Flexible Online Energy Accounting in TinyOS

Simon Kellner

System Architecture Group
Karlsruhe Institute of Technology
kellner@kit.edu

Abstract. Energy is the most limiting resource in sensor networks. This is particularly true for dynamic sensor networks in which the sensor-net application changes its hardware utilization over time. In such networks, offline estimation of energy consumption can not take into account all changes to the application's hardware utilization profile and thus invariably returns inaccurate estimates. Online accounting methods offer more precise energy consumption estimates. In this paper we describe an online energy accounting system for TinyOS consisting of two components: An energy-estimation system to collect information about energy consumption of a node and an energy-container system that allows an application to collect energy-consumption information about its tasks individually. The evaluation with TinyDB shows that it is both accurate and efficient.

Keywords: energy accounting policy tinyos.

1 Introduction

Energy still is the most critical resource in sensor networks. Limitations on energy supply as well as on other resources have led to operating system designs that offer only minimalistic hardware abstractions. The core of TinyOS, for example, is an event-based system that helps application developers in dealing with asynchronous hardware requests, and little else. One effect of this design decision is to make developers more considerate about hardware usage and therefore energy consumption. TinyOS makes it hard to actively wait for a hardware event to occur, while making it easy to react to the same event, which is the more energy-efficient approach in most situations.

One approach to designing sensor-net applications that meet pre-defined energy consumption requirements is to develop an application whose hardware utilization pattern is simple enough to allow predictions on the application's energy consumption. Global parameters of such applications can then be changed to accommodate energy consumption requirements. But the lack of convenient hardware abstractions does not necessarily limit developers in creating complex applications. A sensor network running the TinyOS-based TinyDB application, for example, allows users to issue (SQL-like) queries to the sensor network at a time of their choosing. Planning the energy consumption of nodes in this network can not be done a-priori, because the energy consumption characteristics of a node running TinyDB change with the queries it processes.

P.J. Marron et al. (Eds.): REALWSN 2010, LNCS 6511, pp. 62–73, 2010.

Control of energy consumption in this scenario is only feasible using online energy accounting on the sensor nodes. Information on the energy consumption of whole nodes, however, does not offer much information. An energy-intensive query might, for example, be only revealed by comparing node energy consumption before and after a query was sent into the network. Energy consumption of queries, on the other hand, can be readily used to decide if a query consumes too much energy and has to be canceled before it wears down the energy supplies of the sensor network.

This paper makes the following contributions:

- An online energy-estimation system for TinyOS that allows sensor nodes to become aware of their energy consumption.
- An energy container system for TinyOS that allows application developers to collect energy-consumption information about control flows in the application.
- A set of accounting policies that can be used to adapt the energy-container system to its purpose as set by the application developer.

The paper is structured as follows: After presenting related work in Sect. 2 we define a usage scenario in Sect. 3 that will be referenced later on. Then we present the design and selected implementation issues of the energy estimation system (Section 4) and the energy container system (Section 5). Section 6 details several accounting policies of our energy-container system. Following an evaluation of our systems in Sect. 7 we conclude with an outline of future work in Sect. 8 and closing remarks in Sect. 9.

2 Related Work

Management of energy in sensor networks has received significant attention in research over the last years, as it concerns the primary resource of such networks.

PowerTOSSIM [7] is similar to our own energy estimation system. It instruments OS components or simulations thereof to track power states and uses an energy model to compute energy consumption for one or more sensor nodes. PowerTOSSIM, however, targets off-line simulation, whereas our instrumentation and model are designed to be used in on-line energy accounting.

AEON [5] is the energy model used in the Avrora [8] simulator. It models the hardware's power states of a MICA2 node. Our energy model is based primarily on the MICAz node and additionally considers transitions between hardware states.

Schmidt, Krämer et al. [6] present another energy model used to make existing simulators energy-aware. Although they mention the potential to use their energy model in online energy estimation, they do not elaborate on that option further.

Dunkels et al. [2] present an energy-estimation system for the Contiki OS. This system is used to estimate energy consumption per hardware components. We employ a similar energy-estimation system and extend it with energy containers to a full energy-accounting system that is able to account energy based on control flows, which may span multiple hardware components.