
Xenofon Fafoutis
DTU Informatics
Technical University of Denmark
xefa@imm.dtu.dk

Nicola Dragoni
DTU Informatics
Technical University of Denmark
ndra@imm.dtu.dk

ABSTRACT

Energy Harvesting (EH) provides a promising solution to one of the biggest problems faced by Wireless Sensor Networks (WSN), namely the energy supply. By harvesting energy from the surrounding environment, the sensors can have an infinite lifetime without any needs for battery recharge or replacement. Battery-powered WSNs are typically designed to maximize the energy conservation in order to postpone as much as possible the inevitable battery depletion. Instead, EH-WSNs are being designed on a different principle. The focus is on maximizing the network performance while operating at a state that is energetically sustainable. In this paper, we present ODMAC, an on demand MAC protocol for EH-WSNs which is able to support individual duty cycles for nodes with different energy profiles. Hence, each node is able to increase its energy consumption, thus its performance, to the level that the energy consumed is at the same level to the energy harvested. The protocol is implemented and evaluated using the OPNET simulator [13].

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design—Wireless communication; C.2.2 [Computer-Communication Networks]: Network Protocols

General Terms
Design, Performance

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] are systems of multiple small and inexpensive embedded devices that can sense, measure and gather information from the environment they are deployed. The greatest problem faced by WSNs is energy. As a result, there is interest in developing systems that are capable of extracting energy from existing environmental sources [2]. Energy Harvesting Wireless Sensor Networks (EH-WSNs) [3] can provide a solution to this problem by harvesting energy that already exists in the surrounding environment. In this way, energy is essentially infinite; however, not always available. Hence, energy harvesting is introducing a change to the fundamental principles based on which protocols for WSNs are designed. Instead of focusing on energy efficient networking protocols that aim to maximize the lifetime of each sensor, the main objective in EH-WSN is to maximize the performance of the network given the rate of energy that is available to be harvested from the environment. In other words, the surplus of harvested energy can be used to improve the performance of the network. Another important element that differentiates EH-WSNs from classical WSNs is that in EH-WSNs some sensor nodes can be more capable than others. This is due to the non-uniformly distribution of ambient energy. As an example, consider a solar-based EH-WSN where some nodes are covered by shadows while others are under direct sunlight. In such environment, the more capable nodes can be used for performing the energy consuming tasks, on behalf of the incapable nodes that need to sleep and recharge.

In order to achieve infinite lifetime \(^1\), a sensor needs to operate at a energy neutral operation (ENO) state [4]. The concept of ENO refers to a state where the energy consumed by a node is always less than or equal to the energy harvested from the environment. This state guarantees infinite lifetime as soon as there are not any hardware failures. However, it is hard to design the sensors to be always in this state since the dynamics of the environmental energy sources are hard to predict. A battery can be used as an energy buffer that can handle such unpredictable dynamics. Vigorito et al. [5] define the state of ENO-Max which refers to the condition where a sensor operates at the maximum performance while maintaining an energy neutral operation. The idea behind ENO-Max (Figure 1) assumes that energy consumption is highly correlated to performance and guarantees that any surplus of harvested energy can be used for maximizing the performance.

Contribution of this Paper. In this paper we present ODMAC, an On Demand Medium Access Control (MAC) protocol for EH-WSNs in which every node is able to operate as close to the ENO-Max state as possible. In brief, ODMAC has the following key features. First, the communication is on

\(^1\)In the context of this paper, the term infinite lifetime is used in the sense that the lifetime of the sensors are not constrained by the power supply. A sensor may still die due to other causes such as hardware failures.
demand, in the sense that no sensor transmits a frame unless the receiver asks for it. Hence, the nodes can adjust the period of these requests, called beacons, in order to converge into an ENO-Max operating state. Additionally, the protocol provides the network administrator with a tool to select whether the energy consumption adjustments should favor either the relaying or the sensing tasks of the sensor in accordance to the application requirements. Furthermore, ODMAC incorporates an opportunistic forwarding scheme which significantly decreases the end-to-end delay.

Outline of the Paper. ODMAC is presented in Section 3 right after a discussion on related work (Section 2). ODMAC has been implemented as a process model in the OPNET simulator [13]. Simulations, presented in Section 4, evaluate its performance and highlight the key trends and trade-offs. Finally, Section 5 concludes the paper.

