Opportunistic Forwarding in Wireless Networks with Duty Cycling

Prithwish Basu
BBN Technologies
Cambridge MA, USA
pbasu@bbn.com

Chi-Kin Chau
University of Cambridge &
University College London, UK
chi-kin.chau@cl.cam.ac.uk

ABSTRACT

Opportunistic forwarding, by which data is randomly relayed to a neighbor based on local network information, is a fault-tolerant distributed algorithm particularly useful for challenged ad hoc and sensor networks where it is difficult to obtain global topology information because of frequent disruptions. Also, duty cycling is a common technique that constrains the RF operations of wireless devices for saving the battery energy and thus extending the longevity of the network. The combination of opportunistic forwarding and duty cycling is a useful approach for wireless ad hoc and sensor networks that are plagued with energy constraints and poor connectivity. However, such a design is hampered by the difficulty of analyzing and controlling its performance, particularly, the end-to-end latency. This paper presents analytical results that shed light on the latency of opportunistic forwarding in wireless networks with duty cycling. In particular, we give approximation formulas and bounds for the expected latency of opportunistic forwarding in presence of duty cycling for general finite network topologies, and an exact formula for a specific regular network topology that captures some common sensor network deployment scenarios. Moreover, our results concern finite-sized networks, and hence, are practically more useful than other asymptotic analyses in the literature.

Categories and Subjects: C.2.1 [Computer-Communication Networks]: *Wireless communication* G.2.2 [Graph Theory]: *Network problems*

General Terms: Theory, Design, Performance

Keywords: Wireless ad hoc and Sensor Networks, Opportunistic Forwarding, Duty Cycling, Random Walk on Graphs

1. INTRODUCTION

In challenged wireless ad hoc and sensor networks where gathering network topology and state information in a timely manner is difficult, simple opportunistic forwarding that requires only *local* information can be a viable approach. In such a scheme, a node relays a data packet to a random neighboring node, until the packet reaches its destination or expires. The simplest scheme in this category is *random walk*, which selects the next hop from among all neighbors with equal probability, and is independent of the past selections. Although random walk based algorithms inevitably incur a higher latency than optimized routing algorithms, they are robust under ad hoc topology changes, and require minimal computational capability. Hence, they are suitable for resource constrained devices.

Also, energy conservation is an important consideration in many battery-powered wireless networks. This is because RF transceivers consume a significant amount of energy not only for transmission and reception but also for performing *idle listening* on the channel [11]. Hence, wireless network designers often implement *duty cycling* on energy-constrained wireless devices. A simple duty cycling scheme is to turn RF transceivers off and on independently and randomly. Then, forwarding can only occur when the transmitting and receiving nodes are *both* awake. This does not require global coordination among the devices, and is more resilient and self-configuring for ad hoc deployments. On the other hand, in periodic/deterministic duty cycling schemes, the on/off schedules of network nodes have to be carefully coordinated such that efficient routing and forwarding can occur; such schemes are less robust than random duty cycling schemes.

This paper considers a duty cycling design called *pseudo-random duty cycling*, which is more efficient than purely random duty cycling and retains the benefits of the latter. Suppose that transmissions occur in slotted time. In pseudo-random duty cycling, by *a priori* exchange of pseudo random number generators (p-RNG) among neighboring nodes and the occasional exchange of a small set of parameter values such as the *seed* and *cycle position* of p-RNG and wake up probability, each node can determine the exact on/off state of all its neighbors. This enables deterministic prediction of future awake timeslots of neighbors and drastically reduces the chance of unsuccessful transmissions [12].

We propose a wireless network design that incorporates both opportunistic forwarding and duty cycling. In this integrated design, a node predicts the first timeslot in which any neighbor is supposed to wake up, and then forwards the packet to it in that timeslot. Ties are broken randomly, when more than one neighbor wakes up in a timeslot. This requires only local (neighbor) information, but no global

CHANTS'08, September 15, 2008, San Francisco, California, USA.

topology information, and hence is useful in situations with limited network connectivity/availability. We refer to this integrated scheme as a *duty cycling random walk*. In addition to being robust against ad hoc topology changes, such a randomized forwarding scheme is able to avoid the formation of routing hot spots and non-uniform energy depletion across the network.

Despite the simplicity of implementation and deployment of this wireless network design, it is generally non-trivial to analyze the performance and control it appropriately. Two key performance metrics that we are concerned with are the latency incurred by a packet due to duty cycling and opportunistic forwarding before reaching its destination, and the overall energy consumption levels that are directly proportional to the duty cycling probability. In this paper, we analyze the performance of duty-cycling random walk as a function of the network size n, and the duty cycling probability, p, and other properties of the topology such as node degree distributions.

The end-to-end latency incurred by a random walk on given networks can be characterized by the expected *hitting time* from a source to its destination. However, since duty cycling changes the fundamental nature of random walk in terms of latency, new analytical results are necessary. We build upon the formulation due to Lovasz [10] and provide bounds and approximation formulas for the expected end-to-end latency for duty cycling random walk in finite wireless networks.

We also apply our results to random geometric graphs, which have been widely-used for modeling diverse wireless ad hoc networks. We show that our approximation formulas are reasonably accurate in this context and hence our analytical results have useful ramifications on analyzing energy-latency tradeoffs in practical sensor network deployment scenarios.

Finally, we study the expected hitting time in a particular class of regular network topology, called r-nearest cycle, which can capture some common sensor network deployment scenarios such as on the banks of a circular lake or building. We derive an exact analytical formula for computing the expected hitting time in r-nearest cycle.

While there have been several analytical studies of the properties of random walks in the extant literature, they are mostly concerned with the asymptotic behavior of *mixing time* or *cover time* of random walks in large networks [2,3,9]; otherwise they propose general theoretical frameworks for arbitrary graphs that do not necessarily pertain to wireless networking [1,4,7]. Our results concern finite-sized networks that are relevant to wireless ad hoc networking, and hence are more useful than the aforementioned asymptotic analyses in the literature.

