Wireless Sensor and Actuator Networks for Energy Efficiency in Buildings

Domenico De Guglielmo
Dept. of Information Engineering
University of Pisa, Italy
E-mail: d.deguglielmo@iet.unipi.it

Giuseppe Anastasi
Dept. of Information Engineering
University of Pisa, Italy
E-mail: giuseppe.anastasi@iet.unipi.it

Abstract — Residential and business buildings account for a very large fraction of the world-wide energy consumption. To improve their energy efficiency, building management systems (BMS) – based on (wireless) sensor and actuator networks – have been proposed. To be effective a BMS must be *responsive*, *robust* and *scalable.* Since its performance is mainly determined by the underlying sensor and actuator network, in this paper we focus on the communication between sensors and actuators. Specifically, to minimize congestion, latency and energy consumption, we propose a de-synchronization algorithm that is able to arrange, dynamically, periodic transmissions from different sensor nodes in a round-robin collision-free style, like in conventional TDMA. Unlike TDMA, however, it does not require synchronization, and is able to adapt to changes in the network topology. Our preliminary results show that the proposed algorithm converges to a steady-state in a limited number of periods.

I. INTRODUCTION

It is estimated that residential and commercial buildings account for approximately 20% of the overall world energy consumption [1]. In USA and Europe this percentage is even much larger, i.e. around 40% [2]. About one third of this energy consumption is wasted due to an improper use of electrical appliances, and could be saved without lowering the level of comfort perceived by people [3]. However, while significant energy savings can be achieved by providing users appropriate feedbacks on personal energy consumptions [4], only relying on people's willingness may not be an effective approach [5]. A more effective solution is using a *Building Management System* (BMS) that automates the energy saving of electrical devices in a building. To this end, a number of solutions, based on wireless sensor and actuator networks, have been proposed [6, 7, 8, 9, 10].

To be effective, a BMS should be *responsive*, *robust* and *scalable* [9]. A key role in determining the performance of a BMS is played by wireless sensors that acquire environmental and context information and report them periodically to a central server or a local actuator node, for decision making. Since the number of deployed sensors in a building may be very large, interferences and congestion can arise, compromising the responsiveness and reliability of the BMS. In addition, environmental sensors are typically powered by batteries with a limited energy budget, while their required lifetime is in the order of several years. Hence, communication protocols should guarantee the reliability and latency required by the application with the minimum energy consumption, even when there is a large number of sensor nodes. In addition, standard solutions are desirable to allow the inter-operability with existing systems. Recently, two standards for wireless sensor networks (WSNs) have been released by the IEEE 802.15.4 WG [11] and ZigBee Alliance [12] respectively, and products compliant to them are largely available on the market. However, a number of studies have emphasized severe limitations in the IEEE 802.15.4 MAC protocol, in terms of *reliability*, *scalability*, *timeliness*, and *energy efficiency* [10], which make it unsuitable for BMSs with a large number of sensor nodes. In terms of timeliness, congestion avoidance and energy efficiency, *Time Division Multiple Access* (TDMA) is considered the optimal solution. However, it requires a strict synchronization among nodes and has a limited *flexibility*, i.e., a change in the network topology may require a new transmission schedule. Recently, de-synchronization algorithms [13] have been proposed for efficient periodic data reporting. De-synchronization can be used to arrange periodic transmissions from different sensor nodes in an interleaved, round-robin style – so as to minimize contention and energy consumption – like in conventional TDMA. Unlike TDMA, however, de-synchronization algorithms do not require a strict synchronization among sensor nodes [13].

Several de-synchronization algorithms have been proposed in the literature, both for single-hop [13, 14] and multi-hop [15, 16] WSNs. However, in all the proposed approaches, sensor nodes adapt their behavior on the basis of information received from other nodes. This makes them vulnerable to packet losses. Also, they are not energy efficient, as sensor nodes need to be active more than necessary (to receive information from other nodes), thus consuming energy. To overcome these limitations, we are considering a *Localized De-Synchronization* algorithm which relies only on local information. This reduces the energy consumption and makes the algorithm suitable for environments where packets can be corrupted or missed. The rest of the paper is organized as follows. Section II describes the proposed Localized De-Synchronization algorithm, while Section III presents some preliminary results about it. Conclusions are drawn in Section IV.

