The Journal of Systems and Software 103 (2015) 238–247

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Designing an open source maintenance-free Environmental Monitoring Application for Wireless Sensor Networks

Manuel Delamo, Santiago Felici-Castell*, Juan J. Pérez-Solano, Andrew Foster

Departament de Informàtica, ETSE, Universitat de València, Avd. de la Universidad S/N, 46100 Burjassot, Valencia, Spain

ARTICLE INFO

Article history:
Received 20 March 2014
Revised 6 February 2015
Accepted 7 February 2015
Available online 11 February 2015

Keywords:
WSN
Environmental monitoring
Open source
Reliability
Real deployment

ABSTRACT

We discuss the entire process for the analysis and design of an Environmental Monitoring Application for Wireless Sensor Networks, using existing open source components to create the application. We provide a thorough study of the different alternatives, from the selection of the embedded operating system to the different algorithms and strategies. The application has been designed to gather temperature and relative humidity data following the rules of quality assurance for environmental measurements, suitable for use in both research and industry. The main features of the application are: (a) runs in a multihop low-cost network based on IEEE 802.15.4, (b) improved network reliability and lifetimes, (c) easy management and maintenance-free, (d) ported to different platforms and (e) allows different configurations and network topologies. The application has been tested and validated in several long-term outdoor deployments with very good results and the conclusions are aligned with the experimental evidence.

1. Introduction

Wireless Sensor Networks (WSN) are distributed networks of small autonomous devices known as motes. In these networks, usually at least one mote called sink, acts as a gateway to connect with other networks. Each mote contains power supply, processing unit, memory, sensors and wireless communications but with many constraints, such as limited energy, bandwidth, memory size and computational capabilities. The motes communicate using multi-hop routing protocols. One of the most important areas in WSN research is the reduction in energy consumption, to allow extended maintenance free operating periods of several months or even years. But the design and implementation of WSN applications remain a nontrivial task.

Our goal is to develop a platform-independent data gathering application for Environmental Monitoring (EM) using existing open source components to create an application completely distributed with the following features: (a) runs in a multihop low-cost network based on IEEE 802.15.4, (b) improved network reliability and lifetimes, (c) easy management and maintenance-free, (d) ported to different platforms and (e) allows different configurations and network topologies. In a typical data gathering network all motes send information from their sensors to the sink.

We follow a complete analysis and design process, from the selection of the embedded operating system to the different algorithms and strategies. The proposed Environmental Monitoring Application (EMA) follows the IEEE 802.15.4 standard (IEEE, 2006) and it is designed to reduce the energy consumption, but maintaining at the same time the reliability of data transmission from source motes to the data sink through a multihop network. EMA has been designed to gather temperature (T) and relative humidity (RH) data, following the rules of quality assurance for environmental measurements and allowing different configurations in the gathering process (different sending periods: 30 s, 10 or 60 min, denoted by EMA (30 s), EMA (10 min) and EMA (60 min) respectively). EMA application is based on TinyOS-2.x (Hill et al., 2000) components and has been tested in several real long term deployments (for several months), in both research and industrial situations. We have designed EMA for the popular TelosB motes (Polastre et al., 2005), but it can run on other platforms as described in the paper. During the tests, we have used different motes with appropriate sensors, antenna and housing. The code of EMA is available at the TinyOS repository (Delamo et al., 2010a).

The rest of the paper is structured as follows. Section 2 shows the related work. Section 3 defines the requirements for accurate data gathering in meteorological studies. In Section 4 the application algorithm is analyzed and in Section 5 the selection of the embedded operating system to implement EMA is considered. Section 6 describes the design process of the application and Section 7 details porting to different hardware platforms. Finally, in Section 8 the real deployments in which EMA has been tested are described, before presenting the conclusions in Section 9.

* Corresponding author. Tel.: +34 963543563.
E-mail addresses: derama@alumni.uv.es (M. Delamo), santiago.felici@uv.es (S. Felici-Castell), juan.j.perez@uv.es (J.J. Pérez-Solano), anfos@alumni.uv.es (A. Foster).

http://dx.doi.org/10.1016/j.jss.2015.02.013
0164-1212/

2. Related work

In the last decade WSNs have been used in many different applications, such as environment, habitat, precision horticulture, seismic, volcano hazard monitoring, coil mine monitoring etc. Most of the following references are analyzed in Strazdins et al. (2013), a recent survey based on real world deployments. Mainwaring et al. (2002) was a pioneering project in WSN for habitat monitoring, based on an outdoor single hop network. Macroscope (Tolle et al., 2005) was another interesting outdoor WSN application for micro-climate monitoring of a redwood tree based on TinyOS (Hill et al., 2000) and TASK software (Tiny Application Sensor Kit) (Buonadonna et al., 2005), with an energy-efficient synchronous duty cycle of 1.3%. These references used motes from the Mica family produced by Crossbow.

In Hartung et al. (2006) a wireless system for monitoring weather conditions in large wildland fire environments is shown. Based on MantisOS with IEEE 802.11 nodes, it measures T and RH with a synchronized duty cycle of 10%.

Dozer (Burri et al., 2007) proposes an ultra-low power data gathering application, based on TinyOS, using TDMA MAC (Time Division Multiple Access Media Access Control) algorithm and multihop routing. Using a sampling period of 2 min, Dozer achieves an average duty cycle of less than 0.2%. On the other hand, Sensorscope (Ingelrest et al., 2010) helps in understanding the difficulties in EM in harsh conditions. This reference shows different WSNs, based on TinyOS, with a global synchronized duty-cycle MAC algorithm and multihop routing. Both Dozer and Sensorscope deployments use the Shockfish TinyNode.

