Tunable Locally-Optimal Geographical Forwarding in Wireless Sensor Networks with Sleep-Wake Cycling Nodes

K.P. Naveen and Anurag Kumar

Dept. of E.C.E., Indian Institute of Science, Bangalore 560 012, India.
{naveenkp, anurag}@ece.iisc.ernet.in

Abstract—We consider a wireless sensor network whose main function is to detect certain infrequent alarm events, and to forward alarm packets to a base station, using geographical forwarding. The nodes know their locations, and they sleep-wake cycle, waking up periodically but not synchronously. In this situation, when a node has a packet to forward to the sink, there is a trade-off between how long this node waits for a suitable neighbor to wake up and the progress the packet makes towards the sink once it is forwarded to this neighbor. Hence, in choosing a relay node, we consider the problem of minimizing average delay subject to a constraint on the average progress. By constraint relaxation, we formulate this next hop relay selection problem as a Markov decision process (MDP). The exact optimal solution (BF (Best Forward)) can be found, but is computationally intensive. Next, we consider a mathematically simplified model for which the optimal policy (SF (Simplified Forward)) turns out to be a simple one-step-look-ahead rule. Simulations show that SF is very close in performance to BF, even for reasonably small node density. We then study the end-to-end performance of SF in comparison with two extremal policies: Max Forward (MF) and First Forward (FF), and an end-to-end delay minimising policy proposed by Kim et al. [1]. We find that, with appropriate choice of one hop average progress constraint, SF can be *tuned* to provide a favorable trade-off between end-to-end packet delay and the number of hops in the forwarding path.

I. Introduction

An important application of wireless sensor networks (WSN) is dense embedded sensing for the purpose of detecting certain infrequently occuring events, such as failures in a large structure, or intrusion into a secure region. Such an event can occur anywhere in a large WSN, and once an event is detected, the alarm needs to be rapidly sent to the sink for further action. In such WSNs, typically the nodes rely on batteries, or energy harvested from their surroundings, and, hence, need to be extremely parsimonius in their use of energy. In order to conserve energy, the nodes operate in sleep-wake cycles; when a node wakes up it performs sensing, and also can assist in forwarding any alarm packets towards the sink. In this paper, we consider the situation in which the sleep-wake cycles of nodes are *not synchronized.* In such a setting, stateful routing is not possible. Instead, if the nodes know their own locations and that of the sink, then it is possible to dynamically select forwarding nodes that are successively nearer to the sink. This is called *geographical routing,* and has been widely studied as a simple scalable

approach for routing in sensor networks [2], [3], [4], [5]. For the purpose of location determination, low cost GPS devices are now becoming available, and can be incorporated in the nodes; alternatively, approximate localization algorithms based on various geometrical principles can also be used (see, for example, [6], [7]). For a survey on routing and localization, see [8], [9]. In this paper we assume that nodes know their exact locations and also the location of the sink.

The relay node selection problem: In geographical forwarding, in our setting, there arises the problem of optimal relay node selection, which we now discuss. One approach is that of *greedy forwarding*, in which an intermediate node forwards the packet to its neighbor node that makes maximum progress towards the sink. This scheme is referred to as Most Forward within Radius (MFR) ([2], [3]). If the node density is large such that every node has a neighbor that is closer to the sink than itself, then the greedy approach can find routes close to the minimum hop paths. Following a minimum hop path is beneficial since it reduces the number of times the network needs to transmit the packet.

However, when the nodes are sleep-wake cycling in an asynchronous manner there is a trade-off between the delay in relay node selection and the progress made towards the sink. For example, if MFR is implemented, then for an intermediate node to forward the packet to a relay node that makes the maximum progress towards the sink, the intermediate node will need to wait for all its neighbors closer to the sink than itself to wake up. This will result in an increase in the delay of the alarm that is being forwarded. In fact, a counterpart to the MFR policy could be the policy that forwards the packet to the first node that wakes up and is nearer to the sink than the intermediate node. In this paper we call this latter policy First Forward (FF), and the MFR policy, simply, Max Forward (MF).

In this paper we study the above trade-off for the following one hop relaying problem. A node needs to forward a packet to the sink. There is a set of neighbors of the node that are nearer to the sink than the node; the *forwarding set*. The nodes are asynchronously sleep-wake cycling according to a certain model. We seek policies for relay node selection so as to minimize the delay in determining the relay node, subject to a constraint on the progress made towards the sink. We assume

that each node has at least one neighbor that is strictly closer to the sink than itself so that greedy forwarding will always find a path to sink. This is a reasonable assumption for large node densities.

Our contributions:

- The problem of minimizing average one hop delay subject to a constraint on the average progress made, when nodes wake up periodically, but not synchronously, is formulated as a Markov decision problem (MDP), and solved to yield the optimal policy which we call Best Forward (BF). See Section IV and Section V.

- In a mathematically simplified setting (*i.i.d.*, exponentially distributed inter-wakeup times) the MDP approach is used to derive a threshold type policy, called Simplified Forward (SF). The threshold is a function of the constraint on progress, and the policy is to transmit to the first node which wakes up and makes a progress of more than the threshold. See Section VI. While such a policy has been proposed heuristically in previous works ([10], [11]), we have derived it from the MDP formulation and we show through simulations that the performance of this policy is close to that of BF. The simulation results are in Section VII.

- Finally, we compare the end to end performance (average delay and hop counts) of the SF policy with the forwarding policy proposed by Kim et al. [1]. The approach of Kim et al. aims to achieve minimum average end-to-end delay, but at the expense of an initial configuration phase. The SF policy, however, does not need any global organization phase, and the progress constraint can be used to *tune* the end-to-end performance to suitably trade-off between end-to-end delay and the number of hops in the forwarding path. These results are reported in Section VII.

