RJ, 20001-970, Brazil
{ fdelicato, paulopires, luci, rust } @ nce.ufrj.br

Abstract

The current sensor networks are assumed to be designed for specific applications, having strongly coupled data communication protocols. The future sensor networks are envisioned as comprising heterogeneous devices assisting to a large range of applications. To achieve this goal, a new architecture approach is needed, having application specific features separated from the data communication protocol, while influencing its behavior. We propose a Web Services approach for the design of sensor networks, in which sensor nodes are service providers and applications are clients of such services. Our main goal is to enable a flexible architecture where sensor networks data can be accessed by users spread all over the world.

1 Introduction

Many scientific applications require the acquisition of precise data measurements over time for a large geographic region of interest. While these measurements can sometimes be accomplished using remote sensing techniques, it is often necessary to collect data using in-situ sensing, where sensors are placed directly in the target area [13].

One potential solution for such an application scenario is the design of large multi-point sensor networks comprised of nodes with sensing, processing (elementary) and communication capabilities. Such systems can use hundreds or thousands of sensor nodes, interconnected by a wireless network and play the role of a highly parallel, accurate and reliable data acquisition system.

Sensors are devices with limited energy and processing capabilities, deployed in an ad-hoc fashion, and they have to operate unattended, since it is unlikely to service a large number of nodes in remote, possibly inaccessible locations. Therefore, energy saving is a crucial requirement in such an environment.

Sensors data are transmitted from multiple acquisition sources toward one or more processing points, which may be connected to external networks. Since sensors monitor a common phenomenon, it is likely to appear significant redundancy among data generated from different sensors. Such a redundancy can be exploited to save transmission energy, throughout filtering and data aggregation procedures in-network. To save further energy, the short-range hop-by-hop communication is preferred over the direct long-range communication to the final destination. Thus, nodes send their own data and their neighbors' data through paths, preferably optimized, to some exit point in the network.

In sensor networks environments, it is more useful form the applications point of view to have nodes identified by the type of sensor device or by their geographical location. Several works [5, 6] have suggested the use of data-centric naming systems, instead of traditional address-centric schemes, like IP. In the data-centric approach, nodes are addressed by the type of data they provide, or by their interest in some type of sensing data.

Current works [5, 6] consider sensor networks as being designed for specific applications, with data communication protocols strongly coupled to the application. In fact, the network requirements, organization, and routing behavior change according to the application. In spite of the application specific behavior of the current sensor networks, many authors [10] envision the future sensor networks as being composed of heterogeneous sensor devices and assisting to a large range of applications. To achieve this goal, a new architectural approach is needed, where application specific requirements are separated from the data dissemination functions. In such an architecture, the components should be loosely coupled, having well defined interfaces. To achieve energy efficiency, applications should be able to dynamically change the network behavior. However, these changes should be expressed in a flexible way, through a common protocol. Such features will allow the design of networks to be independent from the applications that will use them.

We propose a service approach for the design of sensor networks. Services are defined as the data provided by the sensor nodes and the applications (for instance, a filtering program) to be executed on those data. Clients access the sensor network submitting queries to those services.

Services are published and accessed using the Web services technology [3]. By adopting the Web Services paradigm, we propose a novel architecture for sensor networks, in which the Web Services Description Language (WSDL) [15] is used to describe data and functionalities of sensor nodes. Sink nodes are Web Services that offer a standard interface for accessing services provided by the network. Queries submitted by user applications are accomplished through such an interface.

Using specific routing protocols for sensor networks, such as direct diffusion [5], we offer a flexible and powerful way of manipulating, extracting and exchanging data from sensor network. Applications access the sensor network and modify the underlying data dissemination behavior through a common and application independent interface provided by the Web Services available on sink nodes.

The present work defines the architectural components as well as the WSDL elements needed to implement a direct diffusion scheme in our architecture. The article is organized as follows. Section 2 covers the background concepts. Section 3 presents the related work. Next, Sections 4 and 5 detail the system architecture and description. Finally, Section 6 outlines the conclusions and future works.

