DEEP: A Deployable Energy Efficient 802.15.4 MAC Protocol for Sensor Networks

Marco Valero, Anu Bourgeois, and Raheem Beyah
Department of Computer Science
Georgia State University
Atlanta, Georgia 30303
{mvalero, abourgeois, rbeyah}@cs.gsu.edu

***Abstract*—IEEE 802.15.4 is a standard designed for low data rate wireless personal area networks (WPANs) intended to provide connectivity to mobile devices. Such devices have considerable storage, energy, and communication constraints. However, they can be used in a variety of applications like home/office automation, healthcare, environmental control and more. To extend the lifetime of the WPAN, we propose a backward compatible energy efficient 802.15.4 MAC protocol (DEEP) for beacon-enabled sensor networks. The implementation of DEEP requires modifications to the Superframe Guaranteed Time Slot (GTS) distribution. This modification optimizes the GTS distribution providing reduced energy consumption. We implemented the improvements to the IEEE 802.15.4 protocol using real sensor nodes in a wireless network. Specifically, we conducted an energy study of DEEP's acknowledgment-based GTS descriptor distribution scheme and compared the results with the standard implementation. Experiments show that DEEP reduces energy consumption up to nearly 50% when 7 devices allocate guaranteed time slots descriptors during normal communication.**

I. Introduction

Given that wireless sensor network (WSN) technology is considered one of the most important technologies for the twenty-first century [1], much research effort has been dedicated to the 802.15.4 MAC protocol. This protocol is designed for personal area networks (PANs) with short distance and low power requirements. PANs can be used for home automation and sensor networking. The protocol supports several network types such as: star, cluster-tree or mesh. The network consists of full function devices (FFDs) that perform network control tasks and reduced function devices (RFDs) that perform data sensing and reporting. Communication is facilitated by the superframe structure that is determined by the PAN coordinator which is an FFD. The superframe consists of a contention access period (CAP), a contention free period (CFP) and an idle period. During the CAP, nodes compete for the channel using slotted carrier sense multiple access with collision avoidance (CSMA-CA). During the CFP, channel access is based on reservations of guaranteed time slots (GTSs).

In our previous work [2] we examined a potential energy-waster: tracking broadcast beacons. Beacons sent by the coordinator are the only broadcast messages in 802.15.4, and a device may receive multiple copies of beacons although only one copy is necessary. The structure of a beacon frame is given in Figure 1, as described in [3]. Transmitting a packet to a large group when it is only intended for a small group can cause unnecessary energy consumption at nodes which are not interested in the packet.

Octs:2	1	4/10	2	1	0/1	var	var	var	2
Frame control	Seq No	Address –ing fields	Superframe Spec	GTS Spec	GTS directions	GTS list	Pending address fields	Beacon payload	FCS

Fig. 1: 802.15.4 Beacon Frame Structure.

In this paper we present implementation details of a deployable 802.15.4 MAC protocol (DEEP) that can reduce energy consumption up to nearly 50% in a network of actual sensor nodes. Besides being one of the few actual deployments of an enhanced 802.15.4 protocol [4], our DEEP protocol implementation is also backward compatible with the original standard. DEEP's changes to the 802.15.4 protocol do not reduce the original functionality of 802.15.4 coordinator nodes since DEEP coordinators only remove GTS descriptors from the beacons when requested. Therefore, regular nodes using the original standard can still communicate with DEEP coordinators using the standard GTS distribution.

We validate our work through several experimental configurations and scale those results through simulations. The main contribution of our work is the presentation of an implemented, backward compatible deployable energy efficient enhancement to the 802.15.4 MAC protocol.

The rest of this paper is organized as follows. The related work is presented in Section II. Section III introduces the DEEP protocol. In Section IV, we describe in detail how we implemented DEEP on real sensors. Section V presents the GTS scenarios and the effects of GTS allocations on beacon sizes. In Section VI we explain the different experiments we ran and show the results. In Section VII, we present an energy consumption analysis and scale the energy savings through simulations. Section VIII concludes the paper.

