A Distributed Wake-up Scheduling for Opportunistic Forwarding in Wireless Sensor Networks

Chul-Ho Lee   Do Young Eun†
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27695-7911
{clee4, dyeun}@eos.ncsu.edu

***Abstract*—In wireless sensor networks (WSNs), sensor nodes are typically subjected to energy constraints and often prone to topology changes. While *duty cycling* has been widely used for energy conservation in WSNs, *random walks* have been popular for many delay-tolerant applications in WSNs due to their many inherent desirable properties. In this paper, we consider an opportunistic forwarding under an asynchronous and heterogeneous duty cycling. We first show that its resulting packet trajectory can be interpreted as a continuous-time random walk, and then provide an analytical formula for its end-to-end delay. Since the extremely large end-to-end delay is still undesirable even for most delay-tolerant applications, we develop a *distributed* wake-up scheduling algorithm in which each node autonomously adjusts its (heterogeneous) wake-up rate based *only* on its own degree information so as to improve the worst-case end-to-end delay. In particular, we prove that our algorithm outperforms pure homogeneous duty cycling, where every node uses the same wake-up rate, in its guaranteed asymptotic upper bound of the worst-case delay for *any* graph. In addition, we show that our proposed algorithm brings out more than 35% performance improvement on average when compared with pure homogeneous duty cycling, under various settings of random geometric graphs via numerical evaluations and independent simulation results.**

I. INTRODUCTION

Random walks have been considered and evaluated as viable solutions for a wide range of applications in wireless sensor networks (WSNs) such as routing/forwarding, data gathering/harvesting, and query process [1], [2], [12], [3], [7]. Such a popularity of random walk-based algorithms originates in inherent and desirable properties of random walks including its simplicity of implementation and deployment, scalability, load-balancing, and robustness to topology changes, though they often lead to larger end-to-end delay. In particular, the random walks are stateless and tend to avoid critical points of failure and non-uniform energy depletion caused by hot-spot formation or congested areas [1] that topology-driven algorithms (e.g., shortest path-based and cluster heads-based) naturally incur. Therefore, the random walk-based algorithms are efficient and applicable for delay-tolerant applications where the end-to-end delay is not a primary concern.

In WSNs, a sleep-wake duty cycling has been adopted for energy efficiency and conservation, since each sensor node is typically equipped with a battery (power-limited) [3]. Specifically, the duty cycling has been used to significantly reduce the cost of idle listening on the channel which is the source of most power consumption along with packet transmission/reception, where in the idle listening, each sensor node keeps its RF transceiver 'on' for any possible packet from its neighbors. Moreover, a *random* sleep-wake duty cycling, where each node randomly and independently performs its duty cycling, is desirable because it is self-configurable and does not require any global coordination, which in turn makes the network more robust to topology changes.

While most of the previous works [1], [2], [12] propose and evaluate the random walk-based algorithms in WSNs without any duty cycling, recent works [3], [7] have begun to consider a simple random walk as a packet forwarding method under the presence of a random sleep-wake duty cycling, where the next node visited is chosen uniformly at random from the current node. Specifically, under a synchronous, homogeneous sleep-wake duty cycling, they propose an opportunistic forwarding in which each node having a packet forwards it to the first awake neighboring node, i.e., the node who turns on its RF transceiver for the first time among the neighbors. They also show that the packet trajectory can be viewed as a simple random walk with heterogeneous sojourn time and derive a closed-form expression of the average delay.

In this paper, we consider the opportunistic forwarding [3], [7], by contrast, under an *asynchronous, heterogeneous* sleep-wake duty cycling where each sensor node i wakes up according to a Poisson process with rate λ_i. The asynchronous duty cycling does not entail any synchronization overhead and complexity, which is necessary for any typical synchronous network operation (e.g. [3], [7]). We first show that the opportunistic forwarding under the asynchronous and heterogeneous duty cycling can be translated into a continuous-time random walk on a graph, or equivalently, a discrete-time random walk with heterogeneous sojourn time in each vertex of the graph. This construction is rather similar to the one in [3], [7] but our case covers much wider class of random walks with different λ_i, while only homogeneous duty-cycling with $\lambda_i = \lambda$ for all i was considered in [3], [7].

