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
Email: {clee4, dyeun}@ncsu.edu

July 2010

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

Random walks have been extensively studied in many disciplines and used as efficient solutions for a wide range of applications in various types of networks such as sampling and file searching in peer-to-peer (P2P) networks [10, 16], routing/forwarding and query process in wireless sensor networks (WSNs) [3, 4, 15, 6, 8], and source-location privacy in WSNs [11], to name a few. The wide-spread popularity and satisfying performance of the random walks-based algorithms are mainly due to their inherent distributed nature and several desirable properties including simplicity of implementation and deployment, scalability, robustness to topology changes. Unlike typical topology-driven algorithms (e.g., shortest path-based or cluster-head based), random-walk based algorithms tend to achieve load-balancing autonomously over the network even in a dynamic setting, which helps avoid critical points of failure and non-uniform energy depletion in WSNs caused by hot-spot formation or congested areas [3, 17].

A simple random walk (SRW), among many other variants in the literature, has been mainly used as an effective packet forwarding method in WSNs without duty cycling [3, 15] and in randomly duty-cycled WSNs [6, 8], where each packet is forwarded from a node to one of its neighbors chosen uniformly at random (in a SRW fashion). 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 [7, 12], 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 [9, 13, 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. As typical WSNs consist of low-cost, low-power sensor nodes with very limited computational capability, it is crucial that sensor nodes avoid such information exchanges through multi-hop communications, not to mention the benefit of smaller delay and power saving for longer network life.

Specifically, in order to overcome the slow diffusion of SRW-based forwarding while saving more power at every sensor, in this paper, we propose a simple yet effective modification, named *Smart Sleep*, on the random duty cycling. Smart Sleep operates as follows: whenever each sensor 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 sensor 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 which the walker moves to one of its neighbors uniformly at random, in p-BRW, the random walker (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 of the stationary distribution of the packet position, we study how to choose the sleep duration T toward better shaping of the distribution of the return time to our advantage, leading to 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 independent 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, while retaining all the aforementioned desirable properties of random walks-based algorithms for dynamic networks.

The rest of this paper is organized as follows. Section II gives preliminaries on network model and base setup for networking operations including opportunistic forwarding and duty cycling. In Section III, we present Smart Sleep protocol and discuss its behavior on a simple topology. In Section IV, we describe a class of p-backtracking random walks along with its properties and analyze delay performance of Smart Sleep under general network settings. After presenting representative simulation results in Section V, we conclude in Section VI.

II Preliminaries

II-A Network Model

Consider n sensor nodes placed on a graph (or network) in which each edge corresponds to a reliable communication link between each pair of n nodes, if exists. Let $G = (\mathcal{N}, \mathcal{E})$ be a graph where $\mathcal{N} = \{1, 2, \dots, n\}$ is a set of sensor nodes and $\mathcal{E}$ is a set of edges with $|\mathcal{E}| = m$. 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., $(i, j) \in \mathcal{E}$ $(i \neq j)$. 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 $i \in \mathcal{N}$ and by d_i the degree of each node i, i.e., $d_i = |N(i)|$. Note that $\sum_{i \in \mathcal{N}} d_i = 2m$.

II-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 [6, 8] as a packet forwarding method, both of which will be used as a base setup for networking operations in this paper. Note that this combination of the opportunistic forwarding and random duty cycling is originally introduced in [8].

We consider a network operating in a synchronous mode, as assumed in [3, 4, 15, 6, 8]. 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 completely 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 [8]. This can be achieved via the exchange of RTS (request to send) and CTS (clear to send) with random waiting time as will be explained below, while this (practical) issue has been ignored in [8].

The (data) packet transmission/reception between two adjacent nodes can occur, only when they both wake up and each of them is aware that the other node is also 'on'. Hence, whenever any node having a packet wakes up at a certain time slot, it transmits a RTS packet to notify its existence. If any of its neighbors wakes up and receives the RTS packet at the same slot, it transmits a CTS packet to acknowledge the reception of the RTS packet to the sender. Here, since it is still possible for many nodes to be awake at the same slot as mentioned before, each node waits for a short random amount of time before transmitting a RTS or CTS packet, while performing carrier sense on the channel, to avoid any possible collisions and to ensure successful packet transmission/reception.

Specifically, during the random waiting time, if an awake node having a (data) packet senses that the channel is busy, then it cancels RTS transmission and seeks next opportunities. If it correctly receives an RTS packet from its neighbor, then it rather prepares to receive a data packet from the sender and waits for another short random time to transmit a CTS packet to the sender. On the other hand, if any node receives a CTS packet from another neighbor during its random wait till its own CTS transmission, it goes to sleep. In this way, even if multiple nodes wake up at the same 'slot' and have the same purpose (waiting for transmission or reception of a data packet), we can randomly break the tie. We here suppose that the duration of each time slot is suitably chosen

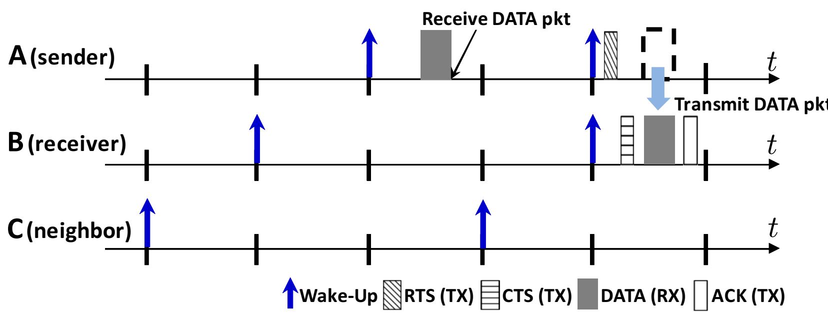


Figure 1: The operation of data transmission via the opportunistic forwarding under the random duty cycling. Here, nodes B and C are the neighbors of node A, and a (data) packet is transmitted from node A to node B.

so as to accommodate all these signallings required for the data transmission. We also suppose that traffic load in the network is light as assumed in [3, 4, 15, 6, 8]. Hence, the interference from concurrent data transmissions is not a critical issue, though the above RTS/CTS exchange with random waiting time can reduce the effect of such interference on the system performance. Fig. 1 depicts an example of the operation of data communication through the opportunistic forwarding with random duty cycling.

II-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.[1] 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 [8], for each node $i \in \mathcal{N}$ having a packet,

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

if $k \in N(i)$; otherwise, zero. In other words, the transition of each packet from a node to one of its neighbors over G is done in a SRW fashion.

