

Energy Awareness in Self-Growing Sensor Networks

Lieven Tytgat, Opher Yaron, Ingrid Moerman and Piet Demeester

Abstract—An ever increasing variety of applications are being addressed by wireless sensor networks, resulting in a continuous proliferation of their deployments, which are in many cases co-located. This development is mostly hindered by the operational complexity involved with management and maintenance of large numbers of small, battery powered wireless sensor devices. The paradigm of energy aware self-growing networks addresses these difficulties. It focuses on power saving which reduces the major maintenance complexity of replacing batteries, and on automatic cooperation between networks which reduces the management complexity. However, cross-network cooperation requires cross-network communication, which is not straightforward as they typically operate on different frequencies. Receiver Directed Transmission is a MAC layer protocol which can bridge this gap, while also minimizing interference and thus reducing the number of transmissions. In this work we study how Receiver Directed Transmission can be combined with Low Power Listening in order to take advantage of the reduced number of transmissions to improve power consumption. We then implement the selected approach on TinyOS and verify its operation experimentally.

Index Terms— Energy awareness, IEEE 802.15.4, MAC, Media Access Control, Power Saving, RDT, Receiver Directed Transmission, Wireless sensor networks

I. INTRODUCTION

One of the major obstacles to large scale adoption of wireless sensor networks remains its relatively high operating expense (OPEX). This cost is strongly influenced by the need to manage and maintain, and particularly replace batteries periodically, in a large number of (sometimes hard-to-reach) devices. In order to reduce OPEX to an acceptable level there is a need to address these two sources of cost.

Except for using batteries with higher capacity, which are of course more expensive, the only way to reduce the cost of battery replacement is to increase the period at which it is needed, i.e. to reduce the power consumption of the nodes. In wireless sensor nodes the single most power consuming component is the radio module. Hence a major contribution to power saving can be achieved by putting the radio in sleep-

mode when it is not needed, which is typically the responsibility of the MAC layer. A variety of well known MAC protocols support sleep-mode, and in most cases there is a direct relation between the average rate of packet transmissions and the percentage of time the node spends in sleep-mode.

It remains, then, to identify a suitable protocol that can reduce the average rate of packet transmissions, and to combine it with a suitable MAC layer protocol that will put the radio to sleep mode when it is not needed. Receiver Directed Transmission (RDT) [1] is a perfect candidate. It is a MAC layer protocol which enables the nodes of a single network to operate on multiple frequencies. In a previous study [2] we use RDT to avoid interference, thus reducing the amount of retransmissions due to reception errors. In this work we elaborate on the combination of RDT with a MAC layer protocol that supports sleep mode, with the purpose of reducing power consumption. RDT makes a perfect candidate in this case, as it is also an enabler of automated management and self-growing [3], thus facilitating further reduction of OPEX. By allowing network nodes to operate on multiple frequencies, RDT in essence also allows separate networks that operate on different frequencies to communicate with each other, thus enabling cross-network communication, which is a prerequisite for cross-network cooperation and self-growing.

In Section II we elaborate on the power consumption of a sensor node and explore possible MAC mechanisms for power saving and cooperation of co-located sensor networks. Section III describes the operating principle of RDT, and Section IV explains the different ways it can be combined with a Low Power Listening (LPL) protocol. In Section V we detail our experimental results and analyze the potential power savings with our combined RDT + LPL implementation. We conclude this paper in section VII.

II. SAVING POWER IN WIRELESS SENSOR NODES

Wireless sensor nodes can typically be in one of three modes of operation at any given time – transmitting, receiving and sleeping (radio module Off). The power consumption when transmitting (P_{TX}) or receiving (P_{RX}) is typically significantly higher than when sleeping (P_{sleep}). For example, Table I details specified and measured values for the popular Tmote Sky wireless sensor node [4] when operating at a supply voltage of 3.3 Volts. Consequently, the most effective way to save power is to maximize the time the node spends in sleep-mode.

Manuscript received may 20, 2012. The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreements n° 257542 (CONSERN project) and n° 258301 (CREW project). It has also received funding from IWT under projects ESSENCES and SYMBIONETS, from IBBT under the project NGWINETS and from Research Foundation Flanders (FWO)

L. Tytgat is a PhD student at Ghent University – IBCN – IBBT, Gaston Crommenlaan 8, 9050 Ghent, Belgium (phone: +32-9--331-4900; fax: +32-9-331-4899; e-mail: lieven.tytgat@intec.ugent.be).

