Towards Energy-Fairness in Asynchronous Duty-Cycling Sensor Networks

ZHENJIANG LI and MO LI, Nanyang Technological University
YUNHAO LIU, Tsinghua University

In this article, we investigate the problem of controlling node sleep intervals so as to achieve the min-max energy fairness in asynchronous duty-cycling sensor networks. We propose a mathematical model to describe the energy efficiency of such networks and observe that traditional sleep interval setting strategies, for example, operating sensor nodes with an identical sleep interval, or intuitive control heuristics, for example, greedily increasing sleep intervals of sensor nodes with high energy consumption rates, hardly perform well in practice. There is an urgent need to develop an efficient sleep interval control strategy for achieving fair and high energy efficiency. To this end, we theoretically formulate the Sleep Interval Control (SIC) problem and find out that it is a convex optimization problem. By utilizing the convex property, we decompose the original problem and propose a distributed algorithm, called GDSIC. In GDSIC, sensor nodes can tune sleep intervals through a local information exchange such that the maximum energy consumption rate of the network approaches to be minimized. The algorithm is self-adjustable to the traffic load variance and is able to serve as a unified framework for a variety of asynchronous duty-cycling MAC protocols. We implement our approach in a prototype system and test its feasibility and applicability on a 50-node testbed. We further conduct extensive trace-driven simulations to examine the efficiency and scalability of our algorithm with various settings.

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

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Wireless sensor networks, duty-cycling, energy-fairness

ACM Reference Format:
Zhenjiang Li, Mo Li, and Yunhao Liu. 2014. Towards energy-fairness in asynchronous duty-cycling sensor networks. ACM Trans. Sensor Netw. 10, 3, Article 38 (April 2014), 26 pages.
DOI: http://dx.doi.org/10.1145/2490256

1. INTRODUCTION

Recent years have witnessed the great success of Wireless Sensor Networks (WSNs). As a promising technique, WSNs have spawned a variety of critical applications in practice. In WSNs, sensor nodes are usually powered by batteries, while frequent replacements of such power sources are normally prohibited. To close the gap between the limited energy supplies of sensor nodes and the long-term deployment requirement in many applications, recent research works suggest to operate sensor nodes in a

This study was supported by Singapore MOE AcRF Tier 2 grant MOE2012-T2-1-070. This study was also supported by NAP M4080783.020, the NSFC Major Program No. 61190110, and NSFC Distinguished Young Scholars Program 61125202.

A preliminary version of this study was presented at IEEE INFOCOM 2012 [Li et al. 2012].

Authors' addresses: Z. Li and M. Li, School of Computer Engineering, Nanyang Technological University; email: {lzjiang; limo}@ntu.edu.sg; Y. Liu, School of Software, Tsinghua University; email: yunhao@greenorbs.com.

DOI: http://dx.doi.org/10.1145/2490256

duty-cycling work mode [Ye et al. 2002]. In duty-cycling WSNs, radios of sensor nodes are controlled on a periodical basis, alternating between the active and dormant states. In the active state, sensor nodes can send or receive data, while in the dormant state they switch radios off to save energy. For instance, with a 5% duty cycle, sensor nodes have radios on only for 5% of the time. The duty-cycling operation therefore significantly reduces the energy consumption rates of sensor nodes and dramatically prolongs the network lifetime.

The duty-cycling operation has been employed in a variety of MAC-layer protocols, which can be basically classified into *synchronous* and *asynchronous* two categories. Typical synchronous protocols, as in Ye et al. [2002, 2006] and Dam and Langendoen [2003], enable sensor nodes to synchronously sleep and wake up, providing intermittent network services. The required time synchronization introduces tremendous communication overhead and computation complicity. Asynchronous protocols, however, allow sensor nodes to operate independently. At an arbitrary time instance, a subset of sensor nodes operates to provide consistent network services. Most asynchronous protocols typically employ *Low Power Listening* (LPL) based approaches [Polastre et al. 2004; Buettner et al. 2006; Liu et al. 2009], including the original LPL technique or some other optimized techniques like *strobed preamble*, to achieve asynchronous data transmissions. The basic principle of those protocols is that prior to the data transmission, a sender transmits a preamble lasting as long as the sleep period (i.e., *sleep interval*) of the receiver. The receiver is, thus, guaranteed to detect the preamble and receive the data. Compared with synchronous protocols, asynchronous protocols are free of time synchronization and robust to network dynamics, which are beneficial for large-scale deployments. Recently, some variant techniques, for example, *Low Power Probing* (LPP), have been proposed to enable receiver-initiated duty-cycling data transmissions. As all those above techniques share similar energy efficiencies, for the sake of clear presentation, we take LPL-based approaches as a vehicle to discuss the energy fairness issue in asynchronous duty-cycling sensor networks, and further extend our analysis and solution to other variant techniques.

Though the asynchronous duty-cycling operation releases the constraint of time synchronization and enables robust sensor networks in dynamic environments [Liu et al. 2011a], there remain excessive challenges for applying such an operation to manage the limited energy supplies of sensor nodes and approach a long network lifetime. First, the choice of sleep interval at any given node affects not only its own energy drain to periodically access the channel, but also the energy consumption of neighbor nodes communicating with it. In particular, by selecting a relatively large sleep interval, one sensor node will poll the channel less frequently with reduced energy drain and vice-versa. On the other hand, as the LPL technique requires that preambles sent from senders should cover the entire sleep periods of receivers, setting a large sleep interval unavoidably increases the energy consumption of packet senders for the current recipient node. Such an energy tradeoff challenges the appropriate choice of sleep intervals for different sensor nodes, and we call the problem *Sleep Interval Control* (SIC). Second, the traffic load usually distributes unevenly and varies in the network in many applications. As the traffic load directly affects the preamble and wake-up time of individual sensors as well, the choice of sleep intervals cannot be determined separately from the traffic load awareness. If the SIC strategy is not well designed, certain nodes could rapidly deplete their energy and become the energy bottleneck, which severely breaks the network-wide energy fairness and thereby shortens the network lifetime. Thus, SIC becomes more challenging as it should be traffic-aware. In addition, the problem will get even worse if the network scale is large, demanding distributed solutions.

There have been excessive studies tailored for achieving the energy fairness to prolong the network lifetime of sensor networks. Nevertheless, they cannot be directly applied to the asynchronous duty-cycling context [Rangwala et al. 2006; Gu and He 2007; S. J. Tang et al. 2011; Chen et al. 2010; Ma et al. 2011; Zhu et al. 2011]. There have also been attempts made towards the SIC problem in duty-cycling WSNs. Most of them, however, investigate bounding the end-to-end transmission delay or adjusting the energy consumption of sensor nodes in a centralized fashion and ignoring the traffic impact [Wang et al. 2010; Merlin and Heinzelman 2009; Park et al. 2010; Zhu 2012]. None of them tackles the SIC problem with a general setting to prolong the network lifetime in a distributed manner. So far as we know, many fundamental issues in SIC are not well understood and an instrument to tackle such problems is still lacking to the community.

In this article, we thoroughly investigate the SIC problem to achieve the min-max energy fairness in asynchronous duty-cycling sensor networks. The contributions of this article are as follows. We propose a mathematical model to describe the energy efficiency of sensor nodes in existing LPL based asynchronous duty-cycling sensor networks, which captures the essential energy tradeoff between senders and receivers. Based on the proposed model, we observe that existing simple sleep interval control mechanisms perform far from the optimal one, and there is an urgent need to develop an efficient SIC strategy. Aiming at dealing with the SIC problem in general, we theoretically formulate such a problem and find out that it is a convex optimization problem. Based on the convex property, we decompose the original problem into suboptimization problems, and develop a distributed algorithm, called GDSIC, to approach the optimal result. In GDSIC, with a solely local information exchange, sensor nodes can determine how to adjust their sleep intervals such that all sensor nodes within the network converge to the optimal sleep interval settings and the maximum energy consumption rate in the network can be minimized. The GDSIC algorithm is self-adjustable to the traffic load variance and is able to serve as a unified framework for a variety of underlying asynchronous duty-cycling protocols. We implement a prototype system on a 50 TelosB Mote testbed. The experiment results validate the feasibility and applicability of the proposed approach in practice. We further conduct extensive and large-scale trace-driven simulations to examine the efficiency and scalability of the proposed algorithm.

The rest of this article is organized as follows: related works are reviewed in Section 2. In Section 3, we model the energy efficiency of sensor nodes and evaluate the traditional SIC strategies. We formulate the SIC problem and propose our solution in Section 4. In Sections 5 and 6, we examine the performance of our approach. We conclude in Section 7.

2. RELATED WORK

In existing literatures, the duty-cycling MAC-layer protocols can be roughly divided into two categories: *synchronous* and *asynchronous* protocols. Typical synchronous protocols include Ye et al. [2002, 2006] and [Dam and Langendoen 2003]. In S-MAC [Ye et al. 2002], sensor nodes are configured with fixed duty-cycle ratios. S-MAC relies on the periodical synchronization among neighbors and a series of synchronizers to cooperate nodes in the network. The network lifetime can be prolonged compared with traditional always-on networks. However, the energy efficiency of S-MAC is usually low. To solve such an issue, T-MAC in Dam and Langendoen [2003] is further proposed. T-MAC can adjust the duration of the active state for each node based on various message rates. Later, in SCP-MAC [Ye et al. 2006], sensor nodes can achieve extremely low duty cycles based on a two-window contention design. The major limitations of synchronous protocols are tremendous communication overhead and computation complicity for time synchronization [Y. Wang et al. 2012]. Asynchronous protocols, on the other hand, allow sensor nodes to operate independently. The first reported asynchronous MAC-layer protocol is B-MAC [Polastre et al. 2004], which applies the original LPL technique.

Afterwards, subsequent protocols, like X-MAC [Buettner et al. 2006], C-MAC [Liu et al. 2009], and WiseMAC [EI-Hoiydi and Decotignie 2005], are essentially similar to B-MAC. However, some optimizations, including the strobed preamble and predictive wake-up techniques, have been introduced in those protocols to further reduce the energy consumption. In X-MAC, senders transmit a series of short preambles instead of one long preamble. Two consecutive short preambles are separated via a brief idle time slot. Whenever receivers wake up and hear the preamble, they will acknowledge senders during those idle time slots. By doing so, the preamble transmission can be stopped and senders can launch the data transmission immediately. C-MAC implements a similar idea by using RTS/CTS. Sensor nodes in WiseMAC utilize feedbacks from the receivers to predict their wake-up times. Then, the length of preambles can be shortened to save energy. Since preambles are sent from senders, aforementioned asynchronous protocols are also referred to as *sender-initiated* protocols. Different from sender-initiated protocols, recently, some *receiver-initiated* protocols have been proposed, such as RI-MAC [Sun et al. 2008], PW-MAC [L. Tang et al. 2011], A-MAC [Dutta et al. 2010], etc., which are mainly designed to improve the channel utilization and unify services, by employing the LPP technique. Based on [Challen et al. 2010], the energy drain in receiver-initiated protocols can be similarly analyzed as the sender-initiated ones. In this article, we take the LPL-based protocols as an instrumental vehicle due to LPL's availability in the standard TinyOS distribution, while dealing with LPP-based protocols as a promising extension.