2. RELATED WORK

Typical carrier sensing protocols, commonly used in Wireless Networks, are not suitable for WSNs because of the huge wastage of energy through idle channel listening and overhearing. Idle channel listening refers the problem of wasting energy on listening to an idle channel to receive potential traffic. Overhearing refers to the problem of wasting energy receiving frames that are destined to other nodes. A MAC protocol designed for energy-constrained networks, such as WSNs, should be able to prevent these energy wastes.

Traditionally, in battery-powered WSNs, energy conservation in MAC layer is mainly achieved by scheduling and low duty cycles. Sensor-MAC (S-MAC) [6] was a milestone protocol for WSNs. S-MAC defines a MAC protocol in which neighboring nodes form virtual clusters that share a common sleeping schedule. The time is divided in active and sleeping periods. All the sensor nodes of the cluster communicate in the active period, essentially saving energy during the sleeping period. The activity periods are scheduled by periodical SYNC packets between the neighbors. Sleeping periods introduce an additional delay to the total end-to-end delay of each packet which is called sleeping delay. S-MAC is straightforward against the energy wasted in idle listening. However, the sensor nodes cannot have an individual duty cycle according to their energy capabilities. In battery-powered WSNs this is not a limitation as the duty cycles are decided based on a trade-off between absolute performance requirements and energy conservation. On the contrary, in EH-WSNs, we want each sensor node to be able to adjust its performance to the availability of ambient energy. Since neither the energy availability nor the energy consuming tasks are uniformly distributed, an approach where every sensor node is free to select its own duty cycle is more suitable. To make matters worse, in S-MAC the activity of all sensors inside a contention neighborhood is not distributed over time. On the contrary, there are periods of inactivity and periods of contention. This property may be the cause of additional delays and energy wastage caused by collisions.

Dynamic Sensor MAC (DSMAC) [7] is a MAC protocol inspired by S-MAC that introduces dynamic duty cycling. The aim of DSMAC is to decrease the latency for delay sensitive applications in battery-powered WSNs. Similarly to S-MAC, all the nodes inside a contention area follow a similar duty cycle. Whenever there are low-delay requirements, the nodes are able to double their duty cycle. All the nodes that decide to double their duty cycle are now able to communicate twice as frequently as before, essentially decreasing the sleeping delays. Furthermore, they are still able to communicate with the nodes that decide not to double their duty cycle, as they are synchronized with them at half frequency.

DSMAC is heading towards the desired direction. However, it cannot provide the full flexibility required to achieve an ENO-Max state. In contrast to ODMAC, nodes are still not able to freely choose their own duty cycle and the communication keeps not being distributed over time.

There is also related work that considers energy harvesting. The authors of [8] studied several fundamental MAC layer approaches on EH-WSNs. An important limitation of their work is that they considered only the case of single-hop WSNs, i.e., networks where the sensor nodes can directly communicate with the sink without packet relaying by intermediate sensor nodes. This is, indeed, a simplified case, as the sink does not have energy constraints. Hence, the sink never sleeps and issues like synchronization and idle listening are not a challenge in the up-link direction. Despite their insightful conclusions, the investigated schemes cannot be applied in multi-hop sensor networks. We stress that ODMAC considers multi-hop topologies.

3. ON DEMAND MAC (ODMAC)

In this Section the basic components of ODMAC are described; starting with an introduction to the basic idea behind ODMAC and a presentation of the basic communication between two nodes.

3.1 Basic Communication and Error Handling

ODMAC uses the carrier sensing scheme in order to support individual duty cycles and it is designed to minimize the energy wastage through idle listening. This is achieved by exploiting the fact that sensor networks are low traffic networks; typically, orders of magnitude lower than WLANs. In particular, ODMAC moves the idle listening time from the receivers to the transmitters. As a result, it is expected
Figure 2: Typical communication between an ODMAC transmitter and an ODMAC receiver.

to yield beneficial results at low traffic conditions. Each ODMAC receiver periodically broadcasts a beacon which indicates to its receivers that it is ready to accept incoming data packet transmissions. All nodes that have queued packets that need to be forwarded to the sink, are listening to the channel waiting for an appropriate beacon. Upon receiving the beacon, the data packet transmission follows. This approach completely eliminates the idle listening from the receiver node. The receiver just spends energy to periodically broadcast control frames, which in this case help the synchronization of the nodes. On the other hand, the transmitter wastes energy while waiting for a beacon. Nevertheless, the energy wastage at the transmitter is not significant in low traffic networks, where the transmitting queues are most of the time empty.