Our analytical results have been validated by extensive simulations. We believe that by using more accurate analyses, wireless network designers and deployers can make appropriate choices regarding the tradeoffs between energy consumption and the resultant end-to-end latency. For instance, if they want to adhere to a certain bound on the average end-to-end latency, our analytical framework can assist them to determine an appropriate duty cycling threshold that meets the energy consumption constraints.

Outline: In Section 2, we first give some technical background and definitions of the expected hitting times of random walks on finite graphs. In Section 3, we derive bounds and approximation formulas for the latency of random walks in the presence of duty cycling for general finite networks. In Section 4, we study the latency for random geometric graphs. In Section 5, we provide an exact formula for the expected hitting time of random walk for r-nearest cycle. Section 7 concludes the paper.

2. EXPECTED HITTING TIME OF RANDOM WALK ON GRAPHS

Random walk is one of the simplest opportunistic forwarding algorithms. The latency can be studied as the hitting time of random walk. There are several approaches to study the expected hitting time of a random walk on a graph. For instance, the effective resistance approach of a graph [1, 9] can only be applicable to vertex transitive graphs, such that the topology at every node of the graph is homogeneous. The discrete Green's function approach in [4, 7] only addresses non-boundary nodes on a graph. In this paper, however, we draw on Lovasz's formula [10], which computes the expected hitting time of random walk between any pair of nodes in a finite graph.

First we define some notations. We consider a *connected* undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Let the number of nodes be $|\mathcal{V}| = n$ and the number of edges be $|\mathcal{E}| = m$.

Denote the adjacency matrix of $\mathcal{G}$ as $\mathbf{A}$ where $\mathbf{A}_{i,j} = \mathbb{1}[(i,j) \in \mathcal{E}]$. For each node $i \in \mathcal{V}$, let d_i be the degree of i. Denote the diagonal matrix of $\mathcal{G}$ as $\mathbf{D}$ where $\mathbf{D}_{i,i} = \frac{1}{d_i}$ and $\mathbf{D}_{i,j} = 0$ for $i \neq j$. Let the transition matrix be $\mathbf{M} = \mathbf{DA}$ where

$$\mathbf{M}_{i,j} = \frac{\mathbb{1}[(i,j) \in \mathcal{E}]}{d_i}$$

For a pair of distinct nodes $s, t \in \mathcal{V}$, we denote the expected hitting time of random walk from s to t as $\mathbf{H}_{s,t}$, which satisfies the following equation:

$$\mathbf{H}_{s,t} = 1 + \frac{1}{d_s} \sum_{(s,i)\in\mathcal{E}} \mathbf{H}_{i,t} \quad \text{and} \quad \mathbf{H}_{t,t} = 0 \tag{1}$$

Define *symmetrized transition matrix* be $\mathbf{N} = \mathbf{D}^{\frac{1}{2}}\mathbf{A}\mathbf{D}^{\frac{1}{2}}$ where

$$\mathbf{N}_{i,j} = \frac{\mathbb{1}[(i,j) \in \mathcal{E}]}{\sqrt{d_i d_j}}$$

Lovasz solved Eqn. (1) with a solution as follows (see [10] Theorem 3.1):

$$\mathbf{H}_{s,t} = 2m \sum_{k=2}^{n} \frac{1}{1-\lambda_k} \Big(\frac{v_{k,t}^2}{d_t} - \frac{v_{k,s} v_{k,t}}{\sqrt{d_s d_t}} \Big) \tag{2}$$

where λ_k and v_k is the k-th largest eigenvalue and the corresponding eigenvector of $\mathbf{N}$. Denote $v_{k,i}$ as i-th entry of eigenvector v_k that corresponds to $i \in \mathcal{V}$. Note that since $\mathbf{N}$ is symmetric, all eigenvalues are real numbers. Since $\mathcal{G}$ is connected, by Frobenius-Perron Theorem, we obtain $\lambda_1 = 1 > \lambda_2$. Hence, Eqn. (2) is well defined.

Lovasz's formula (Eqn. (2)) enables more efficient evaluation of the expected hitting time than simulations. We emphasize that Lovasz's formula is applicable to arbitrary graphs, while other approaches (e.g. effective resistance) appear to be only applicable to limited classes of graphs.

3. RANDOM WALKS WITH DUTY CYCLING

In this section, we extend random walk to the setting with pseudo-random duty cycling. We assume that time is divided into slots and slot boundaries are synchronized (and re-synchronized) using one of many techniques [5, 6]. Let p be the probability that a node is awake in a particular time slot, and $1-p$ be the probability of being asleep in a time slot. In pseudo-random duty cycling [12], nodes are assumed to have shared knowledge of a pseudo-random number generator (p-RNG), which is assumed to have happened out-of-band and may even have been programmed into the system before bootstrapping.

If a sender becomes aware of the seed and the cycle position of the receiver's p-RNG, it can *deterministically* predict the slot in which the receiver is going to be awake and listening. Thus the sender does not need to attempt to transmit in each slot with probability p; instead it can determine how many slots into the future the receiver will become awake, and only then transmit. Meanwhile, the sender can go to sleep until the receiver is supposed to wake up. This not only reduces the energy cost but also reduces the latency of waiting for an awake slot[1].

In this paper, we investigate the forwarding latency in the *low traffic regime* in which the effect of interference due to concurrent wireless transmissions and queuing delays is negligible. This is not unreasonable since most sensor networks with low duty cycles are expected to have low traffic volumes.

We assume the following forwarding rules at nodes performing duty cycling random walk:

1. Forward data to the earliest neighbor to wake up, based on the pseudo-RNG computation for the given wake-up probability.

2. If multiple neighbors wake up in the same time slot, data will be forwarded to a randomly chosen awake neighbor with uniform probability.

We refer to such random walk as *duty cycling random walk*. Note that these forwarding rules do not alter the resultant transition probabilities, which are the same as those for simple random walk on finite graphs. In other words, a duty cycling random walk amounts to first selecting a neighbor with equal probability and then sleeping for the number of slots until that neighbor wakes up (then the sender wakes up and transmits to that neighbor). We formally prove this property in the Appendix.