The energy issue in sensor networks has drawn people's attention in the past several years. Gu and He [2007] propose DSF to optimize the expected energy consumption for data forwarding in low duty-cycling sensor networks. Guo et al. [2009] introduces an opportunistic scheme to achieve a rapid and energy-efficient flooding in duty-cycling wireless sensor networks. Although the routing path can be optimized based on those previous studies, traffic loads are still unevenly distributed. As a result, we still need to design solutions to balance the energy fairness of the entire network. On the other hand, to achieve sustainable operations, a series of works have exploited the sensor networks with a harvested power management [Hu et al. 2009] or powered by ultracapacitors. Zhu et al. [2009] first investigate the leakage-aware energy synchronization in such networks. Then, the study in Zhu et al. [2010] extends to explore the capacitor-driven energy storage and sharing for a long-term operation. Gu et al. [2009] further examines how to integrate the capacitor-powered sensor networks with the duty-cycling operation. However, due to the high cost of capacitors and the design complicity, such a new networking paradigm has not been widely adopted in large-scale sensor networks.

There are also some primary efforts to control sleep intervals in WSNs. Wang et al. [2010] propose Dutycon to achieve a dynamic duty cycle control for the end-to-end delay guarantee. The study in Zhu [2012] bounds the communication delay in energy harvesting sensor networks. In both Merlin and Heinzelman [2009] and Park et al. [2010], multiobjective optimization formulations are introduced, covering transmission reliability, end-to-end delay, and energy consumption. Optimization problems are solved by classical methods in a centralized manner. IDEA in Challen et al. [2010] integrates multiple networking services, like LPL adjustment, energy-aware routing, and localization. Sensor nodes balance the local energy consumption in a heuristic fashion and it is not clear how close the achieved performance is to the optimal result. As energy is the most significant issue limiting the network performance [Dutta et al. 2008], different from previous works, we focus on controlling sleep intervals to achieve the min-max energy fairness so that the network lifetime can be notably prolonged. To make our approach practical, we require that the solution should be completely distributed and self-adjustable to the traffic variance, which is common in many applications. In addition, we also require that our solution can serve as a unified framework applicable to a

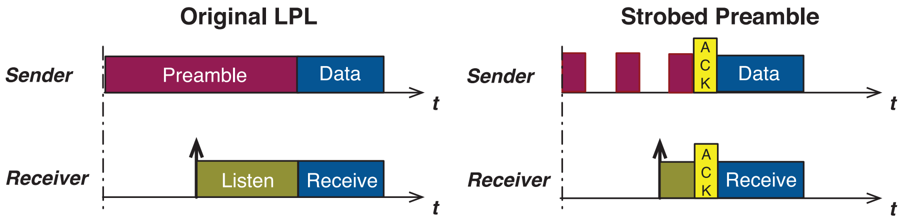

Fig. 1. Illustration of the LPL technique.

variety of asynchronous MAC protocols. So far as we know, such an instrument is still lacking.

3. PROBLEM SPECIFICATION AND DESIGN CHALLENGES

In this section, we mathematically characterize the energy efficiency[1] of sensor nodes running LPL-based asynchronous protocols, and evaluate existing sleep interval control strategies in practice.

As depicted in Figure 1 (left), a sender transmits a *long* preamble prior to the data transmission with the original LPL. After the receiver wakes up and detects the preamble, it keeps awake to receive data. Later, such a working mechanism has been further optimized due to the low energy efficiency at the receiver side, and the most representative example is the *strobed preamble* technique. As shown in Figure 1 (right), instead of sending a long preamble, a serials of short preambles are sent such that intended data can be transmitted without waiting until the end of the long preamble. Since such a technique notably increases the energy efficiency and it is robust to dynamic environments, it has been widely used in large-scale WSNs in practice, like GreenOrbs [Liu et al. 2011a], and released as the default LPL-based MAC protocol in TinyOS. As optimized techniques are proposed based on the original LPL design, we first investigate energy consumption rates of sensor nodes with the original LPL technique in this section, then we observe that later proposed protocols can be transformed to its special cases. Before we proceed, for any sensor node (e.g., i) in the network, we introduce two notations:

—r_i is the overall energy consumption rate of node i,
—T_i^{slp} is the sleep interval of node i.

In general, each r_i is the summation of energy consumption rates for packet transmitting (r_i^{tx}), packet receiving (r_i^{rc}), channel polling (r_i^{cp}) and overhearing (r_i^{oh}) at sensor node i. As a result, r_i can be expressed by:

$$r_i = r_i^{tx} + r_i^{rc} + r_i^{cp} + r_i^{oh}. \tag{1}$$

After specifying each term in Eq. (1), we obtain a general expression for the overall energy consumption rate of any sensor node i in the following theorem.

THEOREM 3.1. *With the LPL technique, the overall energy consumption rate at any sensor node i can be unified by*

$$r_i \;=\; \lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i, \tag{2}$$

[1] Without loss of generality, we focus on the rate of energy consumption (i.e., the energy drain in one unit time) in this section, as the total energy consumption can be obtained by multiplying the rate and the time duration.

where node j receives the packets sent from node i, λ_i, γ_i, ζ_i, and τ_i are coefficients to simplify the expression of r_i.

The detailed derivation of Theorem 3.1 can be found in Appendix A. Based on Theorem 3.1, we will (1) evaluate existing sleep interval control strategies in practice; and (2) identify design challenges for the SIC problem.

3.1. Problem Specifications

To the best of our knowledge, most deployed WSNs in practice employ the identical sleep interval setting due to the design and implementation simplicities. However, it is well known that the in-network traffic load is usually unevenly distributed [X. Wang et al. 2012; Du et al. 2011; Wang and Liu 2011; Liu et al. 2011b] and recent measurement studies, like Liu et al. [2011a], have also reported such a phenomenon. We observe that such a simple strategy may lead to heterogenous energy drains and hardly achieve the energy fairness in the network. As a result, the network lifetime will be severely limited.

THEOREM 3.2. *The identical sleep interval setting usually results in heterogeneous energy consumption rates in practice.*

The rigorous interpretation to Theorem 3.2 can be found in Appendix B, while we briefly explain Theorem 3.2 here. According to Theorem 3.1, we can demonstrate that the energy consumption rate r_i of any sensor node i is mainly determined by its outgoing (transmitting) traffic rate f_i^{tx} when all sensor nodes are set an identical sleep interval. As mentioned previously, the network traffic in practice is normally heterogenous. Therefore, sensor nodes in heavy traffic regions are prone to suffer more frequent preamble time and longer data receiving time. As a consequence, those sensor nodes tend to run out of energy first, and traffic loads are prone to dominate the lifetime of sensor nodes when all sleep intervals are set to be equal. In Section 3.2, we will conduct a case study in data collection to further validate such a conclusion.

Theorem 3.2 essentially demonstrates that due to the inherent *uneven* nature of traffic loads in practice, the widely adopted sleep interval setting policy in existing sensor networks fails to gain a good performance in terms of the energy efficiency. To deal with such an issue, sleep intervals should be controlled dynamically with respect to sensors' energy draining speeds and traffic load variances. An intuitive solution is to increase the sleep interval of a sensor node greedily if its energy consumption rate becomes higher [Challen et al. 2010]. The rationale behind is that prolonging the sleep interval of this sensor node compensates its fast energy consumption. However, as we will show in Theorem 3.3, the hardness of the SIC problem is beyond such an intuition.

THEOREM 3.3. *The greedy SIC strategy by increasing sleep intervals of sensor nodes with large energy consumption rates hardly achieve the min-max energy fairness in WSNs.*

In Figure 2 and Figure 3, we show the energy consumption rate of a sender node with respect to different sleep interval settings. The upper figure in either Figure 2 or Figure 3 depicts that for any sensor node i, how does r_i in Eq. (2) vary with T_i^{slp} when T_j^{slp} is fixed. If we focus on each individual sensor node, its energy consumption rate is indeed decreased in some scenarios when the sleep interval increases. As depicted in Figure 3 (upper), the strobed preamble technique belongs to this category. However, there exist sufficient exceptions. For instance, sensor nodes adopting the original LPL technique in the region with high traffic loads as shown in Figure 2 (upper). It hinders above intuitive heuristics to be applied directly in general.

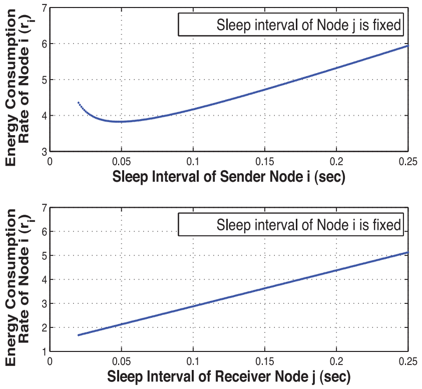

Fig. 2. r_i vs. sleep intervals with the original LPL technique.

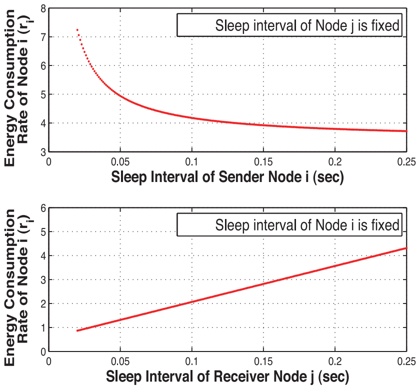

Fig. 3. r_i vs. sleep intervals with the strobed preamble technique.

On the other hand, from the network perspective, Eq. (2) implies that after a sensor node increases its sleep interval, energy consumption rates of its senders increase accordingly. The lower figure in either Figure 2 or Figure 3 depicts how does r_i in Eq. (2) vary with T_j^{slp} when T_i^{slp} is fixed. As a matter of fact, the sleep interval adjustment of one sensor node will trigger senders to tune their own sleep intervals as well. In the greedy strategy, energy drains of sensor nodes are balanced within neighborhoods, which essentially follows the "water-leveling" mechanism. By doing so, energy consumption rates of sensor nodes could be converged to a compromised value. On the other hand, the initial sleep interval setting has implicitly defined an interval, within which the min-max energy fairness can be adjusted by the greedy strategy. However, there is no guarantee that the optimal min-max energy fairness falls within the formed interval exactly. Therefore, the greedy strategy is not always effective, which challenges the algorithm design for SIC. In the next section, we will revisit both Theorems 3.2 and 3.3 to validate those conclusions by a concrete case study.