II. DE-SYNCHRONIZATION ALGORITHM

We consider a single-hop sensor network, where each sensor node has to report data periodically to a sink node (i.e., central server or local actuator node). Without losing in generality, we can assume that the reporting period, T, is fixed and common to all sensor nodes. We also assume that each node has to report one data packet per period. In order to obtain a collision-free schedule, sensor nodes should select non-overlapping time intervals for transmitting their packets (as in the conventional TDMA). This is easily

978-3-901882-46-3 © IFIP, 2012

achieved if sensor nodes can exploit some centralized information (e.g., an allocation pattern notified by the sink node), or exchange information with other sensor nodes. Instead, our localized algorithm takes a different approach as it provides a *simple*, *fully localized,* and *adaptive* mechanism through which sensor nodes can *autonomously* decide the time interval within the period T to be used for transmitting their packet, in order to obtain a collision-free schedule. Specifically, the goal of our algorithm is to adaptively and quickly converge to a schedule where sensor nodes use non-overlapping transmission intervals.

The algorithm operates at the application layer and assumes that the reporting period T is divided into N_s slots, of equal duration. As shown in Figure 1, the length of a *slot* is such that it can accommodate the transmission of a data packet and the related acknowledgement. Before using a slot for data transmission, a node has to achieve the right to use it. To this end, there is a preliminary de-synchronization phase where sensor nodes contend for acquiring a certain slot. At the end of this phase, each node has acquired the right to use one slot. To implement contention, during the de-synchronization phase slots are accessed by sensor nodes with a different access pattern, shown in Figure 2 (the slot size is the same in both phases).

Figure 1. Slot access pattern during the data reporting phase.

Figure 2. Slot access pattern during the de-synchronization phase.

Basically, during the de-synchronization phase, nodes contending for a generic slot σ, try to transmit a *fake* packet in that slot, using the contention period. If a sensor node wins the contention (i.e., it transmits successfully), then it is authorized to access slot σ in all subsequent periods *without* contention (i.e., slot σ is viewed as its data slot) to transmit its data packets. A complete de-synchronization is achieved as soon as all sensor nodes have achieved their own slot. It must be pointed out that, although time is slotted, a strict synchronization between sensor nodes is not required (see below).

Ideally, the contention is always solved – i.e., there is always one winner, irrespective of the number of contending nodes. Under this assumption, the most quick method to de-synchronize sensor nodes is to let they *all* contend for each slot. As a result of each contention, one node is accommodated, while the remaining ones contend for the next slots. Hence, the de-synchronization process takes just one period. In practice, this scheme is unfeasible and we can only approximate it. In our algorithm we use a *random backoff time* to solve contentions, and assume that it can take a number of discrete values (hence, collisions can occur). After waiting for the chosen backoff time, a competing sensor node i checks the status of the channel. Three possible outcomes can occur, namely (**i**) *successful transmission*, (**ii**) *busy channel*, and (**iii**) *collision*. If a successful transmission occurs, node i is the winner of the contention and, hence, it acquires the right to use slot σ in all subsequent periods. In addition, to get priority over all the other nodes, in the next periods node i will always access slot σ with a backoff time equal to 0 (i.e., hereafter, slot σ will be viewed by node i as a data slot). This also reduces energy consumption and packet latency at node i. If the channel is found busy after the backoff time, it means that one or more sensor nodes have generated a shorter backoff time. Thus, node i has to try the next slot (i.e., $\sigma+1$) in the current period. Finally, when a collision is experienced by node i, it means that two or more nodes have selected the same backoff time for contention in slot σ. In principle, node i could either retry the next slot (i.e., $\sigma+1$) in the current period, or retry the same slot (i.e., σ *)* in the next period. The rationale behind the latter option is that if the number of colliding nodes is limited, the contention will be very likely solved at the next period. Another option for a colliding node i, would be re-trying slot $\sigma+1$ in the current period with probability p_r and defer contention to slot σ in the next period with probability ($1-p_r$). This is also the most general case (the previous ones can be derived from it using an appropriate value for p_r).

