Poster Abstract: Energy Optimization Framework in Wireless Sensor Network

Niloufar P. Esfahani

Alberto E. Cerpa

Electrical Engineering and Computer Science
University of California, Merced
{nesfahani, acerpa}@ucmerced.edu

ABSTRACT

We present a holistic architecture for energy management in sensor networks. Our architecture is based on a model-driven approach which attempts to (a) establish functional relationships across different components of the software stack and the interrelated parameters based on empirical data, (b) use the maximum sensor value and time-synchronization errors acceptable by the users of the sensor network application as input to establish minimum quality of service requirements, and (c) optimize the parameter values of all the software modules within the node's application stack to minimize total energy consumption for each sensor node. We explore the trade-offs of the design space by using a non-trivial application that includes sensing, time synchronization and routing modules and show that when using our architecture, we can provide energy savings in the average of 37% to 76% while still maintaining quality of service both in terms of the expected sensing and time-synchronization errors. We further show that even when using modules that perform *significantly* better than others with default values (e.g. ORW $\gg$ CTP), we can still reduce overall energy consumption by properly adjusting the parameters of lowest performance modules and provide energy savings in the average of 30% to 43%.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communication; C.4 [PERFORMANCE OF SYSTEMS]: Performance attributes

Keywords

Component, Energy, Optimization

1. INTRODUCTION

There have been many efforts geared towards improving energy efficiency of different components of the system including data sampling, query processing, radio communication and processor duty cycling. However, these individual components may not be flexible to adapt to changing network and application dynamics. Perhaps more importantly,

Figure 1: System Architecture

each component is designed to optimize energy consumption from a component-centric point of view, without considering the inter-relationships among the different components in the sensor stack. We take a fresh look at this problem by considering the energy trade-offs caused by the use of multiple components simultaneously, and provide a framework that optimizes the overall system energy consumption in a holistic way, based on the inter-relationships among components and the user constraints that bound the expected application quality of service.

In many applications, most of the energy consumption of wireless sensor networks comes from radio communication. In our work, we use the number of packet transmissions (TX) as a proxy that approximates energy consumption¹. Our goal in this work is to reduce the number of packet transmissions (TX) while maintaining quality of service constraints based on user input. We do this by finding functional parametrized relationships based on empirical data among the different components of the sensor software stack, and minimize the TX based on user constraints. Our goal is to find the optimal parameters that allows minimization of the system energy consumption.

2. SYSTEM ARCHITECTURE

We have used a non-trivial application consisting of three core components: time synchronization, sensing and routing modules. These components are representative of multiple data collection sensor network applications and provide a good sample of the design space. We use FTSP [4] for time synchronization between different nodes, and CTP [2] for multi-hop radio communication. The sensing application is similar to MAUVEDB [1]. Sensor nodes sense an environmental parameter (e.g. temperature) over time, and build their own temperature model. Each node builds a simple linear regression model between the temperature and time, and calculates the best fit. If the difference between a value

¹We plan to incorporate radio duty-cycling and actual energy measurements in future work

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author(s). Copyright is held by the owner/author(s).

SensSys '15, November 1–4, 2015, Seoul, South Korea..

ACM 978-1-4503-3631-4/15/11.

DOI: <http://dx.doi.org/10.1145/2809695.2817904>.

sensed by the node and the model prediction is larger than a specific threshold defined by the user, then the application forwards the outlier data sample towards the sink, and the node recalculates the temperature model. While a sensor node could send every sample value for maximum accuracy, there is a trade-off between the number of transmissions required (and their energy usage), and the accuracy that can be achieved. By optimizing only for the required user's accuracy needed, we can minimize the overall energy consumption. Similarly, time synchronization accuracy and data error thresholds can be manually-defined by the user, to ensure that minimum application requirements are met while energy is minimized.