2 Background

The dissemination of information in a sensor network is done by nodes performing measurements and relaying data through neighboring nodes to reach some network exit point, called *sink*. Data sent by different nodes can be aggregated in order to reduce redundancy and minimize the traffic and thus the energy consumption. To enable data aggregation in network in an efficient way, application-specific code, such as data caching and collaborative signal processing should occurs as close as possible to where data is collected. Such a processing depends on attribute-identified data to trigger application-specific code and hop-by-hop processing of data [4].

Attribute-based naming systems offer a data-centric approach, which differs from the traditional address-centric approaches.

2.1 Data-Centric Communication

Data-centric addressing has been proposed for sensor networks, where nodes are identified by the data generated by them or by their geographical location. Directed diffusion [5] is an example of a data-centric protocol specifically designed for sensor networks. In directed diffusion, individual nodes reduce the sampled waveform generated by a target into a relatively coarse-grained “event” description [5]. Such a description contains a set of attributes. Applications requesting data send out interests through some sink in the network. These interests are also represented as a set of attributes. If the attributes of the data generated by source nodes match these interests, a gradient is setup within the network and data will be pulled toward the sinks. Intermediate nodes are capable of caching and transforming data.

Despite of the advantages of data-centric addressing in sensor networks, recent works [5, 6] assume that the data representation is application-specific or offer schemes with low flexibility and expressivity. Such approaches require a strong coupling between the data or interest representation model and the application querying the network.

We can envision a class of future sensor networks as being accessed by several different applications submitting queries through arbitrarily localized sinks. To enable such a scenario, the network should be accessed through a common and application independent interface.

The present work suggests a Web services approach for architecting wireless sensor networks. We propose the use of a service description language - WSDL [15] and the associated protocol - SOAP [19], both accepted as Internet standards, as being the basis for describing and communicating data and interests on a flexible way in a sensor network. The next section briefly introduces the Web services technology.

2.2 The Web Services Technology

Web services can be defined as modular programs, generally independent and self-describing, that can be discovered and invoked across the Internet or an enterprise intranet. Web services are accessed via ubiquitous Web protocols and data formats, such as Hypertext Transfer Protocol (HTTP [1]) and XML [17]).

The Web services technology is based on a flexible architecture named SOA (Service-Oriented Architecture [3]), which defines three roles: a service requestor, a service provider and a service registry. A service provider is responsible for creating a service description, publishing that service description to one or more service registries, and receiving Web services invocation messages from one or more service requestors.

Besides these roles, three operations are defined as part of SOA architecture: publish, find and bind.

The publish operation is an act of service registration or service advertisement. The find operation allows the service requestor to state search criteria, such as type of service. The service registry matches the find criteria against its collection of published Web services descriptions. The bind operation embodies the client-server relationship between the service requestor and the provider [3].

Besides to comply with the SOA pattern, the Web service technology can be factored into three protocols stacks [3]: the wire stack (or exchange format), the description stack and the publish and discovery stack.

The wire stack represents the technologies that determine how a message is sent/received from the service requestor to the service provider. The stack is composed of three levels. The first level is a network protocol, which can be an Internet wire protocol, such as HTTP [1], or sophisticated enterprise-level protocols. The second level is the data encoding mechanism that is based on XML. The third level refers to XML messaging layers. For XML messaging, Web services use SOAP [19], which acts as a wrapper to XML messages, guaranteeing a solid, standard-based foundation for Web services communication.

The description stack provides aspects of a service that are important to the service requestor. In Web services, XML is the basis of service description. The XML Schema specification (XSD) [18] defines the canonical type system. Besides this level, the next levels of the stack are the descriptions of the service interface, the service concrete mapping and the service endpoint. An endpoint defines the network address where the service itself can be invoked. All of those levels use WSDL [15], which is an XML-based language for describing the interface of Web services. WSDL is a very flexible model for services descriptions but it is also rather verbose. A typical sensor device has very limited capacities. So, a more compact mechanism for data representation is needed. One example of such a mechanism is the WAP Binary XML Content Format (WBXML [16]). This format defines a compact binary representation for XML [17], intended to reduce the size of XML documents for transmission and to simplify parsing them.