Algorithm 1: Localized De-synchronization Algorithm

```
1   Choose a slot $\sigma$ in [1, $N_s$] randomly;
2   Try slot $\sigma$ (using a random backoff B); // contention slot
3   Wait (Notification);
4   Switch (Notification);
5     Case SUCCESS:
6       Use slot $\sigma$ in all subsequent periods
        with backoff B=0;                        //data slot
7     Case CHANNEL-BUSY:
8       Re-try slot $(\sigma+1)\bmod N_s$ (with random backoff B);
9     Case: COLLISION
10      Re-try slot $\sigma+1$ in the current period (with random
        backoff B) with probability $p_r$
11      Defer contention to slot $\sigma$ in the next period (with
        random backoff B) with probability (1- $p_r$);
```

Algorithm 1 shows the specific actions performed by a generic sensor node *i*. Initially, node i selects a random slot σ, within the current period, to try contention. This is aimed at spreading contention trials within the period, thus increasing the success probability. Since this initial randomization very rarely provides a complete de-synchronization of sensor nodes, the de-synchronization process follows. Specifically, node i contends for slot σ using a random backoff delay *B*, and waits for the corresponding ACK message (lines 2-3). Depending on the received notification (line 4), node i either acquires the right to use slot σ (lines 5-6), or realizes that a failure has occurred. In the latter case, it behaves in a different way, depending on whether a CHANNEL-BUSY (lines 7-8), or COLLISION (lines 9-11) notification has been received.

III. PRELIMINARY RESULTS

In this section we show some preliminary results obtained by solving an analytical model of the proposed algorithm, based on a discrete time Markov chain (details are omitted due to space limitations). We also ran simulation experiments using the ns2 simulation tool [17]. We assumed that the proposed algorithm is implemented in IEEE 802.15.4 sensor nodes. Hence, we considered a bit rate of 250 Kbps, and a data (ack) frame size equal to 127 (11) bytes. For each simulation experiment, we performed 10 independent replications, each of which consisted of 500 de-synchronizations. We derived confidence intervals using the *independent replication* method and 99% confidence level.

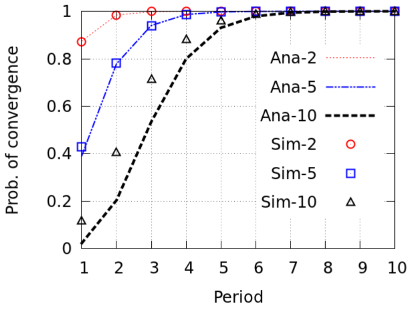

Figure 3. Convergence time of the de-synchronization algorithm for different number of sensor nodes.

Figure 3 shows the probability of reaching a complete de-synchronization for different number of sensor nodes (and available slots). Analytical results are limited to the case $N = 10$ (due to the computational complexity). We can see that there is some discrepancy between simulation and analytical results for N=10. This is due to the initial randomization performed by the algorithm (line 1 in Algorithm 1), which is not considered in the analytical model[1]. As expected, the initial randomization increases the probability of de-synchronization. However, if we look at the 95-th percentile of the distribution (i.e., number of periods required by the algorithm to provide a complete de-synchronization with a probability of, at least, 0.95), we see – from Table I – that analytical and simulation results are very similar. Table I also shows that the proposed algorithm converges in a time which increases with the number of sensor nodes, with a slope less than linear. It should be emphasized that values shown in Table I give the total time taken to obtain a *complete* de-synchronization. Of course, individual sensor nodes may get their own slot in considerably less time.

TABLE I. 95% CONVERGENCE TIME.