In this setup, we provide an analytical formula for the average delay performance of the opportunistic forwarding with the duty cycling, and then address how to control the wake-up rate λ_i for each node i in a *distributed* manner so as to improve the delay performance. Since extremely large end-to-end delay is prohibitive even for most delay-tolerant

† This work was supported in part by National Science Foundation under grants CNS-0831825, CCF-0830680, and CAREER Award CNS-0545893.

applications, we propose a distributed wake-up scheduling algorithm with heterogeneous duty cycling in which each node i autonomously adjusts its wake-up rate λ_i based only on its own degree information to improve the worst-case average end-to-end delay for any given network. By extending a recent result in [9], we prove that our proposed algorithm brings out performance improvement from $O(n^3)$ (guaranteed under pure homogeneous duty cycling) to $O(\sqrt{d_{max}}n^2)$ in the asymptotic upper bound of the worst-case average delay for *any* graph, where n is the total number of nodes and d_{max} is the maximum degree over the network. In particular, we also compare the exact performance of worst-case average delay of our proposed algorithm with that of pure homogeneous duty cycling through numerical evaluations and independent simulation results. We then show that our algorithm leads to more than 35% performance improvement on average under various settings of random geometric graphs that have been widely used in the literature for modeling diverse wireless sensor or ad-hoc networks [1], [2], [4], [3], [7].

II. Preliminaries

A. Network Model

There are n sensor nodes placed on a graph whose edges correspond to the reliable communication links between any pair of n nodes, if exist. We formally define a graph $G = (\mathcal{N}, \mathcal{E})$ with a set of sensor nodes $\mathcal{N} = \{1, 2, \dots, n\}$ and a set of edges $\mathcal{E}$, where $|\mathcal{N}| = n < \infty$. If a sensor node $i \in \mathcal{N}$ can reliably communicate with other sensor node $j \in \mathcal{N}$, then an edge between sensor nodes i, j exists, i.e., $(i, j) \in \mathcal{E}$ $(i \neq j)$. Note that we assume G is an undirected and connected graph. In other words, the communication link between sensor nodes i, j is symmetric and there is at least one (routing) path from each sensor node to every other sensor nodes. Let $N(i) = \{j \in \mathcal{N} : (i, j) \in \mathcal{E}\}$ be the set of neighbors of node $i \in \mathcal{N}$ and $d(i) = |N(i)|$, i.e., $d(i)$ is the degree of node i.

B. Opportunistic Forwarding under An Asynchronous, Heterogeneous Sleep-Wake Duty Cycling

We explain how an opportunistic forwarding performs under an *asynchronous, heterogeneous* sleep-wake duty cycling. The principle of the opportunistic packet forwarding is the same as the one in [3], [7], while the underlying duty cycling is different, but more beneficial.

In the asynchronous and heterogenous duty cycling, each node i sleeps (i.e., completely turns off its RF transceiver or is in an 'off' state) and independently wakes up (i.e., turns on the RF transceiver or is in an 'on' state) according to a Poisson process with rate parameter $\lambda_i > 0$. In other words, the inter-wake-up time (the duration of 'off' state) for each sensor node i is drawn from an independent exponential distribution with rate λ_i. Again, this asynchronous sleep-wake duty cycling is self-configurable and does not need any (global) synchronization overhead and complexity, which is appealing for ad-hoc networking systems. Note also that in contrast to [3], [7], we allow different λ_i for each sensor node i rather than simply using the same $\lambda_i = \lambda$ for all nodes i.