II. Related Work

The work done in the contention free period of the superframe focused primarily on the fairness of the scheduling algorithm, and how best to utilize and allocate GTSs among participating devices [7], [8]. In [7], an adaptive GTS allocation mechanism is proposed, noting the starvation possibility present in the current 802.15.4 design. In [8] the authors proposed a GTS distribution based on priorities. The coordinator keeps track of the transactions made by the nodes and assigns a priority to each of them. In [6], the authors introduced an energy consumption estimation model used to calculate the power consumption of the different radio transceiver states (e.g., transmission, reception). We used part of this work to estimate the energy consumption in our simulations in Section VII. On beacon-enabled WPANs, a sensor node requesting GTSs is vulnerable to miss the GTS descriptor in the beacon due to various reasons such as sleeping. In [2], we used simulations to analyze the energy that nodes spend on tracking a GTS descriptor for the first time up to aGTSDescPersistenceTime (default: 4) superframes. If the device receives no GTS descriptor within aGTSDescPersistenceTime time after sending the request, it concludes that the allocation request has failed.

In this paper we introduce a deployable energy efficient 802.15.4 MAC protocol (DEEP). In contrast to the related work, DEEP is backward compatible, is evaluated using experimentation and simulations, and is more comprehensive than our previous work as it considers the energy savings after the GTS descriptor is successfully received.

III. DEEP Protocol

In the IEEE 802.15.4 standard all devices in the PAN must track at least one beacon before transmitting or receiving data. The coordinator periodically transmits a beacon at the interval defined by aBeaconOrder. If the device has data to transmit and requires dedicated bandwidth to transmit such data, it sends a GTS request to the coordinator. The coordinator will allocate a GTS slot to the device if a slot is available, and all subsequent beacon frames will contain the GTS descriptor defining the device address, GTS slot and direction.

DEEP uses an ACK-based GTS descriptor distribution that consists of removing the descriptor from the beacon once the device requesting GTS allocation acknowledges its reception. This GTS distribution is backward compatible with the original standard since the coordinator only removes GTS descriptors from the beacons when it receives a GTS acknowledgment. A DEEP coordinator includes all unacknowledged descriptors in the beacons. This allows RFDs using the standard GTS distribution to communicate with the coordinator.

Upon receiving the beacon with the GTS descriptor, acknowledging its reception and saving this information, the device will schedule the pending packet to be transmitted at the allocated GTS slot. Algorithm 1 shows the steps followed by the device and coordinator to transfer information regarding GTS allocation and update of the GTS descriptor list.

The acknowledgment frame format used in the new distribution is the same as that in the 802.15.4 specification and the sequence number sent by the devices in the ACK is the assigned starting guaranteed time slot number. Although the acknowledged descriptors will not be included in the beacon, the coordinator will keep a record of all the descriptors and the devices are still informed of the final CAP slot and the number of assigned GTSs through the superframe specification.

Algorithm 1 DEEP's GTS distribution

```
The coordinator (CO) allocates GTS slots for devices
CO transmits a beacon with the GTS descriptors (GTSds)
Turn on the receiver and keep it on in allocated GTS slots
The devices requesting the GTS slot receive the beacon
if The GTS slot is of transmit type then
    if The device has data to send then
        Transmit data in the GTS slot allocated for this device
    else
        Transmit an ACK to CO in the GTS slot allocated for
        this device. The seq number in ACK uses the GTS
        Starting Slot assigned to that device*
    end if
else
    if The GTS slot is of receive type then
        The device transmit an ACK to CO in the GTS slot
        allocated for this device. *
        Turn receiver on
    end if
end if
if CO receives the ACK/data in the dedicated GTS slot
then
    It stops the dissemination of the GTSd in beacons
else
    CO repeats the above procedure from the second step
end if
```

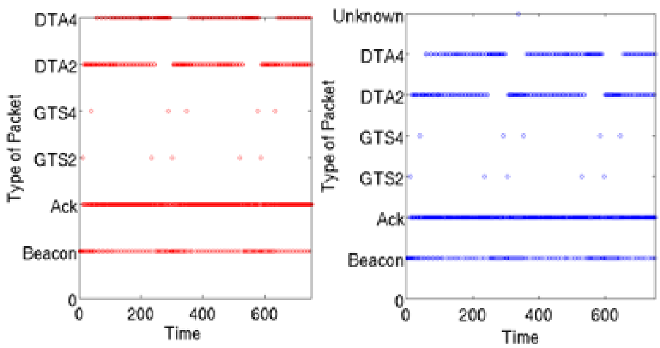
IV. Implementation of DEEP on Sensors

A. Implementation Details

The actual sensors used in our implementation are Tmote Sky from Moteiv. Tmote Sky features the Chipcon CC2420 radio for wireless communications, which is controlled by the Texas Instruments MSP430 microcontroller.

