Lessons from a Pilot Deployment of Energy Efficient Data Collection Protocols in Wireless Sensor Networks

C Edordu and Y Yang
Department of Electrical and Electronic Engineering, University College London, Gower Street, WC1E 6BT, UK
Email:c.edordu@ee.ucl.ac.uk,y.yang@ee.ucl.ac.uk

Abstract. This study reports results of, as well as experiences and engineering challenges encountered, during a pilot deployment of a Dual Prediction and Probabilistic Scheduler (DPPS) for energy efficient data collection (EEDC). The deployment was motivated by sensor network supplier, Senceive Ltd., whose researchers wanted to incorporate more intelligence into their firmware to facilitate more cost effective remote asset management. The process of embedding and deploying DPPS, highlighted the importance of re-evaluating the assumptions typically overlooked in computer simulations and emphasises the need for sensor network researchers as a whole to reconsider the usefulness of EEDC protocols through real-life experimental deployment.

1. Introduction
General energy efficient data collection (EEDC) protocols in wireless sensor networks are used in a plethora of environments including habitat monitoring [1] as well as building monitoring and control [2]. In remote locations, energy consumption is highlighted as a critical challenge because sensor nodes have a finite battery life which when expiated, is expensive or impossible to replace [3]. Therefore effective management of sensor resources in a network requires essential techniques to extend the operational lifetime and enhance the cost effectiveness of sensor node deployment. In order to attain such energy conservation, the conventional practice within the research community is to limit radio usage which is assumed to account for the highest proportion of energy consumption [4]. Further savings are made by reducing the amount of sensing which can be as energy consuming as data transmission. For example specialised low power measurement units such as airflow sensors, pressure sensors and accelerometers, consume between 50-60mW for sensing tasks which is similar to the power used for transmission in MICA2 nodes [5].

In 2008, Senceive Ltd., a wireless sensor network (WSN) supplier and firmware provider, was interested in embedding more intelligence into proprietary mesh networking protocols to facilitate more energy efficient monitoring of phenomena. This is achieved with data collection algorithms of which there are two main kinds: event-based and periodic data collection algorithms. In event-based data collection, the amount of communication used by the network is reduced by suppressing unnecessary measurements. Event-based protocols are evaluated in terms of the radio usage as reflected by the ratio of the number of sent samples to the total number of possible measurements. Using an event-based policy conserves vital battery support and enhances the longevity of a data...
collection system. Alternatively, a periodic data collection policy may be used whereby periodic samples are transmitted. Periodic collection is advantageous because it enables greater flexibility in analysis of raw sensor data. Period-based collection protocols are evaluated in terms of sensor usage which is the ratio of the number of selected readings to the total number of possible measurements.

The work presented in this paper is concerned with the embedding and deployment of a Dual Prediction and Probabilistic Scheduler (DPPS) [6]. DPPS builds on earlier work done in eSENSE [7], where the authors developed an energy efficient stochastic scheduler. By imposing accuracy constraints in the collection process and using the same prediction model at both the sensor side and the server side, DPPS enhances reductions in the sensor usage and radio usage. The rest of this paper is organised as follows: motivation behind this study is discussed in the next section. Section 3 covers the experimental setup including the hardware and software used. In section 4, the results of the deployment are discussed and Section 5 reports the lessons learned from the deployment. Section 6 makes concluding remarks.

2. Application Context and Motivation
This preliminary practical study was carried out because Senceive Ltd., a company which manufactures wireless sensor networks typically for industrial and environmental monitoring, was interested in building more intelligence into their networks by using algorithms that could adapt data collection so as to limit network traffic and also extend battery life on sensor nodes. Energy is conserved in the collection process because some constraints on accuracy can be accepted whereby communication is reduced by filtering and only important data are transmitted.

A dual prediction scheme (DPS) is important for achieving energy conservation because it compares predictions \( \hat{X}_t \) against real samples \( X_t \) so that any deviations are bounded at the server by a maximum error threshold \( e_{\text{max}} \) [8]. This is possible because each sensor node both measures and forecasts readings. For example, at time \( t \), a forecast function \( f(X_t) \) predicts reading \( \hat{X}_t \) at both the sensor and server ends. The sensor then takes an actual reading \( X_t \). If the deviation \( \hat{e}_t = |X_t - \hat{X}_t| \) is within the predetermined error threshold \( e_{\text{max}} \), the forecast \( \hat{X}_t \) is accepted as satisfactory. \( \hat{X}_t \) is active at the server, that is it is used as a reading in which \( \hat{X}_t \approx X_t \), and no transmission is made from the sensor. Energy is saved because transmission to the server only occurs when the deviation \( \hat{e}_t \) at the sensor exceeds \( e_{\text{max}} \). To further conserve energy, DPS can be combined with a scheduler as witnessed in DPPS [6], which adapts the interval between readings so that only important data are collected.