3.2. Case Study in Data Collection

Data collection is one important networking service for WSNs [Werner-Allen et al. 2008]. In data collection, to receive network-wide data, *data collection tree* [Gnawali et al. 2009] and *Directed Acyclic Graph (DAG)* are two major approaches proposed in existing literatures. While each sensor node has only one data receiver in a collection tree, in DAG each node can forward data to multiple receivers closer to the sink [Lin et al. 2008]. In this section, we focus on the data collection tree since DAG can be similarly analyzed. We will examine both two approaches in Section 6 via experiments and simulations.

LEMMA 3.4. *For any node i that is l-hop away from the sink node in a data collection tree, its outgoing traffic can be approximated by:*

$$f_i^{tx}(l) \;=\; \rho(L^2 - (l-1)^2 d^2)/(2l-1)d^2, \tag{3}$$

where d is the average distance of one hop, ρ indicates the average traffic generation rate in the network and L is the maximum distance from the network boundary to the sink.

The detailed derivation of Lemma 3.4 is given in Appendix C.

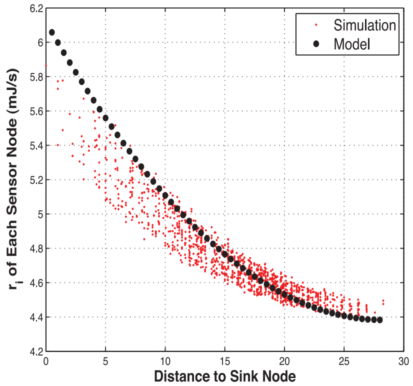

Fig. 4. r_i of each sensor node with the identical sleep interval policy.

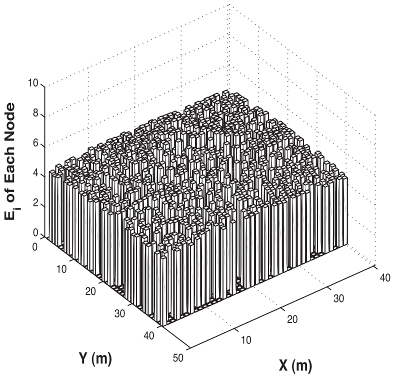

Fig. 5. Distribution of energy consumption rate with the greedy policy.

3.2.1. Revisiting Theorem 3.2. According to Lemma 3.4, sensor nodes closer to the sink consume their energy exponentially faster than other distant nodes. On the other hand, as we have mentioned, when the identical sleep interval setting is adopted, the energy consumption rate of one sensor node i is mainly determined by f_i^{tx}. Therefore, those heavy traffic burden nodes tend to run out of energy first, and those nodes are usually located close to the sink node. We perform a simulation study in Figure 4, in which the Y-axis of a red dot represents the energy consumption rate of one sensor node. As indicated by Eq. (3), the traffic load is relatively high in the region near the sink node. Therefore, sensor nodes in such a region consume energy much faster, which is consistent to our previous discussion.

3.2.2. Revisiting Theorem 3.3. We apply the greedy sleep interval control strategy for the sensor network and illustrate the energy consumption rate of each sensor node after the network becomes stable in Figure 5. In the greedy strategy, sensor nodes adjust their sleep intervals such that their energy consumption rates are set as the average value of their neighbors, which is an intuitive way to achieve the min-max energy fairness in the network. Compared with the identical sleep interval setting policy, the greedy strategy effectively reduces the maximum energy consumption rates of the network. The achieved energy fairness (i.e., 4.9 mJ/s) is within the interval implicitly formed by the maximum (i.e., 5.8 mJ/s) and minimum (4.4 mJ/s) values in Figure 4. However, in such an example, the optimal min-max fairness is 2.8 mJ/s. Figure 5 shows that the greedy strategy leads the network converging to a suboptimal value that is far above the optimal result, which will cause a non-negligible gap between the achieved network lifetime and the optimal performance.

3.3. Design Challenges

Based on these discussions, we can summarize the design challenges for the SIC problem as follows.

- Increasing the sleep interval of one sensor node does not necessarily reduce its own energy consumption rate.
- A sensor node increases its own sleep interval to save energy; nevertheless, energy consumption rates of the packet senders of the current receiver may increase.
- The achieved energy fairness may be far away from the optimal result if sleep intervals are not carefully controlled.

In the next section, we will introduce our solution to deal with those challenges to achieve an optimized sleep interval control.

4. PROBLEM FORMULATION AND ALGORITHM DESIGN

The sensor network is modeled as an undirected graph $G = \{V, E\}$, where V and E represent the sets of sensor nodes and wireless links, respectively. According to Theorem 3.1, the energy consumption rate of an arbitrary sensor node i in the network can be expressed as $r_i = \lambda_i \cdot T_j^{slp} + \gamma_i / T_i^{slp} + \zeta_i \cdot T_i^{slp} + \tau_i$, where j is the receiver[2] of node i. To control the energy consumption rate in the network, we introduce a set of variables R_is and require that $\lambda_i \cdot T_j^{slp} + \gamma_i / T_i^{slp} + \zeta_i \cdot T_i^{slp} + \tau_i \leq R_i$ for each i. As previously mentioned, by determining an appropriate sleep interval for each sensor node, the Sleep Interval Control (SIC) problem aims at minimizing the maximum energy consumption rate (i.e., the min-max energy fairness) in the network to prolong the network lifetime, which can be captured by the model from Eq. (4) to Eq. (6) as follows:

$$\min \quad \max_i \{R_i\} \tag{4}$$

$$\text{such that} \quad \lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i \leq R_i, (i, j) \in E, \tag{5}$$

$$0 < T_i^{slp}, i \in V. \tag{6}$$

Constraint (5) specifies that the energy consumption rate of each sensor node is bounded from above by the variable R_i. Constraint (6) guarantees that sleep intervals have positive values. The coefficients λ_i, γ_i, ζ_i, and τ_i $i \in V$, are all positive as well. Thus, constraints (5) and (6) implicitly ensure that $R_i > 0$. In the end, the objective function (4) minimizes the maximum R_i so that the global min-max energy fairness can be achieved in the network.

A straightforward way to obtain the optimal SIC result based on this formulation is as follows.

—Each sensor i measures its own traffic load, calculates λ_i, γ_i, ζ_i, and τ_i, and reports the calculated coefficients to a central information collector, for example, the sink.
—Based on the harvested information from each sensor node, the sink globally solves Eqs. (4) to (6).
—The sink node disseminates the optimal sleep interval setting to the entire network.
—To be traffic variance-aware, these three steps are repeated periodically or triggered via the sink when traffic dynamics are detected.

However, such a centralized solution normally incurs tremendous communication overhead and complicated cooperation among sensor nodes, which hinders the scalability and applicability of the solution for large-scale WSNs. To overcome those limitations, we now introduce a distributed approach to perform the sleep interval control at each individual sensor node's side.

4.1. Distributed Sleep Interval Control Problem

We decompose the original SIC problem for each sensor node and focus on a local structure of an arbitrary sensor node i in the network. As depicted in Figure 6, node

[2] At the current stage, we focus on the case, in which sensor node i has one receiver only. Such a scenario is common in practice and it can be found when packets are transmitted following a tree structure, for example, CTP [Gnawali et al. 2009]. However, our proposal is not limited to the tree structure, and we will discuss the multireceiver case in Section 4.3.

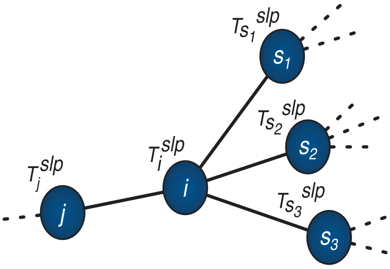

Fig. 6. Local structure for sensor i.

j is the receiver of sensor i and node s_k, $k = 1, 2, \ldots, K$, is a sender of sensor i, where K is the total number of potential senders. By exchanging information with those neighboring nodes, sensor node i can determine its local-optimal sleep interval based on the formulation from Eq. (7) to Eq. (10). As T_i^{slp} affects energy consumption rates of both node i and its senders in Eqs. (8) and (9), the variable R_i bounds the energy drain in the local region from above to control the energy trade-off between i and each sender s_k. Similar to the original SIC problem, R_i in the objective function (7) is minimized to obtain the local min-max energy fairness. We denote Eqs. (7) to (10) as the Distributed SIC (D-SIC) problem. The following lemma reveals the essence of both SIC and D-SIC problems.

$$\min \quad R_i, \tag{7}$$

$$\text{such that} \quad \lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i \leq R_i, (i, j) \in E \tag{8}$$

$$\lambda_k \cdot T_i^{slp} + \frac{\gamma_k}{T_k^{slp}} + \zeta_k \cdot T_k^{slp} + \tau_k \leq R_i, (k, i) \in E \tag{9}$$

$$0 < T_{slp}^i, i \in V. \tag{10}$$

LEMMA 4.1. *The SIC problem and the D-SIC problem are both convex optimization problems.*

Conclusions made by Lemma 4.1 are clear as all constraints and objective functions in both SIC and D-SIC problems are convex. In the D-SIC problem, the total amount of constraints is bounded by the number of senders of sensor node i. According to Liu et al. [2011a], each sensor node only needs to solve one local D-SIC problem with a small number of constraints (e.g., <8). As a result, a variety of mature and lightweight techniques can be adopted in practice, such as the interior-point method [Boyd 2004], in which the optimal result can be found within guaranteed iterations. Even D-SIC problems can be solved locally, there remains one critical issue not answered yet: how to ensure that such distributed computations eventually lead to the global optimal result? The answer will be given when we introduce the Distributed SIC (DSIC) algorithm in the next section.