O. Yaron, I. Moerman and P. Demeester are with IBBT-IBCN-UGent, Gaston Crommenlaan 8, 9050 Ghent, Belgium (e-mail: {opher.yaron, Ingrid.moerman, piet.demeester}@intec.ugent.be).

TABLE I
TMOTE SKY TYPICAL POWER CONSUMPTION

Parameter	Spec. Nominal (mW)	Spec. Max (mW)	Measured (mW)
P_{TX}	64	69	62.8
P_{RX}	72	76	65.0
P_{sleep}	6	8	5.3

The mode at any given time is determined by the MAC protocol that the wireless sensor node employs. A variety of MAC protocols that periodically go into sleep-mode exist in the literature. Naturally, the lower the throughput in a wireless sensor network, the longer will the nodes be able to spend in sleep-mode, and consequently the lower will their power consumption be. A typical example is illustrated in Fig. 1, which compares the power consumption of the popular S-MAC and B-MAC protocols in a specific scenario [6].

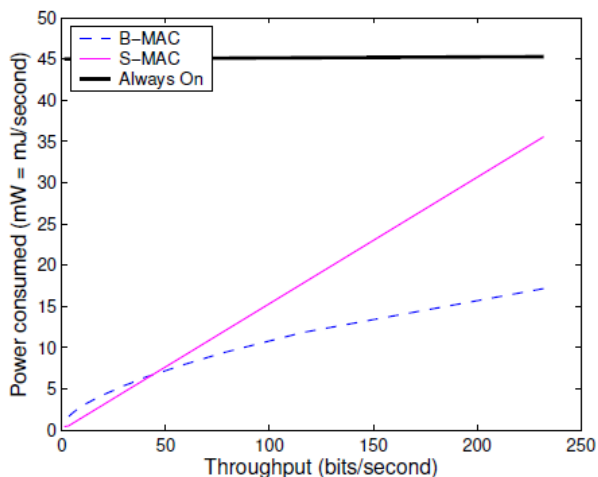


Fig. 1: Power consumption of S-MAC and B-MAC

At low throughputs S-MAC outperforms B-MAC with a small margin (up to 25% in this case), but as the throughput grows B-MAC outperforms S-MAC by growing margins. This difference in behavior is typical, as S-MAC and B-MAC belong to two different classes. S-MAC is a representative of the class of synchronized protocols, where the sleep periods of all nodes in a network are synchronized. Such protocols are more efficient when the throughput is low, as they can use long sleep periods with no penalty, but as the throughput grows the overhead of keeping synchronization between the nodes grows linearly. B-MAC, on the other hand, is a representative of the class of non-synchronized protocols. In this class a node that has a packet to send must transmit for at least the complete duration of the sleep period, to guarantee the destination node wakes-up and learns there is a packet for him. When the throughput is low, this brings to higher power consumption due to the tradeoff between longer sleep periods and the resulting longer transmission times necessary. When the throughput grows, however, the sleep periods get shorter, and the relative penalty of transmission length decreases. Moreover, in this work we also focus on cross-network communication, which is an important enabler for self-growing. The need for synchronization severely increases the complexity of enabling cross-network communication,

therefore we select the non-synchronized approach. More specifically, we use the Low Power Listening (LPL) protocol implemented in TinyOS [5], which is a variant of B-MAC.

The basic idea in LPL is to minimize the time a node is in receive mode. An LPL receiver has a cycle time T during which it sleeps as much as possible, and only wakes up once to listen if a transmitter is sending it a packet. When a node has a packet to send, it transmits it repeatedly for at least one complete cycle time T , making sure that the receiver will have woken up at least once in the meantime. The receiver will therefore wake up for at least one packet time + the time between two consecutive packets, denoted as t . This operating principle is illustrated in Fig. 2.

