Smart Sleep: Sleep More to Reduce Delay in Duty-Cycled 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*—A simple random walk (SRW) has been considered as an effective forwarding method for many applications in wireless sensor networks (WSNs) due to its desirable properties. However, a critical downside of SRW – slow diffusion or exploration over the space, typically leads to longer packet delay and undermines its own benefits. Such slow-mixing problem becomes even worse under random duty cycling adopted for energy conservation. In this paper, we study how to overcome this problem without any sacrifice or tradeoff, and propose a simple modification of random duty cycling, named *Smart Sleep*, which achieves more power-saving as well as faster packet diffusion (or smaller delay) while retaining the benefits of SRW. We also introduce a class of p-backtracking random walks and establish its properties to analytically explain the fast packet diffusion induced by Smart Sleep. We further obtain a necessary condition to achieve an optimal performance under our Smart Sleep, and finally demonstrate remarkable performance improvement via independent simulation results over various network topologies.**

I. INTRODUCTION

A simple random walk (SRW), among many other variants in the literature, has been actively used as an effective packet forwarding method in WSNs without duty cycling [3], [13], [12] and in randomly duty-cycled WSNs [5], [7], where each packet is forwarded from a node to one of its neighbors chosen uniformly at random. The wide-spread popularity of such SRW-based algorithms are mainly due to its inherent distributed nature and several desirable properties including simplicity of implementation and deployment, scalability, robustness to topology changes. They also tend to achieve load-balancing autonomously over the network, which helps avoid critical points of failure and non-uniform energy depletion in WSNs caused by hot-spot formation or congested areas.

However, the SRW has one critical drawback – slow mixing or slow diffusion over the space, which in turn leads to longer delay to reach the destination. There have been several works addressing how to overcome this drawback and the resulting system performance in other literature. In [6], [9], the authors study on the fast diffusion of a class of random walks as node mobility and its impact on the mobility-induced information spreading in mobile ad-hoc networks (MANETs). It has been also studied in [8], [11], [2] how to speed up the random walk (faster mixing or quicker convergence to its stationary distribution). A common underlying theme here is to steer the random walk toward the same 'direction' so as to avoid revisiting (or backtracking) previously-visited nodes (or places), thereby leading to more efficient exploration of the space and faster information delivery. Thus motivated, in this paper, we study how to achieve *faster diffusion or smaller delay of packets* in randomly duty-cycled WSNs while at the same time also achieving *power saving of every sensor node*, without requiring any multi-hop communications to collect topological or geographical information.

Specifically, in order to overcome the slow diffusion of SRW-based forwarding while saving more power at every sensor, we propose a simple yet effective modification, named *Smart Sleep*, on the random duty cycling. Smart Sleep operates as follows: whenever each node successfully forwards a packet to one of its neighbor, it goes to sleep for T seconds, making itself unavailable in the network. This temporary 'forced' sleep right after forwarding a packet reduces the chance of the same packet coming back to the same sensor (backtracking) for a while, thereby facilitating faster exploration for other nodes and 'speeding up' the packet for faster delivery. After this sleep period of T seconds, the sensor resumes its normal random duty cycling, preparing itself for forwarding/receiving other packets. Too large value of T will put many sensors into sleep for a long time and outweigh the benefit of faster diffusion of the packet that leaves such a long 'trail', thus slowing down the delivery of other packets in the network overall.