Before we proceed, we particularly investigate the D-SIC problem for a set of asynchronous protocols based on LPL with the strobed preamble technique, including X-MAC, C-MAC, and so on, because its previously mentioned significance in practice. Due to the special properties, sensor nodes with this type of protocols can avoid using iterative algorithms to solve their own D-SIC problems; instead, close-form expressions can be obtained to further simplify the system design. Eqs. (8) and (9) with the strobed preamble technique in the D-SIC problem can be replaced by Eqs. (11) and

(12), respectively.

$$\lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \tau_i \leq R_i, (i, j) \in E, \tag{11}$$

$$\lambda_k \cdot T_i^{slp} + \frac{\gamma_k}{T_k^{slp}} + \tau_k \leq R_i.(k, i) \in E. \tag{12}$$

As γ_i and T_i^{slp} are both positive, Eq. (11) implies $R_i > \lambda_i \cdot T_j^{slp} + \tau_i$, which further yields: $T_i^{slp} \geq \gamma_i/(R_i - \lambda_i \cdot T_j^{slp} - \tau_i)$. On the other hand, since $\lambda_k > 0$, based on Eq. (12), we can further obtain $T_i^{slp} \leq (R_i - \gamma_k/T_k^{slp} - \tau_k)/\lambda_k$. Then, for each sender k, we have:

$$\frac{\gamma_i}{R_i - \lambda_i \cdot T_j^{slp} - \tau_i} \leq T_i^{slp} \leq \frac{R_i - \gamma_k/T_k^{slp} - \tau_k}{\lambda_k},$$

$$\Rightarrow\ R_i^2 - \phi_i^{j,k} \cdot R_i + \omega_i^{j,k} \geq 0, \tag{13}$$

$$\Rightarrow\ \overline{R_i} = \max_k \left\{ \left(\phi_i^{j,k} + \sqrt{(\phi_i^{j,k})^2 - 4\omega_i^{j,k}} \right) \Big/ 2 \right\}, \tag{14}$$

where $\overline{R_i}$ indicates the selected upper bound for energy consumption rates in the local region, $\phi_i^{j,k} \triangleq \lambda_i \cdot T_j^{slp} + \tau_i + \gamma_k/T_k^{slp} + \tau_k$, and $\omega_i^{j,k} \triangleq (\lambda_i \cdot T_j^{slp} + \tau_i)(\gamma_k/T_k^{slp} + \tau_k) - \gamma_i \cdot \lambda_k$. Note that $(\phi_i^{j,k})^2 - 4\omega_i^{j,k}$ can be transformed to $(\lambda_i \cdot T_j^{slp} + \tau_i - \gamma_k/T_k^{slp} + \tau_k)^2 + 4\gamma_i \cdot \lambda_k > 0$. As a result, roots of R_i in Eq. (13) always exist. Then, we have

$$\overline{T}_i^{slp} = \left(\overline{R_i} - \gamma_{k'}/T_{k'}^{slp} - \tau_{k'}\right)/\lambda_{k'}, \tag{15}$$

where $k' = \arg\max_k \{(\phi_i^{j,k} + \sqrt{(\phi_i^{j,k})^2 - 4\omega_i^{j,k}})/2\}$.

4.2. The DSIC Algorithm Design

To deal with the traffic load variance, at any sensor node i, SIC is performed on a periodical basis or triggered when traffic dynamics are detected. Before the algorithm execution, sensor node i collects necessary information from neighbor nodes, which includes the current sleep interval T_j^{slp} of its receiver j, λ_k, γ_k, ζ_k, and τ_k of each sender k. Such information is used to update R_i and T_i^{slp} by locally solving the D-SIC problem from Eqs. (7) to (10), or Eqs. (11) to (15) if the strobed preamble technique is adopted. To reduce the communication cost, these parameters can be obtained from regular information exchanges of some underlying services, like link estimations or CTP beacons. One point worth noting is that if there are dynamics in the network traffic, the energy consumption rate of some sensor node (e.g., node i) may suddenly become greater than R_i. In this case, R_i will not be a valid upper bound and the DSIC algorithm will not perform correctly. To cope with such an issue, we propose a remedial solution as follows. After node i detects its current energy consumption rate becoming greater than R_i, it needs to increase R_i such that R_i can still bound its energy consumption rate from above. In our implementation, R_i will be reset slightly greater than its current energy consumption rate. However, since R_i should bound energy consumption rates within the local region of node i from above, the increasing of R_i solely according to node i's energy consumption rate cannot guarantee that R_i is greater than the energy consumption rates of its neighbors. Therefore, we further rely on the exchanging of R_i to solve such an issue. Once receiving R_k from each neighbor k, node i updates R_i to be $\max\{R_i, \max_k\{R_k\}\}$. After setting an appropriate R_i, the DSIC algorithm can correctly perform operations as if the traffic loads were stable.

ALGORITHM 1: The DSIC Algorithm at Sensor Node i

```
   Input  : Current $R_i$ and sleep interval $T_i^{slp}$.
   Output: Updated $R_i$ and $T_i^{slp}$, denoted as $\overline{R}_i$ and $\overline{T}_i^{slp}$.
1  Collect $T_j^{slp}$ and $R_j$, where $j$ is the receiver of sensor node $i$.
2  Collect $\lambda_k$, $\gamma_k$, $\zeta_k$, $\tau_k$, and $R_k$ from each sender $k$.
3  Locally solve the D-SIC problem from Eqs. (7) to (10) and obtain the updated $\overline{R}_i$ and $\overline{T}_i^{slp}$.
4  if $\overline{R}_i < R_i$ then
5      Set $R_i$ to be $\overline{R}_i$;
6      Update the sleep interval $T_i^{slp}$ to $\overline{T}_i^{slp}$;
7      Inform the updated $T_i^{slp}$ to its senders;
   else
8      Keep both $R_i$ and $T_i^{slp}$ unchanged;
   end
```

After $\overline{R}_i$ and $\overline{T}_i^{slp}$ are updated, sensor node i first checks whether the new $\overline{R}_i$ is smaller than the current R_i. If so, i adjusts its sleep interval to be $\overline{T}_i^{slp}$. In addition, R_i will be replaced by $\overline{R}_i$ for the next updating. Otherwise, i takes no action. The detailed description of the DSIC algorithm is given in Algorithm 1. When $\overline{R}_i < R_i$, the adjustment of sleep interval will decrease the maximum energy consumption rate in the local region of node i. By iteratively executing the algorithm by all the sensor nodes, eventually, no sensor nodes can further decrease their energy consumption rates without compromising the maximum energy consumption rate achieved within its local region. In other words, their energy consumption rates in this process are converging towards a common value, and such a common value keeps decreasing. Moreover, by exchanging R_i, different sensor nodes will adjust their own R_i to the maximum one within its neighborhood. As a matter of fact, the exchanging of R_i will cause the largest upper bound to eventually spread to the entire network. Thus, each local optimization process is finally constrained by the maximum upper bound in the network, and the updating operations in DSIC essentially adjust the sleep interval based on the reduction of this maximum upper bound stored locally. In other words, the DSIC solves the global optimization problem in a distributed manner with a local exchange of R_i. After the performance of DSIC converges, the energy consumption rate of no sensor node can be further reduced. It implies that the optimal result has been achieved. From a theoretical perspective, a rigorous interpretation to the correctness of our algorithm is given in the following theorem.

THEOREM 4.2. *By the execution of Algorithm* 1 *at each sensor node, the maximum energy consumption rate in the network approaches to be minimized.*

PROOF. Line 4 in Algorithm 1 indicates that whenever the sleep interval is updated, R_i becomes smaller, which results in the decrease of the maximum energy consumption rate in each local region. On the other hand, the original SIC problem and the D-SIC problem are both convex, and each D-SIC is a subproblem of SIC. To finish the proof, we assume that the maximum energy consumption rate in the network converges to $\mathscr{R}$, which is different from the optimal result $\mathscr{R}^*$. Clearly, $\mathscr{R} > \mathscr{R}^*$. Now, we prove this theorem by contradiction. If the maximum energy consumption rate converges to $\mathscr{R}$ by our algorithm, it indicates that there does not exist any R_i to further reduce the current maximum rate of the energy drain in the network, implying $\mathscr{R}$ be a local minimum point. However, the original problem is convex, such a conclusion yields that $\mathscr{R}$ must be a global optimal point as well [Boyd 2004], which is a contradiction. □

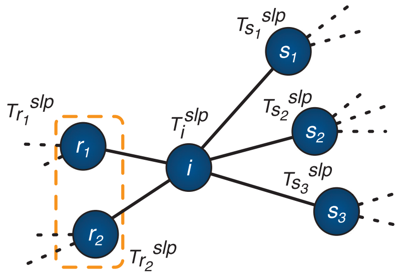

Fig. 7. Multireceiver scenario.

4.3. Discussions

4.3.1. Multireceiver scenario. So far, we have focused on the case, in which each sensor node i has only one receiver. Such a case corresponds to the packet transmission following a tree-based routing structure. As aforementioned, packets, however, can be transmitted following a DAG as well, in which there may exist more than one potential receiver. Without loss of generality, we assume sensor node i has n_i potential receivers.

We can slightly alter our previous analysis and reach a General Distributed SIC (GDSIC) algorithm, which can support multiple receivers in general. GDSIC can be simply extended from the DSIC algorithm, and the basic principle is as follows. Since preambles sent from sensor node i must cover the sleep interval of each potential receiver r_j for $1 \leq j \leq n_i$, the length of i's preamble can be determined by $\max_j\{T_{r_j}^{slp}\}$ (Figure 7). Thus, the multireceiver case is accordingly transformed to an equivalent single receiver case as shown by Figure 8, in which the sleep interval of the single virtual receiver equals to $\max_j\{T_{r_j}^{slp}\}$. We can then modify T_j^{slp} as $\max_j\{T_{r_j}^{slp}\}$ in line 1 of Algorithm 1 and apply the DSIC algorithm for the sleep interval control. Due to the page limit, the detailed algorithm is given in Algorithm 2.

ALGORITHM 2: The GDSIC Algorithm at Sensor Node i

```
Input  : Current $R_i$ and sleep interval $T_i^{slp}$.
Output: Updated $R_i$ and $T_i^{slp}$, denoted as $\overline{R}_i$ and $\overline{T}_i^{slp}$.
Collect $T_{v_{r_j}}^{slp}$ and $R_{r_j}$ from each receiver $v_{r_i}$ of sensor node $i$;
Assign $T_j^{slp}$ in the D-SIC problem by $\max_j\{T_{v_{r_j}}^{slp}\}$;
Collect $\lambda_k$, $\gamma_k$, $\zeta_k$, $\tau_k$, and $R_k$ from each sender $k$.
Locally solve the D-SIC problem from Eqs. (7) to (10) and obtain the updated $\overline{R}_i$ and $\overline{T}_i^{slp}$.
if $\overline{R}_i < R_i$ then
    Set $R_i$ to be $\overline{R}_i$;
    Update the sleep interval $T_i^{slp}$ to $\overline{T}_i^{slp}$;
    Inform the updated $T_i^{slp}$ to its senders;
else
    Keep both $R_i$ and $T_i^{slp}$ unchanged;
end
```

4.3.2. Considering Residual Energy Budgets. In previous discussions, we focus on minimizing the maximum energy consumption rate in the network. However, after sensor nodes are deployed and have worked for a period, the residual energy budget at each