Number of nodes	Model	Simulation
2	2	2.00 (±0.00)
5	4	3.80 (±0.43)
10	6	5.10 (±0.32)
20		8.00 (±0.41)
30		10.50 (±0.54)
40		12.70 (±0.50)
50		14.80 (±0.43)

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we have focused on automated management systems for building energy efficiency, which rely on wireless sensor and actuator networks. To ensure the properties of *responsiveness*, *robustness* and *scalability* required by the application, we have proposed a localized algorithm for desynchronizing periodic transmissions of sensor nodes so as to obtain a round-robin, free-of-collision schedule, like in TDMA. Unlike previous similar solutions, the proposed de-synchronization algorithm is fully localized, i.e., sensor nodes adapt dynamically their behavior, on the basis of local information only. Our preliminary results show that (i) the proposed algorithm is able to converge in a limited number of periods, and (ii) the convergence time increases with the number of sensor with a slope less than linear. We are currently investigating possible improvements to further speed up the convergence time of the algorithm. We are also comparing it with previous similar solution, both in terms of performance and convergence time.

REFERENCES

[1] Energy Information Administration, "International Energy Outlook 2010 - Highlights", http://www.eia.doe.gov/oiaf/ieo/highlights.html.

[2] US DOE Energy, "Buildings Energy Data Book", http://buildingsdatabook.eren.doe.gov, 2009.

[3] US DOE Energy Information Administration, "Commercial Buildings Energy Consumption Survey", 2003. www.eia.doe.gov/emeu/cbecs.

[4] M. Chetty, D Tran, R. E. Grinter, "Getting to Green: Understanding Resource Consumption in the Home", *Proc. UbiComp 2008*, New York, USA 2008.

[5] X. Jiang, M. Van Ly, J. Taneja, P. Dutta, D. Culler, "Experiences with A High-Fidelity Wireless Building Energy Auditing Network", *Proc. ACM SenSys* 2009, Berkeley, USA, 2009.

[6] X. Jiang, S. Dawson-Haggerty, P. Dutta, D. Culler, "Design and Implementation of a High-Fidelity AC Metering Network", *Proc. ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN 2009)*, San Francisco, USA, 2009.

[7] L. Yeh, Y. Wang, Y. Tseng, "iPower: an Energy Conservation System for Intelligent Buildings by Wireless Sensor Networks", *International Journal of Sensor Networks*, Vol. 5, N. 1, 2009.

[8] J. Lu, T. Sookoor, G. Ge, V. Srinivasan, B. Holben, J. Stankovic, E. Field, K. Whitehouse, "The Smart Thermostat: Using Wireless Sensors to Save Energy in Homes", *Proc. ACM BuildSys 2010,* Zurich, CH, Nov. 2010.

[9] A. Marchiori, Q. Han, "Distributed Wireless Control for Building Energy Efficiency", *Proc. ACM BuildSys 2010,* Zurich, CH, Nov. 2010.

[10] G. Anastasi, M. Conti, M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in 802.15.4 Wireless Sensor Networks", *IEEE Transactions on Industrial Informatics*, Vol.7, N.1, Feb. 2011

[11] IEEE Standard for Information technology, Part 15.4; Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society, 2006.

[12] ZigBee Alliance, The ZigBee Specification version 1.0 (Q4/2007).

[13] J. Degesys, I. Rose, A. Patel, R. Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks", Proc. *Int'l Conference on Information Processing in Sensor Networks*, Cambridge, USA, April 25-27, 2007.

[14] R. Pagliari, Y. Hong, A. Scaglione, "Bio-Inspired Algorithms for Decentralized Round-Robin and Proportional Fair Scheduling", *IEEE Journal on Selected Areas in Communications (J-SAC)*, Vol. 28, N. 4, May 2010.

[15] J. Degesys, R. Nagpal, "Towards Desynchronization of Multi-hop Topologies", IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008), Venice, Italy, October 20-24, 2008

[16] H. Kang, J. Wong, "A Localized Multi-Hop Desynchronization Algorithm for Wireless Sensor Networks", Proceedings of IEEE INFOCOM 2009, Rio de Janeiro, Brazil, April 19-25, 2009.

[17] Network Simulator Ns2, http://www.isu.edu/nsnam/ns.

---

[1] We also ran simulation experiments without the initial randomization. In those conditions analytical and simulation results almost overlap.