To set the stage for analytical treatment of Smart Sleep protocol, we introduce a class of *p-backtracking random walks* (p-BRW) on a general graph, which captures such dynamics of packet transitions – less backtracking to the previously visited node. Contrary to SRW, in p-BRW, a random walk (currently at node i) remembers the previous position and goes back to this previously visited node with probability p_i; otherwise, it moves to any one of other neighbors equally likely. We prove that the stationary distribution of the p-BRW is invariant with respect to the choice of p_i. This immediately implies that the average return time of p-BRW to a given node is also invariant. We then illustrate how the packet trajectory under our Smart Sleep can be best described by p-BRW with some backtracking probabilities $\{p_i\}$, which is generally a function of T and underlying network topology. By exploiting the close relationship among p_i, T, and other network characteristics and at the same time by leveraging the invariance property, we study how to choose the sleep duration T, leading to

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

better delay performance as well as transmission cost, while power saving due to additional sleep is self-evident. We then derive a necessary condition for the optimal sleep duration T^* in minimizing the packet delay, and then demonstrate the significant performance improvement through numerical simulations over various network topologies. Therefore, in our Smart Sleep, the delay and power becomes *no longer a typical tradeoff* and both can be improved together.

II. Preliminaries

A. Network Model

Consider n sensor nodes placed on a graph (or network), denoted as $G=(\mathcal{N},\mathcal{E})$, where $\mathcal{N}$ is a set of sensor nodes and $\mathcal{E}$ is a set of edges with $|\mathcal{E}|=m$. If a node $i\in\mathcal{N}$ can reliably communicate with other node $j\in\mathcal{N}$, then an edge between nodes i,j exists, i.e., $(i,j)\in\mathcal{E}$ $(i\neq j)$. Throughout the paper, we assume G is an undirected and connected graph. In other words, each communication link is symmetric and there exists at least one routing path from each node to every other nodes. We also define by $N(i)$ a set of neighbors of node i and by d_i the degree of i, i.e., $d_i=|N(i)|$. Note that $\sum_{i\in\mathcal{N}} d_i = 2m$.

B. Opportunistic Forwarding in Randomly Duty-Cycled WSNs

We explain a typical random duty cycling that each sensor node performs for energy/power conservation, and an opportunistic forwarding proposed in [5], [7] as a packet forwarding method, both of which will be used as a base setup for networking operations in this paper.

We consider a network operating in a synchronous mode, as assumed in [3], [13], [12], [5], [7]. Specifically, time is divided into slots and slot boundaries are synchronized (or can be re-synchronized). By the random duty cycling, we mean that each node independently wakes up (or turns on its RF transceiver) with probability $q>0$ at each time slot; otherwise, it sleeps (or turns off its RF transceiver) for the time slot with probability $1-q$. In addition, while each sensor node conducts this random duty cycling at each time slot, it forwards a packet, if it has, to one of its neighbors through the following opportunistic forwarding method. Whenever a sensor node having a packet wakes up, it opportunistically transmits the packet to *any* one of its neighbors *if* the neighbor also wakes up at the same slot; otherwise, the node having a packet looks for other next opportunities to forward the packet. When there are multiple neighbors waking up at the same slot, the tie will be randomly broken, i.e., one of the multiple awake neighbors will be chosen uniformly at random, as mentioned in [7]. We refer to our technical report [10] for more details on the practical issue regarding how to break the tie randomly.

C. Simple Random Walk Model For Opportunistic Forwarding

We here explain how the 'transitions' of each packet over the network G, governed by the opportunistic forwarding with random duty cycling, can be considered as a SRW*, while the packet stays at each node for some random amount of time before it is forwarded to another node. Suppose that node $i\in\mathcal{N}$ has a packet to transmit. Let $I_i\in N(i)$ be one of the neighbors of i that receives the packet from i or equivalently the first awake node in $N(i)$ that node i can find for the first time, while performing the random duty cycling. Also let W_i be the waiting time (in the number of time slots) for the packet at node i until being forwarded to I_i. As mentioned in [7], for each node $i\in\mathcal{N}$ having a packet,

$$\mathbb{P}\{I_i=k\}=1/d_i \tag{1}$$

if $k\in N(i)$; otherwise, zero. One can also observe that node i having a packet can transmit the packet to any of its neighbors with probability $q_i \triangleq q(1-(1-q)^{d_i})$ at each time slot. Thus, the waiting time W_i is geometrically distributed with mean

$$\mathbb{E}\{W_i\} = 1/q_i = [q(1-(1-q)^{d_i})]^{-1}. \tag{2}$$

Therefore, the transition of each packet from a node to one of its neighbors through the opportunistic forwarding follows a SRW, with heterogeneous random sojourn time [5], [7], where the heterogeneity comes from varying degrees d_i, $i\in\mathcal{N}$.

