Demo Abstract: Flexible Hardware/Software Platform for Tracking Applications

Junaid Ansari, José Sánchez, Marina Petrova, Janne Riihijärvi, Ossi Raivio, Krisakorn Rerkrai, Christine Jardak, Frank Oldewurtel, Matthias Wellens, Lili Wu, and Petri Mähönen

Department of Wireless Networks, RWTH Aachen University
Kackertstrasse 9, D-52072 Aachen, Germany
Email: jan@mobnets.rwth-aachen.de

Abstract

In this demonstration paper we show and describe a flexible hardware and software platform for tracking applications. The architecture presented is extendible both on hardware and software sides, allowing for easy inclusion of sensors and signal processing algorithms of different types. During the demonstrations, examples of the software and hardware blocks developed within this generic architecture will be shown, and insight to the design choices and development issues will be given. Both outdoor and ”table-top” versions of the possible demonstrations will be described.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design– *Wireless communication, Distributed networks*

General Terms

Algorithms, Design, Performance, Experimentation

Keywords

Localization, Tracking, Wireless sensor networks

1 Introduction

Target tracking has always been one of *the* applications of wireless sensor networks. Several systems for this have been designed and demonstrated, especially in the context of vehicle tracking. However, as is common in WSN research, solutions described so far in the literature have been very bespoke, heavily optimized for a particular application.

In this abstract we describe a demonstration of a *flexible* and *modular* WSN platform for studying different tracking approaches for a variety of scenarios. The platform consists on the hardware side of a daughterboard that attaches to the MoteIV's Telos platform, allowing for a variety of sensors to be considered. In this case the daughterboard consists of magnetometer board and a PIR (Passive Infra Red) board as shown in Fig. 1. On top of the hardware platform, a flexible software architecture is specified enabling a number of different data processing and communications solutions to be employed. This architecture has been populated with a number of hardware and software components, allowing fault-tolerant, effective and accurate tracking applications to be created.

The rest of this abstract is structured as follows. In section 2, we give a brief overview of the hardware platform developed, followed by the discussion of the software modules actually deployed in the demo given in section 3. Description of the proposed demonstrations are given in section 4.

2 Hardware Design

Hardware design is clearly influenced by requirement of long autonomous operation typically found in WSNs. To achieve this, a very modular low power architecture has been chosen, where a daughterboard is populated with the necessary sensors and signal conditioning circuitry. The mainboard (MoteIV's Telos platform) provides the necessary computing and communication power.

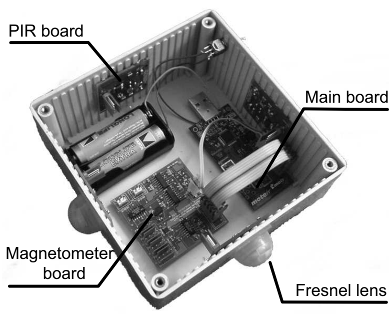

Figure 1. View of the sensor node.

Two different types of sensors are present in the present incarnation of the system in order to make fault-tolerant object detection and classification possible. First, a two-axis magneto-resistive sensor translates changes in the Earth's magnetic field into differential voltages, which then can be amplified using a classical instrumentation amplifier configuration. The digitally controlled potentiometer guarantees the possibility of calibration. Secondly, four pyroelectric sensors are placed orthogonally for 360-degree coverage and better estimation of object location. Fresnel lenses are used in our design in order to focus thermal energy, obtaining a longer detection range and an approximate beam-width of 90 degrees per sensor. Additional energy saving can be achieved by software control of the power routing mechanism included in the daughterboard. It is possible to individually control power supply of magneto-resistive and pyroelectric sensors.

3 Software Design and Implementation

The software architecture continues the modular approach adopted in the hardware platform design. All data gathering is accomplished via a sensor layer abstraction, making it easy to modify the exact sensor collection used in a particular application. The sensor layer interfaces with the data processing path via standard interfaces all the way from initial data filtering to (local) event detection. The event detection module in the end interfaces with communications subsystem, decoupling the data representation and sending process from the actual event detection and parameterizations. This makes it very straightforward to exchange different algorithms for both local and distributed event detection without disturbing other system components.

