

On the Impact of Local Computation Over Routing Performance in Green Wireless Networks

Stefano Basagni^{*}, Valerio Di Valerio[†], Georgia Koutsandria[†], Chiara Petrioli[†]

^{*} ECE Dept., Northeastern University, Boston, MA

E-mail: basagni@ece.neu.edu

[†] Computer Science Department, University of Rome “La Sapienza,” Rome, Italy

E-mail: {divalerio, koutsandria, petrioli}@di.uniroma1.it

Abstract—Superior performance in wireless sensor networks is obtained by taking key protocol decisions based on the outcome of local *learning*-based computations, informing nodes on past and expected availability of resources. This paper investigates the impact on protocol performance of local computational requirements of learning techniques. We consider a recent routing solution, named WHARP, which makes decentralized and proactive decisions based on a Markov Decision Process (MDP) that takes into account key parameters of wireless green networks, including energy harvesting capabilities, and wake-up radio technology. We show that in these scenarios solving the MDP incurs energy expenditures by far superior to that required by wireless communication, even at very high data traffic. In order to maintain the performance advantages of the learning-based protocol machinery, we propose a heuristic solution that closely approximates the MDP trading off optimality for considerably lighter computational requirements. We compare the performance of the heuristic-based WHARP (called W-HEU) to that of the MDP-based WHARP that uses the standard Backward Value Iteration (W-BVI) through GreenCastalia-based simulations with real computational energy measurements. Our results show that W-HEU outperforms W-BVI on key metrics such as energy consumption and packet delivery ratio, making up for the lost optimality of BVI through the remarkable energy savings of its lighter computational requirements.

I. INTRODUCTION

Wireless sensor networks (WSNs) are typically made up of several low cost and low power devices—the sensor nodes. Nodes gather sensory information and communicate wirelessly with other nodes to forward the sensed data towards a network data collector—the sink. Nodes can be equipped with a variety of sensors, making them suitable for a wide range of applications with different characteristics and requirements. Although the use of WSNs is becoming quite widespread, their deployment comes with a number of limitations, such as the finite energy and constrained computing resources of their nodes. These limitations have prompted the adoption of energy-efficient and energy-aware solutions at both hardware and software level. For instance, in order to offset the limitations of batteries (e.g., leakage, relatively short lifetime), alternate methods of energy provisioning have been proposed. These methods include energy scavenging technologies through which nodes harvest energy from the environment, i.e., energy from the sun or wind, and store it in energy storage devices for future use [1]. However, harvesting

sources often exhibit a discontinuous behavior making energy harvesting still insufficient for supporting a wide variety of WSN applications. To help in saving further energy, especially that imposed by wireless communication, sensor nodes can be provided with an advanced and emerging technology that has been shown remarkably effective for reducing energy wastage: Wake-up radios [2], [3]. Nodes are equipped with two transceivers: The main transceiver (main radio) used only for packet exchange, and a low-power wake-up transceiver used to “notify” neighboring nodes when they should activate their main radio. As shown in [2], [3], and also later in this paper, wake-up radio-based communication affects a very small fraction of the energy budget of a node, while achieving considerable energy savings and abating end-to-end data latency incurred by other low power listening (e.g., duty cycling-based) hardware solutions.

Along with hardware advancements, the energy consumption challenge has hastened the design of protocols at all layers of the networking stack that are energy aware and aim to ensure energy sustainability. Among the most promising protocol design techniques, the past decade has shown the increasingly intensive adoption of techniques based on various forms of machine learning [4]. These techniques have the advantage of exploiting historical data to improve the overall performance of the network on given tasks such as routing, scheduling, channel access, etc.. In fact, superior performance is being obtained by taking key protocol decisions based on the outcome of local learning computations that are based on past and expected availability of resources. This allows local optimized choices and obtains global performance gains. However, learning techniques can suffer from high computational costs as nodes drain a considerable percentage of their energy budget to run sophisticated software procedures, predict accurate information and determine optimal decision.

In this work we consider networks with the energy provisioning and conserving technologies mentioned above. More specifically, we consider *green wireless networks*, i.e., WSNs whose nodes are equipped with energy harvesting capabilities and wake-up radios. For these networks we want to investigate the impact of the energy consumption of local computations, especially those induced by machine learning-

based techniques, on the overall network performance. To the best of our knowledge, this is the first work of its kind. In fact, while machine learning-based solutions have flourished for a wide variety of WSN applications, the demonstration of their performance effectiveness does not consider the cost of running the learning methodologies on which they are based.

We consider a recent routing solution, named WHARP (for Wake-up and HARvesting-based energy-Predictive forwarding) that achieves remarkable performance while maintaining nodes operational for the longest period of time [5]. WHARP combines energy prediction techniques and a Markov Decision Process (MDP) to allow each node to decide whether to participate to data forwarding or not proactively. In its original version, WHARP uses a standard method for the solution of the MDP, namely, the Backward Value Iteration (BVI) algorithm [6]. In this paper, we set to study the impact of running BVI periodically for guiding WHARP operations. As our experimental investigation shows that BVI imposes heavy energy consumption, we set out to design a heuristic solution for solving the MDP that is computationally lighter than BVI. Our method trades off the optimality of methods like BVI for greater energy savings. As a consequence, nodes are operational for a longer time, and despite suboptimal decisions the overall network performance improves considerably. Particularly, this paper makes the following contributions.

- We present a new solution method for the MDP at the core of the WHARP forwarding strategy. The solution does not always provide the *optimal decision* that BVI would provide (it is a heuristic solution). However, since its computational requirements are much lighter of those of BVI, WHARP with our heuristic (W-HEU for short) obtains better energy consumption and overall superior network performance.
- We measure the execution time of both methods on real hardware, namely, the MagoNode++ mote that features energy harvesting and wake-up radio capabilities [7].
- We extend the GreenCastalia simulator [8] to also model the energy spent to execute the solution methods based on the measured values.
- We compare the performance of W-HEU to that of WHARP using BVI (W-BVI) through GreenCastalia-based simulations in several scenarios for green wireless networks with varying parameters such as traffic, network density, and energy harvesting sources.

Due to the lower computational energy requirements, W-HEU allows nodes to stay operational for a longer time, independently of the specific scenario considered. Particularly, every node can perform sensing and forwarding duties for *at least 72%* of the simulation time. This is remarkably better than when nodes run W-BVI, where they are instead operational for *at most 72%* of the simulation time.