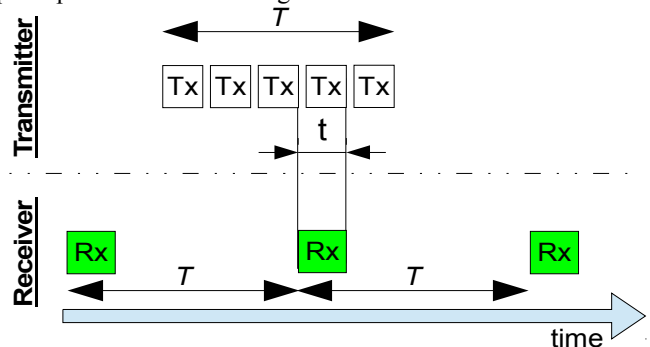


Fig. 2: The LPL operating principle

III. RECEIVER DIRECTED TRANSMISSION

In order to minimize power consumption, it still remains to reduce the throughput at the MAC layer as much as possible. One way to do this is to reduce retransmissions of packets, by avoiding interference and resulting unsuccessful packet receptions. We propose to achieve this by utilizing Receiver Directed Transmission (RDT). RDT was studied in [1] as a mechanism to improve network throughput by using multiple frequency channels concurrently. In RDT, each node employs a single radio transceiver, and is assigned a channel to which it is listening whenever it does not transmit – its *quiescent* channel. To send a packet, the node tunes its radio to the quiescent channel of the intended receiver, transmits the packet, and then retunes to its own quiescent channel.

If the transmitter does not know the quiescent channel of the destination, or there is more than one destination (e.g. broadcast traffic), then the packet needs to be transmitted on all possible channels, as illustrated in Fig. 3. The total transmit time – denoted as the transmit cycle time – will obviously be increased by a factor of the total number of channels.

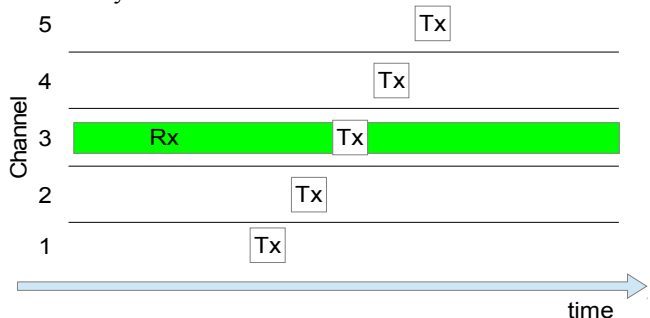


Fig. 3: Transmission to node with quiescent channel 3, which is unknown to the transmitter. The packet is transmitted on all channels.

In an earlier work we already studied RDT as a distributed mechanism for avoiding localized interference [2]. In this paper we leverage on the capability of RDT to reduce interference (and resulting packet retransmissions), but also emphasize its advantage as a cross-network communication enabler. By enabling devices on different channels to communicate with each other, RDT facilitates cross-network communication, which, as mentioned earlier, is an important enabler for self-growing.

IV. COMBINING RDT WITH LPL

The two possible approaches for combining RDT with LPL are depicted in Fig. 4. RDT can either be implemented as a communication protocol layer below LPL (Fig. 4a) or above it (Fig. 4b). Within this section we assume broadcast traffic, resulting in RDT multiplying the packet on all used frequencies, denoted k . In sake of simplicity we assume the use of 3 channels within this paper, thus $k=3$.

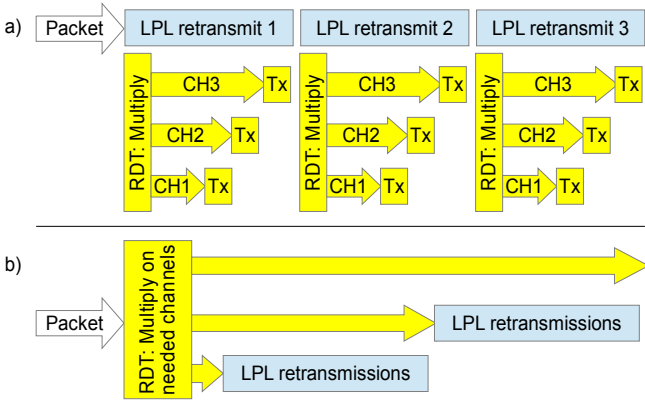


Fig. 4: a) LPL above RDT. b) RDT above LPL

In the first alternative, LPL receives a packet from the higher layer. It then delivers this packet to RDT multiple times. Each time RDT receives the packet from LPL, it transmits it in all necessary channel(s). For example, a broadcast packet results in every LPL delivery being transmitted on all channels, as illustrated in Fig. 5.

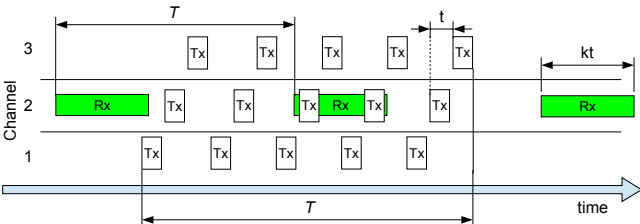


Fig. 5: Combining RDT and LPL case 1: LPL above RDT

Hence, the listen window of the receiver needs to be at least as long as it takes for RDT to transmit a single packet on all channels, which is k times longer than the original LPL. The transmit cycle time can remain identical to the original LPL cycle time T .

