Chapter 6
A Walk through Sensor Network Programming Models

Mukaddim Pathan
Australian National University, Australia & Telstra Corporation Limited, Australia

Doug Palmer
CSIRO ICT Centre, Australia

Ali Salehi
CSIRO ICT Centre, Australia

ABSTRACT
The widespread use of sensor networks in a range of applications has given rise to various programming approaches. It is a long standing concept that the application programmer of a sensor network should be shielded, as much as possible, from the questions of communications and systems architecture, that is, the low level details. In this chapter, the authors survey existing programming models for sensor networks and classify them by following a layered approach, based on their level of abstractions. With their approach they endeavor to lay out a clear guideline to choose the right approach for programming sensor networks for application development, thus potentially alleviating the heavy burden for programmers.

INTRODUCTION
Today, cheap and smart devices with multiple on-board sensors, networked with wireless links are available from research groups and companies. Due to the small size of these devices (also known as motes) pervasive computing concept is now far from just an imaginary idea. Sensor networks (Akyildiz, et al., 2002a; Yick, et al., 2008) gather data from the physical world and deliver it to a central node or sink where the data can be processed and analyzed. There are many applications of sensor networks, ranging from environmental monitoring to precision agriculture, animal tracking, volcano monitoring, weather forecasting, military situation awareness, structural monitoring, coal mine safety, medical emergency management, and traffic control. A sensor network typically has the following characteristics:
• Homogeneous or heterogeneous structure, comprising a large number of sensor devices, a small processing unit, and a wireless communications mechanism.
• Deployment of cheaper devices, rather than more powerful devices since data density and redundancy are key requirements (Costa, et al., 2007).
• Little or no infrastructure, beyond base station data sinks (Yick, et al., 2008).
• Short, low speed communication range between devices and the use of multi-hop routing to deliver messages. Communications may be asymmetric and unreliable (Woo, et al., 2003).
• Battery-powered devices with a limited energy budget, possibly backed by energy scavenging and solar cells (Estrin, 2001; Yick, et al., 2008).
• Unreliable devices that may stop functioning either due to physical damage or because they have run out of energy.
• Dense, overlapping data sets and in-network processing of data to reduce network contention and energy consumption.

With a fast moving world of building powerful sensors, the processing cost is becoming cheaper every day. Sensor devices are powerful enough to act like small computers autonomously managing their own resources and collaborating with other wireless nodes to provide detailed observations of the physical world. To rapidly build flexible sensing programs, researchers have proposed various programming models. A programming model is simply a development platform which constitutes a compiler for a domain specific language and a set of services. Services introduced by a programming model are the ones that are commonly required by the applications in which the programming model is intended to be used. As an example services which might be provided by a programming model are execution model (thread-based or event-based), routing algorithm and time synchronization. Research community has proposed a number of programming models for the wireless sensor networks. One of the most well known contributions of the research community to the programming models used for wireless sensor networks is the TinyOS system. Using the right programming model can have enormous impact in the development, deployment and maintainability of sensor networks. As for instance, choosing a programming model that is event-based requires a different set of expertise and imposes different type of restrictions compared to a thread-based programming model.

Motivation and Contributions

Existing literature reports a few reviews of prominent programming models for sensor networks. In early works (Hadim & Mohamed, 2006; Römer, 2004), authors have taken a broader view of sensor networks and attempt to develop a taxonomy and comparison of different approaches. Mottola and Picco (Mottola & Picco, 2010) present a survey of programming approaches for sensor networks and identify the fundamental requirements programming platforms must deal with. Rubio et al. (2007) provides a broader middleware-support classification by covering a range of approaches and highlighting some open research challenges and future directions. Chatzigiannakis et al. (2008) provide a categorization of middleware platforms for sensor networks, using functionalities and challenges that will play a crucial role in sensor network software development. In another work, Henricksen and Robinson (2006) undertake a survey of programming approaches for sensor networks focusing on software engineering abstractions. Most of the existing surveys are based on a taxonomy with only a few dimensions, by centering around node-centric programming and macroprogramming model. In this chapter, we present a goal-based categorization of sensor network programming models, i.e. what are the low to high-level system goals achieved by
Related Content

Aspect-Oriented Programming and Aspect.NET as Security and Privacy Tool for Web and 3D Web Programming
www.igi-global.com/chapter/aspect-oriented-programming-aspect-net/49524?camid=4v1a

Background on Context-Aware Computing Systems
www.igi-global.com/chapter/background-on-context-aware-computing-systems/170235?camid=4v1a

A Taxonomy of Online Game Security
www.igi-global.com/chapter/taxonomy-online-game-security/16910?camid=4v1a

Rich-Club Phenomenon of the Internet Topology
www.igi-global.com/chapter/rich-club-phenomenon-internet-topology/16891?camid=4v1a