In addition to the architecture and interface specification, several example modules have been implemented within this framework. The hardware abstraction layer for the sensor board developed has been implemented together with basic filtering and event detection functions. Additional services such as time synchronization and routing necessary for most tracking applications are completely decoupled from the above software framework, and existing solutions can be used with no or only minor modifications.

4 Demonstration Description

The layered approach used in the software architecture development allows easy modification, reusability of various modules and adaptability to given application scenarios. The WSN software stack is implemented in TinyOS [1], whereas the data gathering algorithm at a gateway PC is implemented in C/C++ in the presented tracking application. The sensor network in our demonstration setup will use FTSP [2] for time synchronization and TinyAODV [3] for routing.

We intend to demonstrate the auto-calibration of the magnetometer circuit according to the environment and the variation in the earth's magnetic field strength. This feature enables our sensor platform to be used flexibly in a wide range of deployment setups.

The raw ADC samples from the sensor board are typically noisy and can result in unwanted detections if not properly filtered. The filtering scheme used for both PIR sensors and magnetometer employs a combination of efficiently implemented filters (like LP, EWMA and Gaussian filters) for noise suppression and also making the system resilient to the long-term drifts in earth's magnetic field strength. We would like to demonstrate the various filtering steps, the PIR and magnetometer signal characteristics of a passing object, the sensitivity of the sensor platform and the performance of our detection algorithm based on threshold crossing in dwell time. These demonstrations and the related visualizations can be accomplished in "table-top" fashion with a small number of sensor nodes connected to a laptop for data retrieval and visualization.

For demonstrating the output of our tracking algorithm for vehicle tracking application using our sensor platform, we also intend to deploy a small subset of our past sensor nodes deployments for an outdoor demonstration (We can also show some aspects and results from our existing large-scale demonstration (up to 100 nodes) that has been tested in realistic outdoor environments). The sensor nodes, deployed in a field of dimension $30m \times 5m$, will detect the vehicle. (The size of the sensor field can be varied according to the availability of space). The system setup for demonstration is depicted in the Fig. 2:

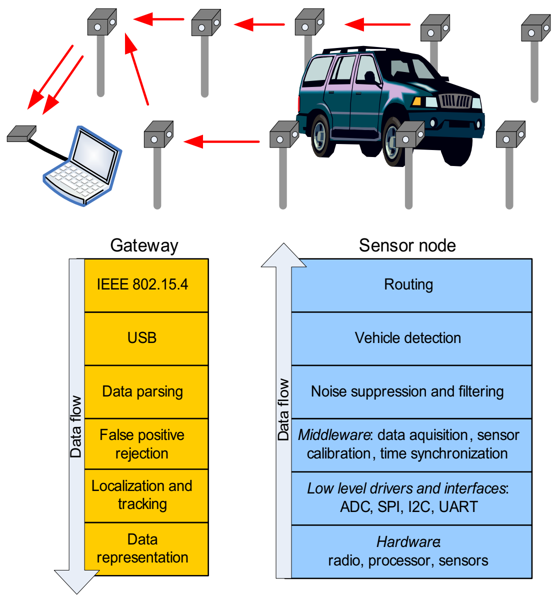

Figure 2. Vehicle tracking demonstration scenario.

A gateway node will collect the detected packets from the sensor nodes and pass these to a gateway PC, which executes the tracking algorithm. Before computing the position of the vehicle, the tracking algorithm employs a false positives rejection scheme to discard the undesired detection packets. The output of the vehicle tracking in the sensor field will be shown in real time on the gateway PC.

In addition, we will show how this system may be used for possible human tracking applications.

Acknowledgment

We would like to thank IABG GmbH and RWTH Aachen University for the financial support.

5 References

[1] TinyOS: An open-source Operating System for the networked sensor regime. http://www.tinyos.net/.

[2] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *Proc. SenSys '04*, New York, NY, USA, 2004.

[3] Tiny AODV. http://tinyos.cvs.sourceforge.net/tinyos/tinyos-1.x/contrib/hsn/. Last visited: 23.08.2006.