Figure 2 depicts the basic communication between an ODMAC transmitter and an ODMAC receiver. Assume that the sensor node B needs to transmit a packet to sensor node A. Node B listens the channel waiting for a beacon. At some point, node A wakes up from its sleeping period and attempts to transmit a beacon. First, node A listens to the channel for an amount of time ($T_{IFS}$). Unless the channel is free throughout all that time, node A returns to a sleeping state. If the channel is free, node A transmits the beacon and initiates a waiting timer, $T_{TX}$, while waiting for incoming packets. If no node transmits during the defined time, node A returns to the sleeping state. Node B receives the beacon, identifies that the beacon originates from node A and attempts to transmit. Before the transmission, node B backs off a random number of time slots, $T_{SLOT}$, in the interval defined by the contention window ($CW$), i.e. $[0, CW - 1]$. Unless the channel remains free, node A quits this beacon and begins listening for the next one. If the channel is free, the node initiates the transmission. After the successful termination of the transmission, node B returns to a sleeping state; unless it has more queued packets. In the latter case, it continues to listening for beacons. In an attempt to minimize the waiting time of the transmitters, which is both energy consuming and performance degrading, Node A does not return into a sleeping state. Instead, after each successful transmission, the receiving node immediately retransmits a new beacon. Hence, contention issues are quickly resolved and transmitters with queued packets are helped.

The purpose of the integrated channel sensing timers, which are heavily influenced by typical wireless protocols (e.g. IEEE 802.11 DCF), is to avoid and handle a series of potential errors. One potential error occurs if a node transmits a beacon while another node has already transmitted a beacon and is waiting to receive a packet. It is important that the communication between a receiver and transmitter remains uninterrupted. This issue is solved by the $T_{IFS}$ sensing time. By reassuring that $T_{IFS} > T_{TX}$, ODMAC guarantees that the aforementioned problem cannot appear and the communication between two nodes will never be interrupted. Another potential error is packet collisions. Indeed, if two transmitters are waiting for the same beacon, a collision may occur. This is solved by the random number of time that the transmitter is backing off before transmitting. Indeed, collisions may still happen if both contending nodes randomly select to back off the same amount of time slots. However, the probability of collisions is significantly low. Even if a collision occurs, ODMAC is able to handle them without crushing. Another potential error occurs if two neighboring nodes happen to transmit their beacons exactly at the same time. The beacons will collide to the nodes that are in range of both the beacon transmitters. The protocol is again able to handle the problem. Eventually, the receiving nodes will go to the sleeping state and the transmitters will keep on waiting for the next beacon.

3.2 Duty Cycles

The end-to-end delay is the first performance metric considered. The period of the beacon, $T_{DC}$, defines the duty cycle of the forwarding tasks of each node. In fact, this attribute controls the trade-off between energy consumption and end-to-end delay. The lower the period is, the more beacon transmissions are transmitted and the higher the energy consumption is. On the other hand, when the beacon period is high, the energy conservation is increased as less energy is consumed in beacon transmissions. Additionally, if the duty cycle period is high the routing protocol may route the packet via another path, essentially helping the node conserving more energy. The price to pay for this energy conservation is an increase to the end-to-end delay of the packet, which denotes a performance decrease. The main task of each sensing node is the actual sensing task. The sensing rate is also considered a performance metric of the system. The sensing period, (or packet generation period, $PGP$), defines the duty cycle of the sensing tasks. This period controls the trade-off between energy consumption and sensing rate. It is important to note that a high sensing rate does not only lead to high energy consumption at the respective node, but it also
imply a higher generation of packets that eventually need to be forwarded to the sink. Hence, it also increases the requirements for forwarding services by intermediate sensor nodes. An important feature of the proposed protocol is that the administrator of the wireless sensor network is free to choose how to utilize the excess of energy harvested by environmental sources. In case of applications that require low delays, such as tracking applications, the administrator may choose to increase the relaying duty cycle, aiming to decrease the delays. In case of applications that are not delay-sensitive but there is a need of a consistent data set, such as monitoring applications for off-line statistical analysis; the administrator may choose to increase the sensing duty cycle, aiming to a larger amount of measurements.