III. *Smart Sleep*: How To Sleep More and Better

In this section, we propose a simple modification on the random duty cycling, which we call *Smart Sleep*, whose operation is defined as follows. Whenever each node successfully forwards a packet to one of its neighbors, it goes to sleep and stays asleep for a constant amount of time slots $T\geq 0$ after which it resumes the random duty cycling, where the parameter T is to be chosen later. Here, when a node goes into this sleep mode for the time duration T, we say that the node is in a 'sleep mode'; otherwise, the node is in a 'normal mode' in which the node performs the random duty cycling. Note that if $T=0$, then it reduces to the original setting of random duty cycling.

One can see that if T is too long, then many sensors nodes would sleep for quite a long time and hence the packet would get delayed longer. In addition, very large value of T would put more sensors into sleep longer, rendering those sleeping sensors unavailable for forwarding of other packets, if any, and thus affecting the transition behavior of those other packets over the network. Observe that the power consumption of each sensor is monotonically decreasing as T increases, thus the advantage of Smart Sleep over the random duty cycling is obvious from the power saving point of view. For packet delay point of view, however, it may seem unclear at first sight whether forcing sensors into more sleep right after forwarding can actually lead to smaller delay in the network. Before going into the details for general set-up, we below demonstrate using a simple network topology how our Smart Sleep influences the dynamics of packet forwarding/delay.

Consider 1-D ring with a set of nodes $\mathcal{N}$. This topology is simple, yet able to capture key dynamics, and only used to obtain qualitative understanding. We look at how a packet of interest travels over 1-D ring in which all nodes are initially in a normal mode. Suppose that a packet of interest is generated

*While the opportunistic forwarding can also work with a variant of the random duty cycling, called a pseudo-random duty cycling [5], [7], the transition of each packet over network G here is still done in a SRW fashion.

at node i and its destination is at least three-hop away from node i. This packet will be forwarded to one of two neighbors of i with equal probability as in (1) and the average waiting (sojourn) time for the packet at node i is simply given by (2), as every node initially performs the random duty cycling. However, once the packet is forwarded to one of neighbors of node i, the behavior of packet transition over 1-D ring and the average sojourn time will be different. In this regard, we first establish a mapping between the packet transition over 1-D ring and the transition of a correlated random walk (CRW) [4], [6] on 1-D ring. In the CRW on 1-D ring, the walk at a vertex moves to its neighbor in the same direction (right or left) that the walk has taken, with probability (say $\tilde{p}$) larger than 0.5; otherwise, it changes the direction and moves to its another neighbor. We also derive the average sojourn time for the packet at each node as a function of T. Let $\tilde{W}_i$ be the sojourn time of the packet at node i until being further forwarded, given that node i has just received the packet from one of its neighbors. Then, we have

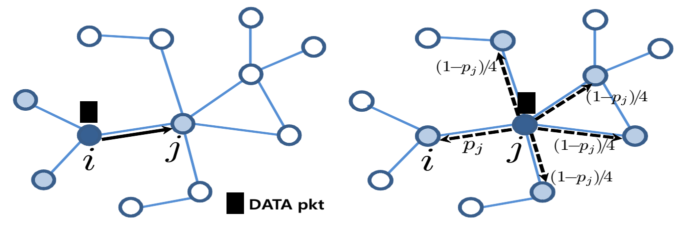
Theorem 1: If G is 1-D ring, the transition of each packet over 1-D ring, under Smart Sleep, behaves as if it follows a CRW with $\tilde{p}=1–0.5(1-q^2)^T>0.5$, while the average sojourn time at node i is given by $\mathbb{E}\{\tilde{W}_i\}=1/q^2\cdot[1-(1-q)\cdot(1-q^2)^T/(2-q)]$ for all $i\in\mathcal{N}$ except the destination. □