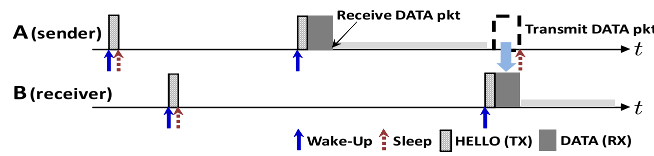

Fig. 1. The operation of data transmission via an opportunistic forwarding under an asynchronous, heterogeneous sleep-wake duty cycling

While performing this duty cycling, if any sensor node i has a packet to transmit, it opportunistically forwards the packet to a first awake node among $N(i)$. Here, any packet transmission/reception between two adjacent sensor nodes can occur, only when they are both on and each of them is aware that the other node is also on. To this end, prior to data communication, when each sensor node i has a packet to transmit, it simply performs an idle listening on the channel until to find a first awake node in $N(i)$. When a node without a packet wakes up, it first conducts a carrier sense on the channel, and then broadcasts a small hello message to notify its 'on' state to any neighboring node having a packet only if the channel is idle; otherwise, it goes to sleep. Fig. 1 depicts the operation of data transmission between two nodes via this opportunistic forwarding. From Poisson process assumption for the beginning instants of 'on' state and short duration for the small hello message, it is unlikely that two or more sensor nodes wake up at the same time, and therefore ignored in our consideration. In addition, since a carrier sensing range (or interference range) is typically twice larger than a transmission range, any awake two-hop neighbors of a transmitting node can detect the presence of a packet transmission if it happens, and thus there will be no collision between a packet transmission and any hello message transmission.

Throughout the paper, we suppose that traffic load in the network is light as assumed in [1], [2], [12], [3], [7]. Hence, an interference from concurrent data transmissions is not an issue, and we measure the delay performance based on the delivery of a single packet. Also, due to the light traffic load and small size of hello message, we assume that time duration required for a packet transmission/reception is relatively negligible compared with the duration of 'off' state for each node.

III. Performance Modeling and Analysis

In this section, we show that the random trajectory of a packet traveling over a given graph (network) G under the opportunistic forwarding with asynchronous and heterogenous duty cycling, can be interpreted as a continuous-time random walk (CTRW) on G, or equivalently, a discrete-time random walk (DTRW) on G with heterogeneous sojourn time. Hence, the forwarding algorithm in our class also retains natural properties of random walk-based algorithms such as simplicity, scalability, load-balancing, and robustness to topology changes. We then define our performance metrics and provide analytical formulas to compute the metrics numerically.

A. Continuous-Time Random Walk on A Graph

We consider a packet of interest which travels over G where its source and destination are not specified here, temporarily. Let $\{X(t) \in \mathcal{N}\}_{t \geq 0}$ be a continuous-time random sequence (or trajectory) of the packet, indicating where the packet is located at time t. One can observe that the sojourn time of $X(t)$ at node i (say W_i), i.e., the waiting time for each node i until to find any first awake node in $N(i)$, is independent of the index of the first awake node among $N(i)$ (say I_i), i.e., the node which receives the packet from node i. Due to the space constraint, we refer to our technical report [10] for details. Then, from this independence, $\{X(t)\}$ becomes a continuous-time Markov chain (CTMC) with state space $\mathcal{N}$ where each state i corresponds to node i currently holding the packet, and it can be also interpreted as a CTRW on G. Then, the transition rate from state i to state j is given by

$$q_{ij} = \begin{cases} \lambda_j & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

for $i \neq j$ and $q_{ii} = -\sum_{k \in N(i)} \lambda_k$. We define by $\boldsymbol{Q} \triangleq \{q_{ij}\}_{i,j \in \mathcal{N}}$ the transition rate matrix of $\{X(t)\}$. Note that the amount of sojourn time the process spends in state i before jumping into one of its neighboring nodes $j \in N(i)$ is simply the waiting time W_i of the packet at sensor node i.