II. RELATED WORK

Zorzi and Rao ([12]) consider a scenario similar to ours: geographical forwarding in a wireless mesh network in which the nodes know their locations, and are sleep-wake cycling. They propose GeRaF (Geographical Random Forwarding), a distributed relaying algorithm, whose objective is to carry a packet to its destination in as few hops as possible, by making as large progress as possible at each relaying stage. Thus, the objective is similar to the MFR algorithm, mentioned above ([2], [3]). For their algorithm, the authors obtain the average number of hops (for given source-sink distance) as a function of the node density. These authors do not consider the trade-off between relay selection delay and the progress towards the sink, which is a major contribution of our work.

Liu et al. ([11]) propose a relay selection approach as a part of CMAC, a protocol for geographical packet forwarding. With respect to the fixed sink, a node i has a forwarding set consisting of all nodes that make progress greater than r_0 (an algorithm parameter). If Y represent the delay until the first wake-up instant of a node in the forwarding set, and X is the corresponding progress made, then, under CMAC,

node i chooses an r_0 that minimizes the expected normalized latency $\mathbb{E}[\frac{Y}{X}]$. The Random Asynchronous Wakeup (RAW) protocol ([10]) also considers transmitting to the first node to wake up that makes a progress greater than a threshold Th. Interestingly, this is also the structure of the optimal policy provided by one of our Markov decision process formulations.

Kim et al. ([1]) consider a dense WSN in which the traffic model and sleep-wake cycling are similar to ours. An occasional alarm packet needs to be sent, from wherever in the network it is generated, to the sink. The nodes are asynchronously sleep-wake cycling. The authors develop an optimal anycast scheme to minimize average end-to-end delay from any node i to the sink. The optimization is also done over sleep-wake cycling patterns and rates. A dynamic programming approach is taken, with the stages being the number of hops to the sink. While the framework is similar to ours, Kim et al. do not consider the objective of spatial progress at each hop, which results in the reduction of hop counts along the forwarding paths, and thus in the reduction of node energy utilization. In our work, we have studied the trade-off, at a typical forwarding stage, between forwarding delay and the distance that the packet covers in the hop.

Rossi et al. ([13]) consider the problem of geographical forwarding in a wireless sensor network in which each node knows its hop distance from the sink. For each link, there is a link cost (for example, energy cost) for forwarding a packet over that link. Thus, there are two end-to-end cost criteria for a forwarding path: the total link cost of the path, and the number of hops in the path. When a node, say i, has a packet to forward to the sink, it has to consider the trade-off between cost reduction and hop distance reduction; note that cost can be reduced by forwarding the packet to a neighbor node with the same hop distance to the sink, but using which the total link cost could be lower. The information available at i is the cost to all its neighbors, and the statistics of the costs-to-go from the neighbors. The major difference in our work is that we have a sequential decision problem *at each stage*, since the costs (wake-up delay) and rewards (progress towards the sink) are revealed as the nodes wake up, and only the statistics are known a priori.

Chaporkar and Proutiere ([14]) consider the problem of a transmitter that needs to transmit over one of several available channels. The transmitter can probe the channels to determine channel state information in order to encode its transmissions. The trade-off is between the time taken to probe and the throughput advantage of finding a good channel. Some important differences between their model and ours are the following. In our work the trade-off is between the time taken to wait for a relay to wake up, and the *spatial* progress the relay makes towards the sink. In [14], the transmitter can use an unprobed channel, whereas in our problem a relay that has not yet woken up cannot be used. In [14], the transmitter can probe the channels in an order that it can choose (e.g., the stochastically best channel first); in our problem the relays wake up in a random order that is not under the control of the transmitter. In [14] it is shown that if the use of an

unprobed channel is not allowed then a one-step-look-ahead rule is optimal. This is similar to the solution we obtain for a simplified version of our model. Note that whereas the concern in [14] is only with one-step relaying, we also study how the one-step policy performs in terms of end-to-end objectives, namely, path delay and path hop count.

III. SYSTEM MODEL

A. Node Deployment

N identical sensor nodes are uniformly deployed in the square region $[0, L]^2$. We take N to be a Poisson random variable of rate λL^2 where λ is the node density. Let x_i, $i = 1, 2, ..., N$, be the locations of the nodes. Additional *source* and *sink* nodes are placed at fixed locations $x_0 = (0, 0)$ and $x_{N+1} = (L, L)$ respectively. Thus including the source and sink nodes, there are a total of $N + 2$ nodes in the disk. r_c is the communication range of each node. Two nodes i and j are called neighbors if and only if $|x_i - x_j| \leq r_c$. The distance between node i and sink $(N + 1)$ is $L_i = |x_{N+1} - x_i|$.

B. The Sleep-Wake Process

To conserve energy, each node performs periodic sleep-wake cycling. The sleep-wake times of the nodes are *not synchronized*. Since we are interested in studying the delay incurred in routing due to sleep-wake cycling alone, we neglect the transmission delay, propagation delay and other overhead delays. This means that if node i has a packet to transmit to its neighboring node j, then i can transmit immediately at the instant j wakes up. We model this by taking the time for which a node stays awake to be zero.

More formally, let T_i, $i = 1, 2, ..., N + 1$ be *i.i.d.* random variables which are uniform on $[0, T]$, where T is the period of the sleep wake cycle. Then node i wakes up at the periodic instants $kT + T_i, k \geq 0$. We define the *waiting time for node i* to wake up at time t as,

$$W_i(t) = \inf\{kT + T_i \geq t : k \geq 0\} - t \qquad (1)$$

C. Forwarding Rules and Assumptions

Forwarding rules dictate the actions a node can take when it has to transmit. We are interested in decentralized policies where a node can take decisions only by observing the activities in its neighborhood (*i.e.*, the disk of radius r_c centered around the node of interest). In this regard we impose some restrictions on the network.