3. Experimental Setup
The empirical implementation of the data collection algorithms were tested in an indoor laboratory environment where temperature in degree Celsius was measured using several PICDEMZ boards setup as shown in Figure 1(a). One sensor node periodically sampled temperature data at a default rate of one second while the other nodes were fitted with either DPPS or eSENSE, the parameters of which were calculated using methods described [6] or [7] respectively. Each transmitted packet contained the node identification number, time, report number (which is the total number of samples measured) and temperature. In total the experiment was run for four hours: two hours for model building and another two hours for evaluation.
3.1. Hardware
The hardware components consisted of Microchip PICDEMZ boards as shown in Figure 1(b). The PICDEMZ demonstration kit is a user friendly platform for ZigBee application design and development. The demonstration board is fitted with an RJ11 connector which provides an interface to Microchip’s MPLAB ICD 2 Debugger. The ICD 2 Debugger allows developers to reprogram or modify code on board the PIC 18F4620 microcontroller unit flash memory. Also part of the demonstration kit is Microchip’s MPLAB IDE which provides the platform for writing and debugging code for application development. Every PICDEMZ board was powered by 9V-170mAh rechargeable batteries. The MCU contained 1.5K RAM, 32K programme memory and 1024 bytes EEPROM. The EEPROM memory was the most critical storage component because it was used for programming the scheduling algorithms as well as storing 360 bytes of parameters.

Other important peripheries located on the board were:
- Temperature sensor - A TC77 thermal sensor was used for temperature data collection
- LED – An LED was also used to check the integrity of the data collection algorithms. This was done by setting the LED to flash when measurements were taken.
- Radio – A CC2420 radio was used at 2.4 GHz. The radio consumed 18.8mA and 17.4mA for receiving and transmitting data respectively.

3.2. Firmware
A simplified version of Senceive Ltd.’s FlatMesh firmware was used for implementing the various algorithms onboard the PIC18F4620 microchip. Senceive Ltd. assisted with this study by providing both the hardware and a simplified version of their FlatMesh firmware for testing the algorithms. The stripped down FlatMesh firmware was geared for point to point communications in a star topology (see Figure 1(a)).

The firmware was used because its modular architecture abstracted the physical layer processes and networking features from the functions in upper layers. In particular the top level application layer
allowed the development and customisation of application based algorithms without knowledge of the underlying framework in lower layers. The firmware was written in C using Micropchip’s MPLAB integrated development environment with a C18 compiler.

4. Experimental Results and Performance Evaluation

As indicated in earlier discussions, the empirical evaluation of performance is done in terms of sensing and radio usage respectively. The results shown in this section correspond to periodic data collection (Figure 2(a)-2(b)) and event-based collection (Figure 3(a)-3(b)).

Figure 2(a) shows time series data as collected periodically when $e_{\text{max}} = 0.1$ for both DPPS and eSENSE protocols. Recall that using the periodic data collection policy, collected measurements are reported to the server for model update and storage. As $e_{\text{max}}$ increases, less data are collected and hence more energy is saved because the sensors spend more time switched off between sensing. Figure 2(b) shows that DPPS has a lower average usage percentage than eSENSE between $e_{\text{max}} = 0.15$ and $e_{\text{max}} = 0.30$. At $e_{\text{max}} = 0.15$, DPPS has an average usage percentage of 6.1% compared to 11% for eSENSE. This indicates that the sensor is used less and hence conserves more energy than eSENSE.

Figure 2.

(a) Temperature time-series as collected using DPPS and eSENSE for a periodic sampling policy
(b) Average sensor usage percentage of DPPS and eSENSE using the periodic sampling policy

Figure 3(a) corresponds to the time series of data collected using DPPS and eSENSE when event-based sampling is used. In event-based sampling, data are transmitted when the deviation between forecasts and real readings are greater than $e_{\text{max}}$ otherwise transmissions are suppressed.