In the second alternative, RDT receives a packet from the higher layer. It delivers the packet to LPL multiple times, each time for a different channel. Whenever LPL receives a packet

from RDT for a specific channel, it transmits it on this channel multiple times, identically to the original LPL operating on this channel. This operation is depicted in Fig. 6

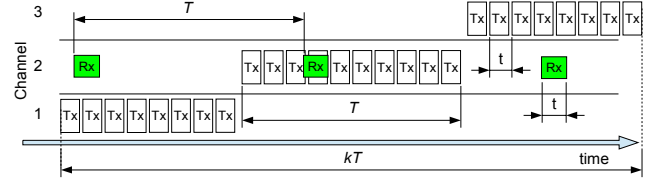


Fig. 6: Combining RDT and LPL case 2: RDT above LPL

The listen window of the receiver is identical to that of the original LPL, but the transmit cycle time is k times longer, i.e. kT .

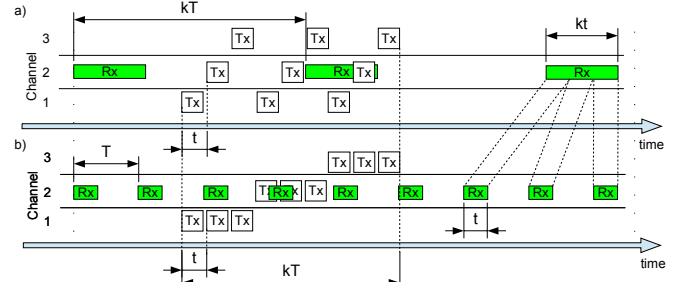


Fig. 7: Comparing case 1 and case 2 with identical link throughputs. a) LPL above RDT, b) RDT above LPL

Comparing the two alternatives with respect to transmitter energy consumption seems advantageous to LPL above RDT. The receiver's energy consumption seems to be in favor of RDT above LPL. However, there is a big difference between the two alternatives with respect to maximum throughput. When we start with the same cycle time T in both cases, the resulting maximum link throughput of the 'LPL above RDT' alternative is k times higher than that of the 'RDT above LPL' alternative, because in the latter the necessary transmit time of a packet is k times longer. To do a fair comparison, we need to compare both cases with identical maximum throughput. We can realize identical maximum throughput by increasing the cycle time of the 'LPL above RDT' alternative to kT , in contrast to a cycle time of T in the 'RDT above LPL' case, as shown in Fig. 7. Now in both alternatives the ratio of time a receiving node is in receive mode is t/T with t the transmission time of one packet; and the transmit cycle time of one packet is kT . Consequently, the two alternatives are practically identical in terms of power consumption.

V. EXPERIMENTAL POWER CONSUMPTION ANALYSIS

We chose the 'RDT above LPL' alternative, and implemented it in TinyOS on tmote sky nodes [4]. The implementation – which is illustrated in Fig. 8 – is running inside the default CC2420 radio stack of TinyOS, making it invisible to higher layer protocols.

When RDT receives a packet from the higher layer protocols, it first looks up the destination's channel(s). It switches the radio channel through the setChannel call, and passes the packet on to the LPL layer, which then takes care of the needed retransmissions. LPL notifies RDT when the transmission is completed. RDT will then either switch to the

next transmission channel if needed, or it will revert back to the receive channel.

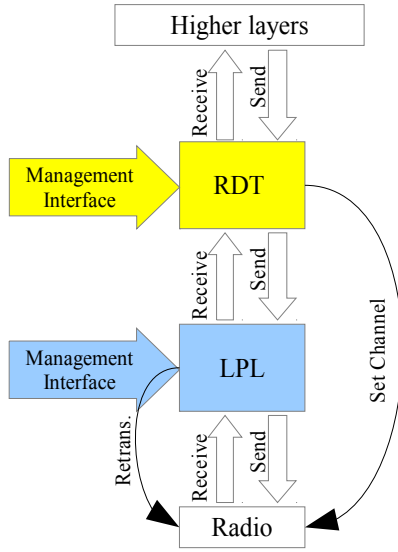


Fig. 8: The implemented architecture: LPL below RDT

TABLE II
MEASURED ENERGY CONSUMPTION

Test set-up	Power Consumption (mW)	Relative Power use
RDT without LPL	65.0	100%
RDT+LPL without transmission	6.1	9.4%
RDT+LPL unicast	18.3	28.2%
RDT+LPL broadcast	41.3	63.5%