The publish and discovery stack corresponds to the directory service for Web services. Service providers need a publication mechanism so that they can provide information about the Web services they offer and service requestors need well-defined find APIs for using such Web services.

3 Related Work

Several works on data centric communication are based on localized algorithms [11] to reduce redundancy, saving energy. Directed diffusion [5] and SPIN [6] are two data-centric localized algorithms for sensor networks. Both assume a strong coupling between the components of data dissemination and application-specific features.

We propose a generic architecture for sensor networks based on well-known standards for data description. By using an underlying data dissemination protocol, such as SPIN or directed diffusion, our architecture provides a flexible and application-independent solution for sensor network design.

Recent works address naming and service discovery for heterogeneous networks of devices. Most of these works rely on IP-based communication, and do not consider dynamic and resource constraint environments such as sensor networks. Universal Plug-and-Play [14] uses a subset of XML to describe resources provided by devices. It is limited to TCP/IP networks. Service Location Protocol (SLP) [8] facilitates the discovery and use of heterogeneous network resources using centralized Directory Agents. Centralized repositories and fixed hierarchies do not fit well to sensor networks. Our proposal is totally distributed and based on lightweight protocols.

Our proposal has similarities with [2, 7, 20], which are database approaches for sensor networks. In [20] the sensor computation capabilities are exploited to execute part of the query processing inside the network, using query proxies. In [2] a SQL-like declarative language is proposed for users posing queries to a sensor networks. In [7] is proposed a sensor network architecture based on the concepts of virtual databases and data-centric routing. The main difference between such works and ours is that we propose a totally distributed service approach, based on ubiquitous standards. Exposing sensor functionalities as services offers a more flexible architecture when comparing to SQL queries. Besides that, our proposal addresses interoperability among different systems, which is not easily achived through a database approach.

4 System Architecture

Our work proposes an architecture for sensor networks based on the Web services technology. Web services are built according to the SOA pattern and they can be described by a trio of interoperability stacks [3]. Sections 4.1 and 4.2 describe the physical components of the proposed system and the roles played for those components in agreement with the SOA pattern. Section 4.3 describes the system elements according to the Web services interoperability stacks.

4.1 System Physical Components

In a generic sensor network, the component nodes can have different functionalities, listed below.

- Specialized sensor devices of different types: detect and collect specific environmental data.
- Routing nodes: receive and transmit data from/to neighbor nodes.
- Aggregator nodes: gather received data before transmitting, in order to save the transmission energy. The aggregation is accomplished through specific filters for each sensor type and for each application.
- Sink nodes: receive queries from applications and extract information from the sensor network to meet the queries.

In our system, the two main physical components are the sensor nodes and the sink nodes. A sensor node contains one or more specialized sensing devices. Furthermore, it has routing and aggregation capabilities. Thus, the routing function is distributed among all nodes. We assume that all the sensor nodes have processing and storage capacities enough to store and execute aggregation filters.

Sink nodes provide application interfaces through which external systems can obtain the information collected by the sensor network. Such interfaces can be accessed locally or remotely. Sink nodes can also aggregate data, but they do not have sensor devices. We assume that they are more powerful regarding to processing and communication capabilities than sensor nodes.

4.2 System Architecture According to SOA

A user application querying data from a sensor network plays the role of a service requestor. Sink nodes act primarily as service providers to the external environment. They provide the service descriptions of the whole sensor network, and they offer access to such services. At the same time, sink node act as requestors to the sensor nodes, requesting their specialized services, in order to meet the user application needs. Sensor nodes are service providers, providing data and filters. Sensor nodes send their services description to sink nodes, thus executing the basic publish operation. Sink nodes also act as registries, keeping a repository with services descriptions of each sensor type existing in the sensor network.

In our system, the functionality of the publish operation is accomplished through the `Publish_content` operation, and the functionalities of find and bind operations are both accomplished through the `Subscribe_interest` operation (see Section 0). Such operation is translated by the sink to a find operation followed by a bind with the sensor nodes that can meet the application request.