Equivalently, we can construct an *embedded* discrete-time Markov chain (DTMC) with transition probability p_{ij} given by

$$p_{ij} = \begin{cases} \mathbb{P}\{I_i = j\} = \frac{\lambda_j}{\sum_{k \in N(i)} \lambda_k} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $\sum_{j \neq i} p_{ij} = 1$ and $p_{ii} = 0$. We define by $\{Y_m\}_{m \geq 0}$ the random sequence of nodes visited and by $\boldsymbol{P} \triangleq \{p_{ij}\}_{i,j \in \mathcal{N}}$ the transition probability matrix of $\{Y_m\}$. Then, we can also interpret the random trajectory of the packet traveling over G as a DTRW on G with heterogeneous sojourn time W_i at state i, where the state transition probability of DTRW is the same as p_{ij} of the embedded DTMC in (2). As a special case, if $\lambda_i = \lambda$ for all $i \in \mathcal{N}$, then the resulting DTRW on G becomes a *simple* random walk, i.e., $p_{ij} = 1/d(i)$, if $(i,j) \in \mathcal{E}$, otherwise $p_{ij} = 0$. Note that this is the case (homogeneous duty cycling) considered in [3], [7] under the synchronous network setting.

B. Performance Metric and Analysis

Consider the *hitting time* (or first passage time) from state (node) i to state j on G, which is defined as

$$H_{ij} \triangleq \inf\{t > 0 : X(t) = j \mid X(0) = i\} \tag{3}$$

for $i, j \in \mathcal{N}$ and $i \neq j$, and $\overline{H}_{ij} \triangleq \mathbb{E}\{H_{ij}\}$. The hitting time represents the delay of unicast where the source node i sends a packet containing its local observation (e.g., temperature) to the destination node j, as similarly used in [3], [7]. Note that the destination node can be a powered sink with Internet connection or a usual sensor node in the scenario [11] where the powered sink is not available due to geographical reasons. To be precise, (3) is proper for the latter case. For the former case, the powered sink may not need to perform the duty cycling and its neighboring nodes can be aware of its presence via a neighbor discovery process [13]. Hence, in this case, the delay of unicast becomes the time taken for a packet from the source node to reach any one of neighbors of the destination, and (3) can be properly modified as the hitting time to the set of neighbors of the destination.

For any given graph (network) G, the mean hitting time $\overline{H}_{ij}$ can be obtained as 'mean time to absorption' for an absorbing chain that can be constructed from the CTMC $\{X(t)\}$. Specifically, suppose that state j is an absorbing state. We can re-label the set of states $\mathcal{N}$ such that transient states $\mathcal{N} \setminus \{j\}$ come first. Then, the transition rate matrix $\boldsymbol{Q}$ of $\{X(t)\}$ can be rewritten as

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{T^0} \\ \boldsymbol{0} & 0 \end{bmatrix},$$

where $\boldsymbol{T}$ is a $(n-1) \times (n-1)$ transition rate matrix, $\boldsymbol{T^0}$ is an $(n-1)$-dimensional column vector, each of whose elements is the transition rate from state i to the absorbing state j, and $\boldsymbol{0}$ is a row vector of zeros. Then, $\overline{H}_{ij}$ can be obtained as [5]

$$\overline{H}_{ij} = -\boldsymbol{\alpha} \boldsymbol{T^{-1}} \boldsymbol{e}, \tag{4}$$

where $\boldsymbol{\alpha}$ is an initial distribution (row) vector in which an element of state i is one and other elements are zeros, and $\boldsymbol{e}$ is a column vector of ones. Hence, one can easily evaluate $\overline{H}_{ij}$ by numerically computing (4).