The remainder of the paper is so organized. In Section II we summarize WHARP and provide the details of the MDP-based policy at its core. Section III presents the details of the proposed heuristic solution method. Performance comparison

results and discussion are introduced in Section IV. Section V discusses works similar to ours. Finally, we conclude this work in Section VI.

II. WHARP: AN MDP-BASED FORWARDING STRATEGY

WHARP provides routing for multi-hop *green wireless networks*, where nodes forward data packets towards a collector node (the *sink*). Nodes are equipped with two transceivers: The main radio for data exchange, and wake-up circuitry for waking up selected neighboring nodes. They can also harvest energy from the surrounding environment. The WHARP packet forwarding strategy is summarized as follows [5].

- 1) When a node i whose hop count-based distance from the sink is l_i has a packet to forward, it broadcasts a wake-up message to wake-up neighboring nodes with hop count $l_i - 1$.
- 2) The recipients of this wake-up message decide whether they should participate to the relay selection process or not based on a Markov Decision Process (MDP) policy. The policy outputs either a *green* decision or a *red* one. Nodes with a *green* output, turn on their main radio, i.e., they wake-up, and wait for a Ready-To-Send (RTS) packet from node i . These nodes are potential relays. Nodes with a *red* decision remain asleep.
- 3) Upon receiving an RTS packet, potential relays reply back to node i with a Clear-To-Send (CTS) packet.
- 4) The packet is sent to the node whose CTS was received first. The recipient is now in charge to keep forwarding the packet to the sink.
- 5) If node i does not receive a CTS from at least one potential relay, steps 1-4 are repeated, for at most K times. If all attempts for finding a potential relay fail, node i drops the packet.

The MDP-based decisions that are at the core of the WHARP strategy are taken locally and periodically (in every *decision epoch*) depending on forecast available energy and on expected energy consumption in the near future. In the remainder of this section we provide a summary of the formulation of the MDP, detailing key elements that are necessary for describing the heuristic solution proposed in Section III.

In deciding whether to be a potential relay or not, the goal of each node is to maximize its ability of forwarding packets without running out of energy. Nodes that have enough energy to forward packets are said to be *active nodes*, while nodes with insufficient energy levels switch to an *all-off* state. We assume that the energy level b_n of a node reflects its state $s_n = b_n$ in decision epoch n , where $b_n \in \mathcal{S} = \{0, \dots, B_{max}\}$ and $1 \leq n \leq N$. Nodes in an *all-off* state are mapped to state $s_n = 0$. The energy harvested by a node in decision epoch n is denoted by h_n , and the energy spent for sensing and for transmitting its own data is denoted by e_n^x . Thus, the available energy for packet forwarding in the n th decision epoch is $e_n = b_n + h_n - e_n^x$.

If with a_f we indicate the availability of a node to forward a packet and with a_d we indicate not being available to do so, deciding whether to be a potential relay or not has the

following consequences on battery energy levels, i.e., on the node state in the following decision epoch depending on the decision taken.

$$s_{n+1} = \begin{cases} e_n & \text{if } a_d \wedge b_n + h_n > e_n^x \\ e_n - e^{tx} & \text{if } a_f \wedge b_n + h_n > e^{tx} + e_n^x \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where e^{tx} represents the energy spent to relay packets from other nodes. State $s_{n+1} = e_n$ models the case of a node that decides not to forward packets from other nodes but that has energy level sufficient for sensing and transmitting its own packets. State s_{n+1} will be $e_n - e^{tx}$ when a node has enough energy for sensing and transmitting its packets and decides to forward packets from neighboring nodes. Finally, if a node does not have sufficient energy for sensing and/or transmitting packets, the node switches to an all-off state, and transits to state $s_{n+1} = 0$. We assume e^{tx} to be a random variable, under a probability distribution $p^{e^{tx}}$ that is continuously estimated by a node during the network operation, independent of the decision epoch n . Conversely, the value of e_n^x is assumed to be constant and equal to a value e^x . This value is estimated at runtime and assumed to be independent of the decision epoch too. The harvestable energy h_n is estimated by using an energy predictor, such as AEWMA [9].

When a node decides to be available for packet forwarding (a_f) in state s_n it transits to state s_{n+1} according to the following probability:

$$P_{s_n \rightarrow s_{n+1}}^{a_f} = \begin{cases} p^{e^{tx}}(e^{tx}) & \text{if } b_{n+1} > 0 \\ \sum_{e^{tx}=e_n}^{\infty} p^{e^{tx}}(e^{tx}) & \text{if } b_{n+1} = 0. \end{cases} \quad (2)$$

Nodes that will have sufficient remaining energy to forward packets from neighboring nodes, i.e., such that $b_{n+1} > 0$, transit to state s_{n+1} with probability $p^{e^{tx}}(e^{tx})$, which is the probability of consuming energy for forwarding packets from other nodes. If instead $b_{n+1} = 0$ (i.e., the node will be all-off), the transition probability corresponds to energy consumption exceeding the node capability of forwarding packets. Finally, when a node chooses not to forward packets (a_d), the transition probability is $P_{s_n \rightarrow s_{n+1}}^{a_d} = 1$, and the next state is uniquely identified by h_n and e_n^x .

As we mentioned earlier, the goal of each node is to be available to forward and generate data packets while remaining active as much as possible. This goal is achieved by providing the node with a “feedback” each time it transits to a new state, indicating how “good” is the transition to that state. This is implemented through a reward function $r(s_n, a_n)$ which provides a positive reward when the node chooses to be available to forward packets (a_f) and a negative one when it transits to an all-off state. The reward when a node decides not to forward packets (a_d) equals to zero. More formally:

$$r(s_n, a_f) = \rho \cdot \sum_{e^{tx}=0}^{e_n} p^{e^{tx}}(e^{tx}) - c \cdot \sum_{e^{tx}=e_n}^{\infty} p^{e^{tx}}(e^{tx}), \quad (3)$$

where ρ is the positive reward that a node receives if it does not run out of energy in the current decision epoch, and c is the cost it incurs instead if it transits to an all-off state. The positive reward ρ and the cost c are weighted by the probability that a node has sufficient energy to forward packets in a decision epoch n .