Observe that at each time slot, node i independently wakes up with probability q and there is at least one awake neighbor with probability $1-(1-q)^{d_i}$ at each slot. In other words, 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

[1]While the opportunistic forwarding can also work with a variant of the random duty cycling, called a pseudo-random duty cycling [6, 8], the transition of each packet over network G here is still done in a SRW fashion.

time slot. Thus, the waiting time W_i is geometrically distributed with mean

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

Therefore, the transition of each packet from node i to one of its neighbors through the opportunistic forwarding follows a SRW, with heterogeneous random sojourn time [6, 8], where the heterogeneity comes from varying degrees d_i, $i \in \mathcal{N}$.

Note that the opportunistic forwarding inherently inherits all the properties of random walks-based algorithms such as simplicity, scalability, load-balancing, and robustness to topology changes. It also operates without any topological or geographical information. In this paper, our goal is to demonstrate that it's always possible to further improve the performance of opportunistic forwarding in *both power and delay together* under the same setting as random walk-based algorithms, retaining all the aforementioned desirable properties while at the same time overcoming the problem of slow mixing (or diffusion) associated with all simple random walk-based algorithms in WSNs.

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 each 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 sensor nodes would sleep for a quite 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 G. 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, for the rest of this section, we 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 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), as every node initially performs the random duty cycling. Then, suppose that node i forwards the packet to node j at time (slot) t' as shown in Fig. 2. From this time on, the packet will stay at i for some random time until forwarding to one of its first awake neighbors. Since node i will be in a sleep mode during $[t', t'+T]$, the packet will be more likely to be forwarded to node k rather than back to i. In what follows, we make this argument precise by computing the probability of forwarding to k.

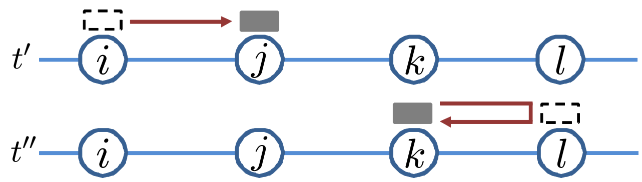


Figure 2: A series of four nodes in 1-D ring

From the definition of random duty cycling, nodes j and k both wake up at each slot with probability q^2. Thus, from time t' onwards, the number of time slots Y until both nodes are awake at the same slot (for possible transmission and/or reception of the packet) is geometrically distributed with mean $1/q^2$. Then, by noting that node i remains asleep until time $t'+T$, the probability that the packet is forwarded to node k (from node j) *no later* than time $t'+T$ is $\mathbb{P}\{Y \le T\}$. If the packet has not been yet forwarded to node k by time $t'+T$, which happens with probability $\mathbb{P}\{Y > T\}$, then from that time on both of nodes i, k are in normal mode and equally likely to receive the packet as in (1). Hence, the probability that the packet is forwarded to node k *after* $t'+T$ is $0.5 \cdot \mathbb{P}\{Y > T\}$. Therefore, the (total) probability that the packet will be eventually forwarded to node k is

$$\tilde{p} \triangleq \mathbb{P}\{Y \le T\} + 0.5 \cdot \mathbb{P}\{Y > T\} = 1 - 0.5(1-q^2)^T > 0.5, \tag{3}$$

and the probability that the packet will be forwarded back to node i is $1-\tilde{p} = 0.5(1-q^2)^T < 0.5$. Similarly, once the packet is forwarded from node j to node k with probability $\tilde{p}$, it will be further forwarded to neighbor l (rather than back to node j) with probability $\tilde{p}$. This operation continues until either the packet reaches its destination or it is not transmitted back to the node that previously forwarded the packet.

Now consider the case of packet being transmitted back to the previous sender. Specifically, the packet, which was forwarded to node l, is now being transmitted back to node k at time (slot) $t'' > t'$ as depicted in the bottom of Fig. 2. This transition is possible only when node k is awake, i.e., its sleep time T (due to forced sleep mode from previous forwarding to l) must have passed by time t''. Thus, at t'', all nodes except node l are in a normal mode. Hence, once the packet is transmitted from node l back to node k which previously forwarded the packet, the transition of the packet over 1-D ring behaves as if the packet is newly generated at node l (in the network where all nodes in a normal mode) and is then forwarded to node k.

From the above observation, one can see that for any node having the packet of interest, its neighbor that previously forwarded the packet is in a sleep mode, while one another neighbor is in a normal mode. Hence, the packet of interest keeps being forwarded to the nodes in the same direction that the packet has followed, with probability $\tilde{p} > 0.5$; otherwise, with probability $1-\tilde{p}$, the packet is forwarded to the node which previously forwarded the packet in a reverse direction. This is exactly the same as the transition of a correlated random walk (CRW) [5, 7] 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 larger than 0.5; otherwise, it changes the direction and moves to its another neighbor (or the previously visited vertex). In particular, [7] 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. Observe that $\tilde{p}$ is increasing in T as shown in (3). Hence, 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 a packet reaches its destination.

However, we should not push this direction indefinitely. As mentioned before, if there are many other packets from different sources or from a single source, many sensor nodes will be unavailable for delivery of other packets for longer duration as T gets very large, and thus the reduction in the number of packet forwardings in the presence of multiple packets is not certain. More importantly, the CRW[2] is a discrete-time random walk where the walk spends one time unit in every node. However, in our setup, while the transition of each packet over 1-D ring behaves as if it follows a CRW, the sojourn time at each node is no longer a constant. We next derive the average sojourn time at each node after packet forwarding as a function of T.

When a packet of interest is initially generated at node i and all nodes are in a normal mode, the average waiting (sojourn) time for the packet at node i is simply given by (2). However, once the packet is forwarded to one of neighbors of node i, the average sojourn time will be different. Let $\tilde{W}_i$ be the sojourn time of the packet at node i (in the number of time steps) until being further forwarded, given that node i has just received the packet from one of its neighbors. Then, it follows that

$$\begin{aligned}\mathbb{E}\{\tilde{W}_i\} &= \mathbb{E}\{Y|Y \leq T\}\mathbb{P}\{Y \leq T\} + (T + \mathbb{E}\{W_i\})\mathbb{P}\{Y > T\} \\ &= \frac{1}{q^2} - \frac{1}{q^2}\frac{(1-q)}{(2-q)}(1-q^2)^T. \end{aligned} \tag{4}$$

The first equality can be obtained by conditioning on whether a neighbor of node i in a normal mode will receive the packet within T slots during which the other neighbor stays in a sleep mode.