Figure 1 shows the overall system architecture. By deploying the application over different sensor nodes, and using arbitrary empirical data for different parameters of the system (e.g. beacon frequency, sampling frequency, time synchronization rate, and error threshold) in acceptable ranges to see how they affect the packet transmissions (TX), we calculate the relationships between them for different modules. TX is tracked for each module in the application including specialized transmissions for time synchronization, beaconing for route discovery, and transmitted data packets for sensing application in each component. As an example, by increasing beacon frequency, more beacon packets are transmitted to maintain the routes (TX in routing component increases), but the improved links make fewer data packet transmissions necessary in sensing application (TX in sensing component decreases). This trade-off in the objective function is an example of why we need to obtain the optimal values in the system. The *Objective Function Calculator* module then sums up these usage statistics and calculate the objective function in linear and non-linear formats (linear and non-linear relationships between TX and related parameters). Once calculated, the objective function along with user-defined constraints are sent to the *Optimizer* module, which calculates optimal values of operation. Parameters are then given to the *Parameter Collector and Distributor* module, which disseminates them to all sensing nodes.

3. EXPERIMENTAL EVALUATION

In our analysis, we determine how the inclusion of our optimized operation parameters affect the amount of radio transmissions that occur, as well of the merits of choosing a linear or non-linear objective function. In order to test how good are the linear and non-linear objective function approximations derived above, we run a 20 minute experiment where we test the number of total transmission packets by using the parameters obtained by each, constrained by different maximum time error and data error thresholds specified by the user. Our experiment consists of a wireless network testbed, consisting of 57 sensor nodes, distributed through a building. Across all time error constraints from 0.5 to $3\mu s$, we see an average of 37% packet reduction using linear objective function parameters, and an average of 47% reduction across a data error range of 1% to 10%. For non-linear objective function we see an average of 76% packet reduction across all time error constraints and in the average of 64% reduction across data error constraints. This is substantial improvement and the user-defined constraints provided to the optimizer ensures that the quality of service of the sensing, time synchronization, and routing are all maintained.

Figure 2: Comparing results of using ORW with CTP as user specified threshold of a) data error changes (top), b) time error changes (bottom)

Another commonly-used data collection communication protocol is ORW [3]. Although similar to CTP in purpose, more advanced duty-cycling allows it to use less energy. We chose to compare the standard ORW network usage to CTP using our optimization framework, as shown in Figure 2. The x axis in Figures 2a and 2b represents the user-defined maximum threshold for data and time error, respectively, and the y axis represents the “cost” of the strategy, in packet transmissions. As shown, in default operation and even using parameters determined by the linear objective function, CTP sends significantly more packets to maintain the same quality of service with ORW. However, by using parameters determined by the non-linear objective function, it is shown to greatly reduce packet transmissions, allowing our application with CTP to reduce packet transmissions from 50% to 10% (average 30%) over a data error range of 1 to 8% (ORW is better than our application for the data error from 8% to 10%), and from 30% to 64% when changing the maximum time error from 0.5 to $3\mu s$ (average 43%). Our preliminary results show that application efficiency can be greatly improved by performing local optimization.

4. CONCLUSIONS

We have presented an architecture for energy management of the wireless sensor nodes in a holistic manner. Our framework is based on a model-driven approach which has 3 steps: 1) Calculating functional relationships among different components and their parameters. 2) Considering the user constraints. 3) Building an objective function based on the functional relationships among components and optimizing based on user constraints. We obtain energy savings in the average of 37% to 76% compared to the components default values. We also show that even when using ORW, which performs better than CTP, we can still reduce overall energy consumption by using the proper parameters and provide energy savings in the average of 30% to 43%.

5. REFERENCES

- [1] A. Deshpande and S. Madden. Mauvedb: Supporting model-based user views in database systems. SIGMOD '06, pages 73–84, New York, NY, USA, 2006. ACM.
- [2] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. SenSys '09, pages 1–14, New York, NY, USA, 2009. ACM.
- [3] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson. Low power, low delay: Opportunistic routing meets duty cycling. IPSN '12, pages 185–196, New York, NY, USA, 2012. ACM.
- [4] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi. The flooding time synchronization protocol. SenSys '04, pages 39–49, New York, NY, USA, 2004. ACM.