The performance of the mean hitting time $\overline{H}_{ij}$ depends on how to perform sleep-wake duty cycling at each sensor node i, or specifically, how to control wake-up rate λ_i of each sensor node i. Note that we do not assume any coordination in network operations and any node can be a source or destination node for the packet delivery. It is also desirable to avoid the extremely large end-to-end delay even for delay-tolerant applications. Hence, in the design of a distributed wake-up scheduling, we will focus on improving the worst-case delay performance $\overline{H}_{max}$, defined as

$$\overline{H}_{max} \triangleq \max_{i,j \in \mathcal{N}, i \neq j} \overline{H}_{ij}, \tag{5}$$

i.e., the maximum mean hitting time over all possible source-destination pairs in the network.

IV. A Distributed Wake-up Scheduling Based on Local Degree Information

In this section, we first examine how to control wake-up rate λ_i of each node i depending on the underlying structure of a given graph (network) G, and then propose a distributed wake-up scheduling algorithm in which each node i autonomously controls its wake-up rate λ_i based only on its own degree information in order to improve (shorten) the maximum mean hitting time $\overline{H}_{max}$ than the case with homogeneous sleep-wake duty cycling $\lambda_i = \lambda$ for all i.

For a fair comparison among all possible heterogeneous duty cycling strategies, we set

$$\frac{1}{\lambda} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\lambda_i} \tag{6}$$

for all possible wake-up rates $\{\lambda_i, i \in \mathcal{N}\}$, and compare the delay performance under $\{\lambda_i\}$ with that under the homogeneous duty cycling with the same wake-up rate λ given by (6) for all nodes in the network. Since sensor node i consumes one unit amount of power whenever it wakes up every $1/\lambda_i$

seconds on average, the above condition implies that the average power consumption rate over the whole network is kept the same under heterogeneous duty cycling with $\{\lambda_i\}$ and under the corresponding homogeneous case with λ in (6).

Recall that the embedded DTMC, $\{Y_m\}$, (or a DTRW on G) under any homogeneous duty cycling becomes a *simple* random walk, as mentioned in Section III. It is well known that the stationary distribution π'_i is proportional to the degree of node i, i.e., $\pi'_i = d(i)/\sum_{k\in\mathcal{N}} d(k)$, where π'_i is the stationary distribution of $\{Y_m\}$. This implies that a packet of interest (a random walk) in the network visits more likely to a sensor node with higher degree. At the same time, if the underlying graph G is a regular graph where $d(i) = d$ for all $i \in \mathcal{N}$, then the stationary distribution is uniform, i.e., $\pi'_i = 1/n$ for all i. Hence, the packet of interest visits every node equally likely and there may not be any benefit by making the wake-up rates λ_i heterogeneous in shortening or minimizing the maximum mean hitting time $\overline{H}_{max}$. This observation can be made rigorous for a complete graph $G = K_n$ (a special case of regular graphs) as shown in the following.

Theorem 1: For a complete graph $G = K_n$, and for a given $\lambda > 0$, $\lambda_i = \lambda$ for all $i \in \mathcal{N}$ is optimal in minimizing $\overline{H}_{max}$ over all $\{\lambda_i\}$ satisfying (6), i.e., $1/n\sum_{i=1}^n 1/\lambda_i = 1/\lambda$. □

Proof: See our technical report [10]. ■

Note that the proof of optimality in Theorem 1 is still non-trivial, though it is straightforward to derive $\overline{H}_{max}$ under the *homogeneous* duty cycling for a complete graph K_n.

However, in general, unless one carefully designs a *coordinated* sensor network, sensor network topologies would be rather heterogeneous (i.e., the degree of each node is different) and form non-regular graphs. Since a packet of interest is more likely to visit a sensor node with higher degree as mentioned above, the homogeneous duty cycling may be no longer optimal. Instead, it would be helpful to control the wake-up rate λ_i of each node i such that the packet visits every node more evenly so as to shorten $\overline{H}_{max}$ under non-regular graphs.