[2] In [5, 7], the CRW was studied in the context of mobility modeling for applications of MANETs.

Due to the symmetry for 1-D ring, the average sojourn time does not depend on i. We refer to Appendix A for the detailed derivation of (4). Here, observe that $\mathbb{E}\{\tilde{W}_i\}$ is increasing in T starting from $\mathbb{E}\{\tilde{W}_i\}=\mathbb{E}\{W_i\}$ for the case of $T=0$. In other words, as the sleep duration 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. In addition, 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 (4). 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 our more in-depth discussion in Sections IV-D and V as to how to suitably choose T in Smart Sleep under general network topologies.

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

In the previous section, we have shown that the transition of a packet of interest on 1-D ring caused by Smart Sleep is the same as that of CRW on 1-D ring. In particular, the packet of interest is less likely forwarded back to the node which previously forwarded the packet. 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.

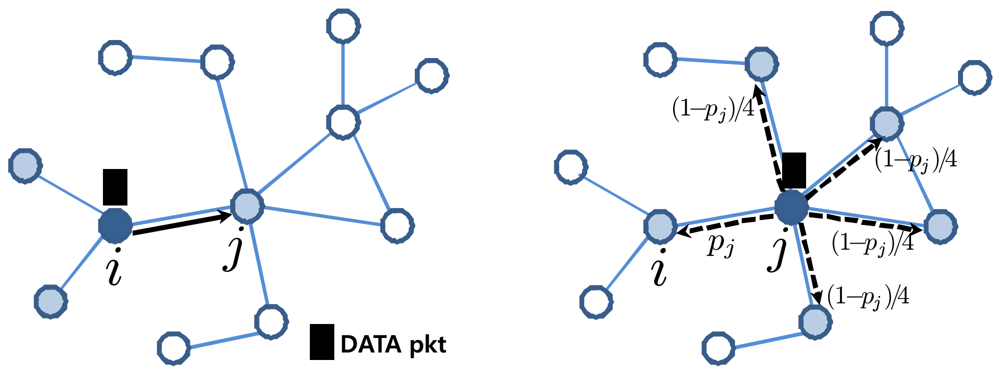
IV-A p-Backtracking Random Walk

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 the current 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

$$\mathbb{P}\{X_{t+1}=k|X_t=j,X_{t-1}=i\} = \begin{cases} p_j & \text{if } k=i, \\ \frac{1-p_j}{d_j-1} & \text{if } (j,k)\in\mathcal{E}, k\neq i, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

for $d_j>1$ and $(i,j)\in\mathcal{E}$ $(i\neq j)$. 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 $j\in\mathcal{N}$ with $d_j>1$. Fig. 3 depicts an example of possible transitions of p-BRW from a node to one of its neighbors.



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

Figure 3: 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 (5) as the next node that it will move to.

The p-BRW includes the following random walks as special cases. If $p_j=1/d_j$ in (5), then the p-BRW reduces to the SRW where the next node is chosen uniformly at random from the neighbors of the current node, i.e.,

$$\mathbb{P}\{X_{t+1}=k|X_t=j\}=\mathbb{P}\{X_{t+1}=k|X_t=j,X_{t-1}=i\}=\frac{1}{d_j},$$

for $(j,k)\in\mathcal{E}$; otherwise, zero. If $p_j=0$ for all node j with $d_j>1$, then the p-BRW reduces to the non-backtracking random walk (NBRW) [2] where the walk at the current node always moves to one of its neighbors except the previously visited node with equal probability if there is any other neighbor.[3] Moreover, if G is 1-D ring and $p_j=p$ $(0<p<1)$ for all j, then the p-BRW reduces to the CRW [5, 7] in which the walk continues in the same direction with probability $1-p$; otherwise, it changes the direction with probability p. When G is 2-D grid, however, the p-BRW is slightly different from the CRW on the same 2-D grid where the walk keeps the same direction (east, north, west, or south) with a certain probability; otherwise, it changes the direction to one of three remaining directions. We note that the CRW can be defined only on a grid structure [5, 7]

[3] In [2] the NBRW is considered only for regular graphs with $d_i=d>3$ for all $i\in\mathcal{N}$.

since the notion of 'direction' becomes unclear on a non-grid topology. In contrast, the p-BRW can be defined on any general topology without such restriction, where its backtracking probability p_i at node i describes a *relative* direction to the previously visited node.

IV-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 (5). 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, $\{Z_t\}_{t\geq 1}$ becomes a Markov chain on the state space $\mathcal{S}$ and its transition probability is given by

$$p_{(i,j)(j,k)} \triangleq \mathbb{P}\{Z_{t+1} = (j,k) \mid Z_t = (i,j)\} = \begin{cases} p_j & \text{if } (j,k) \in \mathcal{S} \text{ and } k = i, \\ \frac{1-p_j}{d_j - 1} & \text{if } (j,k) \in \mathcal{S} \text{ and } k \neq i, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

for each $(i,j) \in \mathcal{S}$ and $d_j > 1$. If $d_j = 1$ for any $j \in \mathcal{N}$, then $p_{(i,j)(j,i)} = 1$ for $i \in N(j)$; otherwise, zero. Note also that $p_{(i,j)(l,k)} = 0$ if $j \neq l$. From the definition of p-BRW, we have

$$\mathbb{P}\{Z_1 = (j,k)\} = \mathbb{P}\{X_0 = j\}\frac{1}{d_j} \tag{7}$$

for each $(j,k) \in \mathcal{S}$, though $\mathbb{P}\{X_0 = j\}$, $j \in \mathcal{N}$, will be specified later. Here, without loss of generality, we can assume

$$\mathbb{P}\{Z_0 = (i,j)\} = \mathbb{P}\{X_{-1} = i, X_0 = j\} = \mathbb{P}\{X_0 = j\}\frac{1}{d_j}, \tag{8}$$

since for any $(j,k) \in \mathcal{S}$

$$\mathbb{P}\{Z_1 = (j,k)\} = \sum_{(i,j)\in\mathcal{S}} \mathbb{P}\{Z_0 = (i,j)\} p_{(i,j)(j,k)} = \mathbb{P}\{X_0 = j\}\frac{1}{d_j}.$$

where the first equality is from conditioning and the second one follows from (6) and (8). Therefore, $\{Z_t\}_{t\geq 0}$ is now a well-defined Markov chain on the state space $\mathcal{S}$ with its initial distribution $\mathbb{P}\{Z_0 = (i,j)\}$ given by (8).