Traffic Model: There is a single packet in the network which is to be routed from the source to sink. At time 0, the packet is given to the source and the routing process begins. The nodes which get the packet for forwarding are called relay nodes. The packet traverses a sequence of relay nodes to eventually reach the sink, at which time the routing ends. Thus there is a single flow and further the flow consists of only one packet. This set up is reasonable, because in sensor networks we can assume that the events are sufficiently separated in time and/or location so that the flows due to two events do not intersect. To avoid multiple packet transmission by different nodes detecting the

same event, the nodes can resolve among themselves to select one node (say the one closest to the sink), which can then transmit. Further, the information about an event comprises its location, and possibly target classification data, which along with some control bits can be easily incorporated in a single packet. This justifies the idea to study the performance of a single packet alone.

Forwarding Set: Each node knows its location and the location of the sink. The *forwarding set* of a node is the set of its neighbors that are closer to the sink then itself. A relay node considers forwarding the packet only to a node in its forwarding set. Each node knows the number of neighbors in its forwarding set, but is not aware of their locations and wake times. While in this paper we assume that each node knows the number of nodes in its forwarding set, it would be desirable to develop forwarding algorithms that do not require even this knowledge. We leave this as future work, but in Section VII-B we provide simulation results on the performance of our algorithm when the node takes the number of nodes in its forwarding region to be just the expected number of nodes.

D. Some Notation

To define a forwarding policy more formally, we begin by setting up some notation. Consider a generic node i which gets the packet to forward at some instant t. Let $\mathcal{S}_i = \{y : |y - x_i| \leq r_c, |x_{N+1} - y| < L_i\}$. $\mathcal{S}_i$ is the set of all points that are within the communication radius of i and are strictly closer to the sink than i (see Fig. 1) (we ignore edge effects by assuming that $\mathcal{S}_i \subset [0, L]^2$). If $x_j \in \mathcal{S}_i$ then the progress made by j is $Z_j = L_i - L_j$. Let N_i be the number of nodes in $\mathcal{S}_i$. Note that $N_i \sim Poisson(\lambda|\mathcal{S}_i|)$, where $|\mathcal{S}_i|$ is the area of the region $\mathcal{S}_i$. Recall that node i knows N_i and hence we focus on the event $\{N_i = K\}$ for some $K > 0$.

Let the indices of the nodes in $\mathcal{S}_i$ be arranged as $i_1, ..., i_K$, such that $W_{i_1}(t) \leq W_{i_2}(t) \leq, ..., \leq W_{i_K}(t)$. The corresponding values of progress are $Z_{i_1}, Z_{i_2}, ..., Z_{i_K}$. For simplicity, from here on we neglect i in the subscript and simply use $W_1(t), ..., W_K(t)$ and $Z_1, Z_2, ..., Z_K$.
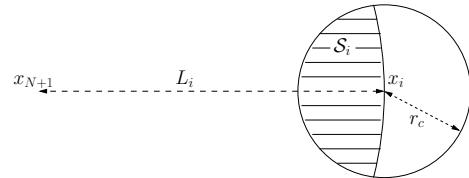

Fig. 1. x_i and x_{N+1} are the locations of node i and sink respectively. L_i is the distance between them, r_c is the communication radius. $\mathcal{S}_i$ is the set of all points that are within the communication radius of node i and closer to sink than i. $\mathcal{S}_i$ is the shaded region in the figure.

The locations of each of these K nodes are uniformly distributed in the region $\mathcal{S}_i$ independent of the others. Hence the progress made by them are *i.i.d.* whose distribution is same as Z. The *p.d.f.* of Z is supported on $[0, r_c]$ and is given by,

$$f_Z(z) = \frac{2(L_i - z) \cos^{-1}\left(\frac{L_i^2 + (L_i - z)^2 - r_c^2}{2L_i(L_i - z)}\right)}{|\mathcal{S}_i|} \qquad (2)$$

Where $|\mathcal{S}_i|$ denotes the area of the region $\mathcal{S}_i$,

$$|\mathcal{S}_i| = \int_0^{r_c} 2(L_i - z)\cos^{-1}\left(\frac{L_i^2 + (L_i - z)^2 - r_c^2}{2L_i(L_i - z)}\right)dz \tag{3}$$

Let $U_1 = W_1(t)$ and $U_k = W_k(t) - W_{k-1}(t)$ for $2 \leq k \leq K$. We refer to $\{U_k\}$ as the *inter-wakeup* times. These are the waiting times between the wakeup instants of sucessive nodes in $\mathcal{S}_i$ (see Fig. 2). Further U_k and Z_k are independent.
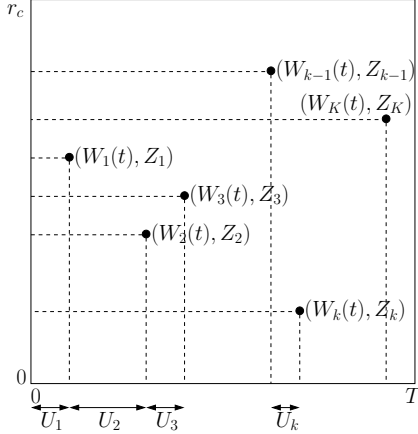

Fig. 2. $(W_k(t), Z_k)$ represents the wake instant and the progress respectively, made by the Node i_k in $\mathcal{S}_i$. These are shown as points in $[0,T]\times[0,r_c]$. U_k is the inter-wakeup time between node i_k and i_{k-1}.

The waiting times $W_1(t), W_2(t), ..., W_K(t)$ are the order statistics of K *i.i.d.* random variables that are uniform on $[0,T]$. The *p.d.f.* of the $k-th$ order statistics is [15, Chapter 2],

$$f_{W_k}(u) = \frac{K!u^{k-1}(T-u)^{K-k}}{(k-1)!(K-k)!T^K} \tag{4}$$

for $0 \leq u \leq T$. Also the joint *p.d.f.* of the $k-th$ and $l-th$ order statistics (for $k < l$) is [15, Chapter 2],

$$f_{W_k,W_l}(u,v) = \frac{K!u^{k-1}(v-u)^{l-k-1}(T-v)^{K-l}}{(k-1)!(l-k-1)!(K-l)!T^K} \tag{5}$$

for $0 \leq u < v \leq T$. Later we will be interested in the conditional *p.d.f.* $f_{U_{k+1}|W_k}$ for $1 \leq k \leq K-1$. Using the above equations we can write ,

$$\begin{aligned} f_{U_{k+1}|W_k}(u|w) &= \frac{f_{W_k,W_{k+1}}(w, w+u)}{f_{W_k}(w)} \\ &= (K-k)\frac{(T-w-u)^{K-k-1}}{(T-w)^{K-k}} \end{aligned} \tag{6}$$

for $0 \leq w \leq T$ and $0 \leq u \leq T-w$.