In addition, a recent work [9] by Ikeda et al. addresses how to design a 'better' (discrete-time) random walk on a graph than a simple random walk, where the transition probability of the random walk is given by

$$\tilde{p}_{ij} = \begin{cases} \frac{d(j)^{-\alpha}}{\sum_{k\in N(i)} d(k)^{-\alpha}} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

for some constant α. In particular, they prove that if $\alpha = 1/2$ in (7), then the maximum mean hitting time, or the maximum average number of transitions taken to visit any given node (state), is $O(n^2)$ for *any* graph. Note that we here use different notation $\tilde{p}_{ij}$ to distinguish it from the transition probability p_{ij} of the *embedded* DTMC, $\{Y_m\}$. This result suggests that a (proper) bias of a transition into a node with *smaller* degree can bring out an improvement of the maximum mean hitting time from $O(n^3)$ (guaranteed under the simple random walk [6], [9]) to $O(n^2)$ for *any* graph. Note that this asymptotic upper bound may be loose depending on underlying graphical structures. Also, (biased) random walks with transition probabilities in the form of (7) have been also studied in the physics literature (e.g., see [8]).

Inspired by the Ikeda et al.'s finding in [9], we propose a distributed wake-up scheduling algorithm in which each node i *autonomously* controls its wake-up rate λ_i based *only* on its own degree information as

$$\lambda_i = \lambda_0 d(i)^{-\beta}, \tag{8}$$

where λ_0 is an initial wake-up rate and β is a tunable parameter. Note that the value of λ_0 can be programmed into each sensor system before bootstrapping, and the degree information can be easily obtained through a neighbor discovery process [13] which takes place right after the network is deployed, and remain up-to-date through data communications as time goes on. Then, from (2) and (8), the transition probability p_{ij} of $\{Y_m\}$ becomes

$$p_{ij} = \begin{cases} \frac{\lambda_j}{\sum_{k\in N(i)} \lambda_k} = \frac{d(j)^{-\beta}}{\sum_{k\in N(i)} d(k)^{-\beta}} & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where now this transition probability p_{ij} becomes identical to $\tilde{p}_{ij}$ in (7). Hence, we arrive to the following asymptotic upper bound of $\overline{H}_{max}$ achievable under our proposed algorithm by extending* the Ikeda et al.'s result [9].

Theorem 2: For *any* graph G, if $\beta = 1/2$, then $\overline{H}_{max} \leq \frac{3\sqrt{d_{max}}n^2}{\lambda_0}$, and hence $\overline{H}_{max} = O\left(\sqrt{d_{max}}n^2\right)$, where $d_{max} \triangleq \max_{i\in\mathcal{N}} d(i)$. □

Proof: See our technical report [10]. ■

Similarly, we can show that any homogeneous sleep-wake duty cycling (or a simple random walk with random sojourn time) guarantees $\overline{H}_{max} = O(n^3)$ for any graph, based on the fact that a discrete-time simple random walk (one unit of sojourn time at every node) guarantees $O(n^3)$ maximum mean hitting time for any graph [6], [9]. Since $d_{max} < n$, we have $O\left(\sqrt{d_{max}}n^2\right) \leq O\left(n^{2.5}\right) < O\left(n^3\right)$, i.e., our proposed algorithm improves its guaranteed asymptotic upper bound of $\overline{H}_{max}$ from $O\left(n^3\right)$ to $O\left(\sqrt{d_{max}}n^2\right)$ for *any* graph.

V. Numerical Evaluations and Simulation Results

In this section, we present the exact performance of $\overline{H}_{max}$ via numerical evaluations of (4) and provide independent simulation results to show that our proposed algorithm significantly outperforms pure homogeneous sleep-wake duty cycling under random geometric graphs, which have been widely used in the literature [1], [2], [4], [3], [7] to model diverse wireless sensor or ad-hoc network topologies. The random geometric graph, denoted as $RGG(n,r)$, is a graph where n nodes are uniformly and independently located in the unit square, and two nodes are connected if they are within distance of r. While $RGG(n,r)$ becomes regular asymptotically for sufficiently large r [4], it typically exhibits heterogeneity in its resulting degree sequence for any given finite value of r. Thus,

*[9] is based on a discrete-time random walk where the walk spends one time unit in every vertex. In contrast, we are dealing with the embedded DTMC $\{Y_m\}$ whose sojourn time in node i is random and different over i.