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 1. *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*

$$\pi_{A_j} = \sum_{(u,v)\in A_j} \pi_{(u,v)} = \frac{d_j}{2m}. \tag{9}$$

Proof. See Appendix B. □

Theorem 1 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 (d_j) 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. This invariance property is a fundamental building block based on which we can obtain other properties of p-BRW and develop methodology as to how to set the backtracking probability p_j for each j, so as to achieve faster diffusion of p-BRW on G and correspondingly smaller delay in our Smart Sleep protocol.

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 the p-BRW on $\mathcal{N}$ starts from its stationary distribution. To see this, observe that $\mathbb{P}\{X_0 = v\} = \pi_{A_v}$ together with (8) and (9) gives $\mathbb{P}\{Z_0 = (u,v)\} = 1/2m$. Now, we define the following two stopping times (or the first hitting times) of $\{Z_t\}$ to the subset A_j:

$$T^+_{A_j} \triangleq \min\{t > 0 : Z_t \in A_j\}, \text{ and } T_{A_j} \triangleq \min\{t \geq 0 : Z_t \in A_j\}.$$

Here, one difference between these two is that the former does not count the case of $Z_0 \in A_j$, while the latter includes this case, i.e., $T_{A_j} = 0$ if $Z_0 \in A_j$. We then define the *mean return time* of p-BRW to node $j \in \mathcal{N}$ when starting at node j as

$$\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\} \triangleq \mathbb{E}\{T^+_{A_j} \,|Z_0 \in A_j\} = \sum_{(u,v)\in A_j} \mathbb{E}\{T^+_{A_j} \,|Z_0 = (u,v)\}\pi_{A_j}(u,v), \tag{10}$$

where $\pi_{A_j}(u,v) \triangleq \frac{\pi_{(u,v)}}{\pi_{A_j}} = \frac{1}{d_j}$ for $(u,v) \in A_j$, and the equality in (10) is from the fact that $\{Z_t\}$ is a stationary chain. This is the average number of (discrete) time steps required for a p-BRW starting at node j to return to j. Similarly, we define the second moment of return time of p-BRW to node j as $\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\} \triangleq \mathbb{E}\{(T^+_{A_j})^2 \,|Z_0 \in A_j\}$. In addition, we define the *mean first hitting time* of p-BRW to node $j \in \mathcal{N}$ from a stationary start as

$$\mathbb{E}_{\boldsymbol{\pi}}\{T_{A_j}\} \triangleq \sum_{(u,v)\in\mathcal{S}} \mathbb{E}\left\{T_{A_j} \,|Z_0 = (u,v)\right\}\pi_{(u,v)}. \tag{11}$$

Here, by the stationary start we mean that Z_0 is drawn from the stationary distribution of $\{Z_t\}$, or equivalently, the p-BRW starts at node v with its stationary distribution π_{A_v}. Note that $\{Z_t\}$

is already a stationary chain. Thus, by consulting the properties of a stationary Markov chain [1], we have

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

$$\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\} = \frac{1}{\pi_{A_j}} = \frac{2m}{d_j}, \quad \textit{and} \tag{12}$$

$$\mathbb{E}_{\boldsymbol{\pi}}\{T_{A_j}\} = \frac{1}{2}\frac{\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\}}{\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\}} - \frac{1}{2} = \frac{d_j}{4m}\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\} - \frac{1}{2} \tag{13}$$

for each $A_j,\ j \in \mathcal{N}$. □

Proof. For any (finite) stationary Markov chain, the following holds [1, Ch.2.,pp.20–21]:

$$\mathbb{E}_{\pi_A}\{T^+_A\} = \frac{1}{\pi_A}, \quad \text{and} \quad \mathbb{E}_{\boldsymbol{\pi}}\{T_A\} = \frac{1}{2}\frac{\mathbb{E}_{\pi_A}\{(T^+_A)^2\}}{\mathbb{E}_{\pi_A}\{T^+_A\}} - \frac{1}{2}$$

for any subset A of the state space. Hence, from Theorem 1, (12)–(13) immediately follows. □

Proposition 1 implies that the mean return time of p-BRW to node $j \in \mathcal{N}$ 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 the return time of p-BRW to node j.

IV-C How To Choose Each Backtracking Probability p_i?

From (13), observe that in order to reduce 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 even 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 invariance result in our Theorem 1 and suitably choosing $\{p_j\}$.

For notational convenience, we first denote the return time of p-BRW to node j as R_j, defined by

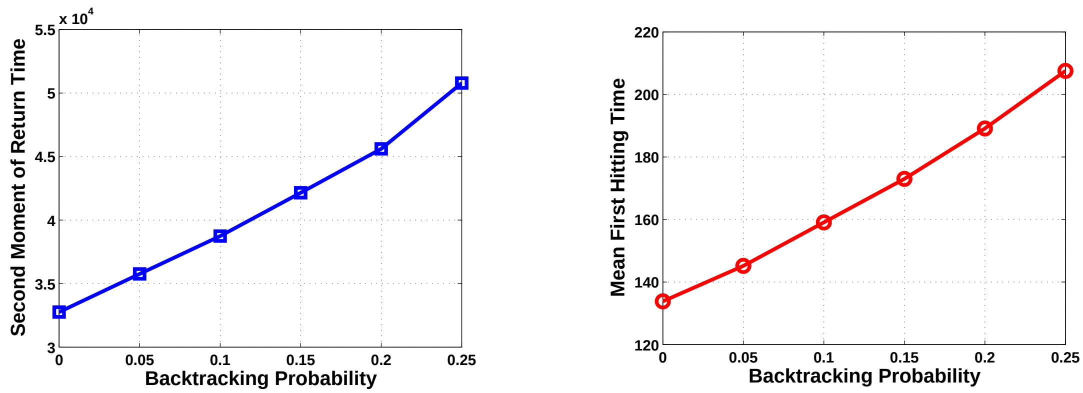
$$R_j \triangleq \min\{t > 0 : X_t = j | X_0 = j\}, \tag{14}$$

where X_t is the location of p-BRW on $\mathcal{N}$ at time $t \geq 0$. Note that from (10) and (14), we have

$$\mathbb{E}_{\pi_{A_j}}\{T^+_{A_j}\} = \mathbb{E}\{R_j\} = \sum_{t=1}^{\infty} \mathbb{P}\{R_j \geq t\} = \frac{2m}{d_j} \tag{15}$$

$$\mathbb{E}_{\pi_{A_j}}\{(T^+_{A_j})^2\} = \mathbb{E}\{R_j^2\} = \sum_{t=1}^{\infty} 2t \cdot \mathbb{P}\{R_j \geq t\} - \mathbb{E}\{R_j\}, \tag{16}$$

where the last equality in (15) is from (12). 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 (15) does not depend on the choice of p_j. To be precise, as the backtracking probability p_i 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 (16), this is always more advantageous toward smaller second moment of the return time $\mathbb{E}\{R_j^2\}$.