ODMAC supports two operating modes, namely the static and the dynamic duty cycle mode. In the static duty cycle mode, the sensing period and the beacon periods of each node are statically set by the administrator. Obviously, ODMAC cannot guarantee that the system would operate at an energy neutral operation state. The selected parameters may lead to an eventual depletion of the energy buffers. In the dynamic duty cycle mode, ODMAC periodically adjusts the two duty cycles in order to reach the ENO-Max state, which is a state where the energy consumed is equal to the energy harvested. As mentioned previously, different applications have different performance requirements. ODMAC offers to the administrator a tool to control if the adjustment of the duty cycle will occur to the sensing or the relaying tasks of the node. In particular, one of the parameters of ODMAC, named $SP_{\text{rel}}$, defines the probability that if there is a need for an adjustment to the duty cycle, this will favor the sensing duty cycle. Obviously, the complementary probability refers to the forwarding tasks. For the duty cycle adjustment, a trivial algorithm is used. In particular, an optimum battery level is selected. Periodically, the battery level is compared to the optimum battery level. When these levels are different, the duty cycle is increased or decreased respectively. The adjustment step is defined by and configurable parameter. Although out of the scope of this project, more sophisticated dynamic duty cycle algorithms [5][9] can be easily incorporated.

3.3 Opportunistic Forwarding
A fundamental problem of ODMAC is that if a receiver is on high duty cycle period, the transmitter will have to wait for a long time essentially wasting energy and increasing the end-to-end delay of the packets. This problem is solved by incorporating an opportunistic forwarding scheme. Instead of waiting for a specific beacon, the ODMAC transmitter opportunistically forwards each frame to the owner of first beacon received as long as it is included in a list of potential forwarders. The selection of the nodes that satisfy the policies and are included in this list is defined by the routing protocol. Apart from the reduction of the waiting time, this approach also introduces an additional benefit. The period of the beacon also controls the packet-relaying load distribution between nodes. A node with a higher beacon period will eventually forward fewer packets than a neighbor with a lower beacon period. Hence, through duty cycle adaptation, the nodes that have access to more environmental energy are able to cover for nodes that are in need for recharging.

In the prototype implementation of ODMAC (Section 4), a trivial routing algorithm is used. The list of acceptable forwarders contains all the nodes that are closer to the sink in terms of number of hops required to reach the destination. Nevertheless, routing protocols that select the set of potential forwarders based on more sophisticated routing metrics (e.g. [10][11]) can be easily incorporated into the system.

3.4 Open Issues
The prototype design of ODMAC has two open issues. First, ODMAC does not acknowledge and retransmit packets. Indeed, the amount of colliding packets is insignificant and can be handled by upper layers. However, wireless communications are notorious for high channel errors that may lead to high packet error rates. Typically, wireless communication protocols solve this issue by MAC layer acknowledgments and retransmissions. The incorporation of a retransmission mechanism in ODMAC and its evaluation in a lossy environment is considered future work.

Another unresolved problem is the well-known hidden node problem, which is typically handled via a CTS/RTS mechanism. In the traffic model of ODMAC, a common result of the hidden node problem is the following scenario. Suppose that there is a node that needs to transmit a packet and it is between two nodes that cannot hear each other. Assume that these two nodes transmit their beacon almost concurrently, so that the beacons do not collide to the node in the middle. The mid-node receives the first beacon and transmits the packet. Apart from the intentional receiver, the packet is also received by the other node that sent the second beacon. ODMAC handles this problem by identifying and discarding these packets. Receiving such packets is a energy wastage factor. However, it is not trivial if the incorporation of a CTS/RTS mechanism is beneficial. Indeed, it needs to be investigated if the energy consumption of the transmission of two additional control packets is lower than the energy consumption of overhearing.

4. SIMULATIONS
In this section we present the simulation results. First, we describe the simulation platform. The simulations performed follow. The first series of simulations demonstrate the correlation of energy consumption to the performance of the system. The second simulations show how the system is able to reach an ENO state. The third group of simulations shows how ODMAC can adapt the performance of the system in various energy harvesting levels using the harvested energy for the different purposes. Finally, the last experiments demonstrate how the system is able to distribute the load to the energetically capable nodes.