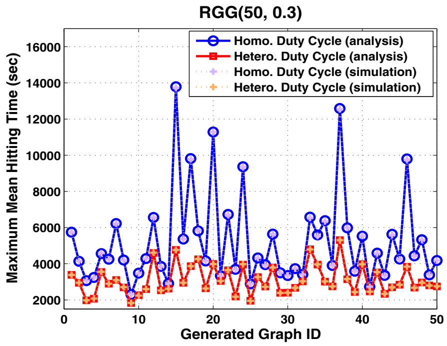

Fig. 2. $\overline{H}_{max}$ under 50 different sample topologies of $RGG(50, 0.3)$.

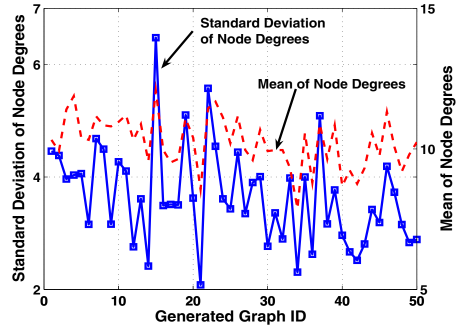

Fig. 3. Statistics of node degrees of 50 different sample topologies of $RGG(50, 0.3)$ used in Fig. 2.

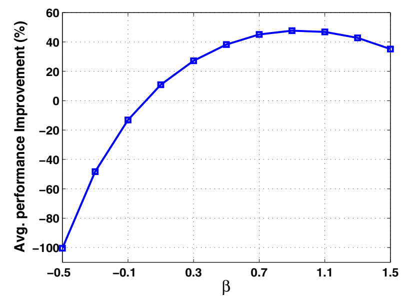

Fig. 4. Average performance improvement in $\overline{H}_{max}$ over 500 different sample topologies of $RGG(50, 0.3)$ while varying β in (8).

we set out to investigate the performance of our proposed algorithm or how the local degree information can be exploited in controlling the wake-up rate λ_i of each node i to improve $\overline{H}_{max}$ under various sample topologies of $RGG(n, r)$.

Fig. 2 shows the numerical results of $\overline{H}_{max}$ by computing (4) and simulation results, obtained through our custom event-driven simulator using C++, for our proposed algorithm (heterogeneous sleep-wake duty cycling) and its corresponding homogeneous duty cycling with λ from (6), under 50 different sample topologies of $RGG(50, 0.3)$. Here, the connectivity of each of 50 sample graphs is ensured, and $\beta = 1/2$ in (8) is set for our algorithm with $\lambda_0 = 0.02$. Since (4) is exact for CTMC, the simulation results precisely match with the numerical computations. Fig. 2 clearly shows that our algorithm outperforms the pure homogeneous duty cycling for all 50 different generations of $RGG(50, 0.3)$. Specifically, an average performance improvement over 50 generated graphs is about 35%, while some case (graph id: 15) exhibits 65% performance improvement (reduction in $\overline{H}_{max}$).

To see any relationship between the heterogeneity of the graph topology and the amount of performance improvement of our proposed algorithm over the simple homogeneous duty-cycling, we plot in Fig. 3 the average and standard deviation of degree sequence for each of 50 sample topologies of $RGG(50, 0.3)$ used in Fig. 2. We observe that the pure homogeneous duty cycling performs poorly under topologies with higher standard deviation of degree sequence (e.g., graph id: 15, 37, 46), while our algorithm not only outperforms the pure homogeneous duty cycling but also tend to 'stabilize' the delay performance regardless of such irregularity in the node degrees of the given topology.