The *optimal policy* π^* , i.e., the decision of being available to forward the packet (*green*) or not (*red*), is the result of solving the following MDP Bellman equations (*value functions*):

$$V_n^{\pi^*}(s) = \max_{a_n \in \mathcal{A}} \left\{ r(s_n, a_n) + \gamma \sum_{s_{n+1} \in \mathcal{S}} P_{s_n \rightarrow s_{n+1}}^{a_n} V_{n+1}^{\pi^*}(s_{n+1}) \right\}, \quad (4)$$

where a_n ranges in the set of *actions* $\mathcal{A} = \{a_f, a_d\}$, and s_{n+1} ranges in the set of all possible states \mathcal{S} . (The discount factor $0 \leq \gamma \leq 1$ models the uncertainty about the future: The farther the reward is in time, the least important it is [5].)

III. SOLVING THE MDP

An MDP can be solved via standard techniques such as Backward Value Iteration or any other method for solving the Bellman equations (Equation 4) [6]. Backward Value Iteration is, for instance, the method used to solve the MDP in [5].

In this section, we present a heuristic method that, differently from known standard techniques, does not impose high computational energy consumption for solving the MDP. We formulate a simple *threshold policy* that output either *green* or *red* only based on the value of the reward function r corresponding to the decision of being available for forwarding the packet (a_f). Particularly:

In each decision epoch n a node computes the reward function $r(s_n, a_f)$. If $r(s_n, a_f) > 0$, then output *green*, otherwise, output *red*.

Our solution is heuristic in nature, in that it does not always provide the *optimal decision* that other solution methods would provide. However, since computing the reward function is much simpler than solving the Bellman equations through value iteration, linear programming or other standard methodologies, we expect lower computational requirements, better energy consumption and superior network performance.

In the remainder of this section we provide the rationale for our heuristic method. Particularly, we prove that when $r(s_n, a_f) \leq 0$ then our heuristic provides the same optimal decision that other optimal solution method would provide. We then show that when $r(s_n, a_f) > 0$, we output the optimal solution only in case the node does not harvest any energy in the n th decision epoch ($h_n = 0$). In case $r(s_n, a_f) > 0$ and $h_n > 0$ the solution we output, namely, *green*, may be sub-optimal. We start by proving the following:

Lemma 1: For each $n = 1, \dots, N$ the value function $V_n^{\pi^*}(s)$ is non decreasing in s .

Proof: Let us define $q_n(k|s_n, a_n)$ as the probability that state s_{n+1} in decision epoch $n + 1$ exceeds $k - 1$, i.e., that the energy level of a node will be greater than or equal to a value k . Formally:

$$q_n(k|s_n, a_n) = \sum_{s_{n+1}=k}^{\infty} P_{s \rightarrow s_{n+1}}^a. \quad (5)$$

We claim that $q_n(k|s_n, a_n)$ is non decreasing in s_n , for all $k \in \mathcal{S}$, $a_n \in \mathcal{A}$, and $n = 1, \dots, N - 1$. We can prove this by contradiction. Let us assume that $q_n(k|s_n, a_n)$ is a decreasing function of s_n , for all k , a_n , and decision epoch n . We now consider two consecutive states s_n^+ and s_n^- , with $s_n^+ > s_n^-$. It follows that

$$q_n(k|s_n^+, a_n) - q_n(k|s_n^-, a_n) < 0. \quad (6)$$

We denote by s_{n+1}^+ and s_{n+1}^- the states in decision epoch $n+1$ in which the system transits from s_n^+ and s_n^- , respectively. We consider two cases, depending on the action taken.

1) $a_n = a_d$. In this case we know that state transitions are deterministic and uniquely identified by h_n and e_n^x , leading to $s_{n+1}^+ > s_{n+1}^-$ with probability 1. We can define $q_n(k|s_n^+, a_d)$ as:

$$q_n(k|s_n^+, a_d) = \begin{cases} 1 & \text{if } s_{n+1}^+ \geq k, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

(We can do similarly for $q_n(k|s_n^-, a_d)$.) Since $s_{n+1}^+ > s_{n+1}^-$, it follows that $q_n(k|s_n^+, a_d) \geq q_n(k|s_n^-, a_d)$, contradicting our assumption.

2) $a_n = a_f$. In this case state transitions are probabilistic and we can define $q_n(k|s_n^+, a_f)$ (and, similarly, $q_n(k|s_n^-, a_f)$) as:

$$q_n(k|s_n^+, a_f) = \begin{cases} \sum_{e^{tx}=0}^{e_n^+ - k} p^{e^{tx}}(e^{tx}) & \text{if } k < e_n^+, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The intuition is that states higher than k can be reached only if they are lower than the overall energy e_n available for packet forwarding. Since $e_n^+ > e_n^-$, it follows that

$$\sum_{e^{tx}=0}^{e_n^+ - k} p^{e^{tx}}(e^{tx}) > \sum_{e^{tx}=0}^{e_n^- - k} p^{e^{tx}}(e^{tx}) \quad (9)$$

and, consequently, $q_n(k|s_n^+, a_f) \geq q_n(k|s_n^-, a_f)$, contradicting again our assumption.

We can conclude that $q_n(k|s_n, a_n)$ is non decreasing in s_n , for all $k \in \mathcal{S}$, $a_n \in \mathcal{A}$, and $n = 1, \dots, N - 1$. The lemma claim follows from plugging this result into Proposition 4.7.3 of [6], which also uses that for each action a_n and epoch n the reward function $r(s_n, a_n)$ is non decreasing in s_n . This is true by construction. \diamond

The above lemma is key to prove the following result.

Theorem 1: For each decision epoch n , $n = 1, \dots, N$, and state $s_n \in \mathcal{S}$ such that $r(s_n, a_f) < 0$, action a_d is optimal.