Proof: See our technical report [10]. ■

We note that [6] analytically showed that as the probability of the walk to follow the same direction that the walk has followed gets higher, the 'diffusion' of the walk over 1-D ring becomes faster. Hence, by observing that $\tilde{p}$ is increasing in T, if the sleep duration T is longer, we can achieve faster diffusion of the packet over the network, which in turn brings out smaller number of packet forwardings required until packet delivery. However, as can be seen from Theorem 1, $\mathbb{E}\{\tilde{W}_i\}$ is increasing in T with $\mathbb{E}\{\tilde{W}_i\}=\mathbb{E}\{W_i\}$ for $T=0$. In other words, as T increases, the number of transitions till delivery is generally decreasing in T but at the same time the average sojourn time at each node is now increasing. Moreover, in the presence of multiple packets, very large value of T prevents each node from being active *in time* to serve other upcoming packets, rendering the average sojourn time much longer than expected. Hence, one has to be careful in choosing T to achieve reduction in actual packet delay. The observation here on a simple 1-D ring topology will be a starting point for more in-depth discussion in the rest of this paper as to how to suitably choose T in Smart Sleep under general network topologies as well as the presence of multiple packets.

IV. p-Backtracking Random Walk and Its Connection to *Smart Sleep*

In this section, we introduce *p-backtracking random walk* (p-BRW), a class of discrete-time random walks that capture such dynamics – less backtracking to the previous visited node, on general graphs. We then derive several properties of p-BRW and explain how to achieve faster diffusion of p-BRW over the graph G. We finally discuss how the p-BRW is related to the transition of a packet of interest over G under Smart Sleep.



(a) time $t-1$ (b) time t

Fig. 1. Illustration of transitions of p-BRW. (a) A p-BRW is located at node i at time $t-1$ and is going to move to node j. (b) At time t, the p-BRW chooses one of the neighbors of node j according to the transition probability in (3) as the next node that it will move to.

A. p-Backtracking Random Walk

The p-BRW is a class of *discrete-time* random walks on G and is defined as follows. A random walk at the current node $i\in\mathcal{N}$ with $d_i>1$ goes back to the previously visited node with probability p_i (we called *backtracking probability* throughout the paper); otherwise, the random walk moves to the next node, chosen uniformly at random among the neighbors of i except the previously visited node. If node i has only one neighbor ($d_i=1$), the walk always returns back to the previously visited node, i.e., $p_i=1$. At $t=0$, the walk initially chooses one of its neighbors uniformly at random, and then continue to use the previously visited node as a 'signpost' to decide the next node that the walk will move to.

Let X_t, $t=0,1,2,\ldots$, be the location of p-BRW over $G=(\mathcal{N},\mathcal{E})$. From the definition of p-BRW, its dynamics can be characterized by

$$\mathbb{P}\{X_{t+1}=k|X_t=j,X_{t-1}=i\}=\begin{cases}p_j & \text{if } k=i,\\ \frac{1-p_j}{d_j-1} & \text{if } (j,k)\in\mathcal{E}, k\neq i,\\ 0 & \text{otherwise,}\end{cases} \quad (3)$$

for $d_j>1$ and $(i,j)\in\mathcal{E}$. If $d_j=1$, then $\mathbb{P}\{X_{t+1}=i|X_t=j,X_{t-1}=i\}=1$ for $i\in N(j)$; otherwise, zero. To avoid triviality, we assume that $0\leq p_j<1$ for all nodes j with $d_j>1$. Fig. 1 depicts an example of possible transitions of p-BRW from a node to one of its neighbors. We also emphasize that the p-BRW is general enough and includes several random walks such as SRW, CRW on 1-D ring [4], [6], non-backtracking random walk (NBRW) on regular graphs [2] as special cases. A detailed discussion on the special cases of p-BRW can be found in our technical report [10].