We measure the power saving this implementation achieves on the w-iLab.t testbed of IBBT [7]. This testbed is deployed in an office environment and among others features power consumption measurements on all nodes. Within the experiments we use a 3.3V supply voltage. We transmit a packet every 5s, and use an LPL cycle time of 1s in all tests, unless explicitly noted otherwise. Table II summarizes the measurement results and the relative power savings achieved.

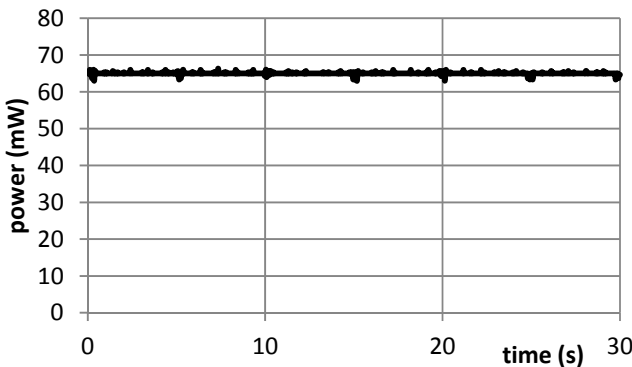


Fig. 9: Power consumption of RDT without LPL

The power consumption of RDT without LPL is depicted in Fig. 9. Without LPL the node never goes into sleep mode, therefore the power consumption is relatively constant. Packet

transmissions are visible as small glitches around 0s, 5s, 10s, etc. The effect of transmissions on the average power consumption is clearly very limited. We measure an average power consumption of 65 mW.

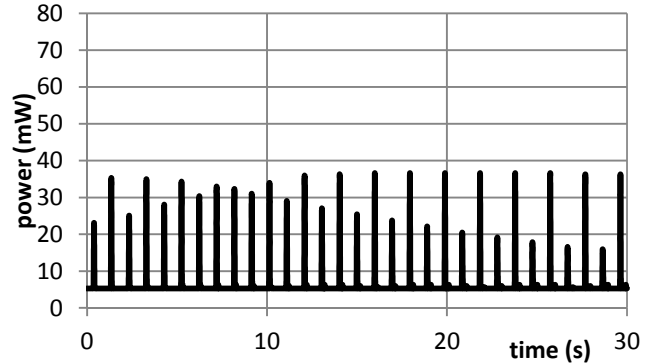


Fig. 10: Power consumption of RDT + LPL without transmission

When we add LPL the node periodically goes into sleep mode, and the power consumption is reduced significantly. Fig. 10 shows the energy consumption of a node that runs RDT + LPL when it is not transmitting packets. Every peak in the diagram is the result of the radio waking up. The average power consumption in this case is 6.1 mW. The differences in the heights of the peaks are measurement artifacts due to the nonzero time it takes to perform reliable current measurements. The actual power consumed during these peaks equals the power consumption of receive mode, i.e. 65 mW. The average cycle time T is 1008 ms.

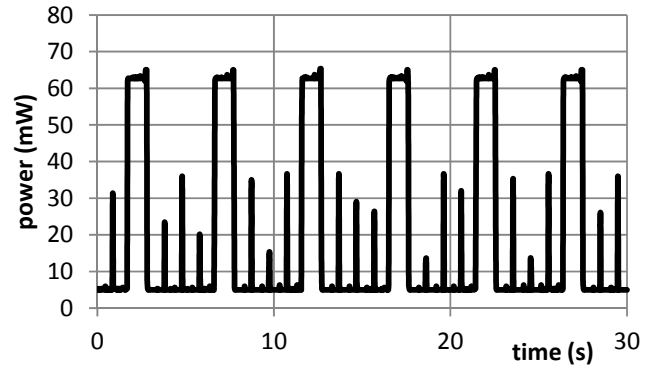


Fig. 11: Power consumption of RDT + LPL with unicast transmission

The power consumption of the RDT+LPL combination when one packet is transmitted every 5 seconds is shown in Fig. 11. We distinguish the receive peaks, also seen in Fig. 10, and the power consumption during transmission. The power consumption of the node during transmission equals 62.8 mW. A transmission lasts in average 1096 ms, which is 88 ms longer than the LPL cycle time. Hence there is sufficient overlap to guarantee the receiver has woken up during the LPL transmit window.