Proof: We prove this theorem by contradiction. Let us assume that when the reward is negative it would be better to transmit. From equations (1) and (4) it follows that:

$$r(s_n, a_f) + \gamma \sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx}) V_{n+1}^{\pi^*}(e_n - e^{tx}) > \gamma V_{n+1}^{\pi^*}(e_n), \quad (10)$$

i.e., the value function associated to a_f is higher than that associated to a_d . Since $r(s_n, a_f)$ is negative, we can write:

$$\sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx}) V_{n+1}^{\pi^*}(e_n - e^{tx}) > V_{n+1}^{\pi^*}(e_n). \quad (11)$$

However, V^{π^*} is non-decreasing by Lemma 1 and $V_{n+1}^{\pi^*}(e_n)$ cannot be lower than the weighted sum of values lower than or equal to $V^{\pi^*}(e_n)$ itself. This contradicts our assumption and ends our proof. \diamond

Theorem 1 allows our solution method to output *red* as the optimal decision, i.e., as if it was computed by standard, yet computationally more expensive techniques for solving MDPs. In the following we show that our method outputs optimal *green* decisions provided that no harvesting happens in epoch n . We start with the following lemma.

Lemma 2: If there is no harvesting, i.e., $h_n = 0$ for each $n = 1, \dots, N$, if $r(s_n, a_f) < 0$ then $V_n^{\pi^*}(s_n) = 0$.

Proof: We proceed by backward induction on the number of epochs. In the last decision epoch N , $V_N^{\pi^*}(s) = \max\{r(s_N, a_f), r(s_N, a_d)\}$. Since $r(s_N, a_d) = 0$, it follows that whenever $r(s_N, a_f) < 0$ it is better to drop packets, and $V_N^{\pi^*}(s) = 0$. We now consider a generic decision epoch n so that $r(s_n, a_f) < 0$. From equations (1) and (4) we can write:

$$V_n^{\pi^*}(s) = \max\{r(s_n, a_f) + \gamma \sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx}) V_{n+1}^{\pi^*}(e_n - e^{tx}), \gamma V_{n+1}^{\pi^*}(e_n)\}. \quad (12)$$

Since there is no harvesting, the energy level in decision epoch $n + 1$ has to be lower than the current energy, independently of the chosen action. As a result, the reward function in the next state s_{n+1} will be negative as well and, by the induction hypothesis and Lemma 1, we can rewrite Equation (12) as:

$$V_n^{\pi^*}(s) = \max\{r(s_n, a_f), 0\}. \quad (13)$$

Since $r(s_n, a_f) < 0$, we have that $V_n^{\pi^*}(s) = 0$. \diamond

We can finally prove the following:

Theorem 2: If there is no harvesting, i.e., $h_n = 0$, for each decision epoch n , $n = 1, \dots, N$, and state $s_n \in \mathcal{S}$ such that $r(s_n, a_f) > 0$, action a_f is optimal.

Proof: We proceed by backward induction on the number of epochs. In the last decision epoch if $r(s_N, a_f) > 0$ it is better to transmit packets, otherwise the reward would be 0. We now assume that, in a generic decision epoch n , $r(s_n, a_f)$ is positive but the optimal action is to drop packets. Our assumption can be expressed by the following equation (a straightforward application of the value function definition):

IV. PERFORMANCE EVALUATION

$$r(s_n, a_f) + \gamma \sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx})V_{n+1}^{\pi^*}(e_n - e^{tx}) < \gamma V_{n+1}^{\pi^*}(e_n). \quad (14)$$

Let us define $s_{n+1} = e_n$, and let us evaluate Equation (14) depending on the value of $r(s_{n+1}, a_f)$.

a) $r(s_{n+1}, a_f) < 0$. In this case we know by the induction hypothesis that $V_{n+1}^{\pi^*}(s_{n+1}) = 0$. Thanks to Lemma 1, Equation 14 becomes

$$r(s_n, a_f) < 0$$

which contradicts the assumption that $r(s_n, a_f)$ is positive.

b) $r(s_{n+1}, a_f) > 0$. In this case we can expand the second term in Equation (14) by exploiting the induction hypothesis as:

$$r(s_n, a_f) + \gamma \sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx})V_{n+1}^{\pi^*}(e_n - e^{tx}) < \gamma \left(r(s_{n+1}, a_f) + \gamma \sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx})V_{n+2}^{\pi^*}(e_{n+1} - e^{tx}) \right). \quad (15)$$

Since there is no harvesting, the energy available in state s_n is greater than or equal to that in state s_{n+1} , which implies that $r(s_n, a_f) \geq r(s_{n+1}, a_f)$. We can simplify the above equation as:

$$\sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx})V_{n+1}^{\pi^*}(e_n - e^{tx}) < \sum_{e^{tx}=0}^{\infty} p^{e^{tx}}(e^{tx})V_{n+2}^{\pi^*}(e_{n+1} - e^{tx}). \quad (16)$$

If we focus on each couple of terms $V_{n+1}^{\pi^*}(e_n - e^{tx})$ and $V_{n+2}^{\pi^*}(e_{n+1} - e^{tx})$, we know that they are both zero if $r(e_n - e^{tx}, a_f)$ is negative (Lemma 2). When it is positive, we can expand $V_{n+1}^{\pi^*}(e_n - e^{tx})$ exploiting the induction hypothesis as:

$$\begin{aligned} V_{n+1}^{\pi^*}(e_n - e^{tx}) &= r(e_n - e^{tx}, a_f) + \\ &\gamma \sum_{e^{tx'}=0}^{\infty} p^{e^{tx'}}(e^{tx'})V_{n+2}^{\pi^*}(e_{n+1} - e^{tx} - e^{tx'}) \\ &> V_{n+2}^{\pi^*}(e_{n+1} - e^{tx}). \end{aligned} \quad (17)$$

This is simply the value function equation computed in state $e_n - e^{tx}$ and decision epoch $n + 1$. The last inequality holds because we know by induction that if the reward is positive it is better to choose action a_f than to drop packets. This proves that Equation (16) cannot hold because we have:

$$V_{n+1}^{\pi^*}(e_n - e^{tx}) > V_{n+2}^{\pi^*}(e_{n+1} - e^{tx}). \quad (18)$$

This contradicts our original assumption, ending our proof. \diamond

Theorem 1 and Theorem 2 provide proof that our heuristic method obtains the same optimal solutions that standard methods for solving MDPs would provide, with the exception of epochs with positive rewards and energy harvesting intake. In this latter case, our *green* decision may be sub-optimal.