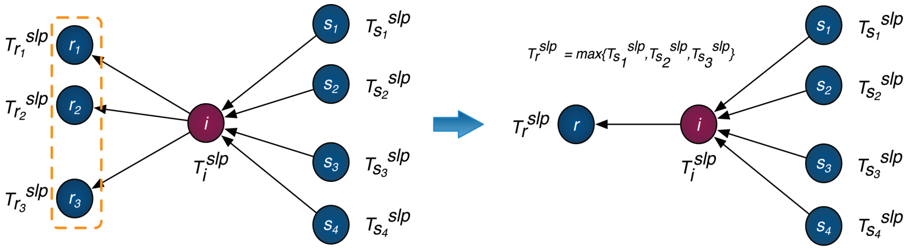

Fig. 8. Illustration of transformation from the multi-receiver scenario to the single-receiver scenario.

node might have already become uneven. Hereby, we show that the original SIC problem formulation can be transformed to consider sensor nodes' residual energy budgets. Denote the residual energy budget of sensor node i as e_i. As a result, the lifetime of node i remains e_i/r_i. With the consideration of the residual energy, we naturally hope to maximize the minimum lifetime of sensor nodes in the network. Mathematically, such a design target can be expressed as follows:

$$\max \quad \min_i\{E_i\} \tag{16}$$

$$s.t. \quad e_i/\left(\lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i\right) \geq E_i, \tag{17}$$

$$0 < T_i^{slp}, 0 < E_i, i \in V. \tag{18}$$

Since Eq. (17) can be rephrased as $\lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i) \leq e_i/E_i$, this formulation is essentially equivalent to the original SIC problem and the detailed proof is as follows. We define R_i to be $1/E_i$. According to Eq. (17), we have:

$$\begin{aligned} & e_i/\left(\lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i\right) \geq E_i, \\ \Rightarrow & \left(\lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \zeta_i \cdot T_i^{slp} + \tau_i\right)\Big/ e_i \leq 1/E_i, \\ \Rightarrow & \lambda_i' \cdot T_j^{slp} + \frac{\gamma_i'}{T_i^{slp}} + \zeta_i' \cdot T_i^{slp} + \tau_i' \leq R_i, \end{aligned} \tag{19}$$

where λ_i', γ_i', ζ_i', and τ_i' equal to λ_i/e_i, γ_i/e_i, ζ_i/e_i, and τ_i/e_i, respectively. Since E_i is positive (indicated by Eq. (18)), R_i in Eq. (19) is positive as well. As a result, the constraints of both the original SIC formulation from Eq. (4) to Eq. (6) and the formulation with the consideration of the residual energy budgets from Eq. (16) to Eq. (18) are shown to be equivalent. Now, we further prove the equivalence of their objective functions.

$$\begin{aligned} \max \quad \min_i\{E_i\} & \Leftrightarrow \min \quad \max_i\{1/E_i\}, \\ & \Leftrightarrow \min \quad \max_i\{R_i\}. \end{aligned}$$

So far, we finish the proof the equivalence of these two problem formulations. As a result, the proposed GDSIC algorithm can be easily extended to consider the residual energy budgets of sensor nodes.

4.3.3. Extension to Receiver-Initiated Protocols. As a most representative technique, Low Power Probing (LPP) has been employed in many receiver-initiated protocols. The

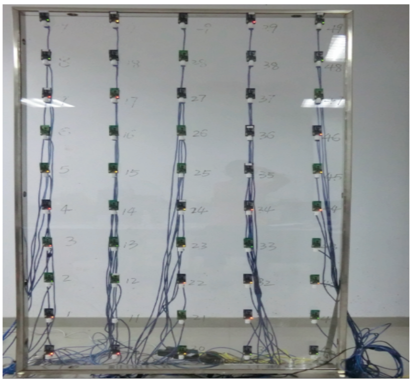

Fig. 9. 10 × 5 grid testbed.

energy efficiencies of LPL and LPP are mainly different from the following two aspects. First, the energy consumption to receive the preamble at the receiver side can be omitted in LPP. Second, the energy consumed for overhearing in LPL should be replaced by obtaining the receiver's predicted wake-up schedule in LPP. After rephrasing the energy consumption rate for each sensor node with LPP based on these two differences, our previous analysis and solution can be seamlessly extended to the receiver-initiated protocols.

4.3.4. SIC for Leaf Nodes. Since leaf nodes in the network have no packet senders, they may fail to obtain an effective sleep interval adjustment based on the GDSIC algorithm. To deal with such a marginal case, in our implementation, those sensor nodes adjust their sleep intervals such that their energy consumption rates are approximately equal to their receivers.

5. EXPERIMENTAL EVALUATION

In previous sections, we elaborate the design principles and important properties of the proposed GDSIC algorithm. In this section, we validate its feasibility and applicability in practice.

5.1. Experiment Setting

We implement GDSIC on TelosB motes and use a 50-node testbed to examine its performance. 50 nodes are organized as a 10 × 5 grid.[3] (See Figure 9.) Due to the experimental space limitation, the power of each TelosB mote is set to be the minimum level and the communication range is about 10 centimeters. Starting from the left-top conner, sensors are placed following the bottom-to-top and left-to-right order based on their IDs. GDSIC is implemented at the application layer, which utilizes two major standard components, LPL and CTP, adopted in the current TinyOS 2.1 package. On the MAC layer, the default protocol, X-MAC, is adopted in the experiment. In the initial five minutes, sensor nodes beacon neighboring nodes to form a stable routing tree rooted at sensor node 0. To increase the depth of the formed routing structure, we manually enforce that the receiver of a sensor node is selected from its adjacent neighbors on the testbed. For example, the parent of node 15 is chosen from nodes 16, 14, 25, 5, 24, 6, 26, and 4. After the initialization phase, sensor nodes inject packets to the network and cooperatively deliver packets to the sink (root) node. The average

[3] Due to the hardware failure, node number 11, in Figure 9, is excluded and only 49 sensor nodes are finally used in the experiment.

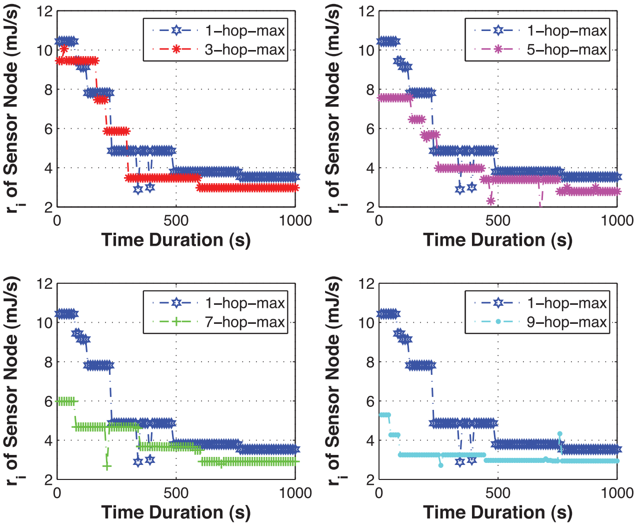

Fig. 10. Energy consumption rate vs. duration.

traffic generation rate is one packet every four seconds and the GDSIC algorithm is triggered every 60 seconds.

5.2. Experimental Results

5.2.1. Energy Consumption Rate vs. Duration Time. The experiment lasts 40 minutes on the testbed. Based on the collected data, we observe that after GDSIC executes for 20 minutes, the system performance becomes relatively stable. For a clear presentation, we mainly demonstrate the transition state of the network *after* the initial phase. In Figure 10, we illustrate energy consumption rates of five representative sensor nodes with hop counts 1, 3, 5, 7, and 9, respectively. Each selected sensor node in Figure 10 experiences approximately the fastest energy draining speed compared with other peering nodes with the same hop count. Figure 10 shows that after 800 seconds, energy consumption rates of those sensor nodes converge to around 3.6 mJ/s, and there is no obvious performance variance afterward. At time 1000 seconds, we take a snapshot of the network and conduct an offline computation. The optimal min-max energy fairness is obtained to be 3.2 mJ/s. The important insights obtained from Figure 10 are two-folds: first, energy consumption rates of sensor nodes in different network positions are well balanced after the network becomes stable, which effectively eliminates the hot spots of the energy consumption within the network. Second, GDSIC has a good convergency speed. In particular, after the initial five minutes, the overall energy consumption rates are decreased to be fairly low within the first 500 seconds. After several extra iterations, the performance converges eventually. According to Figure 10, we find that the stabilized energy consumption rates of sensor nodes near the sink node are still slightly greater than other distant sensor nodes in GDSIC and such a performance gap is difficult to be further closed but remains to be small. Compared with the equal sleep interval setting policy, the min-max energy fairness has been notably improved by GDSIC.

5.2.2. Snapshot of Energy Consumption Rates. In Figure 11, we illustrate the snapshot of the energy consumption rate of each sensor node in GDSIC and compare them with the traditional identical sleep interval setting strategy (EQUAL). According to Figure 11, we can observe that most sensor nodes in GDSIC achieve similar energy consumption rates after executing the GDSIC algorithm, and only a small number of sensor nodes

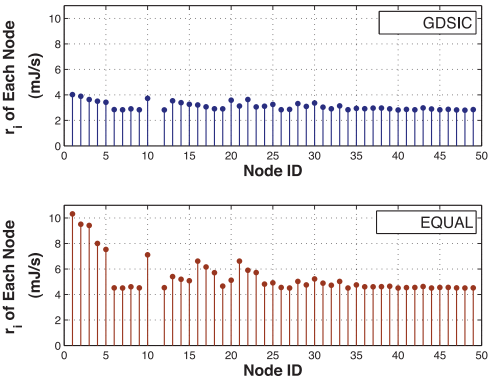

Fig. 11. Snapshot of energy consumption rates.

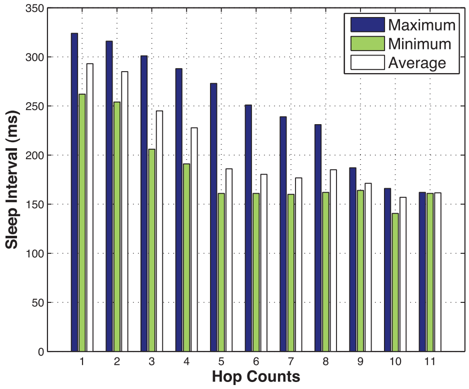

Fig. 12. Snapshot of sleep intervals.

close to the sink (with heavier traffics) suffer slightly higher energy consumption rates. However, compared with EQUAL, the min-max energy fairness has been improved by GDSIC up to 64.1%, and the obtained energy fairness is close to the optimal result. In addition, the average energy energy consumption rate in GDSIC also outperforms EQUAL by 37.2%.

5.2.3. Snapshot of Sleep Intervals. In Figure 12, we depict statistics of sleep intervals of the sensor nodes in GDSIC according to their hop counts. The statistics are obtained after the network becomes stable. In this experiment, sensor nodes are configured with the default sleep interval, that is, 512 ms, initially. Figure 12 indicates that all sensor nodes should increase their sleep intervals so that the obtained min-max energy fairness can break the barrier formed by the sleep intervals selected initially, which is different from the intuitive suggestions of the greedy strategy. On the other hand, from Figure 12, we can observe that the trend, that is, the sleep interval should be set large if the sensor node is close to the sink node carrying higher traffic loads, holds after the network becomes stable. However, such a trend only reflects a statistic result. If we focus on each individual node pair, such a trend does not always exist. Such results validate the hardness of the SIC problem, where heuristics can be hardly borrowed to trivially achieve the optimal sleep interval control.