(a) Second moment of return time

(b) Mean first hitting time

Figure 4: Effect of different backtracking probabilities p on the second moment of return time to a given node and the mean first hitting time to that node under a 2-D torus with $n = 11 \times 11$.

Take a 2-D torus (a regular graph with $d = 4$) with n nodes for example. From Proposition 1, the mean return time to each node j is $\mathbb{E}\{R_j\} = n$. Let $p_i = p \in [0, 1)$ for all $i \in \mathcal{N}$. Then, from symmetry, $\mathbb{P}\{R_j \geq t\}$ does not depend on j anymore, so we can conveniently drop the subscript j from our notation for the return time R_j. Fig. 4 shows the second moment of return time $\mathbb{E}\{R^2\}$ and the mean first hitting time to any given node under a 2-D torus with $n = 11 \times 11$, empirically obtained via numerical simulations while varying p. As mentioned before, if $p = 0.25$, the p-BRW reduces to the SRW on a 2-D torus. Thus, we only consider the case of $0 \leq p \leq 0.25$ to see the behavior of p-BRW compared with that of the SRW. As seen from Fig. 4, the aforementioned argument holds. In other words, the second moment of return time to a given node is *increasing* in the backtracking probability p, and so is the mean first hitting time to that node (from a stationary start).

IV-D From p-Backtracking Random Walk To Smart Sleep

We explain how the p-BRW is related to the transition of a packet of interest on G under Smart Sleep protocol. Consider a packet of interest being forwarded from node i to node j. Node i then immediately goes to sleep for T slots while all the other neighbors of j (except node i), i.e., all nodes in $N(j)\backslash\{i\}$, are in a normal mode. Similarly as was done in Section III, it follows that the number of time slots Y_j until successful transmission from j to a node in $N(j)\backslash\{i\}$, is geometrically distributed with mean $1/q'_j$, where $q'_j \triangleq q(1-(1-q)^{d_j-1})$. Also, if the packet has not been forwarded to any node in $N(j)\backslash\{i\}$ within T slots (during which node i has been in a sleep mode), i.e., $Y_j > T$, then node i resumes its normal random duty cycling and the packet can be forwarded back to node i with probability $1/d_j$. Hence, the probability that the packet backtracks to node i is $\frac{1}{d_j}\mathbb{P}\{Y_j > T\} < \frac{1}{d_j}$ for $T > 0$. Consequently, the packet will be forwarded to a node in $N(j)\backslash\{i\}$ with probability $\frac{1}{d_j-1}(1-\frac{1}{d_j}\mathbb{P}\{Y_j > T\}) > \frac{1}{d_j}$. These transition probabilities are the same regardless of which node is the one that previously forwarded the packet to node j, provided that all neighbors of node j except the previous forwarder are in a normal mode. One can also see that this transition behavior is the same as that of p-BRW from a node j to one of its neighbors, where the backtracking probability at node j is $p_j = \frac{1}{d_j}\mathbb{P}\{Y_j > T\}$. Note that p_j becomes smaller for larger T, possibly leading to faster diffusion of the single packet over G.

In a general (non-tree like) graph G, situation is more subtle as there may exist several paths routed to each node, over which the packet can traverse before reaching its destination. Unlike the case of 1-D ring in Section III, when a packet of interest reaches node l, there might be multiple neighbors which are still in their sleep modes. Fig. 5 shows an example of this case with large sleep duration T, where the packet reaches node l after traversing path $i \to j \to k \to l$, only to realize that in addition to node k, node i is still in its sleep mode (caused by the packet itself earlier). The packet at node l now sees 'less-than-usual' transition probabilities to both nodes i and k, whereby p-BRW allows only one such case (different probability of backtracking to its previously visited node while all others are equally likely).

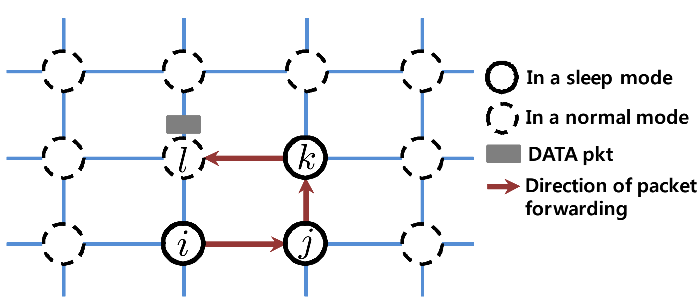


Figure 5: A snapshot of a part of 2-D grid topology when a packet of interest reaches node l via path $i \to j \to k \to l$. If T is very large, nodes i, j, k can be still in a sleep mode by the time the packet reaches l.

However, we maintain that this case will be unlikely with reasonably chosen (not too large) value of T. For example, in Fig. 5, observe that the average sojourn time of the packet at each node is lower bounded by $[q(1-(1-q)^4)]^{-1}$, achieved when $T=0$. Thus, if $T \leq 3[q(1-(1-q)^4)]^{-1}$, then the case in Fig. 5 will not arise. This is even more so, considering that the probability of the packet traversing over the path $i \to j \to k \to l$ (of length 3) is no larger than $1/3^3 \approx 0.038$, and it is even more unlikely with the increase of path-length. Thus, for suitably chosen values of T, the p-BRW can well approximate the transition behavior of the packet over a general graph in Smart Sleep, and all the previous results implying faster diffusion of packet under Smart Sleep still hold.

Necessary condition for the sleep duration T***:*** As mentioned before, too large values of T put many sensors consecutively inactive as relay nodes for the delivery of *other* packets for a long time, which in turn eventually leads to longer packet delay. On the other hand, too small values of T would render Smart Sleep protocol behave just like the usual SRW-based forwarding, thus losing all the benefit of faster diffusion and smaller delay in p-BRW. A moment of thought here thus suggests that the sleep duration T 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. 6 for illustration.