We demonstrate the benefits of the heuristic solution method presented in Section III through an extensive and diverse set of simulation-based experiments. Particularly, we compare the performance of WHARP where the MDP is solved by the Backward Value Iteration method (W-BVI), with that of when the MDP is solved by our heuristic (W-HEU). The protocol and both solution methods have been implemented in the open source GreenCastalia simulator [8]. GreenCastalia is an extension of the Castalia simulator [10] that is used on top of OMNeT++ to accurately model energy-related aspects of WSNs that benefit from energy harvesting technologies. We further extended its capabilities to model a prototype of wake-up radio [2].

In the following we start by describing the way we determine the energy cost of computing each solution method. We then introduce the simulation scenarios and the metrics we investigate. We conclude the section by presenting our simulation results.

A. Computational energy cost

Commonly with most simulators used in WSN research, GreenCastalia does not take into account the energy consumption needed for computational purposes. This is platform and algorithm-dependent. In our experiments we used the MagoNode++ mote [7], which has energy harvesting and wake-up radio capabilities. The mote is shown in Fig. 1. In order to establish the energy consumption of computing both MDP solution methods, we implemented them in TinyOS (the operative system used by the MagoNode++), and measured their execution time. Using these execution times we computed the corresponding computational energy consumption according to the specifications of the MagoNode++, which uses the ATmega256RFR2 microcontroller, consuming 4.1mA at 3V when in active mode. We then extended GreenCastalia to take into account the computational energy consumption measured on the MagoNode++ by extending the simulator resource manager to take into account the measured computational energy consumption.



Fig. 1. A MagoNode++ mote.

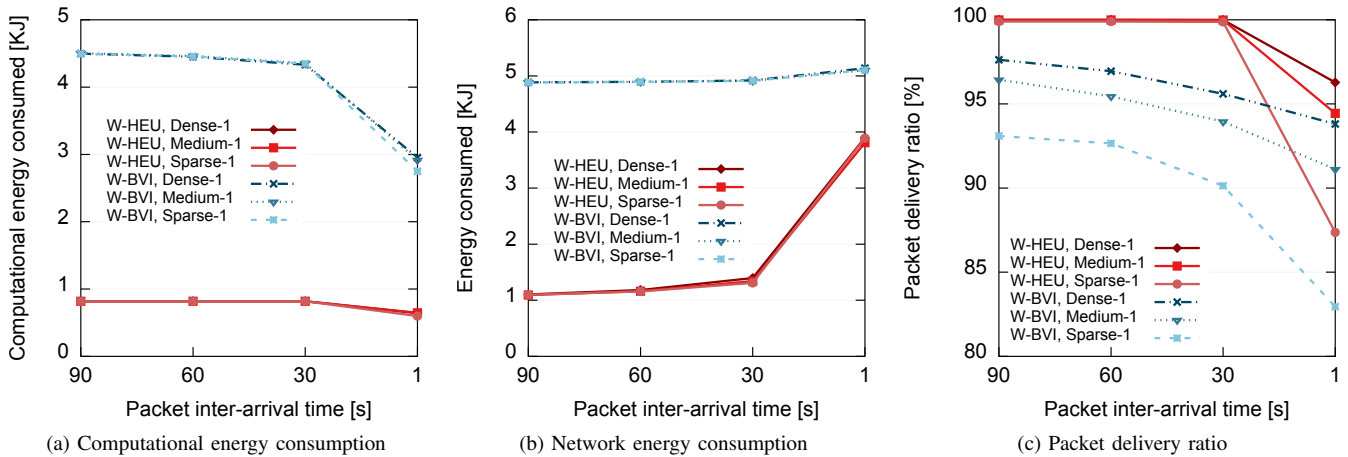


Fig. 2. W-HEU vs. W-BVI for homogeneous harvesting sources in sparse, medium, and dense networks.

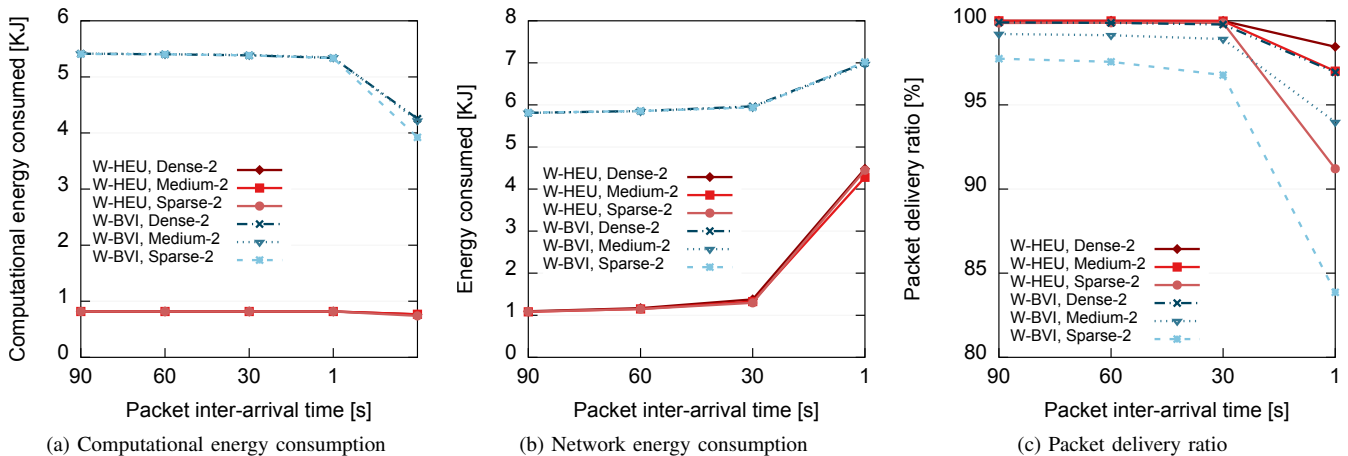


Fig. 3. W-HEU vs. W-BVI for heterogeneous harvesting sources in sparse, medium, and dense networks.