3.1 Per hop Latency

Now we estimate the expected per hop latency of random walks in the presence of pseudo-random duty cycling. Suppose node i is waiting to forward data, which has a set of d_i neighbors, denoted as $\mathsf{Nb}_i \subseteq \mathcal{V}$. Each of these neighbors is performing pseudo-random duty cycling with probability p. Let $W^{(i)}$ be the random variable denoting the number of slots at i before one of the neighbors of i wakes up. Therefore $W^{(i)} = \min\{W_1, W_2, \ldots, W_d\}$, where W_j is the waiting time random variable for neighbor $j \in \mathsf{Nb}_i$.

LEMMA 1. For all $i \in \mathcal{V}$,

$$\mathbb{E}[W^{(i)}] = \frac{1}{1-(1-p)^{d_i}}$$

PROOF. Note that each W_j is a geometrically distributed random variable with parameter p. For $t \geq 1$, the per-hop latency probability distribution is:

$$\begin{aligned}\mathbb{P}\{W^{(i)} \geq t\} &= \mathbb{P}\{\min(W_1, W_2, \ldots, W_{d_i}) \geq t\} \\ &= \mathbb{P}\{\forall j \in \mathsf{Nb}_i, W_j \geq t\} \\ &= (1-p)^{(t-1)d_i}\end{aligned}$$

Since $W^{(i)}$ is non-negative, the expected per-hop latency until data is forwarded to some awake neighbor is:

$$\begin{aligned}\mathbb{E}[W^{(i)}] &= \sum_{t=1}^{\infty} \mathbb{P}\{W^{(i)} \geq t\} \\ &= \sum_{t=1}^{\infty} (1-p)^{(t-1)d_i} = \frac{1}{1-(1-p)^{d_i}}\end{aligned}$$

□

Clearly the expected per-hop latency decreases with decreasing node degree d_i. For extremely low duty cycling rates (i.e. small values of p), we obtain: $\mathbb{E}[W^{(i)}] \approx \frac{1}{pd_i}$.

3.2 Latency with Duty Cycling

Since each node is scheduling its wake-up slots independently from each other with equal wake-up probability p, it is equivalent to first selecting a neighbor with equal probability and waits for $W^{(i)}$ slots to forward data (see the Appendix for the formal proof).

For a pair of nodes $s, t \in \mathcal{V}$, we denote the expected hitting time of random walk with pseudo-random duty cycling from s to t as $\mathbf{L}_{s,t}$. By Lemma 1, $\mathbf{L}_{s,t}$ satisfies:

$$\mathbf{L}_{s,t} = \frac{1}{1-(1-p)^{d_s}} + \frac{1}{d_s} \sum_{(s,i)\in\mathcal{E}} \mathbf{L}_{i,t}$$

Unlike $\mathbf{H}_{s,t}$, it is much harder to solve for $\mathbf{L}_{s,t}$ since the per hop latency is not homogeneous at all nodes. Hence, we derive bounds for $\mathbf{L}_{s,t}$ that helps us to estimate the expected end-to-end latency with pseudo-random duty cycling.

THEOREM 2. Given a graph $\mathcal{G}$, the expected hitting time of random walks with pseudo-random duty cycling from s to t is bounded by:

$$\frac{\mathbf{H}_{s,t}}{1-(1-p)^{d_{\max}}} \leq \mathbf{L}_{s,t} \leq \frac{\mathbf{H}_{s,t}}{1-(1-p)^{d_{\min}}}$$

where $d_{\min}$ and $d_{\max}$ are the minimum and maximum node degrees of $\mathcal{G}$.

PROOF. Clearly, for all $i \in \mathcal{V}$,

$$\frac{1}{1-(1-p)^{d_{\min}}} \geq \frac{1}{1-(1-p)^{d_i}} \geq \frac{1}{1-(1-p)^{d_{\max}}}$$

Hence, the expected per-hop latency is bounded as:

$$\frac{1}{1-(1-p)^{d_{\max}}} \leq \mathbb{E}[W^{(i)}] \leq \frac{1}{1-(1-p)^{d_{\min}}} \qquad (3)$$

Therefore, the expected end-to-end latency, summing over an average $\mathbf{H}_{s,t}$ hops, is bounded similarly. □

[1]Note that the processors of nodes are on, even if the transceiver may be off at certain times, and hence such computations are feasible.

3.3 Approximations

Although it is hard to solve for $\mathbf{L}_{s,t}$ exactly, we can provide two better approximations based on the following heuristics:

1) We define the first approximation as:

$$\tilde{\mathbf{L}}^{(1)}_{s,t} \triangleq \frac{\mathbf{H}_{s,t}}{1-(1-p)^{\bar{d}}}$$

where $\bar{d}$ is the average node degree of $\mathcal{G}$.

We consider a sample path of random walks with pseudo-random duty cycling from s to t on $\mathcal{G}$, $(s = v_1, v_2, ..., v_\ell, t)$. The hitting time is

$$L_\ell = \sum_{i=1}^{\ell} \frac{1}{1-(1-p)^{d_{v_i}}}$$

It is easy to check that $f(d) = \frac{1}{1-(1-p)^d}$ is a convex function. By Jensen's inequality, we have:

$$\frac{1}{\ell}\sum_{i=1}^{\ell} \frac{1}{1-(1-p)^{d_{v_i}}} \geq \frac{1}{1-(1-p)^{\frac{1}{\ell}\sum_{i=1}^{\ell} d_i}}$$

When ℓ is large or the separation between s and t is large, it is reasonable to expect that:

$$\frac{1}{1-(1-p)^{\frac{1}{\ell}\sum_{i=1}^{\ell} d_i}} \approx \frac{1}{1-(1-p)^{\bar{d}}}$$

Hence, when n is large, we obtain a better approximation of the lower bound as:

$$\mathbf{L}_{s,t} = \mathbb{E}[L_\ell] \geq \frac{\mathbb{E}[\ell]}{1-(1-p)^{\bar{d}}} = \tilde{\mathbf{L}}^{(1)}_{s,t}$$