E. Single Hop Policy

Decision process begins at the instant t at which node i gets the packet to forward. This is *stage* $k = 0$. The $k-th$ ($k \geq 1$) decision instant is the time at which node i_k wakes up.
A *Single Hop (SH) policy* π is a sequence of mappings $\{\mu_k^\pi : 0 \leq k \leq K\}$, where $\mu_0^\pi : \{(0,0)\} \to \{0\}$ and for $k \geq 1$ $\mu_k^\pi :$ $[0,T] \times [0, r_c] \to \{0, 1\}$. π should also satisfy $\mu_K^\pi(w, b) = 1$. **The function μ_k^π maps the state at stage k to an action** 0 **(continue) or** 1 **(stop)**. Let $D^\pi(t)$ and $Z^\pi(t)$ denote the delay incurred and progress made by node i using policy π. Forwarding rules for node i, using policy π are as follows:

- At stage 0, node i has to wait for further nodes to wake up. We represent this by allowing the only *state* at stage 0 to be $\mathbf{0} = (0, 0)$ and the corresponding action to be to 0 (continue to wait) *i.e.*, $\mu_0^\pi(\mathbf{0}) = 0$.
- If $L_i \leq r_c$, then wait for sink to wake up and transmit to it. In this case, the delay and progress made are $D^\pi(t) = W_{N+1}(t)$ and $Z^\pi(t) = L_i$ respectively.
- Otherwise (*i.e.*, if $L_i > r_c$), wait for the nodes in $\mathcal{S}_i$ to wake up. When node i_k wakes up ($1 \leq k \leq K$), evaluate $p = \mu_k^\pi(W_k(t), b_k)$ where $b_k = \max\{Z_1, ..., Z_k\}$. If $p = 1$, then transmit to the node $i_{\arg\max\{Z_1,...,Z_k\}}$. The delay incurred is $D^\pi(t) = W_k(t)$ and the progress made is $Z^\pi(t) = b_k$. If $p = 0$, ask the node which makes the most progress so far to stay awake, put the other node to sleep and wait for further nodes to wake up.
- The requirement $\mu_K^\pi(w, b) = 1$ in the definition of π ensures that node i transmits at or before the instant the last node wakes up.

Since the distribution of $\{(W_k(t), Z_k) : 1 \leq k \leq K\}$ are not dependent on the value of t, the average values of $D^\pi(t)$ and $Z^\pi(t)$ also do not depend on t. Hence to compute these average values we can, without loss of generality, take $t = 0$ and use D^π and Z^π to simplify the notation.

Let Π represent the class of all SH policies. **Note that many policies are excluded from class Π. For instance, the policy which waits for all the nodes to wake up and then transmits to the one which makes least progress does not belong to the class Π. This is because for a policy in Π, transmission is allowed only to the node that makes the most progress so far.** We would like to explicitly mention two SH policies namely Max Forward (MF) and First Forward (FF):
A node using **Max Forward** policy will wait for all the nodes in its forwarding set to wake up and then transmit to the one which makes most progress. We use π_{MF} to represent this policy. For this policy, $\mu_k^{\pi_{MF}}(w, b) = 1$ if and only if $k = K$. This policy obtains maximum delay and maximum progress among all other policies in class Π.
A node using **First Forward** policy will always transmit to the node in the forwarding set which wakes up first irrespective of the progress made by it. π_{FF} is used to represent this policy. For this policy, $\mu_1^{\pi_{FF}}(w, b) = 1$. π_{FF} obtains minimum delay and minimum progress among all the policies in class Π.

IV. PROBLEM FORMULATION

From here on, without loss of generality we fix $T = 1$ and $r_c = 1$. Let $\mathbb{P}_K$ (where $K \geq 1$) denote the probability law conditioned on the event $\{N_i = K\}$ *i.e.*, $\mathbb{P}_K(.) = \mathbb{P}(.|N_i = K)$. Similarly we define the conditional expectation $\mathbb{E}_K$. Define $\gamma_{MF} = \mathbb{E}_K[Z^{\pi_{MF}}]$ and $\gamma_{FF} = \mathbb{E}_K[Z^{\pi_{FF}}]$, average progress made by the MF and FF policies respectively.

Our interest in this work are, at a relay node i with $N_i = K$, to minimize the average delay subject to a constraint on the average progress achieved. More formally the problem is,

$$\min_{\pi \in \Pi} \quad \mathbb{E}_K[D^\pi] \tag{7}$$
$$\text{s.t.} \quad \mathbb{E}_K[Z^\pi] \geq \gamma$$

where $\gamma \in [0, \gamma_{MF}]$.

This formulation embodies the one-step tradeoff between the need to forward the packet quickly while attempting to make substantial progress towards the sink. The parameter γ controls the tradeoff. A large γ indicates our desire to make large progress in each step, which will come at a cost of a large one hop forwarding delay.

To solve the problem in (7), we consider the following unconstrained problem,

$$\min_{\pi \in \Pi} \quad \mathbb{E}_K[D^\pi] - \eta \mathbb{E}_K[Z^\pi] \tag{8}$$

Where $\eta > 0$. Let $\pi_{BF}(\eta)$ (Best Forward) be the optimal solution for this problem.

Lemma 1: For a given γ in problem (7), suppose there is an η_γ such that $\mathbb{E}_K[Z^{\pi_{BF}(\eta_\gamma)}] = \gamma$, then $\pi_{BF}(\eta_\gamma)$ is optimal for the problem in (7) as well.