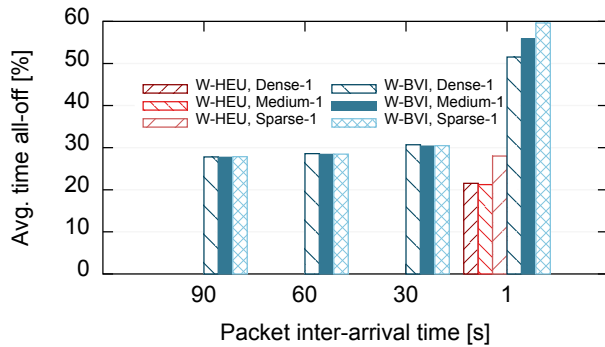
B. Simulation scenarios and parameters

We consider WSNs made up of 64 sensor nodes (modeled as MagoNode++) equipped with on-board Sensirion SHT1x sensors to perform temperature measurements [11]. Nodes are positioned according to a randomized grid deployment, i.e., they are laid down as a 16×4 grid where the actual location of each node is randomly displaced from the precise grid point by 10%. The size of the grid deployment area differs depending on the considered spatial node density. In this work we consider three different sizes: 1) $324 \times 80\text{m}^2$ (sparse scenario); 2) $224 \times 56\text{m}^2$ (medium density scenario), and 3) $160 \times 40\text{m}^2$ (dense scenario). The sink node is placed at the bottom left corner of the deployed area. Traffic is generated as a result of sensing measurements. We stipulate that the per node inter-arrival time between packets ranges in the set of four values $\{90, 60, 30, 1\}$ seconds, corresponding to low, medium, high and very high traffic. Each sensor node acts as both a relay and a source. Data packets have a total size of 58B and they are transmitted at 250Kbps. The energy model considered in our simulations is based on the MagoNode++ mote. The transmission range of the main transceiver is set to 60m. The average path loss between nodes is the result of an estimation based on the lognormal shadowing model [12].

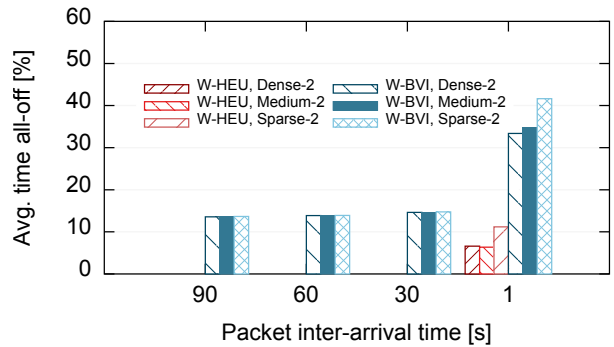
The additive interference model is used to determine packet collisions, according to which interference at the receiver is calculated as a linear sum of the simultaneous transmissions of multiple nodes. The wake-up radio is modeled based on the specifications of a wake-up radio receiver prototype of our design that we tested and characterized [2]. Wake-up sequences are sent at +10dBm using the low-power CC1101 transceiver from the Texas Instruments [13], they are of size 1B and they are transmitted at 1Kbps. The power consumption of the wake-up receiver is $1.071\mu\text{W}$. Its sensitivity is -55dBm , leading to a maximum wake-up range R_w of 45m. This model also considers the power consumption of the integrated ultra-low power microcontroller (MCU) used to perform wake-up addressing, which consumes $0.036\mu\text{W}$ and $54\mu\text{W}$ in idle and active states, respectively. Harvested energy is stored in a supercapacitor with maximum operating voltage of 2.3V and capacitance of 50F [14].

We consider different harvesting sources scenarios.

1) *Homogeneous harvesting sources*: In this scenario, we assume that nodes harvest energy using micro wind turbines. The harvesting traces used in our simulators are real traces collected in Rome, Italy, for a period of one month during summer.



(a) All-off ratio: Homogeneous harvesting sources



(b) All-off ratio: Heterogeneous harvesting sources

Fig. 4. W-HEU vs. W-BVI: Per node average all-off time for increasing traffic (% of the simulation time).

2) *Heterogeneous harvesting sources*: Half of the nodes are equipped with solar cells and the remaining harvest energy using micro wind turbines. Solar harvesting traces are obtained from the National Renewable Laboratory at Oak Ridge [15], while wind harvesting traces are the same as in the homogeneous harvesting scenario.

C. Investigated metrics

The following metrics are investigated to evaluate the performance of the two solution methods.

- 1) The *network computational energy consumption*, defined as the total amount of computational energy spent by all nodes.
- 2) The *network energy consumption*, defined as the total amount of energy spent by all nodes.
- 3) The *packet delivery ratio* (PDR), defined as the percentage of packets successfully delivered to the sink.
- 4) The *all-off ratio*, defined as the average percentage of simulation time a node was all-off.

(We do not show results for metrics such as route lengths and end-to-end data packet latency as we observed that they are not noticeably affected by changing the solution method.)

All results have been obtained by averaging the outcomes of 100 simulation runs, each of duration T_s of 4 days. This number of runs obtains a 95% confidence with 5% precision. In order to evaluate steady-state performance, all metrics are collected after the network setup phase, including hop count determination and training times for the energy predictor.

D. Simulation results

1) *Network computational energy consumption*: Fig. 2a and Fig. 3a show the total computational energy spent by the network for increasing traffic and for homogeneous and heterogeneous energy harvesting sources, respectively. Independently of node density and harvesting source, W-BVI requires at least 5.5 times more energy to compute the solution of the MDP compared to W-HEU. Overall, independently of the network density, the consumption trends of both solution methods are similar. Their actual performance, however, depends on the energy harvesting source. This is mainly because of the higher energy harvesting rates in the case of heterogeneous harvesting sources. The total energy spent for

computational purposes decreases for increasing traffic. This is more noticeable for W-BVI than for W-HEU because by using the former method nodes consume considerably more energy, which sends a higher number of nodes to the all-off state (see also Fig. 4a). As a consequence, with so many nodes all-off, the computational energy consumption is more limited. We notice that at the highest traffic, W-HEU also shows a drop in computational energy consumption. However, even in this case, nodes running W-HEU consume an average of 82% less energy that those using W-BVI (heterogeneous harvesting sources, Fig. 3a).

2) *Network energy consumption*: The total energy consumption incurred by the two solution methods is depicted in Fig. 2b and Fig. 3b. W-HEU always consumes less energy than W-BVI. This is mainly due to the higher computational energy consumption that W-BVI requires to solve the MDP. The performance gap is more noticeable at the lowest traffic, where W-BVI consumes 81% more energy that W-HEU (heterogeneous harvesting sources, Fig. 3b). At higher traffic, and despite lower computational energy consumption, both solution methods consume more energy. This is because of the higher energy toll imposed by dealing with a (considerably) higher number of packets to transmit/receive. However, because of its lower computational energy expenditure, W-HEU manages to consume approximately up to 1.6 times less energy than W-BVI (highest traffic, Fig. 3b). These results show rather clearly the importance of considering the computational energy costs, as their impact on the energy spent by the network is quite remarkable (see also Fig. 5 and discussion below).