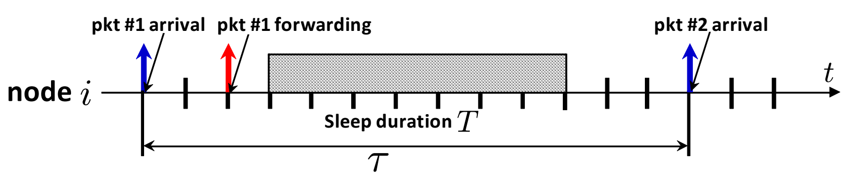


Figure 6: A relationship between the sleep duration T and the interval τ between two consecutive packets arriving to sensor i.

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.[4] Consider a randomly tagged sensor node i. From the invariance property in Proposition 1 (see (12)), the mean return

[4]The performance of SRW-based algorithms [3, 15, 6, 8] is typically measured based on the delivery of a single packet, i.e., $\Lambda D(0) \approx 1$.

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

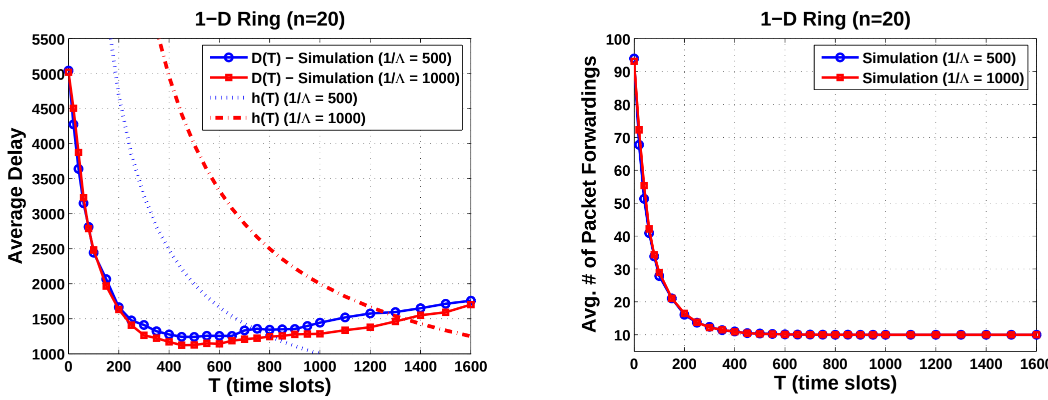
$$T \approx \mathbb{E}\{\tau\} \approx \frac{n\mathbb{E}\{\tilde{W}\}}{\Lambda D(T)} = \frac{\mathbb{E}\{\tilde{W}\}}{\lambda D(T)}, \tag{17}$$

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 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 (17) is informative, and in particular, we will show later in Section V 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 (17) 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.

V Numerical Simulations

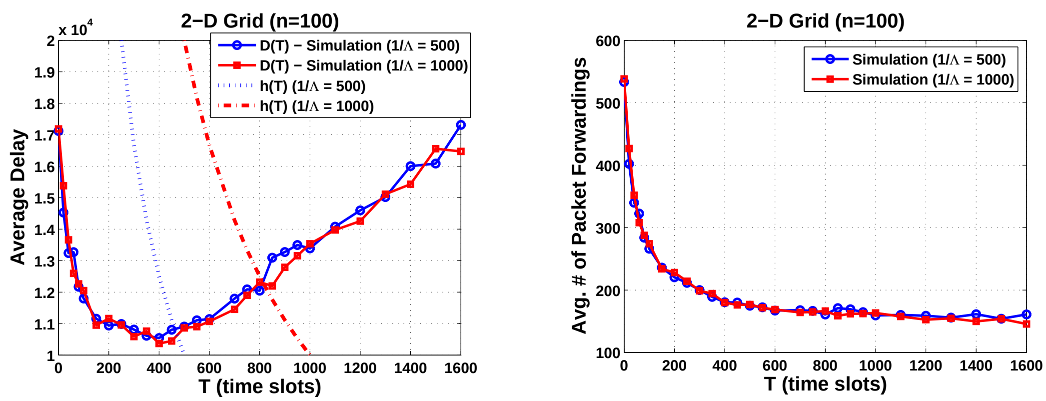
In this section, we present representative simulation results to demonstrate and quantify performance improvement of Smart Sleep under a large range of $T > 0$ compared with the SRW-based forwarding ($T = 0$). Our metrics of interest are packet delay, transmission cost (total number of packet forwardings), and the amount of power-saving of each sensor. We conduct simulations over 1-D ring, 2-D grid, and random geometric graph on our custom event-driven simulator using C++. The random geometric graph, denoted as $RGG(n, r)$, is a widely used graph in the literature [3, 4, 6, 8], in which n nodes are uniformly and independently located in a square and two nodes are connected if they are within distance of r.

We use the following common setups for simulations. First, as used in [6, 8], we measure the performance for the farthest source-destination pair under each instance of graph, where the length between two nodes is the number of hops over the shortest path connecting the two nodes. We test under two different arrival rates ($1/\Lambda$=500 or 1000 slots). Each simulation runs until 50 packets are delivered, and each data point reported here is the average of 300 independent simulations (i.e., average over 15000 packet deliveries). We also measure the average wake-up frequency per each sensor to quantify the amount of power-saving under Smart Sleep. In our scenario, we designate a sensor node as the destination simulating a situation without any powered sink [14, 6, 8], but we



(a) Average Delay $D(T)$ (b) Transmission cost

Figure 7: Performance comparison on the packet delivery for the farthest s-d pair while varying T under 1-D ring with $n = 20$.

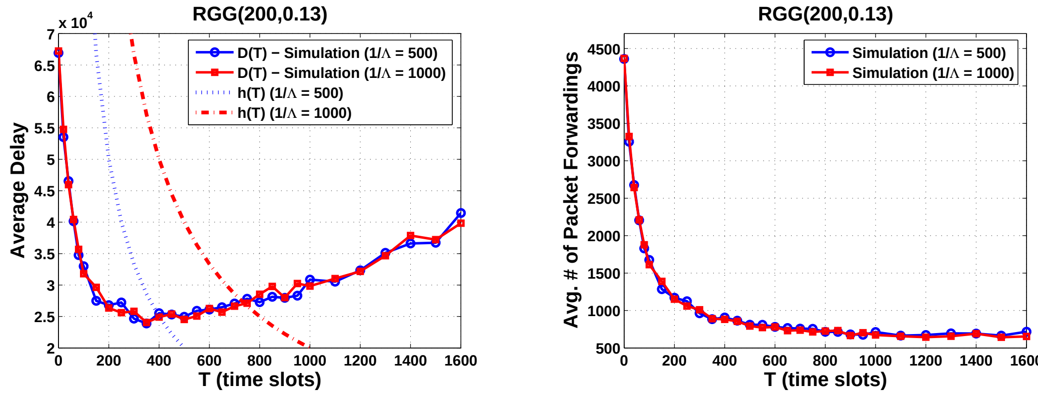


(a) Average Delay $D(T)$ (b) Transmission cost

Figure 8: Performance comparison on the packet delivery for the farthest s-d pair while varying T under 2-D grid with $n = 100$.