2) We define the second approximation as:

$$\tilde{\mathbf{L}}^{(2)}_{s,t} \triangleq \left(\sum_{i=1}^{n} \frac{d_i}{1-(1-p)^{d_i}}\right)\left(\sum_{k=2}^{n} \frac{1}{1-\lambda_k}\left(\frac{v^2_{k,t}}{d_t} - \frac{v_{k,s}v_{k,t}}{\sqrt{d_s d_t}}\right)\right)$$

We recall that the stationary distribution of a random walk on a connected graph is given by $\pi(i) = \frac{d_i}{2m}$ [1, 9]. Suppose the random walks from s to t is close to reaching the stationary distribution. Then, we obtain:

$$\begin{aligned}\mathbf{L}_{s,t} &\approx \mathbf{H}_{s,t}\sum_{i=1}^{n} \frac{\mathbb{P}\{\text{random walk hits } i\}}{1-(1-p)^{d_i}} \\ &= \mathbf{H}_{s,t}\sum_{i=1}^{n} \frac{\pi(i)}{1-(1-p)^{d_i}} \\ &= \tilde{\mathbf{L}}^{(2)}_{s,t}\end{aligned}$$

COROLLARY 3.

$$\tilde{\mathbf{L}}^{(2)}_{s,t} \geq \tilde{\mathbf{L}}^{(1)}_{s,t}$$

PROOF. This follows from the fact that function $f(x) = \frac{x}{1-(1-p)^x}$ is convex and the application of Jensen's inequality. □

As we shall see in Section 6, both $\tilde{\mathbf{L}}^{(1)}_{s,t}$ and $\tilde{\mathbf{L}}^{(2)}_{s,t}$ appear as relatively accurate approximations to $\mathbf{L}_{s,t}$, in the simulations with respect to random geometric graphs.

4. RANDOM GEOMETRIC GRAPHS

Random geometric graphs are widely-used for modeling diverse wireless ad hoc networks, in which nodes are randomly placed in a confined area, and communication links are established between nodes that are within a pre-defined transmission radius. In this section, we especially study the hitting time of random walks on random geometric graphs.

We denote a random geometric graph as $\mathcal{G}_{\mathsf{geo}}(n, \mathsf{r})$, which is an ensemble of n-node graph such that the position of each vertex is independently uniformly distributed on a 2D unit square area, and there is an edge between a pair of nodes if they are within transmission radius r. First, we draw on the following lemma from [3] (Lemma 10) that shows that the degrees of a random geometric graph are concentrated on the mean degree.

LEMMA 4. Given a random geometric graph $\mathcal{G}_{\mathsf{geo}}(n, \mathsf{r})$, such that $\mathsf{r} = \Omega(\sqrt{\frac{\log n}{n}})$. Then the degree of every node i is

$$d_i = \bar{d}(1+o(1)) = n\pi\mathsf{r}^2(1+o(1))$$

with high probability.

The following corollary confirms that $\tilde{\mathbf{L}}^{(1)}_{s,t}$ is a good approximation of $\mathbf{L}_{s,t}$ for large random geometric graphs.

COROLLARY 5. Given a random geometric graph $\mathcal{G}_{\mathsf{geo}}(n, \mathsf{r})$, such that $\mathsf{r} = \Omega(\sqrt{\frac{\log n}{n}})$. Then,

$$\mathbf{L}_{s,t} = \frac{\mathbf{H}_{s,t}}{1-(1-p)^{n\pi\mathsf{r}^2(1+o(1))}}$$

with high probability.

PROOF. This follows from Theorem 2 and Lemma 4. □

5. R-NEAREST CYCLES

In this section, we derive an exact analytical formula of the expected hitting time in a particular deterministic class of *regular* network topology, called r-nearest cycle, which is useful to some common sensor network deployment scenarios.

Although Lovasz's formula is very general, it is not straightforward to gain insights for specific classes of graphs useful to the design of wireless networks. In general, the evaluation of eigenvalues and eigenvectors of symmetrized transition matrix $\mathbf{N}$ is non-trivial. Nonetheless, we found that the evaluation of eigenvalues and eigenvectors of $\mathbf{N}$ is straightforward for special regular topologies such as the r-nearest cycle. Given a cycle of n nodes denoted as $\mathcal{C}_n$, we construct an r-nearest cycle (denoted as $\mathcal{C}^r_n$) as the graph with edges between nodes and their r-nearest left and r-nearest right neighbors on $\mathcal{C}_n$. Thus, the degree of each node i is $d_i = 2r$.

An r-nearest cycle can be regarded as one dimensional projection of a geometric wireless network topology, where nodes are connected when they are within some distance r. Hence, the fundamental insights for r-nearest cycle will be useful for understanding the two dimensional case in wireless networking.

The symmetrized transition matrix $\mathbf{N}$ of an r-nearest cycle is a *circulant* matrix [8], which consists of row entries as

shifts of the next row entries. A circulant matrix is in form of:

$$\begin{pmatrix} a_1 & a_2 & \dots & a_{n-1} & a_n \\ a_n & a_1 & \dots & a_{n-2} & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_3 & a_4 & \dots & a_1 & a_2 \\ a_2 & a_3 & \dots & a_n & a_1 \end{pmatrix}$$

It is easy to diagonalize circulant matrices. Let ϵ be the primitive n-root of 1. That is, ϵ is the complex number:

$$\epsilon = \cos\left(\tfrac{2\pi}{n}\right) + \mathbf{i}\sin\left(\tfrac{2\pi}{n}\right)$$

For $1 \le k \le n$, define a number:

$$\lambda_k = a_1 + a_2\epsilon^{k-1} + \cdots + a_n\epsilon^{(n-1)(k-1)} \tag{4}$$

and a column vector:

$$v_k = \frac{1}{\sqrt{n}}\Big(1,\ \epsilon^{k-1},\ \epsilon^{2(k-1)},\ \dots,\ \epsilon^{(n-1)(k-1)}\Big)^T \tag{5}$$

By substitution, we obtain:

$$\begin{pmatrix} a_1 & a_2 & \dots & a_{n-1} & a_n \\ a_n & a_1 & \dots & a_{n-2} & a_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_3 & a_4 & \dots & a_1 & a_2 \\ a_2 & a_3 & \dots & a_n & a_1 \end{pmatrix} \begin{pmatrix} 1 \\ \epsilon^{k-1} \\ \epsilon^{2(k-1)} \\ \vdots \\ \epsilon^{(n-1)(k-1)} \end{pmatrix} = \lambda_k \begin{pmatrix} 1 \\ \epsilon^{k-1} \\ \epsilon^{2(k-1)} \\ \vdots \\ \epsilon^{(n-1)(k-1)} \end{pmatrix}$$

Hence, λ_k and v_k are the k-th eigenvalue and eigenvector of a circulant matrix, and hence, can be used for substitution in Eqn. (2).