4.3 Interoperability Stacks

In our system, the **wire stack** is composed of the SOAP protocol and an underlying data dissemination protocol, the directed diffusion [5] protocol. The **description stack** is based on WSDL documents. The functionalities of **publish and discovery stack** are accomplished by a software module executing in sink nodes. Sections 0 and 0 detail the wire and description stacks. We do not describe the discovery stack in detail since it is not relevant for our current work.

4.3.1 The Wire stack: the communication framework. Users applications interested in submitting queries to the sensor network must access some sink node. Applications must generate a SOAP message describing the user interests. Such a message is generated based on the network WSDL documents stored in the sink repository.

All the communication inside the sensor network is accomplished using direct diffusion and formatted as SOAP messages. The sending and receiving of SOAP messages by a SOAP node is mediated by a binding to an underlying protocol. SOAP messages can be transported using a variety of underlying protocols. In our system we define a SOAP-Diffusion binding. To reduce the messages size, thus saving energy in sending/receiveing, the XML compact binary representation [16] is adoted for SOAP messages exchanged inside the sensor network

The SOAP module and the directed diffusion module must be present in every node in the network.

SOAP Module. The SOAP module in our system is composed of three main components: the SOAP engine, a set of handles and a binding with the underlying protocol. The SOAP engine acts as the main entry point into the SOAP module. It is responsible for coordinate the SOAP message's flow through the handles and for ensuring that the SOAP semantics are followed. Handles are the basic building blocks inside the SOAP module and they represent the messages processing logic. Three kinds of handles are defined. Common handles are responsible for messages serialization, header and attachments processing, , data type conversion, and others basic functions. The transport handle `Matching_Data` was specifically built for the sent and receive of messages through the directed diffusion protocol. The handle `Matching_Filter` represents the activation of application-specific filters inside the network. More details about the use of specific handles are described in Section 5 .

Sink nodes contain common handles only. Sensor nodes contain, besides common handles, the transport handle `Matching_Data` and the specific handle `Matching-Filter`.

Directed Diffusion Module. For the communication among all the sensor network components we use the directed diffusion protocol [5].

The current directed diffusion model [5] consists of a core diffusion layer, a diffusion library and the application layer, which includes applications and filters. The core diffusion layer is used to receive/send out packets from/into the network. The library provides a interface for the overlying application classes for publishing data and subscribing interests [12].

Our system uses the core diffusion layer as its basic data dissemination protocol. The functionalities of gradients configuration, matching data to interests and matching data to filters are part of the diffusion. We keep those functionalities but we change the representation model for data, interests and filters. Attributes describing data, interests and filters are represented through the WSDL language, and the matching functions are carried out by SOAP handles. Gradients and application-specific filters are implemented as software modules.

4.3.2 The Services Description Stack. The generic services provided by a sensor network are described through a WSDL document. In that document, port type elements contain the two types of service descriptions: services provided by sensor nodes and services provided by sink nodes. Each service port type contains operations. Operations contain parameters, defined in the document through messages and elements. Bindings of operations definitions to their concrete implementation should be defined according to the underlying protocol. A port identification, indicating the place containing the operation implementation, can be done through any unique identifier, as a device address.

The operations defined for the Web services specified in our system are: (i) **`Publish_Content`**, used by the sensor node to create and disseminate a SOAP message containing its service descriptions; (ii) **`Publish_Data`**, used by sensor nodes to create SOAP messages communicating generated data; (iii) **`Subscribe_Interest`**, used by an application to submit a query to a sink node; (iv) **`Subscribe_Filter`**, used by an application in a sink node to inject a new filter in the network.

5 System Description

Sensor networks have an initial setup stage comprising of four different phases: deployment, activation, local organization and global organization [13]. For energy saving, sensor nodes reside in a sleep state until the deployment, and they need to undergo an activation phase after they are scattered in the target area. The local organization phase includes the neighbors' discovery. During the global organization phase, nodes establish the communication path to some sink in the network. It is essential that all nodes reach a sink through some path so that their data can be delivered to the application. After the organization phase, each node is supposed to know and distinguish the nearby nodes. Any unique identifier can be used as a node identificator.