B. Properties of p-BRW

One can see that $\{X_t\}$ with state space $\mathcal{N}$ is not a Markov chain due to its memory of the previous state as shown in (3). However, by augmenting the state space, we can construct a Markov chain for the random sequences of nodes visited by the p-BRW as follows. We define by $\mathcal{S}$ a set of directed edges, i.e., $\mathcal{S}\triangleq\{(i,j):i\in\mathcal{N},j\in N(i)\}$ and $(i,j)\neq(j,i)$ in general. Note that $|\mathcal{S}|=2|\mathcal{E}|=2m$. Let $Z_t\triangleq(X_{t-1},X_t)$ for $t\geq 0$. Then, $\{Z_t\}_{t\geq 0}$ becomes a Markov chain on the state space $\mathcal{S}$ and its transition probability $p_{(i,j)(j,k)}\triangleq\mathbb{P}\{Z_{t+1}=(j,k)\mid Z_t=(i,j)\}$ is given by

$$p_{(i,j)(j,k)} = \begin{cases} p_j & \text{if } (j,k) \in \mathcal{S} \text{ and } k = i, \\ \frac{1-p_j}{d_j-1} & \text{if } (j,k) \in \mathcal{S} \text{ and } k \neq i, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

for each $(i,j) \in \mathcal{S}$ and $d_j > 1$. If $d_j = 1$, then $p_{(i,j)(j,i)} = 1$ for $i \in N(j)$; otherwise, zero. Note that $p_{(i,j)(l,k)} = 0$ if $j \neq l$. In addition, its initial distribution is given by

$$\mathbb{P}\{Z_0 = (i,j)\} = \mathbb{P}\{X_0 = j\} \cdot 1/d_j,$$

for each $(i,j) \in \mathcal{S}$, where $\mathbb{P}\{X_0 = j\}$ will be specified later.

Let $\boldsymbol{\pi} \triangleq [\pi_{(u,v)}, (u,v) \in \mathcal{S}]$ be the stationary distribution of $\{Z_t\}$ on $\mathcal{S}$ and $\pi_A \triangleq \sum_{(u,v)\in A} \pi_{(u,v)}$ be the probability of $\{Z_t\}$ being in a subset $A \subseteq \mathcal{S}$ in the steady-state. For each $j \in \mathcal{N}$, let $A_j \triangleq \{(i,j) \in \mathcal{S} : i \in N(j)\}$ be the set of directed edges incident to (and directed toward) node j. Note that $\{A_j\}_{j\in\mathcal{N}}$ forms a partition of $\mathcal{S}$ and $|A_j| = d_j$ in the original undirected graph G. Thus, it follows that π_{A_j} is the probability of the p-BRW being at node j in the steady-state, as the walk has to traverse one of those edges in A_j to reach node j. We now have the following result.

Theorem 2: For any choice of $p_j \in [0,1),\ j \in \mathcal{N}$, the stationary distribution of $\{Z_t\}$ is uniform over $\mathcal{S}$, i.e., $\pi_{(u,v)} = \frac{1}{2m}$ for all $(u,v) \in \mathcal{S}$. Consequently, we also have

$$\pi_{A_j} = \sum_{(u,v)\in A_j} \pi_{(u,v)} = d_j/2m.$$

Proof: See our technical report [10]. ∎

Theorem 2 says that $\{Z_t\}$ of the p-BRW has the same uniform stationary distribution on $\mathcal{S}$ *regardless of* the amount of backtracking at each node j, or equivalently, the stationary distribution is *invariant* with respect to $\{p_j\}_{j\in\mathcal{N}}$. In particular, the steady-state probability of the p-BRW being at node j, $\pi_{A_j} = d_j/2m$, is proportional to the degree of node i, which is the same as that of SRW. This allows us to freely choose $\{p_j\}$ as desired while keeping their stationary distribution the same as if the walk is the SRW on the same graph.