6. TRACE-DRIVEN SIMULATION EVALUATION

We conduct comprehensive and large-scale simulations to further examine the efficiency and scalability of GDSIC. We evaluate the system performance of GDSIC in comparison with the optimal policy (OPT), the greedy strategy (GREEDY), and the equal sleep interval strategy (EQUAL). In GREEDY, sensor nodes adjust sleep intervals such that their energy consumption rates are set as the average value of their neighbors. To test a realistic network setting, simulations are conducted with a real trace harvested from GreenOrbs [Liu et al. 2011a]. GreenOrbs is a long-term and large-scale wireless sensor network deployed in the forest, which contains 433 nodes and has continuously worked over one year. From the harvested trace over 6 months, we observe that the dynamics of wireless links result in fluctuating of the network topology. To mimic the link estimation for real data transmissions, we filter out lossy links with small RSSI values. In particular, links with the packet reception ratio lower than 30% or RSSI smaller than −80 dB are excluded by the filter. By doing so, we obtain a stable network topology for simulations in Figure 13. The topology includes 6567 links with relatively good qualities.

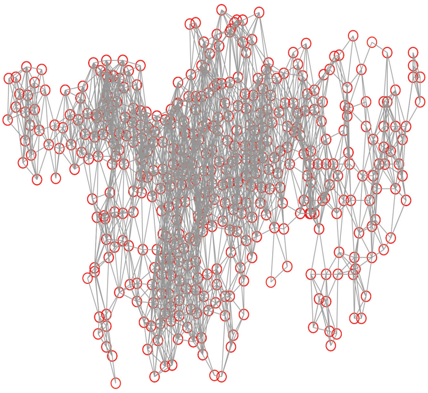

Fig. 13. GreenOrbs topology.

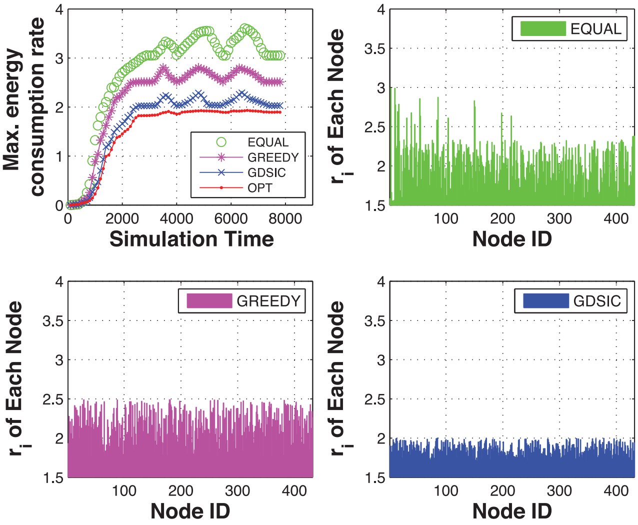

Fig. 14. Maximum energy consumption rates

6.1. Experimental Setting

In the trace, sensor nodes are deployed in a 700m × 200m rectangle field with the default transmission power. Parameters of sensor nodes are set based on the TelosB mote specification [TelosB 2004]. To collect network-wide data, both DAG and data collection tree are investigated in the simulation. Packet retransmissions due to the link loss are considered in the simulation. The sink node is placed at $(-200.2, 115.7)$ and the default traffic generation rate is one packet every ten seconds. To mimic the traffic dynamics in real applications, we manually trigger the traffic variance and investigate the impact of the traffic dynamics. The default MAC-layer protocol is X-MAC, while we also study the GDSIC strategy over a variety of other asynchronous protocols, adopting the original LPL, strobed preamble and predictive wake-up techniques.

6.1.1. Maximum Energy Consumption Rates. In Figure 14, we first investigate the achieved maximum energy consumption rates (min-max energy fairness) with different approaches on top of DAG. We simulate an 8000-second data collection process. During three time intervals of $[3500, 4000]$, $[4500, 5000]$, and $[6000, 6500]$, we double traffic generate rates of sensor nodes in four random regions and each region roughly contains 10% of total sensor nodes. In Figure 14 (up-left), we observe that EQUAL incurs much larger min-max energy fairness than both GREEDY and GDSIC all the time. When traffic varies, its min-max fairness fluctuates significantly. As unveiled by Theorem 3.2, such a fluctuation is mainly caused by the traffic dynamics since sleep intervals are set to be identical. Compared with EQUAL, GREEDY improves the achieved energy fairness by 16.7% on average and the fluctuation of GREEDY is smoother. However, we can find that there is still a clear gap between GREEDY and OPT, which needs to be closed. In Figure 14, GDSIC effectively closes such a gap and outperforms EQUAL and GREEDY by up to 32% and 21%, respectively.

To further investigate energy consumption rates of sensor nodes in different systems, we take a snapshot at time 4000 and show the instant energy consumption rate of each sensor node in Figure 14 (up-left). Similar to Section 5, in EQUAL, the energy consumption rate is not well balanced. The energy drains of certain sensor nodes are much faster than other sensors, which will potentially limit the lifetime of the network. The difference between the maximum and minimum energy consumption rates in EQUAL is up to 48.9%. Different from EQUAL, both GREEDY and GDSIC result in a well-balanced energy consumption rate over the entire network. Statistics show that differences between the maximum and minimum rates to consume energy in GREEDY and GDSIC are only 29.5% and 14.9%, respectively. However, we notice that the average rate of the energy drain in GREEDY is still high, and there is a nonignorable gap between GREEDY and GDSIC. Figure 14 provides a good indication that GDSIC has achieved the best performance in terms of both the min-max energy fairness and average energy consumption among three approaches.

6.1.2. CDF of Energy Consumption Rates. According to Figure 14, we further illustrate the CDF of energy consumption rates for different strategies in Figure 15. As expected, the energy drain of GDSIC distributes within a narrow interval. In addition, the average value of GDSIC is also the smallest one compared with other two strategies. On the contrary, the energy consumption rates of sensor nodes in EQUAL spread over a wide region from 1.42 to 2.99 mJ/s and certain sensor nodes suffer relatively high speeds of the energy consumption, which will limit the lifetime of the network. Another important piece of information delivered by Figure 15 is that, although the distribution of GREEDY is within a narrow region as well, its average value is greater than EQUAL. As the initial rate differences in the network might be large, when the sensor nodes with high energy consumption rates greedily adjust their own sleep intervals, they unavoidably increase the energy consumption rates of their children. Such a greedy strategy could lead to a suboptimal result that is far away from the optimal result.

6.1.3. Network Yield. Network yield in Figure 16 is defined as the percentage of sensor nodes which are reachable from sink node. Figure 16 shows that even the average energy consumption rate of GREEDY is greater than that of EQUAL, GREEDY still has a larger network yield than EQUAL all the time, since there is no obvious energy bottleneck in GREEDY. Figure 16 provides a good indication to the importance of minimizing the maximum energy consumption rate in the network. Different from EQUAL and GREEDY, GDSIC performs closer to OPT all the time and achieves an excellent min-max energy fairness performance. From statistics, network yield of GDSIC, on average, is greater than EQUAL and GREEDY by 50.8% and 35.9%, respectively.

6.1.4. Traffic Load Dynamics. In Figure 17, we compare the system performance of GDSIC with EQUAL, GREEDY, and OPT under different traffic generation rates in the network. In each setting of the traffic generation rate, EQUAL always suffers from the largest energy consumption rate than other schemes. Compared with OPT, its

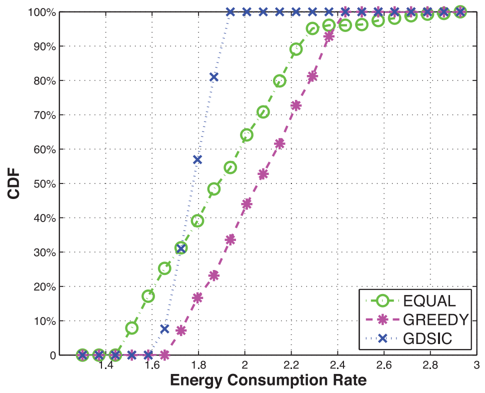

Fig. 15. CDF of energy consumption rates.

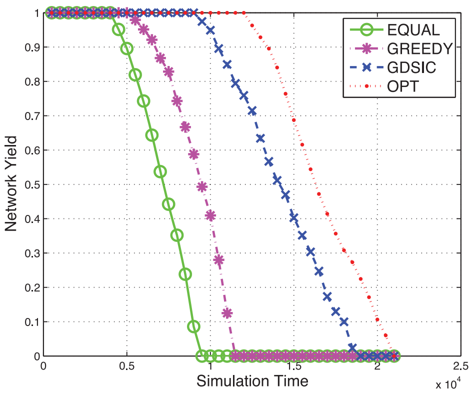

Fig. 16. Network yield.

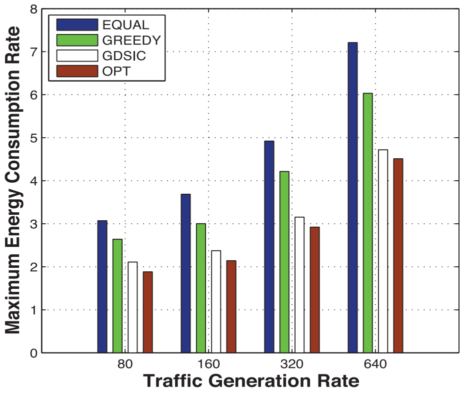

Fig. 17. Maximum energy consumption rates under different traffic generation rates.

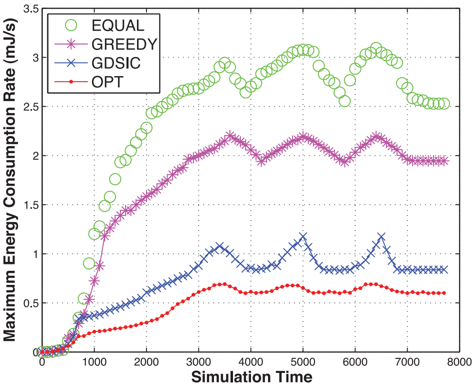

Fig. 18. Maximum energy consumption rates in the data collection tree.

performance distortion reaches as high as 58%. Although GREEDY can improve the energy fairness achieved by EQUAL in all cases, there are still clear performance gaps between GREEDY and OPT. From the results, we find that the performance distortion of GREEDY is up to 33%. In Figure 17, GDSIC effectively narrows such a gap down to only 6% and outperforms EQUAL and GREEDY by 35% and 22%, respectively. Figure 17 indicates that GDISC can achieve a good performance under different traffic loads in the network.