Our system operates according to three steps. Step 1 is the network initial configuration and it occurs during the local and global organization phases just described. Steps 2 and 3 are based on the working stages of directed diffusion protocol [5].

5.1 Step 1: Initial Set Up

In our system, during the organization phase, nodes exchange SOAP configuration messages, describing the services (data and filters) supplied by the them. Such messages include the node and network identification (the latter used when there are several interconnected sensor networks), a TTL (sensor time-to-live), sensor type (s), geographical location, current amount of energy, maximum and minimum confidence degrees, maximum and minimum acquisition intervals, filters existent in the node and specific information of each sensor type. The SOAP configuration message is broadcasted in the network using the diffusion core functionality. When a sensor node receives a configuration message, it can decide to transmit it or not. If the message describes a sensor type matching its own features or if a similar message has already been sent before, the node does not need to transmit it again. Sinks keep entries for each different sensor type, therefore their repositories scale with the number of sensor types.

Sink nodes store the content of received configuration messages in a local repository based on soft-state. Sinks periodically exchange messages, so that all the sinks in the network contain the same information.

Since configuration messages traverse intermediaries nodes until reaching a sink, such nodes can also store messages and exploit their content, for example, extracting geographic and energy information when disseminating interests through the network.

5.2 Step 2: Interest Advertisement

Applications requesting data from a sensor network should subscribe an interest in some sink. An interest contains the sensor type, the data type, the geographical location of interest, the acquisition interval and the acquisition duration. For time critical applications, a threshold value can be included, as a limit from which the sensors must inform data, regardless the current acquisition interval.

Applications can request the activation of application-specific filters existent in nodes. Furthermore, new filters can be injected as programs in the network. A filter contains an identifier and a list of data types with their respective values. The filter identifier is used to trigger the execution of a program already existent in the sensor node when such a node receives data matching the values specified in the filter. When injecting a new filter, the program filter itself is transported as an attachment in a SOAP message (SOAP attachment capacity [19]).

SOAP messages advertising interests are disseminated in the sensor network using the diffusion core and the diffusion gradient functionalities [12]. When a sensor node receives an interest message, the handle `Matching_Data` in the SOAP module verifies if the interest matches to some data provided by the sensor. The handle extracts from the message all the parameters needed for configuring the gradients, according to the diffusion model adopted [5].

5.3 Step 3: Data Advertisement

A sensor generates data in an initial acquisition interval specified in the configuration message. The sensor only sends SOAP data advertisement messages if there is some active gradient representing an interest matching its own data type. The sensor changes the acquisition interval according to the SOAP interest messages received. When detecting data for which it has received an interest, the sensor will issue a data advertisement message, which is delivered to the underlying diffusion protocol.

SOAP messages advertising data contain the data type, the instance (or value) of that type that was detected, the sensor current location (sensors can be mobile), the signal intensity, the confidence degree in the accomplished measurement, a timestamp, and the current sensor amount of energy.

The message dissemination involves a matching stage among data and interests, and the possible execution of filters. The matching data to interest stage is accomplished by the handle `Matching_Data` (step 2). The handle `Matching_Filter` matches data to filters and dispatches the filters execution whenever it is necessary. The resulting (possibly aggregated or filtered) data is delivered to the diffusion layer as a new SOAP data advertisement message to be disseminated along the network.

6 Conclusions and Future Works

In this paper, we presented a service based architecture for designing sensor networks. We claim that the future wireless sensor networks should provide a ubiqutuos, standardized access through a common and application independent interface. The contributions of this work are three-fold. First, we propose generic architecture where the data communication functionality is separated from the application-specific processing. Second, we have defined a Web services approach for wireless sensor networks, where sink nodes are modeled as Web Services that expose the services provided by the network using a standard service interface. Third, we propose the use of the WSDL language and SOAP protocol, already recognized as Internet standards, as the mechanisms for describing services and formatting messages used by the underlying communication protocol.