One of the challenges in working with the 802.15.4 MAC protocol is that, while much research has been conducted in the area, many of the implementations of the standard are proprietary undocumented blackboxes [4] that cannot be modified. However, we were able to find two open source 802.15.4 implementations by Atmel and Open-zb [9]. The Atmel implementation was developed in the C language for the AT86RF230 transceiver, but it lacks features like GTS, MAC-Security and MAC-Routing. The Open-zb implementation supports the Chipcon CC2420 transceiver and was developed in nesC/TinyOS v1.15 and is the implementation that we used to do our experiments.

We modified the GTS distribution process by having the GTS requesting devices send an acknowledgment right after the reception of their descriptor in the beacon. Since the coordinator cannot identify the device sending the acknowledgment, we use the GTS starting slot assigned to each node as the sequence number in the ACK packet. This allows the coordinator to identify and remove the descriptor from



(a) Original 802.15.4. (b) DEEP.

Fig. 2: Event Diagrams.

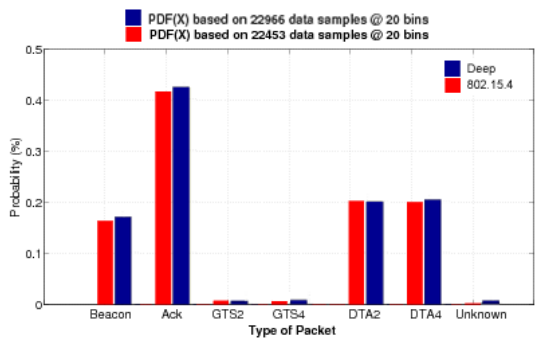
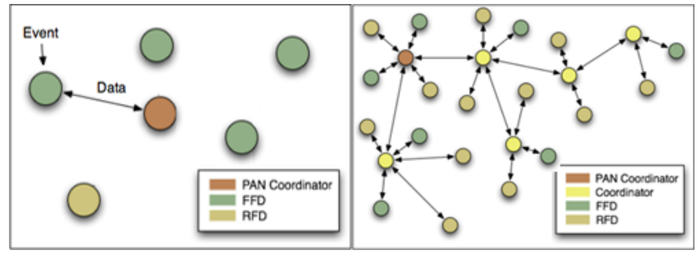


Fig. 3: Packets Distributions 802.15.4 and DEEP.



(a) Scenario 1. (b) Scenario 2.

Fig. 4: Scenarios Using GTSs.

the next beacons. When a device gets a descriptor with a GTS allocation, it saves the slot information and uses it to transmit data without processing the GTS characteristics of the following beacons. Thus, the coordinator only includes GTS descriptors in the beacons when there is a new GTS allocation request or after rearrangement of the GTS slots due to a GTS deallocation.

B. Validating DEEP

In order to verify that DEEP still behaves as the original standard, we ran experiments with both implementations using several nodes and one coordinator. We captured all the packets to analyze the communications pattern and the distributions (Texas Instruments CC2420 Evaluation Board/Evaluation Module in conjunction with the TI Chipcon packet sniffer). This is important because though we reduced the size of the packets, the general traffic pattern should be the same. Figure 2a shows the communication pattern of the 802.15.4 while Figure 2b shows DEEP's. Figure 3 shows the packet distributions for both the original standard and DEEP, respectively when two nodes and one coordinator are used.

V. GTS Analysis

A. Scenarios Using GTSs

Guaranteed time slots might be required in different scenarios. Let us consider two important scenarios that use GTSs. In the first scenario (Figure 4a), sensor nodes send data to the coordinator right after a specific event has occurred. For instance, nodes send to the coordinator a notification along with the monitored value when the temperature goes below 60 degrees in a home temperature monitoring system. In this scenario, the transmitting nodes send a GTS request to the coordinator and after receiving the descriptor, the data is sent. After transmission, the coordinator can deallocate the GTS descriptor without an explicit request. With our proposed changes, we will not have any energy savings in this scenario since the GTS slot is needed just once per event.