Proof: Since $\pi_{BF}(\eta_\gamma)$ is optimal for the problem in (8),

$$\mathbb{E}_K[D^{\pi_{BF}(\eta_\gamma)}] - \eta_\gamma \mathbb{E}_K[Z^{\pi_{BF}(\eta_\gamma)}] \leq$$
$$\mathbb{E}_K[D^\pi] - \eta_\gamma \mathbb{E}_K[Z^\pi], \text{ for all } \pi \in \Pi$$

i.e., $\mathbb{E}_K[D^{\pi_{BF}(\eta_\gamma)}] \leq \mathbb{E}_K[D^\pi] - \eta_\gamma(\mathbb{E}_K[Z^\pi] - \gamma)$

Therefore for any π such that $\mathbb{E}_K[Z^\pi] \geq \gamma$, we have

$$\mathbb{E}_K[D^{\pi_{BF}(\eta_\gamma)}] \leq \mathbb{E}_K[D^\pi]$$

∎

In the subsequent sections we focus on solving the problem in (8).

V. OPTIMAL POLICY FOR THE EXACT MODEL

To solve the problem in (8), we develop it in a Markov Decision Process (MDP) framework [16]. $\mathcal{X} = [0,1]^2 \bigcup \{\psi\}$ is the state space (recall that $T = 1$ and $r_c = 1$). ψ is the terminating state. $\mathcal{C} = \{0,1\}$ is the control space where 1 is for *stop* and 0 is for *continue*. A small change to the π defined earlier in section (III-E), is the inclusion of ψ in the domain of μ_k^π. Let (w_k, b_k) be the state at stage k where b_k is the best (maximum) progress made by the nodes waking up until stage k *i.e.,* $b_k = \max\{Z_1, ..., Z_k\}$. Conditioned on being in state (w_k, b_k) at stage k, transition to the next state depends on w_k through U_{k+1} whose *p.d.f.* is $f_{U_{k+1}|W_k}(.|w_k)$ (Equation (6)). The other disturbance component Z_{k+1}, is independent of the (w_k, b_k). *p.d.f.* of Z_{k+1} is f_Z (Equation (2)). We define the conditional expectation,

$$\mathbb{E}_{(W_k = w_k)}[.] = \mathbb{E}_K[.|W_k = w_k]$$

Then using expression (6) we can write,

$$\mathbb{E}_{(W_k = w_k)}[U_{k+1}] = \frac{1 - w_k}{K - k + 1} \tag{9}$$

Initial state $s_0 = \mathbf{0}$ and initial action $a_0 = 0$ always. Therefore the next state is $s_1 = (U_1, Z_1)$ and the cost incurred at stage 0 is $g_0(\mathbf{0}, 0) = U_1$. If $a_k \in \mathcal{C}$ is the action taken at stage $1 \leq k \leq K - 1$, then the next state s_{k+1} is,

$$s_{k+1} = \begin{cases} (w_k + U_{k+1}, \max\{Z_{k+1}, b_k\}) & \text{if } a_k = 0 \\ \psi & \text{if } a_k = 1 \end{cases}$$

and the one step cost function is,

$$g_k((w_k, b_k), a_k) = \begin{cases} U_{k+1} & \text{if } a_k = 0 \\ -\eta b_k & \text{if } a_k = 1 \end{cases} \tag{10}$$

If the state at stage k is ψ then $s_{k+1} = \psi$ and $g_k(\psi, a_k) = 0$ irrespective of a_k. Also if s_K is the state of the system at the last stage, there is a cost of termination, $g_K(s_K)$ given as,

$$g_K(s_K) = \begin{cases} 0 & \text{if } s_K = \psi \\ -\eta b_K & \text{otherwise} \end{cases}$$

The total average cost incurred with policy π is,

$$J_\pi(\mathbf{0}) = \mathbb{E}_K\left[\sum_{k=0}^{K-1} g_k(s_k, \mu_k^\pi(s_k)) + g_K(s_K)\right]$$

The expectation in the cost function above is taken over the joint distribution of $\{(U_k, Z_k) : 1 \leq k \leq K\}$. Note that,

$$J_\pi(\mathbf{0}) = \mathbb{E}_K[D^\pi] - \eta \mathbb{E}_K[Z^\pi]$$

Therefore the optimal cost is,

$$J^*(\mathbf{0}) = \min_\pi J_\pi(\mathbf{0}) = J_{\pi_{BF}(\eta)}(\mathbf{0})$$

Let $J_k(w, b)$ be the optimal cost to go when the system is in state (w, b) at stage $1 \leq k \leq K$. When the stage is K (*i.e.,* all the nodes have woken up), then invariably transmission has to happen. Therefore,

$$J_K(w, b) = -\eta b$$
$$= -\eta \max\{b, \phi_K(w, b)\} \tag{11}$$

where, we define $\phi_K(w, b) = 0$ for all (w, b). Next when there is one more node to wake up (*i.e.,* stage is $K - 1$) then both actions, $a_{K-1} = 1$ and $a_{K-1} = 0$ are possible. Therefore,

$$J_{K-1}(w, b) = \min\left\{-\eta b, \mathbb{E}_{(W_{K-1} = w)}[U_K + J_K(w + U_K, \max\{b, Z_K\})]\right\}$$

The terms in the min expression are the costs when $a_{K-1} = 1$ (stop) and $a_{K-1} = 0$ (continue) respectively. Using the expression for J_K in (11) we obtain,

$$J_{K-1}(w, b)$$
$$= \min\left\{-\eta b, \mathbb{E}_{(W_{K-1} = w)}[U_K - \eta \max\{b, Z_K, \phi_K(w + U_K, \max\{b, Z_K\})\}]\right\}$$
$$= -\eta \max\{b, \phi_{K-1}(w, b)\} \tag{12}$$

where,

$$\phi_{K-1}(w, b) = \mathbb{E}_{(W_{K-1} = w)}[\max\{b, Z_K, \phi_K(w + U_K, \max\{b, Z_K\})\} - \frac{U_K}{\eta}] \tag{13}$$

The following lemma is obtained easily.