6.1.5. Energy Consumption Rate in a Data Collection Tree. From Figures 14 to 16, all three strategies are operated over DAG. From Figures 18 to 20, we further demonstrate the system performance in a data collection tree. Compared with Figure 14, we plot the variances of the min-max energy fairness of all three strategies over time and observe similar system behaviors. Figure 18 shows that GDSIC can achieve the best performance among these three strategies and approach to the optimal result as well. However, as the throughput of the data collection tree is usually smaller than that of DAG, the energy consumed to collect data becomes smaller accordingly. Figures 14 and 18 jointly demonstrate that GDSIC can achieve a good performance in both DAG and tree, which are two major routing structures in current WSNs.

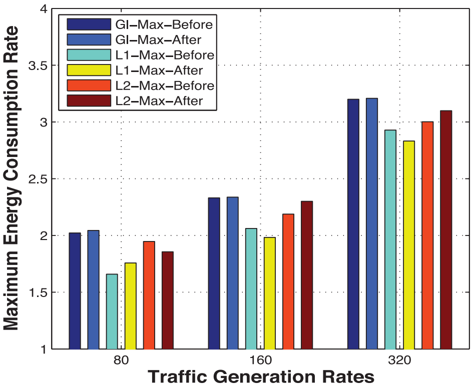

Fig. 19. Maximum energy consumption rates with routing dynamics.

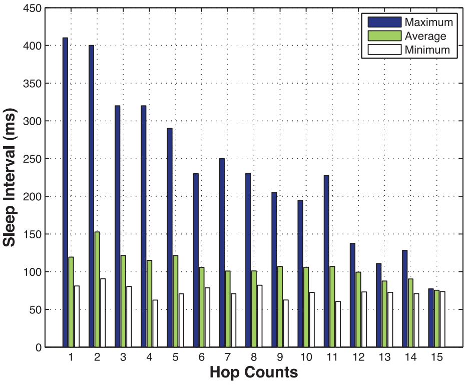

Fig. 20. Snapshot of sleep intervals under different hop counts.

6.1.6. Energy Consumption Rates with Routing Dynamics. In Figure 19, we evaluate the system performance under three different traffic generation rates in the network. For each configuration of the traffic generation rate, after sensor nodes have achieved the min-max energy fairness with GDSIC algorithm, we artificially change the underlying routing structure to examine the system performance against the route change. In particular, routing structures in two randomly selected regions are changed and each region contains around 15% of total sensor nodes. In each local region, sensor nodes are enforced to re-pick different parent nodes on the routing tree. In addition, we require that one selected region is close to the sink node and another one is relatively far away from the sink node. After the route change, sensor nodes further execute the GDSIC algorithm to achieve the min-max energy fairness again. In Figure 19, we plot the maximum energy consumption rates of the entire network and each local region before and after the route change. For instance, "L1-Max-After" in the legend of the figure indicates the maximum energy consumption rate of the first local region after the route change. From Figure 19, we can see that the maximum energy consumption rate of each local region may change due to the route change. Nevertheless, the change is usually slight. It is because the traffic burden does not change in each local region and the GDSIC algorithm can adjust each node to approach to the min-max fairness again after the route change. Moreover, although two local regions are selected from different positions of the network, their energy consumption rates can still be adapted to be close to the maximum energy consumption rate in the network. Therefore, the proposed algorithm can achieve good energy fairness even with the route changes in the network.

6.1.7. Distribution of Sleep Intervals in Data Collection Tree. In Figure 20, the x-axis indicates the hop counts of sensor nodes and the y-axis represents statistic results for sleep intervals of sensor nodes with the same hop count. A traditional belief states that the energy bottleneck in the data collection is always around the sink node. However, according to the statistics, not all the sensor nodes close to the sink node will adapt to longer sleep intervals, and the sensor nodes far away from the sink node do not necessarily set shorter sleep intervals either. Although on average the trend of sleep intervals of sensor nodes decreases as the hop count increases, such a trend is a statistic result only. If we focus on each individual sensor node, we find that it is nontrivial to determine an appropriate sleep interval, especially in a distributed manner. In addition, the SIC problem becomes even complicated in practical WSNs, since not all heavy load regions are located near the sink node.

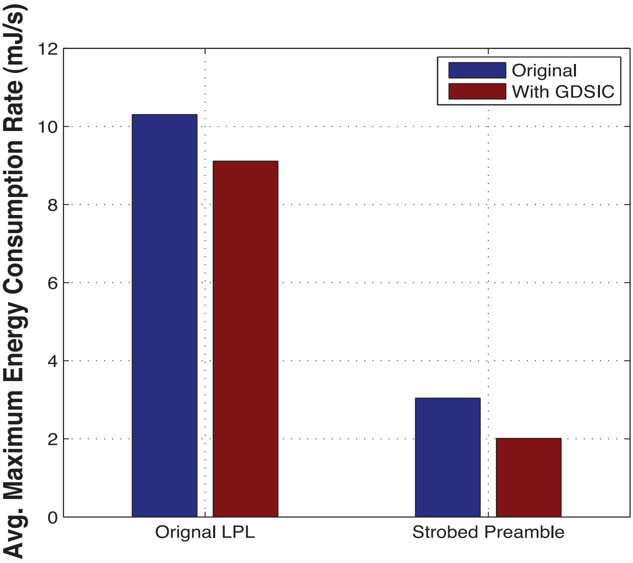

Fig. 21. Avg. maximum energy consumption rate vs. Different protocols.

6.1.8. GDSIC over Different MAC Protocols. In this set of simulations, we further extend GDSIC to other asynchronous MAC-layer protocols, adopting the original LPL technique. Figure 21 demonstrates that compared with the original LPL technique, the strobed preamble techniques improve the energy efficiency of the network. In addition, after the GDSIC is applied to each type of protocols, the achievable min-max energy fairness has been improved by 11.6% and 33.8%, respectively. As a unified framework, on average, GDSIC has gained more than 20% performance improvement, and it is highly beneficial for asynchronous protocols to adopt GDSIC for improving the energy fairness and prolonging the the network lifetime.

7. CONCLUSION AND FUTURE WORK

In this article, we investigate the problem of achieving the min-max energy fairness in asynchronous duty-cycling sensor networks. We aim at optimal sleep interval control for sensor nodes so as to achieve min-max energy fairness. We propose a mathematical model to describe energy efficiency of such networks and observe that traditional sleep interval setting and control strategies hardly perform well in practice. Towards developing an efficient control strategy, we formulate the SIC problem as a convex optimization problem. By utilizing the convex property, we decompose the original problem, which yields to a distributed algorithm GDSIC. In GDSIC, the network-wide min-max energy fairness can be achieved in a distributed fashion. The proposed solution serves as a unified framework applicable to a variety of underlying asynchronous protocols. One possible future work of this article is to implement and evaluate the performance GDSIC in large-scale sensor network deployments.

APPENDIXES

In these Appendixes, we provide complete proofs that are omitted in Section 3.

A. PROOF OF THEOREM 3.1

Before one sensor node i transmits a packet to another node (e.g., j), i should send a preamble covering sensor j's sleep interval first, then followed by the intended packet as well as the *ACK* message. Thus,

$$r_i^{tx} = f_i^{tx} \times \left(T_j^{slp} + L_{pck} \times t_{txb} + T_{ack}^{tx}\right) \times e_{tx}, \tag{20}$$

where f_i^{tx} is the outgoing (transmitting) traffic rate, L_{pck} indicates the packet length, L_{ack} represents the *ACK* message length, t_{txb} is the time to transmit one digital bit, T_{ack}^{tx} is the time to send out an *ACK* message, and e_{tx} refers to the *electric power* in transmission.

When receiving one packet, sensor i needs to wait until the end of the preamble before the receiving of the intended packet and the *ACK* message. Since sensor i may wake up to poll the wireless channel starting from any portion of the preamble, on average, we have:

$$r_i^{rc} = f_i^{rc} \times \left(T_i^{slp}/2 + L_{pck} \times t_{rcb} + T_{ack}^{rc}\right) \times e_{rc}, \tag{21}$$

where f_i^{rc} is the incoming (receiving) traffic rate, t_{rcb} is the time to receive one digital bit, T_{ack}^{rc} is the time to receive an *ACK* message, and e_{re} indicates the *electric power* in receiving.

The energy consumption in overhearing can be described based on r_i^{rc}. During overhearing, sensor i needs to wait until the end of the preamble sent from the sender as well. However, different from packet receiving, at the end of the preamble, sensor i becomes aware of the occurrence of overhearing and switches to sleep immediately. As a result, we have:

$$r_i^{oh} = f_i^{oh} \times \left(T_i^{slp}/2\right) \times e_{rc}, \tag{22}$$

where f_i^{oh} is the overhearing traffic rate.

Based on Buettner et al. [2006], Challen et al. [2010], and Langendoen and Meier [2010], the energy consumption for regular channel polling can be expressed as:

$$r_i^{cp} = \left(B - f_i^{tx} - f_i^{rc} - f_i^{oh}\right) \times \left(T_{poll}/T_i^{slp}\right) \times e_{poll}, \tag{23}$$

where B is the maximum bandwidth (e.g., 256 kbps in the Zigbee protocol) and T_{poll} is the time to poll the channel and e_{poll} is the *electric power* in idle listening. According to Eqs. (20) to (22), we can verify that Eq. (2) holds after some mathematical computation. As mentioned in Section 3, the original LPL has been further optimized due to the low energy efficiency at the receiver side, and the most representative example is the *strobed preamble* technique. Now, we discuss how does the energy consumption rate of each sensor node that applies such a technique can be unified by Theorem 3.1 as follows.

Before the intended packet transmission, sensor node i sends out a series of short preambles separated by short pauses instead of a long preamble for the energy saving purpose. During one sleep interval of receiver j, a strobed preamble may contain at most $\lfloor T_j^{slp}/(L_{spre} \times t_{txb} + T_{ack}^{tx})\rfloor$ complete *short preamble* and *short pause* pairs, where L_{spre} indicates the length of the short preamble. Similarly as before, when the intended receiver (e.g., j) wakes up and receives a short preamble, it will reply node i by an early *ACK* message. Upon receiving this *ACK* message, the sender i stops sending preambles and sends the data packet. Therefore, we have:

$$r_i^{tx} = f_i^{tx} \times \left[T_j^{spre} + L_{pck} \times t_{txb} + T_{ack}^{tx}\right] \times e_{tx},$$

where

$$T_j^{spre} \triangleq \left\lfloor \frac{T_j^{slp}}{L_{spre} \times t_{txb} + T_{ack}^{tx}} \right\rfloor \times \left(\frac{L_{sp} \times t_{txb} + T_{ack}^{tx}}{2}\right),$$

and

$$r_i^{rc} = f_i^{rc} \times \left[\frac{3}{2}(L_{spre} + T_{ack}) + L_{pck} \times t_{rcb} + T_{ack}^{rc}\right] \times e_{rc}.$$

Parallel to Eqs. (23) and (22), r_i^{cp} and r_i^{oh} can be similarly analyzed. Therefore, r_i in this category can be expressed as:

$$r_i = \lambda_i \cdot T_j^{slp} + \frac{\gamma_i}{T_i^{slp}} + \tau_i, \tag{24}$$

which is a special case of Eq. (2). In other words, Eq. (24) can be unified by Theorem 3.1 as well.