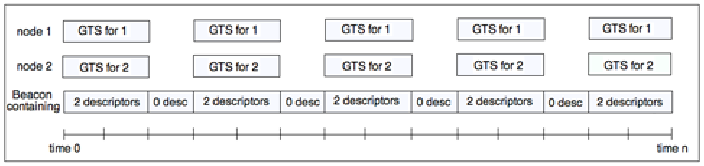
The second scenario (Figure 4b) is when the nodes are constantly transmitting data to the coordinator(s). Similar to the first scenario, each node will request a GTS and send data after receiving the descriptor. In this case, nodes will hold the assigned time slot until they finish transmitting data. Then, they send an explicit GTS deallocation request to inform the coordinator of the end of the transmission so it can reuse that time slot. An example of this scenario is tracking people in a battlefield. Nodes send data to the coordinator when the person is moving and end the transmission when the person stops.

To compare the energy consumption of the IEEE 802.15.4 standard with and without the proposed changes, we conduct experiments and simulations using both the original standard and DEEP for 2 and 3 GTS requesting nodes and we ran 4 different configurations, where each node used the GTS for a period of 6, 12, 18, or 32 superframes, stopping for half of the period and repeating the same process 5 times.

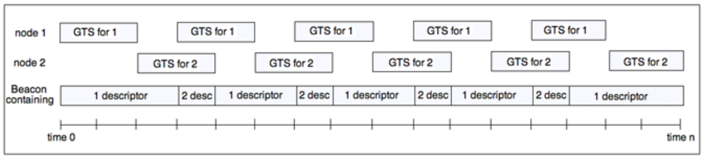
The idea of using the GTS for 6, 12, 18 and 32 superframes is to simulate varying amounts of data that nodes need to transmit. If a node uses a GTS for 32 superframes, it means that it has more data to transmit than a node that holds it for 6 superframes.

B. Effects of GTS Allocations on Beacon Sizes

The coordinator has fifteen guaranteed time slots available to allocate up to seven devices [3]. The size of the beacons and therefore the energy consumed when the nodes receive such beacons depends on when GTS allocations requests are generated. For example, if 2 RFDs request a GTS at the same time, the subsequent beacons will contain two descriptors (19 bytes). But if one device requests a guaranteed time slot right after the other devices deallocates its GTS, the beacons will contain one descriptor (16 bytes). In contrast to the original



(a) GTS Allocation Best Case.



(b) GTS Allocation Worst Case.

Fig. 5: Effects of GTS Allocations on Beacon Sizes.

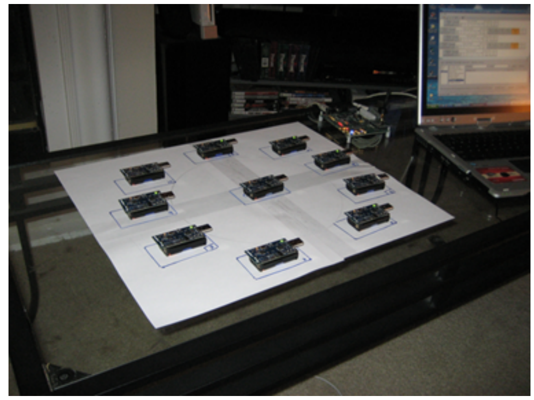


Fig. 6: Experiments Topology.

standard, our proposed GTS distribution mechanism removes the GTS descriptors from the beacons after acknowledgment, guaranteeing that for most of the communication time we will have 12-byte beacons (no descriptors in them). Assuming that there are many iterations of the allocation-deallocation process, Figures 5a and 5b show the best case and worst case scenarios for GTS allocations when using the standard GTS distribution mechanism for 2 GTS requesting nodes.

VI. Experiments

We ran over 20 hours of experiments with both the original 802.15.4 standard and DEEP, using new batteries for each trial. In this section we present the most relevant experiments concerning the energy consumption.

A. *Energy Measurement Approach*

We had two methods for measuring voltage in batteries. The first one was to use the Oscilloscope application. On the telosb platform the Oscilloscope application instantiates a component called VoltageC, which reads data from the micro controller unit internal voltage. The second method was to use a digital multimeter to measure voltage in the batteries. Before and after the experiments we measured voltage in batteries.

B. *Experiment Setup*

In [10], the node distances-energy consumed relation is analyzed showing that the energy spent increases when the distance between communicating nodes increments due to re-arrangements of the transmit power. The energy consumption for transmission depends on the different transmit power (8 levels on the Chipcon CC2420); 17.4mA is the maximum. The power consumed when receiving packets is 19.7mA.