We suppose that the Markov chain $\{Z_t\}$ is a stationary Markov chain, i.e., Z_0 is chosen from the stationary distribution $\boldsymbol{\pi}$ ($\mathbb{P}\{Z_0 = (u,v)\} = \pi_{(u,v)} = 1/2m$). This is equivalent to assuming that a p-BRW starts at node v with the steady-state probability π_{A_v} ($\mathbb{P}\{X_0 = v\} = \pi_{A_v}$). Then, from Theorem 2, we establish an *invariance* property for the *mean return time* of p-BRW to node $j \in \mathcal{N}$. Here, the mean return time to node j, denoted as $\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\}$, is the average number of (discrete) time steps required for the p-BRW starting at node j to return to j. In addition, we show a relationship between the *mean first hitting time* of p-BRW to node j from a stationary start and the first two moments of return time of p-BRW to node j, which will later be a basis for the control of p_j. By the mean first hitting time to node j from a stationary start, denoted as $\mathbb{E}_{\boldsymbol{\pi}}\{T_{A_j}\}$, we mean the average time steps taken for the p-BRW to reach node j when the p-BRW starts at node v with probability π_{A_v}. We also denote by $\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\}$ the second moment of return time to node j. Due to space constraints, we refer to our technical report [10] for the rigorous definitions and mathematical arguments. Then, we have

Proposition 1: For any choice of $p_j \in [0,1), j \in \mathcal{N}$, we have

$$\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\} = \frac{1}{\pi_{A_j}} = \frac{2m}{d_j}, \text{ and} \quad (5)$$

$$\mathbb{E}_{\boldsymbol{\pi}}\{T_{A_j}\} = \frac{1}{2}\frac{\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\}}{\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\}} - \frac{1}{2} = \frac{d_j}{4m}\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\} - \frac{1}{2}. \quad (6)$$

Proof: See our technical report [10]. ∎

Proposition 1 implies that the mean return time of p-BRW to node j is *invariant* regardless of the values of backtracking probabilities $\{p_j\}_{j\in\mathcal{N}}$. Moreover, the mean first hitting time of p-BRW from a stationary start to node j ($\mathbb{E}_{\boldsymbol{\pi}}\{T_{A_j}\}$), or the average delay of a packet generated from randomly chosen source to destination j under p-BRW, depends only on the first two moments of return time of p-BRW to node j.

C. How To Choose Each Backtracking Probability p_i?

From (6), observe that in order to reduce the mean first hitting time or the average delay of a packet, we need to choose each backtracking probability p_i such that the second moment of return time to node j ($\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\}$) gets smaller whenever possible. Unfortunately, however, computing the second or any higher moment of the return to a node in a closed form is extremely difficult even for a SRW on general graphs [1]. Even worse, the sequence of visited nodes under the p-BRW, $\{X_t\}$, itself is not a Markov chain. Nonetheless, we demonstrate below that it is still possible to 'shape' the distribution of the return time toward smaller second moment by resorting to the above invariance result and suitably choosing $\{p_j\}$.

For notational convenience, we first denote the return time of p-BRW to node j as R_j, where X_t is the location of p-BRW on $\mathcal{N}$ at time $t \geq 0$. Then, we obtain the following:

$$\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\} = \mathbb{E}\{R_j\} = \sum_{t=1}^{\infty} \mathbb{P}\{R_j \geq t\} = \frac{2m}{d_j}, \text{ and} \quad (7)$$

$$\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\} = \mathbb{E}\{R_j^2\} = \sum_{t=1}^{\infty} 2t \cdot \mathbb{P}\{R_j \geq t\} - \mathbb{E}\{R_j\}, \quad (8)$$