For an r-nearest cycle $\mathcal{C}_n^r$, the entry a_i of symmetrized transition matrix $\mathbf{N}$ is:

$$a_i = \begin{cases} \frac{1}{2r} & \text{if } 2 \le i \le r+1 \text{ or } n-r+1 \le i \le n \\ 0 & \text{otherwise} \end{cases}$$

Next, we substitute Eqns. (4)-(5) into Lovasz's formula, Eqn. (2), to evaluate the hitting time of random walk on $\mathcal{C}_n^r$. The k-th eigenvalue of symmetrized transition matrix $\mathbf{N}$ for $\mathcal{C}_n^r$ by Eqn. (4) is:

$$\begin{aligned} \lambda_k &= \tfrac{1}{2r}\sum_{j=1}^{r}\cos\left(\tfrac{2\pi j(k-1)}{n}\right) + \mathbf{i}\sin\left(\tfrac{2\pi j(k-1)}{n}\right) \\ &\quad + \tfrac{1}{2r}\sum_{j=1}^{r}\cos\left(\tfrac{2\pi (n-j)(k-1)}{n}\right) + \mathbf{i}\sin\left(\tfrac{2\pi (n-j)(k-1)}{n}\right) \\ &= \tfrac{1}{r}\sum_{j=1}^{r}\cos\left(\tfrac{2\pi j(k-1)}{n}\right) \end{aligned} \tag{6}$$

Although λ_k is a real number, the entries of eigenvector in Eqn. (5) are not always real numbers, and hence not useful in evaluating Eqn. (2). However, when n is even, the first entry of the k-th eigenvector in Eqn. (5) is $v_{k,1} = \frac{1}{\sqrt{n}}$, while the $(\frac{n}{2}+1)$-th entry is:

$$\begin{aligned} & v_{k,\frac{n}{2}} = \tfrac{1}{\sqrt{n}}\epsilon^{(k-1)\frac{n}{2}} \\ &= \tfrac{1}{\sqrt{n}}\Big(\cos\left(\tfrac{2\pi(k-1)\frac{n}{2}}{n}\right) + \mathbf{i}\sin\left(\tfrac{2\pi(k-1)\frac{n}{2}}{n}\right)\Big) \\ &= \tfrac{1}{\sqrt{n}}\Big(\cos((k-1)\pi) + \mathbf{i}\sin((k-1)\pi)\Big) = \tfrac{(-1)^{(k-1)}}{\sqrt{n}} \end{aligned} \tag{7}$$

Thus, $v_{k,\frac{n}{2}}$ is always a real number.

In fact, the first and the $(\frac{n}{2}+1)$-th entries in an eigenvector correspond to a pair of farthest apart nodes on $\mathcal{C}_n^r$. It is the maximum expected hitting time, which can serve as the upper bound for other pairs of nodes.

THEOREM 6. Given an r-nearest cycle $\mathcal{C}_n^r$ and n is even, the maximum expected hitting time of random walk is:

$$\mathbf{H}_{1,\frac{n}{2}+1} = \sum_{k=0}^{\frac{n-2}{2}} \frac{4r}{(2r+1) - \frac{\sin\left(\frac{\pi(2k+1)(2r+1)}{n}\right)}{\sin\left(\frac{\pi(2k+1)}{n}\right)}} \tag{8}$$

PROOF. We remark that $2m = 2rn$. By substitution of Eqns. (6)-(7) into Eqn. (2), we obtain:

$$\begin{aligned} \mathbf{H}_{1,\frac{n}{2}+1} &= 2m\sum_{k=2}^{n}\frac{\frac{1-(-1)^{(k-1)}}{2rn}}{1 - \frac{1}{r}\sum_{j=1}^{r}\cos\left(\frac{2\pi j(k-1)}{n}\right)} \\ &= \sum_{k=0}^{\frac{n-2}{2}}\frac{2}{1 - \frac{1}{r}\sum_{j=1}^{r}\cos\left(\frac{2\pi j(2k+1)}{n}\right)} \\ &= \sum_{k=0}^{\frac{n-2}{2}}\frac{4r}{(2r+1) - \frac{\sin\left(\frac{\pi(2k+1)(2r+1)}{n}\right)}{\sin\left(\frac{\pi(2k+1)}{n}\right)}} \end{aligned}$$

where the last inequality follows from the identity:

$$1 + 2\sum_{j=1}^{r}\cos(jx) = \frac{\sin\left(r+\frac{1}{2}\right)x}{\sin(\frac{x}{2})}$$

□

Fig. 1(a) illustrates the trend of Eqn. (8) with respect to different network size, n, whereas Fig. 1(b) illustrates the same with respect to the number of nearest neighbors, r. It is observed that the expected hitting time scales almost quadratically with n, but is inversely proportional to r.

Increasing the number of nearest neighbors r in r-nearest cycles is an analogue of increasing the transmission radius r in random geometric graphs. Hence, an implication of our result is that increasing the transmission radius in a wireless network will roughly decrease the latency in inverse proportion, while increasing the number of relay nodes in a wireless network will roughly increase the latency quadratically.

We remark that there are no known results using the effective resistance and discrete Green's function for computing the hitting time in r-nearest cycles. Our result appears to be an original contribution to this area.