We fixed the transmit power levels in our experiments by placing 9 nodes equidistant from the coordinator. Out of the 9 nodes we only used 3 (or 2) nodes to request GTSs and communicate with the coordinator while the other 6 (or 7) were IDLE nodes. We assume that all the nodes within communication range of the coordinator have the same transmit power and therefore spend the same amount of energy when sending packets and receiving the beacons. Figure 6 shows the topology used for the experiments.

C. *Experiment Details*

For the sake of space and simplicity, here we explain all the steps followed to compare the protocols and calculate the energy savings for one experimental configuration. An experimental configuration is the combination of one of the 4 configurations in Section V and a fixed number of GTS requesting nodes (from 1 to 7). In this case, we used 3 GTS requesting nodes, 1 coordinator and configuration 4 from Section V where each node uses the GTS for 32 superframes. We called this combination: Experimental Configuration 4 or EC4.

Using EC4 we ran one experiment using the original 802.15.4 and another with DEEP. All the packets transmitted throughout the experiments were captured using the TI Chipcon packet sniffer and saved in a psd binary file. We then used our custom C++ program to parse the binary file and extract important information to evaluate the performance of both standards. Our parser generates a text file with details of the packets as shown in Table I, where we see that the amount of packets transmitted on both experiments are very close. Figure 7 shows the goodput for both experiments and the total beacon bytes transmitted. Although the amount of beacons is about the same, we can see a notable difference in the amount of total beacon bytes transmitted. This is a result of our changes, where we send smaller-sized beacons most of the time by removing GTS descriptors.

TABLE I: Packet Details for Experimental Configuration EC4.

Original 802.15.4		**DEEP**	
Total packets: beacons=300 gts=30 acks=967 data=946 unknown=3 total=2246	Detail of packets: beacons=**300** gts3=10 gts4=10 gts5=10 data3=316 data4=313 data5=317	Total packets: beacons=301 gts=30 acks=935 data=946 unknown=4 total=2216	Detail of packets: beacons=**301** gts3=10 gts4=10 gts5=10 data3=317 data4=315 data5=314
Goodput: **1892** bytes	Total beacon bytes: **6267**	Goodput: **1892** bytes	Total beacon bytes: **4482**

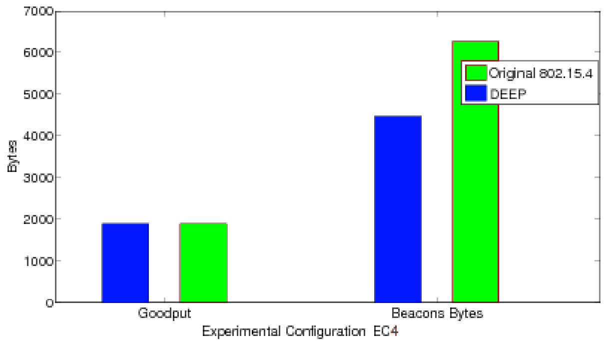


Fig. 7: Goodput and Beacons Bytes Transmitted with Experimental Configuration EC4.

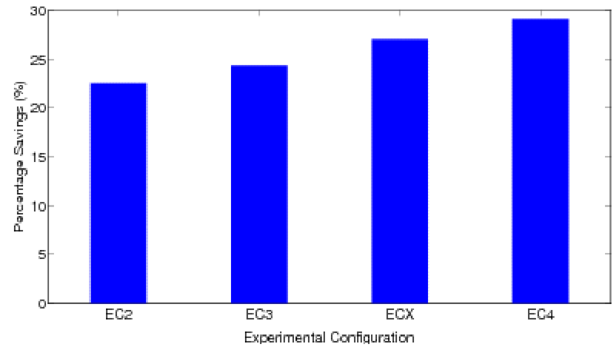


Fig. 8: Percentage of Energy Savings for Different Experimental Configurations.

The voltage percentage change is shown in Table II.

TABLE II: Voltage Percentage Change for EC4.