A broadcast packet needs to be transmitted on all channels. In this experiment there are 3 channels configured for RDT, therefore each broadcast transmission lasts 3 times longer than unicast, as shown in Fig. 12.

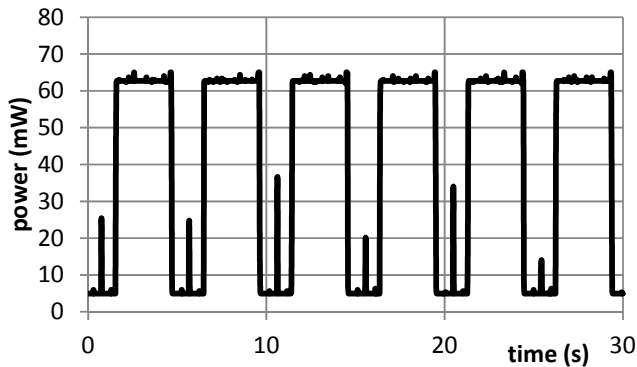


Fig. 12: Power consumption of RDT + LPL with broadcast transmission

VI. FUTURE WORK

Within this paper we have studied the energy consumption of a single node. However, we have not studied the energy consumption of a complete network. Especially the exchange of quiescent channel information to neighboring nodes will determine the final energy gains. We will elaborate on this in future work.

LPL reduces the average power consumption in receive mode. However, the time a node spends in transmit mode increases with this approach. Therefore the transmit power consumption becomes more important. Using transmit power adjustment can reduce the transmit power, but remains future work.

VII. CONCLUSION

Sensor networks are deployed worldwide, resulting in more and more co-located sensor networks. Operational costs can be significantly reduced in such cases by self-growing, due to reduced management costs and power consumption. Within this paper we propose to use RDT as a self-growing enabler, as well as a mechanism to reduce the amount of packetloss resulting from interference. However, with current state-of-the-art RDT implementations the radio is always on, incurring unnecessarily high power consumption. We propose to reduce the power consumption by combining RDT with LPL.

Both RDT and LPL are MAC layer protocols, and will therefore interact with each other. We investigate the different plausible combinations, and compare the two alternatives of RDT running above LPL and LPL running above RDT. We show that for identical maximal link throughputs the two alternatives have identical power consumptions.

We selected to implement RDT above LPL, as it does not require any modification of the standard LPL implemented in TinyOS. We measured the power consumption on the IBBT w-iLab.t wireless testbed, and conclude that the power consumption of a receiver running RDT+LPL with a 1s cycle time brings power saving of 90.6%. An RDT+LPL transmitter sending unicast packets at 5 second intervals results in power saving of 71.8%; and a transmitter sending broadcast packets to three different channels achieves a 36.5% power saving.

REFERENCES

- [1] Shacham N., King P.: "Architectures and performance of multichannel multihop packet radio networks.", *IEEE Journal on Selected Areas of Communication*, JSAC-5(6):1013-1025, 1987
- [2] L. Tytgat, O. Yaron, S. Pollin, I. Moerman P. Demeester, "Analysis and experimental verification of interference avoidance mechanisms in IEEE 802.15.4", submitted to *IEEE/ACM Transactions on Networking*, aug. 2012
- [3] Koudouridis, G.P.; Hedby, G.; Chin, W.H.; Merentitis, A.; Stamatiatos, M.; Alonistioti, N.; Yaron, O.; "Enablers for Energy-Aware Cooperative Decision and Control in Wireless Networks," *Vehicular Technology Conference (VTC Spring)*, 2011 IEEE 73rd, vol., no., pp.1-5, 15-18 May 2011, doi: 10.1109/VETECS.2011.5956575
- [4] Moteiv Tmote Sky data sheet, online: http://www.snm.ethz.ch/pub/uploads/Projects/tmote_sky_datasheet.pdf, accessed may 15, 2012
- [5] D. Moss, P. Levis, "BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking", *Stanford Information Networks Group Technical Report SING-08-00*, 2008
- [6] Polastre, J., Hill, J., And Culler, D., "Versatile low power media access for wireless sensor networks", In *Proceedings of the 2nd International Conference on Embedded Networked Sensor System (SenSys)*, 2004
- [7] w.iLab.t portal – CREW project, online: <http://www.crew-project.eu/wilabt>, accessed may 15, 2012