where the last equality in (7) is from (5). While the precise relationship between $\{p_j\}_{j\in\mathcal{N}}$ and $\mathbb{P}\{R_j \geq t\}$ for all t is beyond reach, we can still *locally* control the shape of $\mathbb{P}\{R_j \geq t\}$ for small t, which in turn affects $\mathbb{P}\{R_j \geq t\}$ for large t as well via the invariance property – the total sum $\sum_{t=1}^{\infty} \mathbb{P}\{R_j \geq t\}$ in (7) does not depend on the choice of $\{p_j\}$. To be precise, as each p_j gets smaller, the p-BRW at the current node is more unlikely to return to the previously visited node over the next few slots, implying that $\mathbb{P}\{R_j \geq t\}$ for small t will be larger and thus $\mathbb{P}\{R_j \geq t\}$ for large t will be smaller. In view of (8), this is always more advantageous toward smaller second moment of the return time $\mathbb{E}\{R_j^2\}$. In [10], we also provide numerical results for a 2-D torus case to support our argument and see the speed-up factor.

D. From p-Backtracking Random Walk To Smart Sleep

In Section III, we demonstrated that the transition behavior of a packet of interest over 1-D ring is exactly the same as the CRW on 1-D ring (see Theorem 1). Note that if G is 1-D ring, the CRW is a special case of p-BRW where the backtracking probability is $p_j = 1-\tilde{p} = 0.5(1-q^2)^T$ for all j. Similarly as was done in Section III, we can show that for suitably chosen values of T, the p-BRW can well approximate the transition behavior of each packet over a general graph G in Smart Sleep, and thus all the previous results implying faster diffusion of packet under Smart Sleep still hold. For more details, we refer to our technical report [10].

Necessary condition for the sleep duration T***:*** It is important to choose the sleep duration T to be long enough so that the packet that caused the sleep won't backtrack for a while, but at the same time short enough such that the sensor can resume its normal mode before another packet comes in. To capture this idea precisely, let τ be the interval between two consecutive packet arrivals to sensor i. When there are multiple packets in the network, chances are that these two packets have different IDs. Then, the above argument leads to $T \approx \mathbb{E}\{\tau\}$. See Fig. 2 for illustration.

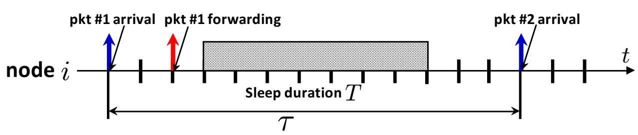


Fig. 2. A relationship between the sleep duration T and the interval τ between two consecutive packets arriving to sensor i.

To capture the inter-dependency among τ, T, and other network parameters, we define by Λ the total aggregate packet arrival rate into the whole network, and by $D(T)$ the average packet delay under Smart Sleep with parameter $T \geq 0$. (This way, $D(0)$ is the average packet delay via SRW-based forwarding.) Then, by viewing Λ and $D(T)$ as the exogenous arrival rate into the system (network) and the waiting time of each packet in the system, respectively, we see from Little's Law that $\Lambda D(T)$ is the average number of packets in the network in the steady-state, where $\Lambda D(0) < n$ to ensure system stability or from light-traffic load condition.† Consider a randomly tagged sensor node i. From the invariance property in Proposition 1 (see (5)), the mean return time of a packet to node i (in the number of packet forwardings) is inversely proportional to the stationary probability of being at node i. If the network of n nodes is roughly regular and if we let $\mathbb{E}\{\tilde{W}\}$ be the average sojourn time of the packet at each node (given that successively arrived different packets do not affect each other), then the actual mean return time to i (in the number of time slots) is $n\mathbb{E}\{\tilde{W}\}$. Since there are $\Lambda D(T)$ number of packets in the network on average, we arrive at

$$T \approx \mathbb{E}\{\tau\} \approx \frac{n\mathbb{E}\{\tilde{W}\}}{\Lambda D(T)} = \frac{\mathbb{E}\{\tilde{W}\}}{\lambda D(T)}, \tag{9}$$