	Original 802.15.4 Percentage Change	DEEP Percentage Change
Coordinator	0.43451	0.27916
Node 3	0.40248	0.27881
Node 4	0.37129	0.27864
Node 5	0.37209	0.27933

We can now calculate the average energy consumed in each experiment which is 0.395% with the original standard and 0.279% with DEEP. Finally we calculate the energy savings which in this case is 29.39%. This experiment was repeated several times using the configurations mentioned in Section V. Using 2 GTS requesting nodes and 1 coordinator we obtained 22.54% savings with configuration 2 from Section V (Experimental Configuration 2 or EC2) and 24.33% savings with configuration 3 (EC3). Using 3 GTS requesting nodes, 1 coordinator and a different configuration not specified in Section V where nodes used the GTS for 30 superframes, we obtained 27.03% savings (we called it Experimental Configuration Extra or ECX). The energy savings results are shown in Figure 8. In the original protocol when there are more nodes allocating GTSs, the size of the beacons are larger since they allocate more descriptors. Also, when nodes have more data to transmit, the descriptors will remain in the beacons for more time, which also implies larger beacons. With our changes, the energy savings increased when we have more nodes allocating GTSs and/or more data to transmit. In all the cases, and with all the configurations, we achieve energy savings with our improved DEEP protocol.

The IEEE 802.15.4 standard specifies that a WPAN can allocate up to 7 GTS requesting nodes. Unfortunately, the open source implementation that we used does not handle more than 3 GTS requesting nodes. Although we have the resources available, we could not exceed the setup of 3 requesting nodes and 1 coordinator. Therefore, in the next section we scaled our sensor network through simulations and compared the results with our experiments to illustrate the energy savings, when using our changes, as the network grew.

VII. Energy Analysis and Simulations

We modeled both the normal GTS algorithm and the GTS optimized algorithm to investigate the energy advantages of DEEP modifications. For both, we simulated all the possible combinations of GTS allocations using 1 coordinator and a range from 2 to 7 reduced function devices to get the best and worst cases. From there we calculated the average amount of bytes that nodes received from beacons and the energy they consumed. The energy parameters: receiver current ΔRX and transmit current ΔTX are based on the CC2420 specifications and R is the transmission rate. These values are used to calculate energy consumption when transmitting and receiving data using equations (1) and (2) respectively.

$$TXEnergy = \frac{Bits}{R} * \Delta TX \tag{1}$$

$$RXEnergy = \frac{Bits}{R} * \Delta RX \tag{2}$$

The beacon frame size is identical between the normal and GTS optimized protocols. Beacon sizes depend on the number of descriptors that the coordinator has allocated. Beacons with no descriptors are 12 bytes. Beacons with 1 to 7 descriptors are 16, 19, 22, 25, 28, 31, 34 bytes respectively.

For the simulations, we focused our attention on the beacons and the energy spent when nodes receive such beacons. This was done since we assumed that all the nodes transmit the same amount of data and therefore spend the same amount of energy on transmissions. We simulated all the cases for GTSs allocation (discussed in part *B* of Section V) to get an average amount of beacon bytes transmitted using each experimental configuration. Figure 9 shows the average number of beacon bytes received by each node when using Configuration 1 and Configuration 4 from Section V. With both configurations we saved more bytes when using the DEEP protocol.

Using equation (2) and the beacons bytes received by each node, we calculate the energy consumption for tracking beacons and compare it with our experimental results. Figure 10 shows the percentage *energy savings* for both experiments and simulations using configurations EC2, EC3 and EC4.

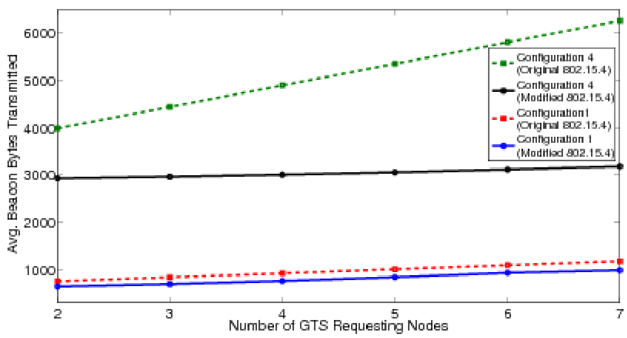


Fig. 9: Comparison of Received Bytes Original vs. DEEP.

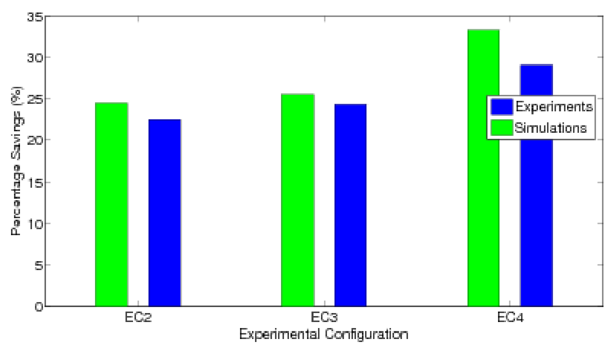


Fig. 10: Percentage Energy Savings Experiments vs. Simulations.