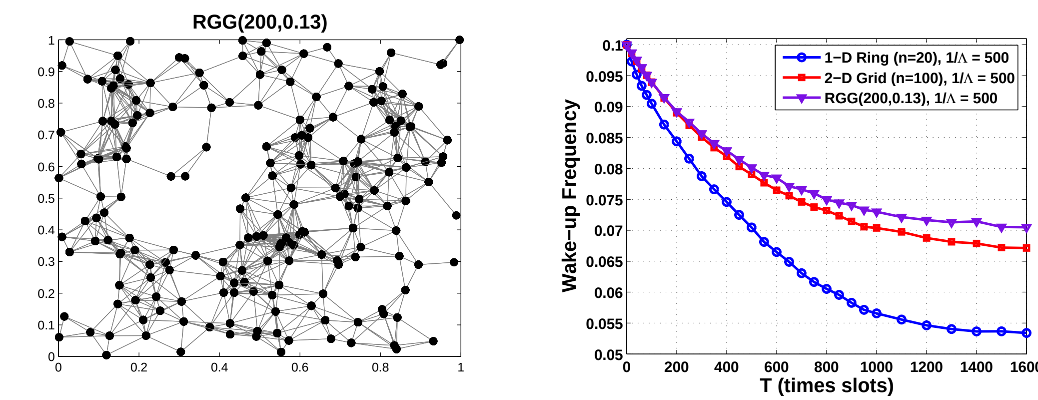
note that the main feature of Smart Sleep doesn't change even in the presence of powered sinks. Throughout the simulations, we set the default wake-up probability as $q = 0.1$. This can be easily programmed into each sensor before network deployment or reconfigured if necessary. Note that our focus here is to extend the sleep period after forwarding for any given q, not to optimally choose q under certain criteria, which is outside the scope of this paper.

Fig. 7 shows the average delay $D(T)$ and transmission cost under a 1-D ring with n=20. Clearly, Smart Sleep offers significant improvement (more than 70%) for both metrics for all $T \in [200, 1000]$ compared with SRW-based one ($T = 0$), though the average delay starts increasing slowly after around $T = 600$ slots. It implies that for $T \lessapprox 600$, the reduction in the number of total packet forwardings achieved through faster diffusion of each packet over the network (the hallmark of p-BRW) weighs more than the slight increase in the average sojourn time of the packet at every



(a) Average Delay $D(T)$ (b) Transmission cost

Figure 9: Performance comparison on the packet delivery for the farthest s-d pair while varying T under a sample graph of $RGG(200, 0.13)$.



(a) A sample topology of RGG (b) Wake-up frequency

Figure 10: (a) A sample topology of $RGG(200, 0.13)$; (b) Measured wake-up frequency for a sensor under various topologies as T varies.

node (see (4)). Note that the condition for the optimal T^* in (17) can be rewritten as

$$D(T) \approx \frac{\mathbb{E}\{\tilde{W}\}}{\lambda} \cdot \frac{1}{T} \triangleq h(T). \tag{18}$$

Note that the average sojourn time of a packet at each sensor $\mathbb{E}\{\tilde{W}\}$ is also a function of T. In Fig. 7(a), we overlay $h(T)$ where the average sojourn time $\mathbb{E}\{\tilde{W}\}$ here for 1-D ring can be computed from (4), and observe that the optimal T^* in minimizing the average delay is located roughly around the intersection of $D(T)$ and $h(T)$, confirming the validity of the condition in (17).

Figs. 8 and 9 show the results for a 2-D grid with $n = 100$ and for a sample topology of $RGG(200, 0.13)$ displayed in Fig. 10(a) while T varies, respectively, but now under two different values of Λ. For 2-D grid, the average delay reduces by around 30% and the transmission cost by 60% for $T \in [200, 600]$. For RGG, both performance metrics improve by more than 60% for $T \in [200, 700]$. Similarly as in Fig. 7(a), we also plot $h(T)$ in all cases. We here use $\mathbb{E}\{\tilde{W}\} = 1/q^2$, corresponding to the case of only one awake neighbor (under duty-cycling with probability q) while all other neighbors are in sleep mode, which is roughly an upper bound of the actual average sojourn time. In all these cases, the plotted $h(T)$ is thus an upper bound of the true values, and the actual intersection point with more accurate value of $\mathbb{E}\{\tilde{W}\}$ would be smaller than what is shown in the figures, suggesting that the optimal sleep duration T^* is again well approximated by the fixed point satisfying (17) or (18). Lastly, Fig. 10(b) shows the average wake-up frequency per each sensor to quantify the amount of power-saving through Smart Sleep measured during the simulation time for all the above topologies with $1/\Lambda = 500$. We see that the additional power-saving (compared with the SRW-based duty-cycling, i.e., $T = 0$) is around 20% for the estimated optimal values T^*. In other words, the observed 30% to 60% reduction in the average delay is achieved when each sensor turns off 20% more than the usual random duty-cycling. In addition, since the packet transmission itself also consumes power, the actual power-saving will be even greater by noting the reduction in the average number of packet forwardings in all cases. We also point out here that in all the simulations, the range of T values for which we enjoy great improvement both in delay and power is fairly broad, implying that our Smart Sleep protocol allows easy configuration and offers robust and superior performance even under inaccurately estimated T^*.

VI Conclusion

Throughout the paper, we have demonstrated that our Smart Sleep protocol, 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, a variation of random walk with past memory, 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, with high potential for distributed and autonomous implementation under dynamic environments. 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, 2004.

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

[5] 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.

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

[7] H. Cai and D. Y. Eun, “Toward stochastic anatomy of inter-meeting time distribution under general mobility models,” in *Proc. of ACM MobiHoc*, 2008.