B. PROOF OF THEOREM 3.2

When the identical sleep interval setting policy is employed, according to Theorem 3.1, the energy consumption rate of any sensor i depends on four coefficients in Eq. (2), that is, λ_i, γ_i, ζ_i, and τ_i. Suppose errors and differences of hardware devices can be ignored among different sensor nodes, by Appendix A, each of these coefficients is mainly determined by f_i^{tx} and f_i^{rc}, where f_i^{tx} and f_i^{rc} represent the outgoing (transmitting) and incoming (receiving) traffic rates of node i, respectively. In addition, due to the fact $f_i^{rc} = f_i^{tx} - \rho$, where ρ is the traffic generation rate, coefficients thus can be further expressed by a set of functions in terms of f_i^{tx}. As previously mentioned, the network traffic in practice is normally heterogenous. Therefore, sensor nodes in heavy traffic regions are prone to suffer more frequent preamble time and longer data receiving time. As a consequence, those sensor nodes tend to run out of energy first, and traffic loads are prone to dominate the lifetime of sensor nodes when all sleep intervals are set to be equal. In Section 3.2, we have conducted a concrete case study in data collection for validating such a conclusion.

C. PROOF OF LEMMA 3.4

Now we consider an arbitrary sub-tree of the whole data collection tree rooted at the sink node. Logically, we organize all sensor nodes into levels based on their hop counts and denote the region containing all sensor nodes with the hop count l as $\mathcal{R}_l$. The average outgoing traffic density of $\mathcal{R}_l$ can be calculated by $\iint_{\mathcal{R}_l}(f^{tx}(l) \times \sigma) \cdot d(x, y)$, where $f^{tx}(l)$ is the average outgoing traffic rate in $\mathcal{R}_l$ and σ is the average node density of the network. On the other hand, the overall outgoing traffic of $\mathcal{R}_l$ originates from $\mathcal{R}_l$ itself plus all other regions with larger hop counts in the same sub-tree. In other words, sensor nodes in $\mathcal{R}_l$ generate their own traffics, meanwhile they also need to relay traffics for the distant sensor nodes from the sink. According to the traffic conservation nature, we have:

$$\iint_{\mathcal{R}_l}(f^{tx}(l) \times \sigma) \cdot d(x, y) = \sum_{j=l}^{L} \iint_{\mathcal{R}_l}(\rho \times \sigma) \cdot d(x, y), \tag{25}$$

where ρ indicates the average traffic stretch in the network and L is the maximum hop count from the network boundary to the sink node. For any sensor node i that is l-hop away from the sink node, after solving Eq. (25), we can obtain Eq. (3) in Lemma 3.4 as follows:

$$f_i^{tx}(l) = \rho(L^2 - (l-1)^2 d^2)/(2l-1)d^2.$$

In addition, based on the fact of $f_i^{tx}(l) = f_i^{rc}(l) + \rho$, we can further derive: $f_i^{rc}(l) = \rho(L^2 - l^2 d^2)/(2l-1)d^2$.

REFERENCES

S. P. Boyd. 2004. *Convex optimization*. Cambridge University Press.

M. Buettner, G. Yee, E. Anderson, and R. Han. 2006. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of ACM, Sensys*. 307–320.

G. W. Challen, J. Waterman, and M. Welsh. 2010. IDEA: Integrated Distributed Energy Awareness for wireless sensor networks. In *Proceedings of ACM Mobisys*. 35–48.

J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiramanz, and X. Shen. 2010. Utility-Based Asynchronous Flow Control Algorithm for Wireless Sensor Networks. *IEEE J. Select. Areas Commun.* 28, 7, 1116–1126.

T. Dam and K. Langendoen. 2003. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of ACM SenSys*. 171–180.

W. Du, F. Mieyeville, D. Navarro, and I. O. Connor. 2011. IDEA1: A validated SystemC-based system-level design and simulation environment for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* 2011, 1, 1–20.

P. Dutta, S. Dawson-Haggerty, Y. Chen, C. J. M. Liang, and A. Terzis. 2010. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of ACM SenSys*. 1–14.

P. Dutta, J. Taneja, J. Jeong, X. Jiang, and D. Culler. 2008. A building block approach to sensornet systems. In *Proceedings of ACM SenSys*. 267–280.

A. EI-Hoiydi and J. Decotignie. 2005. Low power downlink MAC protocols for infrastructure wireless sensor networks. *ACM Mobile Netw. Appl.* 10, 5, 675–690.

O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. 2009. Collection tree protocol. In *Proceedings of ACM SenSys*. 1–14.

Y. Gu and T. He. 2007. Data Forwarding in extremely low duty-cycle sensor networks with unreliable communication links. In *Proceedings of ACM SenSys*. 321–334.

Y. Gu, T. Zhu, and T. He. 2009. ESC: Energy synchronized communication in sustainable sensor networks. In *Proceedings of IEEE ICNP*. 52–62.

S. Guo, Y. Gu, B. Jiang, and T. He. 2009. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In *Proceedings of ACM MobiCom*. 133–144.

W. Hu, N. Bulusu, C. T. Chou, S. Jha, A. Taylor, and V. N. Tran. 2009. Design and evaluation of a hybrid sensor network for cane toad monitoring. *ACM Trans. Sens. Netw.* 5, 1, 4.

K. Langendoen and A. Meier. 2010. Analyzing MAC protocols for low data-rate applications. *ACM Trans. Sensor Netw.* 7, 2, 19.

Z. Li, M. Li, and Y. Liu. 2012. Towards energy-fairness in asynchronous duty-cycling sensor networks. In *Proceedings IEEE Infocom*. 801–809.

H. Lin, M. Lu, N. Milosavljevic, J. Gao, and L. Guibas. 2008. Composable information gradients in wireless sensor networks. In *Proceedings of ACM/IEEE IPSN*. 121–132.

S. Liu, K.-W. Fan, and P. Sinha. 2009. CMAC: An energy efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. *ACM Trans. Sens. Netw.* 5, 29.

Y. Liu, Y. He, M. Li, et al. 2011a. Does wireless sensor network scale? A measurement study on GreenOrbs. In *Proceedings of IEEE Infocom*. 873–881.

Y. Liu, Y. Zhu, Lionel M. Ni, and G. Xue. 2011b. A reliability-oriented transmission service in wireless sensor networks. *IEEE Trans. Parall. Distrib. Syst.* 22, 2100–2107.

Q. Ma, K. Liu, X. Miao, and Y. Liu. 2011. Opportunistic concurrency: A MAC protocol for wireless sensor networks. In *Proceedings of IEEE DCOSS*.

C. J. Merlin and W. B. Heinzelman. 2009. Schedule adaptation of low-power-listening protocols for wireless sensor networks. *IEEE Trans. Mobile Comput.* 9, 5 (2009), 672–685.

P. Park, C. Fischione, and K. H. Johansson. 2010. Adaptive IEEE 802.15. 4 protocol for energy efficient, reliable and timely communications. In *Proceedings of ACM/IEEE IPSN*. 327–338.

J. Polastre, J. Hill, and D. Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of ACM SenSys*. 95–107.

S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. 2006. Interference-aware FaireRate control in wireless sensor networks. In *Proceedings of ACM SIGCOMM*. 63–74.

Y. Sun, O. Gurewitz, and D. Johnson. 2008. RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of ACM SenSys*. 1–14.

L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson. 2011. PW-MAC: An Energy-Efficient Predictive-Wakeup MAC Protocol for Wireless Sensor Networks. In *Proceedings of IEEE Infocom*. 1305–1313.

S. J. Tang, X. Y. Li, X. Shen, J. Zhang, G. Dai, and S. K. Das. 2011. Cool: On coverage with solar-powered sensors. In *Proceedings of IEEE ICDCS*. 488–496.

TelosB. 2004. (2004). http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf.

L. Wang and W. Liu. 2011. Navigability and reachability index for emergency navigation systems using wireless sensor networks. *Tsing. Scie. Tech.* 16, 6, 657–668.

X. Wang, L. Fu, X. Tian, Y. Bei, Q. Peng, X. Gan, H. Yu, and J. Liu. 2012. Converge-cast: On the capacity and delay tradeoffs. *IEEE Trans. Mobile Comput.* 11, 970–982.

X. Wang, X. Wang, G. Xing, and Y. Yao. 2010. Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks. In *Proceedings of IEEE IWQoS*. 1–9.

Y. Wang, Y. He, X. Mao, Y. Liu, Z. Huang, and X. Li. 2012. Exploiting constructive interference for scalable flooding in Wirelessnetworks. In *Proceedings of IEEE Infocom*. 2104–2112.

G. Werner-Allen, S. Dawson-Haggerty, and M. Welsh. 2008. Lance: Optimizing high-resolution signal collection in wireless sensor networks. In *Proceedings of ACM SenSys*. 169–182.

W. Ye, J. Heidemann, and D. Estrin. 2002. An energy-efficient MAC protocol for Wireless Sensor Networks. In *Proceedings of IEEE Infocom*. 1567–1576.

W. Ye, F. Silva, and J. Heidemann. 2006. Ultra-Low Duty Cycle MAC with schedulled Channel Polling. In *Proceedings of ACM SenSys*. 321–334.

T. Zhu, Y. Gu, T. He, and Z. L. Zhang. 2010. eShare: A capacitor-driven energy storage and sharing network for long-term operation. In *Proceedings of ACM SenSys*. 239–252.

T. Zhu, Z. Zhong, Y. Gu, T. He, and Z. Zhang. 2009. Leakage-aware energy synchronization for wireless sensor networks. In *Proceedings of ACM MobiSys*. 319–332.

Y. Zhu. 2012. Statistically bounding detection latency in low-duty-cycled sensor networks. *Int. J. Distrib. Sens. Netw.*

Y. Zhu, Y. Liu, and L. M. Ni. 2011. Optimizing event detection in low duty-cycled sensor networks. *Wireless Netw.* (2011), 1–15.

Received January 2012; revised April 2012 and January 2013; accepted May 2013