Although there is a small difference in the percentage of energy savings between the simulations and the experiments, the simulations results are very consistent with the experiments, validating our use of simulations to scale our analysis. The energy savings grow when the number of GTS requesting nodes increases and also when nodes have more data to transmit. Also, the simulations focus only on the energy spent when tracking beacons since we assume that all the nodes spend the same energy on transmissions.

Figure 11 shows the percent difference in energy savings when receiving beacons against the baseline scenario. The baseline scenario is the 802.15.4 protocol without the GTS optimizations. The x-axis shows the number of superframes that each node holds the GTS, which represents the amount of data to transmit. The y-axis shows the percent difference in energy savings of each sensor receiving beacons. The results show that DEEP's GTS optimizations increases energy savings in all the scenarios and the savings grow when we have more GTS requesting nodes and more data to transmit. We also notice that the proposed changes reduced energy consumption, from 15% savings up to nearly 50% when 7 nodes allocate guaranteed time slots descriptors.

VIII. Conclusion

In this paper we presented and implemented a backward compatible energy efficient 802.15.4 MAC protocol (DEEP). We conducted experiments with real sensors and simulations

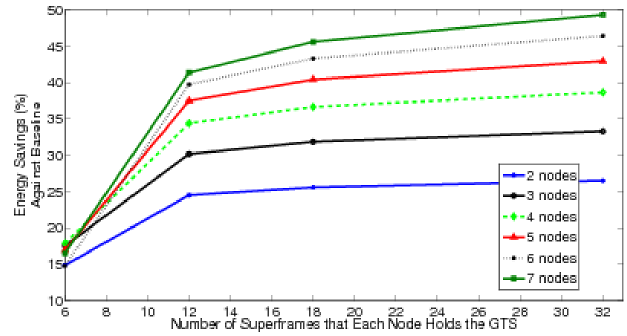


Fig. 11: Percent Difference in Energy Savings against the baseline scenario.

using different configurations and scenarios. We compared the performance and energy consumption of DEEP against the standard. The results show that our improved implementation has the same functionality as the original 802.15.4 and the GTS optimizations increase energy savings in all the scenarios. The energy savings range from 15% up to nearly 50% according to the number of GTS requesting nodes.

References

[1] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, pp. 1247–1256, August 2003.

[2] A. Bourgeois and L. Cheng, "Energy efficiency of different data transmission methods in ieee 802.15.4: study and improvement," in *Wireless Pervasive Computing, 2007. ISWPC'07. 2nd International Symposium*, 2007.

[3] IEEE, *IEEE 802.15.4 Standard Specification for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, 2003.

[4] J. Flora and P. Bonnet, *Never Mind the Standard: Here is the TinyOS 802.15.4 Stack*, 2006.

[5] Y. Zhang, P. Xu, G. Bi, and S. Bao, "Analysis of energy efficiency and power saving in ieee 802.15.4," in *Wireless Communications and Networking Conference, 2007.WCNC 2007.*, pp. 3330 – 3334, 2007.

[6] A. Barberis, L. Barboni, and M. Valle, "Evaluating energy consumption in wireless sensor networks applications," in *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007.*, pp. 455 – 462, 2007.

[7] Y.-K. Huang, A.-C. Pang, and T.-W. Kuo, "AGA: Adaptive GTS allocation with low latency and fairness considerations for IEEE 802.15.4," in *Communications, 2006 IEEE International Conference on*, vol. 9, (Istanbul), pp. 3929–3934, June 2006.

[8] Y.-K. Huang and A.-C. Pang, "A comprehensive study of low-power operation in ieee 802.15.4," in *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, (New York, NY, USA), pp. 405–408, ACM, 2007.

[9] A. Cunha, A. Koubaa, R. Severino, and M. Alves, "Open-zb: an open-source implementation of the ieee 802.15.4/zigbee protocol stack on tinyos," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference*, pp. 1 – 12, 2007.

[10] R. J. W. Howitt, I; Neto and J. Conrad, "Extended energy model for the low rate wpan," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, p. 8 pp., 7-10 Nov. 2005.

[11] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, "Energy efficiency of the IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives," in *Design, Automation and Test in Europe, 2005. Proceedings*, pp. 196–201, 2005.