6. SIMULATIONS AND EVALUATIONS

In this section, we present some representative simulations results to show that the approximations $\tilde{\mathbf{L}}_{s,t}^{(1)}$ and $\tilde{\mathbf{L}}_{s,t}^{(2)}$ are relatively accurate. We simulated random walks with pseudo-random duty cycling on random geometric graphs and large 2D lattices, and then compared those results with the values obtained from the approximation formulas. Specifically, we simulated a large number of random duty cycling instances on each graph with different p and then measured the average end-to-end latency.

First, we simulated duty cycling random walks on small networks – for a 10 node random geometric graph with wake up probability, $p = 0.1$, we found that the simulated value of average end-to-end latency for all s, t pairs was 40.736 slots (averaged over 1000 runs), while $\tilde{\mathbf{L}}_{s,t}^{(1)} = 39.54$ slots.

We then considered larger networks – we simulated the duty cycling random walk on a 100 node random geometric

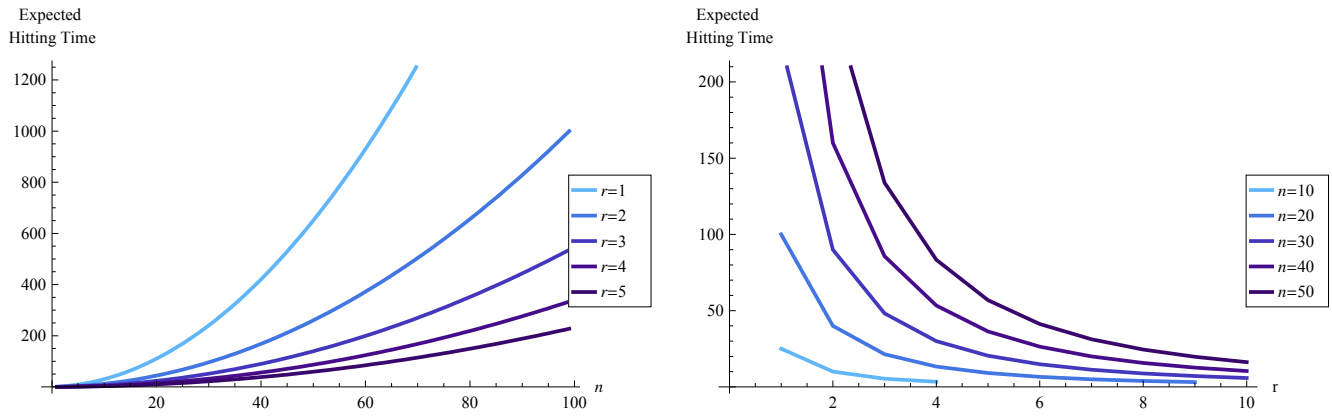

Figure 1: Expected hitting time for the farthest pair of nodes in r-nearest cycles (Eqn. 8) as a function of: (a) network size n; (b) number of nearest neighbors r.

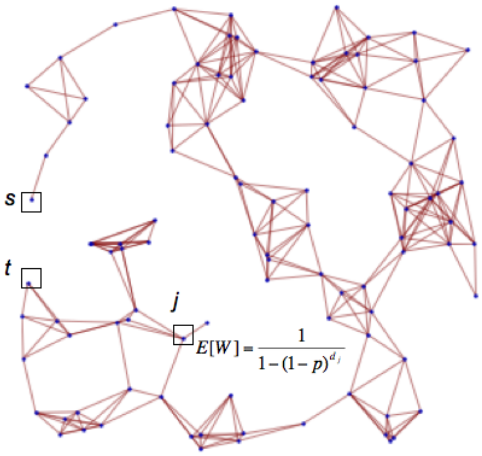

Figure 2: A Sparse random geometric graph.

graph shown in Fig. 2 (1500 independent simulation runs) with wake up probability, $p = 0.1$. The diameter of this graph is 22 hops and we chose the farthest pair of nodes as source and target[2]. The average latency in simulations was 7892 slots, whereas $\tilde{\mathbf{L}}^{(1)}_{s,t} = 7770$ slots. Hence, the approximation formula, $\tilde{\mathbf{L}}^{(1)}_{s,t}$, is within 1.5% of deviation.

We also compared the simulation and analytical results for average latency on a regular grid graph, a 10×10 Manhattan 2D lattice ($n = 100$), and considered the path from a node at one corner to the node at the opposite corner of the lattice (the diameter is 18 hops). We ran 1000 random simulations with $p = 0.1$. The average latency in simulations was 1715 slots, whereas $\tilde{\mathbf{L}}^{(1)}_{s,t} = 1734$ slots. Again the approximation formula, $\tilde{\mathbf{L}}^{(1)}_{s,t}$, is within 1.1% of deviation. The latency histogram is shown in Fig. 3.

Finally, we studied the effect of pseudo-random duty cycling on different values of p on a 100 node random geometric graph shown in Fig. 4. The diameter of this graph is 14 hops. In the same figure we plot the four different per-hop latency estimates. We observe that $\tilde{\mathbf{L}}^{(1)}_{s,t}$ and $\tilde{\mathbf{L}}^{(2)}_{s,t}$ are very close to each other. We plotted the comparison of these two estimates with the simulation results (averaged over 1500

[2]Note that we chose to present the simulation results for the farthest pair of nodes since it takes a long time to simulate all possible (s, t) pairs in large networks.

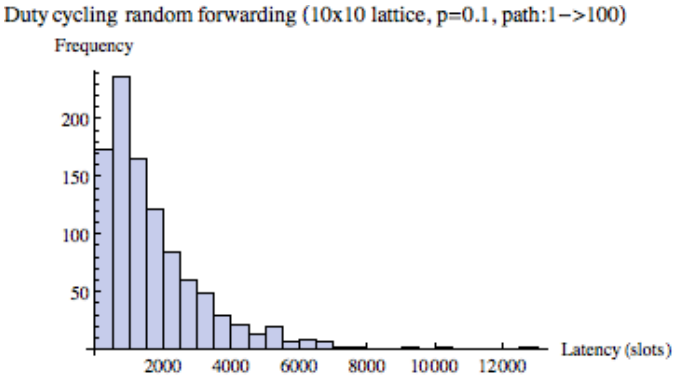

Figure 3: Latency histogram for a 10×10 2D lattice.