Lemma 2: For every $1 \leq k \leq K - 1$, the following equations holds,

$$J_k(w, b) = -\eta \max\{b, \phi_k(w, b)\} \qquad (14)$$

where,

$$
\begin{aligned}
\phi_k(w, b) &= \mathbb{E}_{(W_k=w)}[\max\{b, Z_{k+1}, \phi_{k+1}(w + U_{k+1}, \\
&\quad \max\{b, Z_{k+1}\})\} - \frac{U_{k+1}}{\eta}]
\end{aligned}
\qquad (15)
$$

Proof: Suppose for some $2 \leq k \leq K - 1$ equations (14) and (15) holds, then following similar lines which was used to obtain (12) and (13) (just replace K by k) we can show that (14) and (15) holds for $k - 1$ as well. Since we have already shown that these equations hold for $k = K - 1$, from induction argument we can conclude that it holds for every $1 \leq k \leq K - 1$. ∎

The structure of the optimal policy is given in the following corollary.

Corollary 3: The optimal policy $\pi_{BF}(\eta)$ is of the following form,

$$
\mu_k^{\pi_{BF}(\eta)}(w, b) = \begin{cases} 1 & \text{if } b \geq \phi_k(w, b) \\ 0 & \text{otherwise} \end{cases}
\qquad (16)
$$

for $1 \leq k \leq K$. Where $\phi_K(w, b) = 0$ for all $(w, b) \in \mathcal{S}$ and for $1 \leq k \leq K - 1$, $\phi_k(w, b)$ is given in equation (15). ∎

Remarks: The optimal policy requires threshold functions $\{\phi_k\}$ which are computionally intensive. For our later numerical work in Section (VII), we discretize the state space into 10^4 equally spaced points and use the approximate values of the functions $\phi_k, 1 \leq k \leq K - 1$ at these discrete points.

VI. OPTIMAL POLICY FOR A SIMPLIFIED MODEL

The random variables $\{U_k : 1 \leq k \leq K\}$ are identically distributed [15, Chapter 2] (but not independent). Their common *c.d.f.* is $F_{U_k}(u) = 1 - (1 - u)^K$. From Fig. 3 we observe that the *c.d.f.* of $\{U_k : 1 \leq k \leq K\}$ is close to that of the *c.d.f.* of an exponential random variable of parameter K and the approximation becomes better for large values of K. This motivates us to consider a *simplified model* where $\{U_k : 1 \leq k \leq K\}$ are distributed as *Exponential(K)*. Further in our simplified model we take these random variables to be independent.

For the simplified model, the cost function (similar to (10)) when the system is in state (w, b) at stage $1 \leq k \leq K - 1$ is,

$$
g_k((w_k, b_k), a_k) = \begin{cases} U_{k+1} & \text{if } a_k = 0 \\ -\eta b_k & \text{if } a_k = 1 \end{cases}
\qquad (17)
$$

We observe that due to the *i.i.d.* inter-wake time assumption the cost function is not dependent on the value of w_k. Also we need not consider conditioning on $W_k = w_k$ unlike in the previous section since the *p.d.f.* of U_{k+1} does not depend on w_k. Hence, the optimal policy for this model is going to be independent of w_k for each k. So we simplify the state space by ignoring the values of w_k for each k, *i.e.,* the state space is $\bar{\mathcal{X}} = [0, 1] \bigcup \{\psi\}$. Control space $\mathcal{C}$ and the other
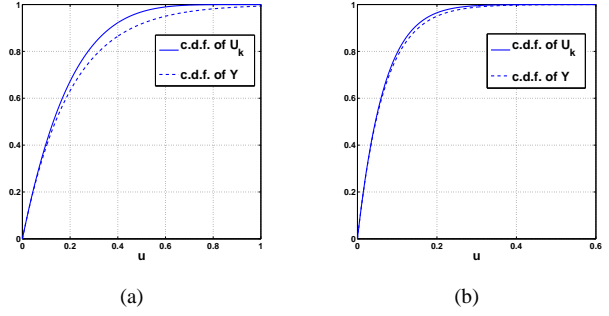

Fig. 3. The *c.d.f.*'s F_{U_k} and F_Y where $Y \sim Exponential(K)$ are plotted for (a) $K = 5$ and (b) $K = 15$.

disturbance component Z_k remain the same. Since the state space is different, we make a small change to the definition of policy π by allowing $\mu_k^\pi : \bar{\mathcal{X}} \to \mathcal{C}$. The state transition and cost functions remain same as in the previous section with (w, b) replaced by b. Let $\pi_{SF}(\eta)$ represent the optimal policy for this model.

Let $J_k(b)$ be the optimal cost to go at stage k when the state is b. Then, for all $b \in [0, 1]$,

$$J_K(b) = -\eta b \qquad (18)$$

Next when the stage is $K - 1$, for $b \in [0, 1]$,

$$
\begin{aligned}
J_{K-1}(b) &= \min\{-\eta b, \mathbb{E}_K[U_K + J_K(\max\{b, Z_K\})]\} \\
&= \min\{-\eta b, \mathbb{E}_K[U_K - \eta \max\{b, Z_K\}]\} \\
&= -\eta \max\{b, \beta_1(b)\}
\end{aligned}
\qquad (19)
$$

where β_1 is a function, which for $b \in [0, 1]$ is given by,

$$
\begin{aligned}
\beta_1(b) &= \mathbb{E}_K[\max\{b, Z_K\}] - \frac{\mathbb{E}_K[U_K]}{\eta} \\
&= \mathbb{E}_K[\max\{b, Z\}] - \frac{1}{\eta K}
\end{aligned}
\qquad (20)
$$

Here we have made use of the fact that $\mathbb{E}_K[U_K] = \frac{1}{K}$ and $Z_K \sim Z$. The *p.d.f.* of Z is given in (2). Evidently, at stage $K - 1$, the optimal action is to stop and transmit the packet if $b \geq \beta_1(b)$ and to continue otherwise. The following results about $\beta_1(b)$ can easily be obtained. A detailed proof is provided in the report [17].