4.1 Simulation Platform
ODMAC is implemented as a process model in the OPNET simulator [13]. For the wireless links and the physical layer the default models of OPNET are used. The transmission rate is manually set to 1Mbps and the transmission power is manually set to 10dBm. The time frames are set to follow the following polices. Initially, $T_{TX}$ needs to be greater than $T_{XX}$, as explained in Section 3.1. The value of $T_{RX}$ needs to provide sufficient time for a successful transmission, even when the maximum back-off time is selected. Hence,
it should be greater than \( CW \) times \( T_{\text{SLOT}} \) plus the transmission and propagation delay. The exact values of these attributes are set using heuristics to find the lowest values that satisfy the given policies. The contention window \( (CW) \) is set to 8. Note that there is traffic only in the uplink direction (i.e., from the sensors to the sink). Finally, the initial battery level is set to be equal to the optimum battery level, so that each sensor converges faster to an equilibrium.

A simplified energy model is also incorporated. The energy model has two aspects: energy harvesting and energy consumption. Focusing on the energy consumption of the communication process, three types of energy consumption are considered: the consumption during transmission, reception and listening. Every time a node is transmitting a packet or receives a packet while having the transceiver on, the respective amount of energy is being deducted from the battery level counter. Note that the energy model also takes into account the energy consumed for the reception of discarded packets. Additionally, whenever a node is listening the channel the respective energy is also consumed. The exact values of the battery and energy consumption parameters, used in the simulations, are based on a study on the energy consumption of a real wireless sensor node [12]. The energy harvesting is modeled trivially. Periodically, the node increases its battery level by a specified amount. Note that all the harvested energy rates used in the simulations are realistic according to [3]. In the simulations that follow, the ratio of the energy harvested over the energy consumed is used as a performance metric. Whenever it is above 1 the system operates at an ENO state. So, the exact value of the battery level is obsolete.

ODMAC has been evaluated on a simple topology. It consists of one sink node and 9 sensor nodes which are placed in three groups of three tiers, as shown in Figure 3. The distance between the nodes are placed accordingly so that the nodes of each group can only communicate directly with the nodes of its neighboring group(s) and the nodes of its own group. Hence, the packets that are generated by the third group need to traverse three hops to reach the sink. Note also that whenever a node needs to forward a packet there are three potential next nodes to choose from. Although this topology is sufficient to demonstrate the basic properties of the protocols, an evaluation of ODMAC in random arbitrary topologies is considered future work.

4.2 Performance vs. Energy Consumption

The first series of simulations demonstrate the strong relation between the energy consumption and the performance of the system. The achievable sensing rate and the end-to-end delay are chosen as performance metrics, as explained in the previous sections. Both of them can be improved if the energy consumption is increased and vice versa.

The sensing rate is relative to energy consumption because an increase to the sensing rate corresponds to an increase in the packet generation rate of the sensing node and an increase to the relaying rate of the intermediate nodes. In the following simulations, all dynamic duty cycling mechanisms are deactivated and all nine sensor nodes have an equal energy harvesting rate of \( 400 \mu J \) per second (in fact, \( 2mJ \) every 3 seconds). The beacon period is also fixed to all sensor nodes at 0.2 seconds. The duration of each simulation is 1 hour. Figure 4 depicts the energy operating state of the sensors 1-B and 2-B for various values of the sensing period \( (PGP) \). Note that the sensing period is equal to all nodes. The y-axis is the harvested to consumed energy ratio. Hence, every value above 1 denotes an ENO state and, thus, infinite lifetime. Considering, that all the sensor nodes are identically configured, all the nodes that belong to the same tier yield the same results. Due to their position, nodes that belong to different tiers have different relaying tasks and consequently different energy consumption levels, as shown in Figure 4. The simulations demonstrate that the energy ratio increases as the sensing period increases (i.e., the energy consumption decreases as the sensing period increases). We can also observe that Sensor 2-B has a higher energy ratio, since it just relays packets originating from tier-3 sensors. In addition to those, Sensor 1-B has to relay the packets generated by tier-2 nodes. Hence, the ENO-Max state is different for nodes that belong to different tiers. In particular, Sensor 1-B maximizes the performance (sensing rate) while maintaining an ENO state at a sensing period of 0.6. Sensor 2-B can sense once every 0.4 seconds while maintaining an ENO state.