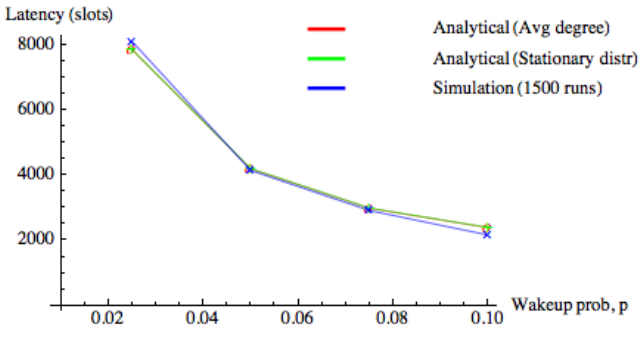

Figure 5: Comparison between approximation formulas and simulations of average end-to-end latency suffered by duty cycling random walks between the farthest pair of nodes on the 100 node graph shown in Fig. 4 against different values of p.

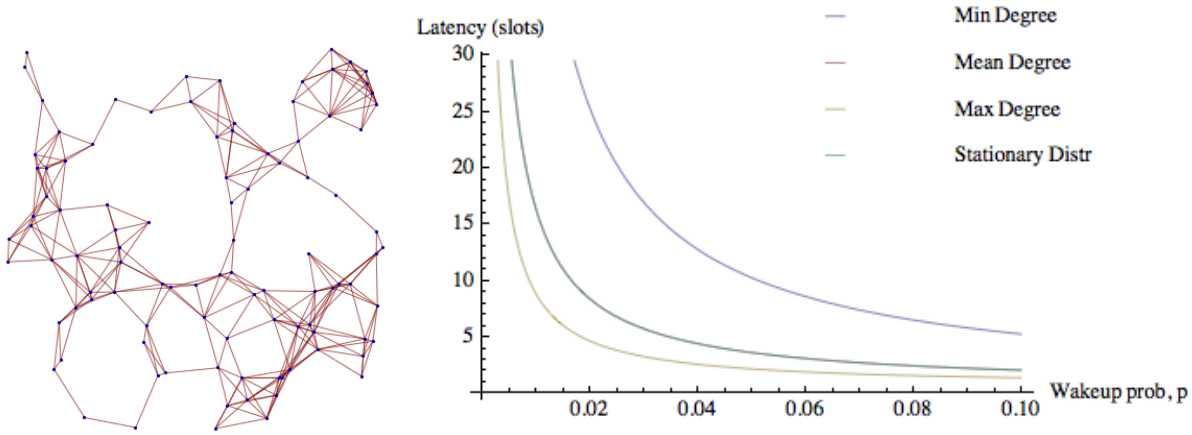

Figure 4: A random geometric graph with $n = 100$, $r = 0.15$, and $p = 0.1$; and the average per-hop latency according to various bounds and approximation formulas: Min and Max curves are given by Equation 3; the Mean curve is given by per-hop component of $\tilde{\mathbf{L}}_{s,t}^{(1)}$ and Stationary Distribution curve refers to the per-hop component of $\tilde{\mathbf{L}}_{s,t}^{(2)}$

.

runs) in Fig. 5 and we can observe that the approximations are reasonably accurate. This is a significant win because the execution of 1500 runs took several hours whereas the latency approximations presented in this paper were computed virtually instantaneously.

7. CONCLUSION

In this paper we presented an approach that combines opportunistic forwarding and duty cycling as a useful paradigm in wireless sensor and ad hoc networks that have poor connectivity and energy constraints. We presented accurate estimation formulas for the latency of opportunistic forwarding in wireless networks in the presence of duty cycling. Our results are for finite-sized networks, which are practically more useful than other asymptotic analyses in the literature. Designers and deployers of wireless sensor networks who desire to exploit the tradeoffs between energy efficiency and latency can put these results to good use. In this paper we have not compared the performance of duty cycling random walks with deterministic duty cycling techniques that gather topology information and perform more intelligent routing. Although the latter is likely to have better latency performance, that can be outweighed by the other benefits of using stateless random approaches, e.g. avoidance of hot-spot formation and non-uniform energy depletion, and fault-tolerance. Comparison with more intelligent duty cycling approaches is a topic of ongoing research.

Acknowledgment

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0002.l. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Chi-Kin Chau is also sponsored by a Croucher Foundation Fellowship.

8. REFERENCES

[1] D. Aldous and J. A. Fill. Reversible Markov chains and random walks on graphs. `http://www.stat.berkeley.edu/~aldous/RWG/book.html`, 1999. Lecture notes.

[2] C. Avin and G. Ercal. On the cover time and mixing time of random geometric graphs. *Theoretical Computer Science*, 380:2–22, June 2007.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Information Theory*, 52(6), June 2006.

[4] F. Chung and S.-Y. Yau. Discrete green's functions. *Journal of Combinatorial Theory, Series A*, 91:191–214, 2000.

[5] L. Dai, P. Basu, and J. Redi. An energy efficient and accurate slot synchronization scheme for wireless sensor networks. *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pages 1–8, 1-5 Oct. 2006.

[6] J. Degesys, P. Basu, and J. Redi. Synchronization of strongly pulse-coupled oscillators with refractory periods and random medium access. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 1976–1980, New York, NY, USA, 2008. ACM.

[7] R. B. Ellis. Discrete green's functions for products of regular graphs, 2003. arXiv:math/0309080v2.

[8] D. Geller, I. Kra, S. Popescu, and S. Simanca. On circulant matrices. `http://www.math.sunysb.edu/~sorin/`. Lecture notes.

[9] D. A. Levin, Y. Peres, and E. L. Wilmer. Markov

chains and mixing times. http://www.oberlin.edu/markov/, 2006. Lecture notes.
[10] L. Lovasz. Random walk on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:1–46, 1993.
[11] C. Park, K. Lahiri, and A. Raghunathan. Battery discharge characteristics of wireless sensor nodes: An experimental analysis. In *IEEE SECON 2005*, September 2007.
[12] J. Redi, S. Kolek, K. Manning, C. Partridge, R. Rosales-Hain, R. Ramanathan, and I. Castineyra. Javelen: An ultra-low energy ad hoc wireless network. *Ad Hoc Networks Journal*, 5(8), 2008.

