Chapter 11

Middleware Support for Wireless Sensor Networks: A Survey

Tales Heimfarth
Federal University of Rio Grande do Sul, Brazil

Edison Pignaton de Freitas
Federal University of Rio Grande do Sul, Brazil

Flávio Rech Wagner
Federal University of Rio Grande do Sul, Brazil

Tony Larsson
Halmstad University, Sweden

ABSTRACT

Wireless sensor networks (WSNs) are gaining visibility due to several sophisticated applications in which they play a key role, such as in pervasive computing and context-aware systems. However, the evolution of WSN capabilities, especially regarding their ability to provide information, brings complexity to their development, in particular for those application developers that are not familiar with the technology underlying and needed to support WSNs. In order to address this issue and allow the use of the full potential of the sensor network capabilities, the use of a middleware that raises the abstraction level and hides much of the WSN complexity is a promising proposal. However, the development of a middleware for WSNs is not an easy task. Systems based on WSNs have several issues that make them quite different from conventional networked computer systems, thus requiring specific approaches that largely differ from the current solutions. The proposal of this chapter is to address the complexity of middleware made for sensor networks, presenting a taxonomy that characterizes the main issues in the field. An overview of the state-of-the-art is also provided, as well as a critical assessment of existing approaches.

INTRODUCTION

The emerging use of WSNs in different application domains is increasing the importance of this research field. The complexity of issues such as dynamic routing, service discovery, resource management, data fusion, in-network processing, among others, are examples of what is being proposed as research efforts needed in this area. The support for applica-
tion development through a middleware layer for WSNs has also received a lot of attention recently. A middleware to support the programmability of WSNs is necessary due to the increasing sophistication of the applications running over them. These cooperative sensing related applications require adequate programming abstractions and services to enable a seamless programming.

The main challenge for programming a WSN arises from the large gap between the high-level functional specification of WSN applications and the complexity of the underlying infrastructure. This motivates search and strive for mechanisms that can simplify the programmability of these applications, hiding the mentioned complexity (Wang, Cao, Li & Das, 2008). Complexity arises mainly due to factors such as the dynamic network topology, constrained resources, and low-level sensor programming. These concerns are opposed to the requirements of the applications, such as flexibility, reliability, and reusability.

The complexity of the WSN infrastructure depends largely on which components and configurations that are selected to build it. Moreover, WSNs have a large design space, which adds additional challenges to the development of their applications (Römer, K. & Mattern, F., 2004). When programming applications, the developer must be aware of the configuration of the underlying network. Examples of very important such concerns about the infrastructure when developing WSN applications are: How are the deployment of the WSN to be made? Which mobility pattern the nodes will have? Which resources will be available in each node? To what extent will the network be homogeneous or heterogeneous? What are the characteristics of the communication device used? Is the network always connected or is it setup in a dynamic fashion at need?

Therefore, the middleware used must address the complexity of the WSN infrastructure and be aware of the different configurations of this infrastructure. In this context, useful abstractions to simplify the network programming should be provided by the middleware, and it should also be able to cover a large part of the WSN design space. However, it is very difficult to provide a homogeneous software platform that supports the development of different applications over different WSN infrastructures.

Key issues that must be addressed by middleware due to the different WSN infrastructures are: energy management, to cope with very small nodes; resource sharing, to use resources efficiently; topology control, due to the different possibilities of deployment; integration of different nodes, in order to cope with the heterogeneity; and adaptability, due to the dynamics of the environment.

From the point of view of the applications, the most important issue of the middleware is the programming model it provides. It should support and make easier the development of applications, while at the same time being as generic as possible. These are often contradictory goals. The abstractions provided to support the programming model are implemented by an underlying set of services.

Since different types of applications may be programmed on the WSN, as area surveillance, object tracking, environmental monitoring, and health care applications, different requirements of the programming model can be recognized. For example, for environmental monitoring a declarative language that allows the expression of queries is a suitable programming model, whereas an event-based programming model is more suitable for event supervision and tracking. It is a big challenge to develop middleware that can support many different types of applications due to the very different requirements that they impose on the programming model.

In this chapter, several state-of-the-art middleware techniques will be presented, taking in account the design space of the WSN infrastructure and the different types of applications. The focus is to provide an insight on how different middleware cope with the infrastructure complexity, giving a