So far we have set $\beta = 1/2$ in (8) as originally suggested by [9]. While our asymptotic worst-case delay guarantee $\overline{H}_{max} = O\left(\sqrt{d_{max}}n^2\right)$ holds under $\beta = 1/2$, for finite sized graphs, it is still possible that our algorithm may perform better under different values of β. Fig. 3 shows the average performance improvement of our proposed algorithm compared with its corresponding homogeneous duty cycling with λ from (6) under various values of β in (8). We use the same $\lambda_0 = 0.02$ as before, and each data point in this figure is obtained by averaging the performance improvements over 500 different sample topologies of $RGG(50, 0.3)$. As seen in Fig. 3, it is possible to achieve close to 50% improvement when β is around 0.9, although $\beta = 1/2$ still results in 38% performance improvement on average, along with asymptotic guarantee for delay reduction for any arbitrary graph as shown in Theorem 2.

VI. Conclusion

In this paper, we have analyzed the end-to-end delay of an opportunistic forwarding under an asynchronous and heterogeneous sleep-wake duty cycling in WSNs, and proposed a distributed wake-up scheduling algorithm in which each sensor node controls its wake-up rate λ_i based only on its own degree information so as to improve the worst-case end-to-end delay. We have proven that our proposed algorithm leads to performance improvement from $O(n^3)$ to $O(\sqrt{d_{max}}n^2)$ in its guaranteed worst-case delay for any arbitrary graph. While our algorithm already achieves significant reduction in delay performance for any sensor network topology, we have also shown that it is still possible to extract even further performance improvement by carefully calibrating the parameter β in our distributed algorithm for a given class of network topologies, which we leave as a future work.

References

[1] C. Avin and C. Brito, "Efficient and robust query processing in dynamic environments using random walk techniques," in *Proc. of ACM/IEEE IPSN*, Berkeley, CA, 2004.

[2] C. Avin and B. Krishnamachari, "The power of choice in random walks: an emprical study," in *Proc. of ACM MSWiM*, Torremolinos, Spain, 2006.

[3] P. Basu and C.-K. Chau, "Opportunistic forwarding in wirelss networks with duty cycling," in *Proc. of ACM CHANTS*, San Francisco, CA, 2008.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Mixing times for random walks on geometric random graphs," in *Proc. of SIAM Workshop on Analytic Algorithmics and Combinatorics*, Vancouver, 2005.

[5] P. Brémaud, *Markov chains: gibbs fields, monte carlo simulation, and queues*. Springer-Verlag, 1999.

[6] G. Brightwell and P. Winkler, "Maximum hitting time for random walks on graphs," *Random Structures and Algorithms*, vol. 1, no. 3, pp. 263–276, 1990.

[7] C.-K. Chau and P. Basu, "Exact analysis of latency of stateless opportunistic forwarding," in *Proc. of IEEE INFOCOM*, Rio de Janeiro, Brazil, 2009.

[8] A. Fronczak and P. Fronczak, "Biased random walks in complex networks: the role of local navigation rules," *Physical Review E*, vol. 80, no. 1, pp. 1–6, July 2009.

[9] S. Ikeda, I. Kubo, and M. Yamashita, "The hitting and cover times of random walks on finite graphs using local degree information," *Theoretical Computer Science*, vol. 410, no. 1, pp. 94–100, Jan. 2009.

[10] C.-H. Lee and D. Y. Eun, "A distributed wake-up scheduling for opportunistic forwarding in wireless sensor networks," Tech. Rep., Mar. 2010, http://www4.ncsu.edu/~dyeun/pub/techrep-globecom10-lee.pdf.

[11] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, 2007.

[12] I. Mabrouki, X. Lagrange, and G. Froc, "Random walk based routing protocol for wireless sensor networks," in *Proc. of InterPerf*, Nantes, France, 2007.

[13] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proc. of ACM MobiCom*, Beijing, China, 2009.