GreenOrbs is a large scale operating WSN for forest monitoring, used in many publications for performance analysis. For example, Liu et al. (2013) is concerned with the scalability properties of WSN. The software is developed on the basis of TinyOS and the Collection Tree Protocol (CTP) (Gnawali et al., 2009) routing protocol using TelosB motes.

It is worth mentioning a complete custom solution, such as Lazarescu (2013) that is a design using the TI CC1150 radio chipset for sensor nodes, based on a 1 hop star topology extended through gateways, with an extremely low consumption (4.7 μA) for long-term monitoring.

In addition, we can find several commercial products for EM such as PowerCast Corp. (2014), Wireless Sensors (2014) and Memsic Inc. (2014), with constrained features and focused on specific target applications. In particular PowerCast Corp. (2014) and Wireless Sensors (2014) use extended star topology with powered gateways (routers). Also, it is worth mentioning the Dust Networks product (Linear Technology, Product Group, 2014) based on the IEEE 802.15.4e standard (IEEE, 2012), that is a modification of the IEEE 802.15.4 to support real time applications, and is WirelessHART-compliant (International Electrotechnical Commission, 2010). With Dust Networks we can implement both star and full mesh topologies with multihop routing, where the network manager continually optimizes the network in terms of routes, radio channel changes and TDM scheduling to maintain a high data reliability while providing many years of battery powered operation.

All these papers gave maturity to WSN and EM research but, in our opinion many things are still open. WSNs still remain a nontrivial task. Due to the complexity and low performance devices of these networks, very few real deployments can be found that are working maintenance-free (Strazdins et al., 2013). Following the requirements explained in Section 3 we have designed EMA. With the related work, we can find some points in common but with many differences. Some references are based on single hop WSN and others use custom software solutions, other differences come from the design for a specific motes platform, others require specific topologies (and centralized decisions to optimize the network performance) or use of TDMA MAC algorithm, that requires tight or even global synchronization. EMA does not require real time monitoring but does require an energy efficient operation, so in our opinion a contention-based MAC protocol with a locally synchronized duty cycle (between parent–child node pairs in the path to the sink) is a more suitable choice than a TDMA protocol that would require higher energy consumption and computational resources to implement the synchronization and optimization of time slots. EMA is completely distributed and it is not limited to specific topologies, such as star or full mesh topologies.

Furthermore, as discussed in Section 5, we will select TinyOS (Hill et al., 2000) as the embedded operating system for the motes, and if we check open-source applications in the TinyOS repository for EM and data gathering, we can only highlight Harvester (Lim et al., 2009), that has been unsupported since 2007. Harvester has been designed only for the TelosB mote with an interesting synchronized MAC, similar to WiseMAC (El-Hoiydi and Decotignie, 2004), using a synchronized version of the Low Power Listening (LPL) protocol (Moss et al., 2007). During the initial stages of our research we used Harvester, but soon realized that it was not completely reliable. In our opinion it is not maintenance free, because it has some bugs that make it crash after some time of operation and requires frequent manual resets of the motes, at least once a week. Also Harvester has great energy consumption because it sends many packets (for topology and status information, besides the EM data packets) with great overhead. Harvester is programmed using Sensor Network Platform Kit (SNPK) available also at the TinyOS repository, designed by the Computer Engineering and Networks Lab of the ETHZ (Zurich).

3. Analysis of the requirements for environmental monitoring

The first step in the analysis process is to determine the requirements for the application. Applications designed for environmental monitoring must collect raw data following a strict set of guidelines, such as those outlined in the US Environmental Protection Agency's Quality Assurance Handbook for Air Pollution Measurement Systems (US Environmental Protection Agency, 2008).

The Centre for Mediterranean Environmental Studies (CMES), a research institute in Paterna (Spain) that collaborates with us on the evaluation of the EMA, requires temperature (T) and relative humidity (RH) data to be collected via a specific method, described below, with the purpose of validating different meteorological models and detecting diverse meteorological phenomena, requiring long term measurements of several months. Basically these measurements are taken in outdoor deployments at random points. This method consists of recording values for T and RH at 10 min intervals, with each value being the average of 20 raw data samples, taken at 30 s intervals throughout the 10 min period. This method is based on the guidelines in (US Environmental Protection Agency, 2008), but with a higher data sampling frequency in order to provide greater accuracy and higher quality of the final results. To achieve good results we need at least 90% of valid measurements (The European Parliament, 2008).

To analyze the accuracy of the measurements, we have compared the T and RH mote sensors as described in Gallart et al. (2011), against a traditional meteorological station available in CMES. The single chip sensor SHT11 (included in TelosB mote (Polastre et al., 2005)) and SHT75 (connected externally to the motes), both manufactured by Sensirion (Sensirion AG, 2014), measure digitally T and RH with a nearly zero mean square error compared with the measurements from the station, using a proper housing as shown in Fig. 1. Also, to understand the response of these sensors to a wide range of phenomena, we have calibrated each sensor in a calibration chamber between 5 and 30° C and between 20 and 90 %RH. The calibration is done by a linear regression, where the different coefficients for each sensor are saved in the external database for later processing.