Figure 3(b) shows the transmission percentages at various error thresholds using DPPS and eSENSE. At $e_{\text{max}} = 0.1$, DPPS has a higher transmission percentage than eSENSE because sensor readings are more likely to deviate from predictions thus prompting increased data transmission. At $e_{\text{max}} = 0.15$, this chance decreases more sharply with DPPS thus a lower transmission percentage is recorded using DPPS compared to eSENSE. When $e_{\text{max}} \geq 0.2$, the transmission percentage of both DPPS and eSENSE are similar because the $e_{\text{max}}$ used is large. In general although DPPS has a lower sensor usage (see Figure 2(b)), results in Figure 3(b) indicate that transmission percentages and thus energy consumption are equivalent for high levels of $e_{\text{max}}$. 
5. Lessons Learned

The practical challenges encountered during the experimental deployment of DPPS highlight the importance for EEDC protocol designers to go beyond computer simulations and consider the ease of implementation of proposed algorithms in real-life. The issues discussed below support this contention:

i. Random number generation - A basic random number generator was implemented of the type inspired by linear congruential methods in which the random number is calculated from its predecessor. In practice this method tended to generate low order bits whose occurrence were less randomised compared to computer simulations. This problem was alleviated by adding a weight to the random number generation process such that the occurrence of higher order bits is more likely [9].

ii. Sensor calibration - Adjacent sensor nodes registered different temperature readings despite being in almost the same location. In computer simulations differing readings are attributed to measurements from separate locations in space. In practice however this may not necessarily be the case. Thus during the deployment, the presence of a central cooling system within the confines of the experimental area influenced sensor readings from neighbouring nodes. This problem was partially alleviated by shielding the nodes from the overhead cooling system using a card box and rotating the sensor nodes so that the onboard sensors were closer together in space. This facilitated the collection of similar readings from nearby nodes.

iii. Packet collision - Although conventional computer simulations for sensor networks assume the presence of a robust collision avoidance mechanism, the application of DPPS did not involve the use of such a scheme during implementation. Rather it was found that an effective method of reducing collisions is to add a random time within the order of a hundredth of a second to the scheduling interval so that no two nodes transmit data simultaneously.
iv. Synchronisation - Whereas simulations assume the synchronisation of all node times, the clocks onboard each node were not synchronised. This caused varying degrees of inaccuracy in the onboard watch crystals and is a direct consequence of the fact that prototypes with hand-soldered components were used. This problem introduced a small phase shift between readings collected from adjacent nodes.

v. Outliers and data corruption - Approximately 1% of packets received were corrupted such that temperature and timing information contained erroneous values. For example some temperature data were too high and caused spikes in the time series plot. Furthermore some corrupted data were received in incorrect formats. While this problem is not critical, the fact that it exists is totally ignored in computer simulations.

6. Conclusion
This paper practically evaluates the performance of energy efficient data collection algorithms in sensor networks. Several challenges were uncovered during real-life deployment which computer simulation alone would not disclose. For example the imperfection of random number generators in sensor boards, poor sensor calibration, packet collision and lack of synchronisation all affected the energy efficiency of the data collection algorithms. The main lesson learned is that these problems are often wrongly assumed to be either resolved or negligible in computer simulations. If energy efficient data collection protocols are to be more effectively deployed, scientists in the WSN research community must strive to use real-life experimentation to evaluate the usefulness and limitations of data collection protocols.

Acknowledgement
The authors would like to thank fellow members of the Communication and Information Systems research group at UCL for helpful suggestions throughout this study. Furthermore the experimental deployment would not have been possible without the cooperation of people from Senceive Ltd, who not only kindly provided the hardware, but gave guidance on how to use it with Senceive Ltd’s Flatmesh firmware.

References
[1] Sikka P, Corke P, Valencia P, Crossman C, Swain D and Bishop-Hurley G 2006 IPSN ’06: ACM/IEEE International Conference on Information Processing in Sensor Networks.
[2] Deshpande A, Guestrin C and Madden S R 2005 Bulletin of the IEEE Technical Committee on Data Engineering 28 40–47 More references
[3] Wang N, Zhang N and Wang M 2006 Computers and Electronics in Agriculture 50 1–14
[4] Pottie G J and Kaiser W J 2000 Communications of the ACM 43 51–58
[5] Honeywell 2008 Honeywell Sensor Datasheet URL http://www.honeywell.com/sensing
[6] Edordu C and Yang Y 2009 Wireless ViTAE ’09: IEEE International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology
[7] Liu H, Chandra A and Srivastava J 2006 IPSN ’06: ACM/IEEE International Conference on Information Processing in Sensor Networks
[8] Borgne Y A L, Santini S and Bontempi G 2007 Journal of Signal Processing 87 3010–3020
[9] Bramer B and Bramer S 1997 C for Engineers 2nd ed (Arnold)