The proposed approach offers high expressiveness and flexibility when designing sensor networks. Our main goal is to provide the underpinning for building more general purpose networks, instead of strictly task-specific ones, in order to assist a large range of users, possibly spread over the world, sharing a common interest in a specific application area. Since energy saving is a key element in WSN design, our proposal makes an effort to keep the amount of spent energy in the same level as current WSN systems. In particular, the energy cost of communication is expected to be significantly higher than the cost of local computation [9]. Therefore, the additional processing needed for parsing SOAP messages should be insignificant to the system. For this reason, our approach addresses energy saving in data transmission by adopting a compact binary XML format in the messages exchanges inside the WSN.

7 References

[1] Fielding, R. et al., "RFC 2616. Hypertext Transfer Protocol - HTTP/1.1. June, 1999". Available in: ftp://ftp.rfc-editor.org/in-notes/rfc2616.txt.

[2] Govindan, R. et al., "The Sensor Network as a Database". Available in: ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/ 02-771.pdf, Tech-Rep 02-771 CS Department, University of Southern California, September 2002.

[3] Graham, S. et al., *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*. Sams Publishing, 2002.

[4] Heidemann, J., et al., "Building Efficient Wireless Sensor Networks with Low-Level Naming", in *Proc. of the Symposium on Operating Systems Principles*, pp. 146-159. Chateau Lake Louise, Banff, Alberta, Canada, ACM. October 2001. Available in: http://www.isi.edu/~johnh/PAPERS/ Heidemann01c.html.

[5] Intanagonwiwat, C., Govindan, R., and Estrin, D., "Directed diffusion: a scalable and robust communication paradigm for sensor networks", in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pp. 56-67, Boston, MA, USA, Aug 2000.

[6] Kulik, J., Heinzelman, R. B., and Balakrishnan, H., "Negotiation-based protocols for disseminating information in wireless sensor networks", *ACM Wireless Networks* 2000. Available in: http://citeseer.nj.nec.com/ 335631.html, 2000.

[7] Madden, S. et al., "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks". Available in: http://www.cs.berkeley.edu /~madden/ madden_tag.pdf.

[8] Perkins, C., "Service Location Protocol White Paper". Available in: http://playground.sun.com/srvloc/ slp_white_paper.html.

[9] Pottie, G. and Kaiser, W., "Wireless Sensor Networks", *Communications of the ACM*, 2000.

[10] Qi, H., Kuruganti, P. T. and Xu, Y., "The Development of Localized Algorithms in Wireless Sensor Networks", Invited Paper, *Sensors 2002*, 2, pp. 286-293, 2002.

[11] Römer, K., Kasten, O., and Mattern, F., "Middleware Challenges for Wireless Sensor Networks", *ACM Sigmobile*, vol. 6, no. 2, 2002.

[12] Silva, F., Heidemann, J., and Govindan, R., "Network Routing Application Programmer's Interface (API) and Walk Through 9.0, 2002". Available in: http://citeseer.nj.nec.com/ silva02network.html.

[13] Ulmer, C., Alkalai, L. and Yalamanchili, S., "Wireless Distributed Sensor Networks for In-Situ Exploration of Mars", Work in progress for NASA Tech-Rep. Available in: http://users.ece.gatech.edu/~grimace/research/reports/ nasa_wsn_report.pdf

[14] Universal Plug and Play: Background. Available in: http://www.upnp.com/ resources/UPnPbkground.htm.

[15] W3C (World Wide Web Consortium) Note, "Web Services Description Language (WSDL) 1.1". Available in: http:// www.w3.org/TR/2001/NOTE-wsdl-20010315.

[16] W3C (World Wide Web Consortium) Note, "WAP Binary XML Content Format". Available in: http://www.w3.org/TR/ wbxml/.

[17] W3C Recommendation, "Extensible Markup Language (XML) 1.0 (Second Edition)". Available in: http://www.w3.org/TR/REC-xml.

[18] W3C Recommendation, "XML Schema Part 0: Primer". Available in: http://www.w3.org/TR/xmlschema-0/.

[19] W3C Note on Simple Object Access Protocol (SOAP) 1.1, Available in: http://www.w3.org/TR/SOAP/.

[20] Yong Yao and J. E. Gehrke. "The Cougar Approach to In-Network Query Processing in Sensor Networks", *Sigmod Record*, vol. 31, no. 3, September 2002.