9. APPENDIX

In this section, we formally show that the forwarding rules due to pseudo-random duty cycling do not alter the resultant transition probabilities, which are the same as those for simple random walk on finite graphs. In other words, a duty cycling random walk amounts to first selecting a neighbor with equal probability and then waiting (while sleeping) for the number of slots until that neighbor wakes up.

THEOREM 7. In duty cycling random walk, the probability of forwarding data from node i to any particular neighbor is $\frac{1}{d}$, where d is the degree of node i.

PROOF. Let $X_1, X_2, \ldots, X_d$ be i.i.d. random variables denoting waiting time (in slots) before neighbors $N_1, N_2, \ldots, N_d$ wake up, respectively.

Let us denote the probability that data from node i is forwarded to node N_1 by $\mathbb{P}(i \rightarrow N_1)$. This is given by the following expression:

$$\begin{aligned}\mathbb{P}(i \rightarrow N_1) &= \mathbb{P}(X_1 < X_2, X_1 < X_3, \cdots, X_1 < X_d) \\ &+\frac{1}{2}\mathbb{P}(X_1 = X_2, X_1 < X_3, \cdots, X_1 < X_d) \\ &+\frac{1}{2}\mathbb{P}(X_1 < X_2, X_1 = X_3, \cdots, X_1 < X_d) \\ &+\cdots \\ &+\frac{1}{2}\mathbb{P}(X_1 < X_2, X_1 < X_3, \cdots, X_1 = X_d) \\ &+\frac{1}{3}\mathbb{P}(X_1 = X_2 = X_3, \cdots, X_1 < X_d) \\ &+\frac{1}{3}\mathbb{P}(X_1 = X_2 = X_4, \cdots, X_1 < X_d) \\ &+\cdots \\ &+\frac{1}{4}\mathbb{P}(X_1 = X_2 = X_3 = X_4, \cdots, X_1 < X_d) \\ &+\cdots \\ &+\frac{1}{d}\mathbb{P}(X_1 = X_2 = X_3 = \cdots = X_d) \qquad (9)\end{aligned}$$

The general term in the above series is:

$$\frac{1}{k}\mathbb{P}(X_1 = X_{\pi_2} = \cdots = X_{\pi_k}, X_1 < X_{\pi_{k+1}}, \cdots, X_1 < X_{\pi_d})$$

where π denotes a permutation of the set $\{2, 3, 4, \cdots, d\}$. In particular, X_1 is exactly equal to $k-1$ other variables $\{X_{\pi_2}, X_{\pi_2}, \ldots, X_{\pi_k}\}$ and is strictly less than the other $d-k$ other variables $\{X_{\pi_{k+1}}, \ldots, X_{\pi_d}\}$. The multiplying factor of $\frac{1}{k}$ exists due to the fact that a k-way tie among simultaneously awake neighbors is broken randomly (uniformly). Also note that there are exactly $\binom{d-1}{k-1}$ terms of this type.

Note than if the tie-breaking procedure is biased more favorably towards certain nodes over others, the $\frac{1}{k}$ term would have to be replaced with a term that reflects this aforementioned bias. In that case the duty cycling random walk will not be equivalent to a simple random walk but will be a weighted version of it, instead. In that case, this formulation can be appropriately modified to compute the resultant transition probabilities which, in general, may no longer be independent of the neighbor.

Now we calculate the value of this term as follows:

$$\begin{aligned}&\mathbb{P}(X_1 = X_{\pi_2} = \cdots = X_{\pi_k}, X_1 < X_{\pi_{k+1}}, \cdots, X_1 < X_{\pi_d} \\ &= \sum_{s=1}^{\infty}\mathbb{P}(X_1 = s)\mathbb{P}(X_{\pi_2} = s)\cdots\mathbb{P}(X_{\pi_k} = s) \times \\ &\qquad \mathbb{P}(X_{\pi_{k+1}} > s)\mathbb{P}(X_{\pi_{k+2}} > s)\cdots\mathbb{P}(X_{\pi_d} > s) \\ &= \sum_{s=1}^{\infty}((1-p)^{s-1}p)^k \times ((1-p)^s)^{d-k} \\ &= (\frac{p}{1-p})^k\sum_{s=1}^{\infty}(1-p)^{ds} \\ &= (\frac{p}{1-p})^k\frac{(1-p)^d}{1-(1-p)^d} \\ &= \frac{p^k(1-p)^{d-k}}{1-(1-p)^d} \qquad (10)\end{aligned}$$

We use the following in the above calculation:

$$\begin{aligned}\mathbb{P}(X_j = s) &= (1-p)^{s-1}p \\ \mathbb{P}(X_j > s) &= \sum_{i=s+1}^{\infty}\mathbb{P}(X_j = i) \\ &= \sum_{i=s+1}^{\infty}(1-p)^{i-1}p = (1-p)^s\end{aligned}$$

From Eqns. (9)-(10) we have

$$\begin{aligned}\mathbb{P}(i \rightarrow N_1) &= \sum_{k=1}^{d}\frac{1}{k}\binom{d-1}{k-1}\frac{p^k(1-p)^{d-k}}{1-(1-p)^d} \\ &= \frac{(1-p)^d}{1-(1-p)^d}\sum_{k=1}^{d}\frac{1}{k}\binom{d-1}{k-1}(\frac{p}{1-p})^k \qquad (11)\end{aligned}$$

Now consider the following expression:

$$\begin{aligned}\sum_{k=1}^{d}\frac{1}{k}\binom{d-1}{k-1}x^k &= \frac{1}{d}\sum_{k=1}^{d}\binom{d}{k}x^k \\ &= \frac{1}{d}(\sum_{k=0}^{d}\binom{d}{k}x^k - 1) \\ &= \frac{1}{d}((1+x)^d - 1) \qquad (12)\end{aligned}$$

Using identity (12) on Eqn. (11), we get:

$$\mathbb{P}(i \rightarrow N_1) = \frac{1}{d}$$

□