The ODMAC beacon transmission period also affects the performance, because the end-to-end delay depends on that. As shown in Figure 2, each transmitter has to wait the receiver to wake up and transmit a beacon before the data
packet transmission. This waiting delay depends on the frequency of the beacon transmissions. On the other hand, a short period of beacon transmissions increases the energy consumption that is caused by the radio transmission and the waiting-for-reply listening time. Additionally, the beacon period control the distribution of the relaying tasks between nodes with different energetic capabilities. Although this functionality affects the end-to-end delay, this is demonstrated in following simulations in which nodes do not have equal energy harvesting rates. In these simulations, all dynamic duty cycling mechanisms are deactivated and all nine sensor nodes have an equal energy harvesting rate of $400\mu J$ per second. The sensing period is now set to 0.6 seconds. Figure 5 shows that as the beacon period increases, the average end-to-end delay increases due to the additional sleeping delay. Note that all generated packets are used for the calculation of the average end-to-end delay regardless how far away from the sink are generated. Additionally, the beacon period is equal to all nine nodes. Figure 6 depicts the harvested to consumed energy ratio of Sensor 1-B for the same values of the beacon period. Due to the fact, that a packet transmission is much more energy consuming than a beacon transmission, the variation of the ratio is not as high as in the previous simulations. Nevertheless, there is a clear increasing trend that shows that more energy needs to be consumed to keep delays at low levels.

The simulated experiments show that there is a significant correlation between energy consumption and performance. ODMAC is a protocol that aims to adapt the performance and the energy consumption levels in an attempt to operate as close to an ENO-Max state as possible.

### 4.3 Achieving ENO-Max state

The next simulations aim to show that the system can achieve a sustainable state that maximizes the performance (ENO-Max). All the sensors are set to harvest energy at a rate of $400\mu J$ per second. The sensing period is set to 0.6 seconds and the beacon period at 0.2 seconds. Initially, the adaptive duty cycle functionality is deactivated (Figure 7). Hence, the values of the task periods are constant throughout the experiment. The simulation runs for 1 hour. Observe that nodes in different tiers operate at different states where either the surplus of energy is not used to increase the performance (Sensors 2-B and 3-B) either coincidentally operate at a state that is very close to the ENO-Max state (Sensor 1-B). Of course under different system parameters there would be some nodes that operate at an unsustainable state.

Then, the adaptive duty cycle functionality is activated (Figure 8). Hence, the values of the task periods act as initial values. The $SProb$ parameter is set to 0.5 indicating the sensing and relaying traffic are equally important. Observe now that the operating state of all the nodes gradually converges to 1, which denotes an ENO-Max state. In addition to that, a number of interesting observations can be made. First, we can see that the sensors that are positioned to the outer layer (represented by Sensor 3-B) converge faster to the ENO-Max state. The reason of that is that these nodes do not have relaying tasks. As a result, their energy consumption does not depend on the sensing rate of other nodes. Eventually, these sensors use the surplus of energy to increase their sensing rate. This action leads to a decrease of the energy ratio of the inner nodes (represented by Sensors 1-B and 2-B). Nevertheless, after some time the inner nodes manage to stabilize the energy consumption by adapting their tasks. Note also that at the end the system requires several hours (simulation runs for 7 hours) for every node to stabilize to an ENO-Max state. These simulations reveal that the system might be unable to reach an ENO-Max state. Under some favorable energy harvesting conditions, the nodes in the outer layer may create vast amounts of traffic that the inner nodes are unable to handle. A solution to this problem is considered future work. Potential approaches include feedback between the nodes or the introduction of packet generation constraints based on the distance to the sink.