3) *Packet delivery ratio*: The packet delivery ratio for the two harvesting source scenarios is shown in Fig. 2c and Fig. 3c. In both scenarios, W-HEU clearly outperforms W-BVI and consistently attains a packet delivery ratio higher than 87%. This is because nodes running W-BVI spend more energy and they tend to go all-off more regularly, as shown in Fig. 4. Nodes running W-HEU remain active in the network for at least 72% of the time allowing higher packet generation rates. As shown in both figures, and especially clearly in Fig. 2c, the PDR of both solutions decreases with increasing traffic. This is because WHARP has to process more packets, deal with more interference and retransmissions, etc., which results in higher energy consumption. The superior perfor-

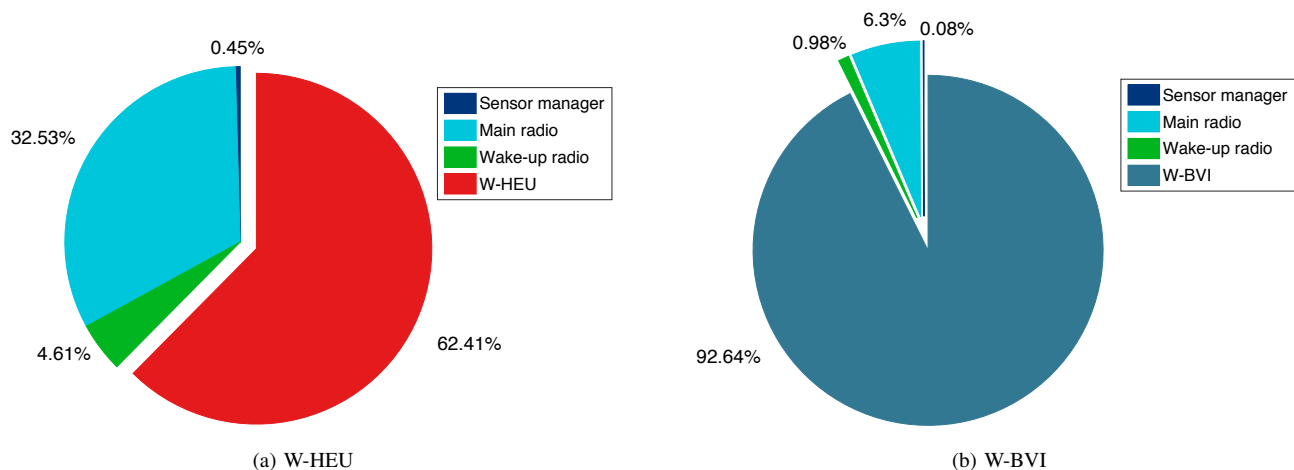


Fig. 5. W-HEU vs. W-BVI: Energy consumption breakdown in networks with packet inter-arrival time of 30s, heterogeneous energy harvesting sources, in a network of medium density (%).

mance of W-HEU depends on the fact that due to the lower energy consumption of the heuristic solution method, more nodes are active, and therefore less packets are dropped for lack of finding potential relays. We notice that the performance of both W-HEU and W-BVI also depends on the density of the network. Specifically, the higher the density, the higher the PDR. This is explained by the higher number of active nodes, which means that a higher number of potential relays are available to the sender to forward its packet. Finally, we notice a difference in the performance of WHARP in scenarios with different energy harvesting sources. In general, the PDR of both W-HEU and W-BVI is higher in the heterogeneous source scenarios. This is because harvesting energy from solar panels provides more energy to the nodes, that are in active state considerably more time than when they only use wind turbines (see also Fig. 4).

4) *All-off ratio*: The average per node all-off ratio is shown in Fig. 4. Independently of traffic and energy harvesting sources, W-BVI experiences higher all-off ratios, which can be up to 3.7 time more than those experienced by W-HEU. This is because of the higher computational cost of solving the MDP using W-BVI. At the lower traffic, nodes running W-BVI go to an all-off state approximately 28 and 14 times more than W-HEU, for homogeneous and heterogeneous energy harvesting sources, respectively. In the heterogeneous case, both solutions perform better than in the case where nodes harvest energy only through wind. This pattern is consistent with the fact that nodes harvest more energy in scenarios with solar harvesting.

To further demonstrate the higher computational energy consumption incurred by W-BVI, we investigate the energy consumption breakdown of different node activities. Fig. 5a and Fig. 5b show the energy consumed by the main radio, the wake-up radio, the sensor manager, and by the solution method for W-HEU and W-BVI, respectively. The figures refer to networks of medium density, with packet inter-arrival time of 30s and heterogeneous energy harvesting sources. The breakdown is expressed as the percentage of the total energy consumption incurred by both solutions (as shown in

Fig. 3b). We observe that in both cases the computational energy expenditure is prevailing over all other forms of energy consumption. The ratio among the energy consumed for local computation vs. that for communication (say, on the main radio) is however visibly different for the two solutions. W-HEU local consumption is almost twice the energy spent for data communications. In case of W-BVI, instead, nodes spend most of their energy for local computational purposes (specifically, 14.7 times more than for data communication). As a result, we note that nodes using W-HEU have more available energy to spend for communication purposes, remain active 100% of the time, and successfully deliver more packets (Fig. 4b and Fig. 3c). The remarkable difference of energy spent for solving the MDP using W-BVI provides further justification of its reduced packet delivery ratio.