Lemma 4:

1) β_1 is continuous, increasing and convex in b.
2) If $\beta_1(0) < 0$, then $\beta_1(b) < b$ for all $b \in [0, 1]$.
3) If $\beta_1(0) \geq 0$, then there is a unique α_η such that $\beta_1(\alpha_\eta) = \alpha_\eta$.
4) If $\beta_1(0) \geq 0$, then $\beta_1(b) < b$ for $b \in (\alpha_\eta, 1]$ and $\beta_1(b) > b$ for $b \in [0, \alpha_\eta)$.

∎

If $\beta_1(0) < 0$, then define $\alpha_\eta = 0$. Otherwise α_η is defined by $\beta_1(\alpha_\eta) = \alpha_\eta$. Then

$$
\mu_{K-1}^{\pi_{SF}(\eta)}(b) = \begin{cases} 1, & \text{if } b \geq \alpha_\eta \\ 0, & \text{otherwise} \end{cases}
$$

We proceed to evaluate J_{K-2}.

$$
\begin{aligned}
J_{K-2}(b) \\
= \min\{-\eta b, \mathbb{E}_K[U_{K-1} + J_{K-1}(\max\{b, Z_{K-1}\})]\} \\
= \min\{-\eta b, \mathbb{E}_K[U_{K-1} - \eta \max\{b, Z_{K-1}, \\
\beta_1(\max\{b, Z_{K-1}\})\}]\} \\
= -\eta \max\{b, \beta_2(b)\} \quad\quad (21)
\end{aligned}
$$

where,

$$
\beta_2(b) = \mathbb{E}_K[\max\{b, Z, \beta_1(\max\{b, Z\})\}] - \frac{1}{\eta K} \quad (22)
$$

Lemma 5: $\beta_2(b) \geq \beta_1(b)$ for any $b \in [0, 1]$. In particular, if $b \geq \alpha_\eta$ then $\beta_2(b) = \beta_1(b)$.

Proof: The first part follows easily because $\mathbb{E}_K[\max\{b, Z\}] \leq \mathbb{E}_K[\max\{b, Z, \beta_1(\max\{b, Z\})\}]$.

Next, if $b \geq \alpha_\eta$ then from Lemma 4, $\max\{b, Z\} \geq \beta_1(\max\{b, Z\})$, so that $\max\{b, Z, \beta_1(\max\{b, Z\})\} = \max\{b, Z\}$. Therefore,

$$
\beta_2(b) = \mathbb{E}_K[\max\{b, Z\}] - \frac{1}{\eta K}
$$

■

Lemma 6: For every $1 \leq k \leq K - 2$ the following holds,

$$
J_k(b) = -\eta \max\{b, \beta_{K-k}(b)\} \quad (23)
$$

where,

$$
\beta_{K-k}(b) = \mathbb{E}_K[\max\{b, Z, \beta_{K-(k+1)}(\max\{b, Z\})\}] - \frac{1}{\eta K}
$$

and has the property, $\beta_{K-k}(b) \geq \beta_{K-(k+1)}(b)$ for any $b \in [0, 1]$. In particular, if $b \geq \alpha_\eta$ then $\beta_{K-k}(b) = \beta_1(b)$.

Proof: Proof is along the lines used to obtain Equations (21), (22) and Lemma 5. ■

Corollary 7: The policy $\pi_{SF}(\eta)$ is of the following form, $\mu_K^{\pi_{SF}(\eta)}(b) = 1$ and

$$
\mu_k^{\pi_{SF}(\eta)}(b) = \begin{cases} 1 & \text{if } b \geq \alpha_\eta \\ 0 & \text{otherwise} \end{cases} \quad (24)
$$

for $1 \leq k \leq K - 1$. ■

Remarks: The policy is a simple one-step-look-ahead rule where at each k ($1 \leq k \leq K - 1$) the policy compares the cost of stopping at k ($C_s = -\eta b$) with the cost of continuing for one more step and then stopping at $k + 1$ ($C_c = \frac{1}{K} - \eta \mathbb{E}_K[\max\{b, Z\}]$). The policy is to stop if $C_s \leq C_c$ (simplification yields, stop if $b \geq \alpha_\eta$), continue otherwise. The policy is to transmit to the first node which makes a progress of more than α_η. If all the nodes, make progress of less than α_η then transmit to the node whose progress is maximum at the instant the last node wakes up.

VII. Simulation Results

A. One Hop Performance

We apply the policies $\pi_{BF}(\eta)$ and $\pi_{SF}(\eta)$ to the actual model and obtain average progress and average one hop delay for $L_i = 10$ and $K = 5$. Expressions for the average values for policies $\pi_{SF}(\eta)$, π_{FF} and π_{MF} are available in the detailed technical report [17]. Since it is difficult to obtain similar analytical expressions for policy $\pi_{BF}(\eta)$, we have performed simulations to obtain these values. In Figs. 4(a) and 4(b) we plot the average values as a function of η. The minimum and maximum values of average delay and progress are achieved by π_{FF} and π_{MF} respectively. From the figures we can observe that for values of η less than $\eta_o = \frac{1}{\mathbb{E}_K[Z]K}$ the performance of $\pi_{SF}(\eta)$ is same as π_{FF}. This is because for η less than η_o, we have $\beta_1(0) < 0$, and therefore the threshold used is $\alpha_\eta = 0$ which is same as that used by π_{FF}.