where $\lambda := \Lambda/n$ is the exogenous packet arrival rate (as a source) *per each sensor node*. Note that obtaining a closed-form expression of $D(T)$ under a general topology and multiple packets would entail rigorous analysis of interacting non-Markovian processes on a general graph, which is clearly beyond the scope of this paper. Still, we find that (9) is informative, and in particular, we can show that the optimal sleep duration T^* in minimizing packet delay is achieved roughly when $T^* \approx \mathbb{E}\{\tau\}$ via extensive simulation results. This confirms our argument that each sensor can stay asleep as much as possible to prevent the return of the same previous packets while not holding off the delivery of other upcoming packets. We also find (9) practically useful in implementing distributed algorithms in which each sensor only needs to independently estimate $\mathbb{E}\{\tau\}$ based on two consecutive incoming packets with different IDs and self-adjust T on the fly, which we leave as a future work.

Due to space limit, the readers are referred to our technical report [10] for simulation results to demonstrate significant performance improvement of Smart Sleep under a large range of T and to confirm our argument on how to choose T^*.

†The performance of SRW-based algorithms [3], [12], [5], [7] is typically measured based on the delivery of a single packet, i.e., $\Lambda D(0) \approx 1$.

V. Conclusion

We have demonstrated that *Smart Sleep*, a simple modification of random duty cycling, can overcome the slow-mixing problem of SRW-based forwarding in randomly duty-cycled WSNs while achieving more power-saving at each sensor. To analytically address the packet dynamics in Smart Sleep, we introduce p-BRW and establish several properties of p-BRW to explain why Smart Sleep leads to smaller delay of each packet over the network. Numerical simulations confirm our reasonings and reveal that Smart Sleep can be made very robust while yielding superior performance. We expect that our reasoning behind Smart Sleep and p-BRW for faster delivery can be also applicable to many other SRW-based algorithms in general networks beyond WSNs.

References

[1] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*. monograph in preparation.
[2] N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin, "Non-backtracking random walks mix faster," *Communications in Contemporary Mathematics*, vol. 9, no. 4, pp. 585–603, 2007.
[3] 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, April 2004.
[4] S. Bandyopadhyay, E. J. Coyle, and T. Falck, "Stochastic properties of mobility models in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 11, pp. 1218–1229.
[5] P. Basu and C.-K. Chau, "Opportunistic forwarding in wirelss networks with duty cycling," in *Proc. of CHANTS*, San Francisco, CA, Sep. 2008.
[6] H. Cai and D. Y. Eun, "Toward stochastic anatomy of inter-meeting time distribution under general mobility models," in *Proc. of ACM MobiHoc*, Hong Kong SAR, China, May 2008.
[7] C.-K. Chau and P. Basu, "Exact analysis of latency of stateless opportunistic forwarding," in *Proc. of IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
[8] F. Chen, L. Lovász, and I. Pak, "Lifiting markov chains to speed up mixing," in *Proc. of ACM STOC*, Atlanta, GA, May 1999.
[9] S. Kim, C.-H. Lee, and D. Y. Eun, "Super-diffusive behavior of mobile nodes and its impact on routing protocol performance," *IEEE Trans. on Mobile Computing*, vol. 9, no. 2, pp. 288–304, 2010.
[10] C.-H. Lee and D. Y. Eun, "*Smart sleep*: sleep more to reduce delay in duty-cycled wireless sensor networks," Tech. Rep., Nov. 2010, http://www4.ncsu.edu/~dyeun/pub/techrep-smartsleep.pdf.
[11] W. Li and H. Dai, "Accelerating distributed consensus via lifting markov chains," in *Proc. of IEEE ISIT*, Nice, France, June 2007.
[12] I. Mabrouki, X. Lagrange, and G. Froc, "Random walk based routing protocol for wireless sensor networks," in *Proc. of InterPerf*, Nantes, France, Oct. 2007.
[13] S. Shakkottai, "Asymptotics of query strategies over a sensor network," in *Proc. of IEEE INFOCOM*, Hong Kong SAR, China, March 2004.