### 4.4 Energy Availability vs. Performance

The third series of simulations are perhaps the most interesting ones, as they demonstrate how the system is able to adjust the performance to the availability of ambient energy. They also demonstrate how the network administrator can select which performance metric to favor in case of energy surplus or deficit. In the following simulations all sensors but one have the dynamic duty cycle functionalities deactivated and are statically set to a sensing period of 0.6 seconds and a beacon period of 0.2 seconds. These nodes also har-
vest energy at a rate of 400\(\mu\)J per second. Sensor 1-B is the only node whose dynamic duty cycle functionalities are activated. Additionally, its harvesting rate is variable. Each simulation runs for 4 hours. The following figures show how the performance of the network is increased when a sensor is exposed to more powerful energy sources and under different values of the \(SProb\) parameter.

Figure 9 shows the results on the average end-to-end delay. If we consider the average delay of all the generated packets the results are misleading. This happens because the adjustments on the sensing rate may create large amounts of packets generated at tier-1. These packets have, indeed, a low delay not because of the performance adaptation but because of the fact that no relaying is required. Hence, we consider the average delay of just the packets generated in tier-3. Generally, there is a decreasing trend that shows that an increase in the energy harvesting rate is used to decrease the delay. The exception to that observation is when the \(SProb\) is set to 0.25. Under that configuration the system favors the sensing tasks. Thus, when the consumed energy is higher than the harvested energy the beacon period is increased. This leads an increase of the beacon period above 0.2 seconds, which is the static period of the other nodes. As a result, the nodes with a lower beacon period tend to relay more packets keeping the average end-to-end delay to 0.2 seconds. In parallel, Sensor 1-B uses the available energy to favor the sensing tasks. When the \(SProb\) is set to 0.25, the average delay is considered the important performance metric. Under this configuration the sensor decreases its beacon period and, eventually, decreases the average delay. In the case that both performance metrics are equally important, the average delay is somewhere in the middle.

It is interesting to observe the results on the sensing period under the same configurations (Figure 10). A decreasing trend to the sensing period as the harvesting rate increases is also visible here. When the sensing rate is favored (\(SProb = 0.75\)), the sensing period of Sensor 1-B falls down to the minimum value of 0.1 seconds for high energy harvesting rates. Observe that how the sensing tasks are favored compared to the relaying tasks under this configurations. On the contrary, when \(SProb\) equals to 0.25 the results are the opposite. In order to compensate for the favored relaying tasks, the sensor has to decrease its own sensing rate. As expected, the sensing period under the \(SProb = 0.5\) configuration is between the two other configurations. Observe, however, that it is much closer to the configuration that
favors the sensing rate. This happens because Sensor 1-B is the only sensor that adapts its performance and all the relaying burdens are carried by it.

4.5 Load Balancing
The last simulations aim to show how the relaying tasks are shared between nodes according to their energy capabilities by adjusting the beacon period. The simulation setup is similar to the previous simulations with the only variation that the adaptation of sensing tasks is deactivated for all nodes. Hence, the only way for Sensor 1-B to adapt its performance to the availability of energy is to adjust the beacon period. Figure 11 depicts the percentage of the total performance to the availability of energy is to adjust the nodes. Hence, the only way for Sensor 1-B to adapt its performance to the availability of energy is to adjust the beacon period. Figure 11 depicts the percentage of the total performance to the availability of energy is to adjust the beacon period.

Figure 11: Load balancing on the relaying tasks of the sensor nodes.

Finally, the protocol incorporates an opportunistic forwarding scheme, in which the transmitter has a list of potential receivers and sends the packet to the node that wakes up first. Apart from decreasing the delays, this approach can also make easy the incorporation of a distributed load balancing technique regarding the forwarding tasks. Additionally, the protocol would perfectly cooperate with a routing protocol that selects multiple potential next hops according to the routing metrics.

5. CONCLUSIONS
This paper presents the design and performance evaluation through simulations of a MAC protocol for Energy-Harvesting Wireless Sensor Networks. The protocol includes the following key properties. The communication is occurred in an on demand manner, in the sense that the transmitter does not transmit a frame unless the receiver asks for it. This way the nodes are minimizing the idle listening time while maintaining a communication scheme that can support individual duty cycles. The simulations show how the protocol allows the sensors to adapt their performance, namely the sensing rate and the packet delay, in order to adapt the energy consumption to the energy sustainable level that maximizes the performance. In addition to that, the protocol provides the network administrator a way to favor one of the performance metrics. The importance of this functionality derives from the fact that different applications have different requirements. Thus, applications that require low delays can set the protocol to favor low end-to-end delays and vice versa.