V. RELATED WORKS

Despite the wide variety of solutions using learning techniques to increase and optimize the performance of WSNs [4], [16], the energy consumption needed for running the learning frameworks is not explicitly addressed in previous works or determined using real hardware. To the best of our knowledge, the investigation presented in this paper is the first to show the significant energy tolls of running learning-based methodologies, and in proposing a viable (heuristic) alternative. Contrary to widely recognized beliefs, our work clearly points out that local computation can be as impactful on protocol and overall network performance as communication. This was somewhat clear to those researchers who focused on a subclass of problems and applications of WSNs, such as those working on multi-media networking, or on implementing secure communications. For instance, when sensing involves images and video, and more generally, in the realm of multi-media WSNs, it is understood that the energy demanded by the sensory equipment is non-negligible. The paper by Tahir and Farrell provides an example of the importance of finding trade-offs between communication and computation energy consumption in wireless multi-media sensor networks [17]. In

networks concerned with the computation and the transmission of large volumes of video frames, Arastouie et al. investigate the trade-off between computation and communication energy that allows the nodes to make the best decision on the level of compression of the frames [18]. It has always been clear that securing WSN communications is a particularly critical task because cryptographic methods are resource intensive, and the nodes of a WSN have limited resources. As a consequence, a significant number of works has investigated the cost of using security WSN protocols, and proposed solutions that trade off the level of security with energy consumption. Lee et al. analyze the computational cost of securing WSN communications by using different security techniques [19]. Their investigation, based on TelosB and MicaZ commercial motes, provide insights about which solution is suitable for which class of WSN applications, based on different energy costs and the specific application. Similarly, Wander et al. compare the energy expenditure and computation time of two public-key algorithms on wireless sensor nodes [20]. Based on their investigation, they show that in prevailing mote micro-controllers (such as the one of the Mica2dots mote) one method allows viable secure communication for its lighter computational requirements.

Whether for multi-media application for securing WSN communications, these works, and others of this nature, show that local computation can be a major source of energy consumption, whose impact on network performance should not be neglected. This is also what we intend to investigate with our work for networks that use machine-learning based protocol design.

VI. CONCLUSIONS

In this paper we investigate the impact of computational requirements of learning techniques on the performance of protocols for data forwarding in green wireless networks. We introduced a heuristic solution method that approximates the Backward Value Iteration (BVI) algorithm used to solve the MDP in the high-performance routing protocol WHARP. We compared the performance of WHARP using our heuristic (W-HEU) and that of WHARP using BVI (W-BVI) through GreenCastalia-based simulations, where we extended the capabilities of the simulator to reflect the energy consumption due to local computations. Results show that W-HEU outperforms W-BVI because of its remarkably lower computational cost: When nodes run W-HEU the network consumes up to 5.35 times less energy than when they run W-BVI. This results in higher operational times and allows nodes running W-HEU to be active for *at least* 72% of the simulation time, while nodes using W-BVI remain active for *at most* 72% of the simulation time.

ACKNOWLEDGMENTS

This project is partially supported by the MIUR Smart City Smartour project and by the MIUR-funded project for

the Italian Technology Cluster for Smart Communities Techs. Stefano Basagni was supported in part by the U.S. Office of Naval Research under grant N00014-17-1-2046.

REFERENCES

- [1] S. Basagni, M. Y. Naderi, C. Petrioli, and D. Spenza, "Wireless sensor networks with energy harvesting," in *Mobile Ad Hoc Networking: The Cutting Edge Directions*. Hoboken, NJ: John Wiley and Sons, Inc., 2013, ch. 20, pp. 701–736.
- [2] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems," in *Proceedings of IEEE INFOCOM 2015*, April 2015, pp. 522–530.
- [3] M. Magno and L. Benini, "An ultra low power high sensitivity wake-up radio receiver with addressing capability," in *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2014, pp. 92–99.
- [4] M. A. Alsheikh, S. Lin, D. Niyato, and H. P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1996–2018, Fourthquarter 2014.
- [5] S. Basagni, V. Di Valerio, G. Koutsandria, C. Petrioli, and D. Spenza, "WHARP: A wake-up radio and harvesting-based forwarding strategy for green wireless networks," in *Proceedings of IEEE MASS 2017*, Orlando, FL, October 24–27 2017, pp. 1–9.
- [6] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.
- [7] M. Paoli, D. Spenza, C. Petrioli, M. Magno, and L. Benini, "MagoNode++: A wake-up radio-enabled wireless sensor mote for energy-neutral applications," in *Proceedings of ACM/IEEE IPSN 2016*, April 11-14 2016, pp. 1–2.
- [8] D. Benedetti, C. Petrioli, and D. Spenza, "GreenCastalia: An energy-harvesting-enabled framework for the Castalia simulator," in *Proceedings of ACM SenSys 2013*, November 11-13 2013.
- [9] D. K. Noh and K. Kang, "Balanced energy allocation scheme for a solar-powered sensor system and its effects on network-wide performance," *Journal of Computer and System Sciences*, vol. 77, no. 5, pp. 917 – 932, 2011.
- [10] A. Boulis, "Castalia: Revealing pitfalls in designing distributed algorithms in WSN," in *Proceedings of ACM SenSys 2007*, November 6-9 2007, pp. 407–408.
- [11] S. AG, "SHT1x datasheet: Humidity and temperature sensor IC," 2011.
- [12] T. S. Rappaport, *Wireless Communications: Principles and Practice*, ser. Prentice Hall Communication Engineering and Emerging Technologies. Upper Saddle River, NJ: Prentice Hall PTR, 2002.
- [13] T. Instruments, "CC1101 datasheet," rev. SWRS0611, 2013.
- [14] Electric double layer capacitors (gold capacitor). [Online]. Available: <http://www.farnell.com/datasheets/2015895.pdf>
- [15] NREL: Measurement and instrumentation data center. [Online]. Available: <http://www.nrel.gov/midc>
- [16] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 68–96, First 2011.
- [17] M. Tahir and R. Farrell, "Optimal communication-computation tradeoff for wireless multimedia sensor network lifetime maximization," in *Proceedings of IEEE WCNC 2009*, April 5-8 2009, pp. 1–6.
- [18] N. Arastouie, M. Sabaei, and H. S. Shahreza, "A novel approach for trade-off between computation and communication cost in wireless sensor networks," in *Proceedings of International Conference on Advanced Communication Technology (ICACT) 2010*, vol. 1, Feb. 7-10 2010, pp. 820–825.
- [19] J. Lee, K. Kapitanova, and S. H. Son, "The price of security in wireless sensor networks," *Computer Networks*, vol. 54, no. 17, pp. 2967–2978, Dec. 2010.
- [20] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications 2005*, March 2005, pp. 324–328.