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

[9] F. Chen, L. Lovász, and I. Pak, “Lifiting markov chains to speed up mixing,” in *Proc. of ACM STOC*, Atlanta, GA, 1999.

[10] C. Gkantsidis, M. Mihail, and A. Saberi, “Random walks in peer-to-peer networks,” in *Proc. of IEEE INFOCOM*, Hong Kong, China, 2004.

[11] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, “Enhancing source-location privacy in sensor network routing,” in *Proc. of IEEE ICDCS*, Columbus, OH, 2005.

[12] 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.

[13] W. Li and H. Dai, “Accelerating distributed consensus via lifting markov chains,” in *Proc. of IEEE ISIT*, Nice, France, 2007.

[14] 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.

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

[16] L. Massoulié, E. L. Merrer, A.-M. Kermarrec, and A. Ganesh, “Peer counting and sampling in overlay networks: random walk methods,” in *Proc. of ACM PODC*, Denver, CO, 2006.

[17] A. Mei and J. Stefa, “Routing in outer space: fair traffic load in multi-hop wireless networks,” in *Proc. of ACM MobiHoc*, Hong Kong SAR, China, 2008.

[18] S. M. Ross, *Stochastic processes*, 2nd ed. John Wiley & Son, 1996.

Appendix A: Derivation of Average Waiting Time $\mathbb{E}\{\tilde{W}_i\}$ in (4)

First, observe that

$$\begin{aligned}\mathbb{E}\{\tilde{W}_i\} &= \mathbb{E}\{\tilde{W}_i|Y \leq T\}\mathbb{P}\{Y \leq T\} + \mathbb{E}\{\tilde{W}_i|Y > T\}\mathbb{P}\{Y > T\} \\ &= \mathbb{E}\{Y|Y \leq T\}\mathbb{P}\{Y \leq T\} + (T + \mathbb{E}\{W_i\})\mathbb{P}\{Y > T\}. \end{aligned} \tag{19}$$

The first equality is obtained by conditioning on whether a neighbor of node i in a normal mode will receive the packet within T slots during which the other neighbor remains asleep. Here, if the neighbor in a normal mode receives the packet from node i within T, the average waiting time becomes $\mathbb{E}\{\tilde{W}_i|Y \leq T\} = \mathbb{E}\{Y|Y \leq T\}$. Otherwise, other neighbor which was in a sleep mode will in a normal mode after T, and both neighbors have an equal chance to receive the packet from node i. Hence, from memoryless property of geometric distributions, the average waiting time is $\mathbb{E}\{\tilde{W}_i|Y > T\} = T + \mathbb{E}\{W_i\}$. Then, after a little computation, we have

$$\begin{aligned}\mathbb{E}\{Y|Y \leq T\}\mathbb{P}\{Y \leq T\} &= \sum_{y=1}^{\infty} y \cdot \mathbb{P}\{Y = y, Y \leq T\} \\ &= \sum_{y=1}^{T} y \cdot \mathbb{P}\{Y = y\} = \frac{1-(1-q^2)^T}{q^2} - T(1-q^2)^T. \end{aligned} \tag{20}$$

By noting that $\mathbb{E}\{W_i\} = [q(1-(1-q)^2)]^{-1}$, we also have

$$(T + \mathbb{E}\{W_i\})\mathbb{P}\{Y > T\} = \left[T + \frac{1}{q(1-(1-q)^2)}\right](1-q^2)^T. \tag{21}$$

Thus, from (19)–(21), we finally have

$$\mathbb{E}\{\tilde{W}_i\} = \frac{1}{q^2} - \frac{1}{q^2}\frac{(1-q)}{(2-q)}(1-q^2)^T. \tag{22}$$

Appendix B: Proof of Theorem 1

Proof. Recall that each state in the state space $\mathcal{S}$ is a directed edge. G is also connected, i.e., there is at least a path (a sequence of nodes) from node u to node v for all $u, v \in \mathcal{N}$ $(u \neq v)$. Hence, there exists at least a directed path (a sequence of direct edges) connecting each directed edge (u', u) to every other directed edges (v, v'), where $u' \in N(u)$ and $v' \in N(v)$. Then, for $0 < p_j < 1$ at node $j \in \mathcal{N}$ with $d_j > 1$ in (6) in addition to $p_j = 1$ for $d_j = 1$, the transition probability $p_{(i,j)(j,k)} > 0$

for any $(i,j),(j,k) \in \mathcal{S}$. Thus, every state in $\mathcal{S}$ is reachable in finite time with positive probability, and the Markov chain $\{Z_t\}$ is irreducible. One can also see that the case of $p_j = p_{(i,j)(j,i)} = 0$ at node $j \in \mathcal{N}$ with $d_j > 1$ in (6), $i \in N(j)$, does not affect the irreducibility of $\{Z_t\}$, because it is not necessary to traverse path $i \to j \to i$ to reach (v,v') from (u',u). Therefore, by noting that the state space $\mathcal{S}$ is finite, i.e., $|\mathcal{S}| = 2m < \infty$, the Markov chain $\{Z_t\}$ is positive recurrent and hence it has a *unique* stationary on $\mathcal{S}$ [18, 1]. Due to this uniqueness, one can easily check that the stationary distribution of $\{Z_t\}$ is $\pi_{(i,j)} = 1/2m$ for each $(i,j) \in \mathcal{S}$ satisfying the following balance equations.

$$\begin{aligned}\pi_{(j,k)} &= \sum_{(i,l)\in\mathcal{S}} \pi_{(i,l)} p_{(i,l)(j,k)} = \pi_{(k,j)} p_{(k,j)(j,k)} + \sum_{i\in N(j)\setminus\{k\}} \pi_{(i,j)} p_{(i,j)(j,k)} \\ &= \pi_{(k,j)} p_j + \sum_{i\in N(j)\setminus\{k\}} \pi_{(i,j)} \frac{1-p_j}{d_j - 1},\end{aligned} \tag{23}$$

for each $(j,k) \in \mathcal{S}$, and $d_j > 1$. If $d_j = 1$ for any $j \in \mathcal{N}$, then $\pi_{(j,k)} = \pi_{(k,j)} \cdot 1$, where $k \in N(j)$. Also, $\sum_{(j,k)\in\mathcal{S}} \pi_{(j,k)} = 1$.

In addition, since $|A_j| = d_j$, it follows that

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

for each $j \in \mathcal{N}$. $\square$