By using a large value of η, a node will value progress more and will end up waiting for better nodes to wake up thus incurring a large delay as well. Hence, delay and progress for both the policies (π_{BF} and π_{SF}) are increasing with η. We can conclude from Lemma 1, that for each policy, BF or SF, and a given η, the corresponding delay value is the minimum that can be obtained using that policy, subject to a constraint on progress equal to the progress value obtained for that η. These corresponding average delay vs. average progress values are shown in Fig. 4(c), for $K = 3, 5$ and 15. Each point on the curve for each K corresponds to a different value of η, which increases along the curves as shown. We see that the performance of the SF policy is close to that of the optimal BF policy, even for small values of K. The way η serves to trade-off one hop progress and delay is clearly shown by these curves.

B. End to End Performance

Although our policies have been developed for one-hop optimality, it is interesting to study their end-to-end performance if they were used, heuristically, at each hop. We compare the end-to-end performance of our policy with the work of Kim et al. [1] who have developed end-to-end delay optimal geographical forwarding in a setting similar to ours. We first give a brief description of their work. They minimize, for a given network, the average delay from any node to the sink when each node i wakes up asynchronously with rate r_i. They show that periodic wake up patterns obtain minimum delay among all sleep-wake patterns with the same rate. A relay node with a packet to forward, transmits a sequence of beacon-ID signals. They propose an algorithm called LOCAL-OPT [18] which yields, for each neighbor j of node i, an integer $h_j^{(i)}$ such that if j wakes up and listens to the $h-th$ beacon signal from node i and if $h \leq h_j^{(i)}$, then j will send an ACK to receive the packet from i. Otherwise (if $h > h_j^{(i)}$) j will go back to sleep. A *configuration phase* is required to run the LOCAL-OPT algorithm.

As before, we fix $r_c = 1$ and $T = 1$ *sec*. Each node wakes up periodically with rate $\frac{1}{T}$ but asynchronously. To make a fair
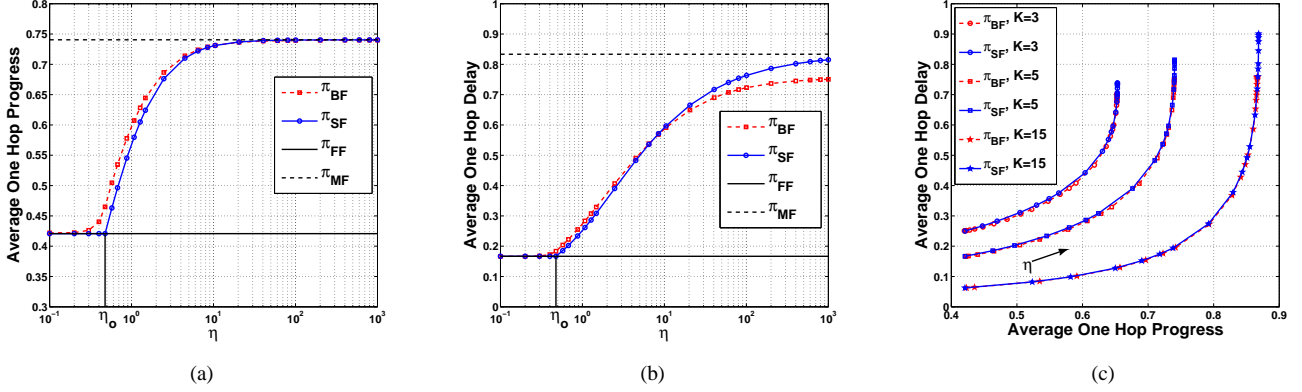
Fig. 4. One Hop Performance: (a): Average one hop progress as a function of η for various policies. The plots are for $L_i = 10$ and $K = 5$. Maximum and minimum progress are achieved by π_{MF} and π_{FF} respectively. (b): Average one hop delay as a function of η for various policies. The plots are for $L_i = 10$ and $K = 5$. Maximum and minimum delay are achieved by π_{MF} and π_{FF}. (c): Average one hop delay vs. the corresponding average one hop progress for the class of policies π_{BF} and π_{SF} are plotted for $K = 3, 5$ and 15. The parameter η controls the delay-progress trade-off. Each point on the curve corresponds to a different value of η which increases along the direction shown.

comparision with the work of Kim et al. we introduce beacon-ID signals of duration $t_I = 5$ *msec* and packet transmission duration of $t_D = 30$ *msec*. We fix a network by placing N nodes randomly in $[0, L]^2$ where $L = 10$. N is sampled from *Poisson(λL^2)* where $\lambda = 5$. Additional source and sink nodes are placed at locations $(0, 0)$ and (L, L) respectively. Further we have considered a network where the forwarding set of each node is non-empty. The wake times of the nodes are sampled independently from *Uniform([0,1])*. Description of the policies that we have implemented is given below.

π_{SF}: We fix γ as a network parameter. Each relay node chooses an appropriate η (in other words, chooses an appropriate threshold α_η) such that the average one hop progress made using the policy $\pi_{SF}(\eta)$ is equal to γ. Note that η depends on node i (*i.e.*, on the values of L_i and K). At a relay node i if γ is less (greater) than the average progress made by π_{FF} (π_{MF}) then we allow node i to use π_{FF} (π_{MF}) to forward. When a node j wakes up and if it hears a beacon signal from i, it waits for the ID signal and then sends an ACK signal containing its location information. If the progress made by j is more than the threshold, then i forwards the packet to j (packet duration is $t_D = 30$ *msec*). If the progress made by j is less than the threshold, then i asks j to stay awake if its progress is the maximum among all the nodes that have woken up thus far, otherwise i asks j to return to sleep. If more than one node wakes up during the same beacon signal, then contentions are resolved by selecting the one which makes the most progress among them. In the simulation, this happens instantly (as also for the Kim et al. algorithm that we compare with); in practice this will require a splitting algorithm; see, for example, [19, Chapter 4.3]. We assume that within $t_I = 5$ *msec* all these transactions (beacon signal, ID, ACK and contention resolution if any) are over. π_{FF} and π_{MF} can be thought of as special cases of π_{SF} with thresholds of 0 and 1 respectively.

$\hat\pi_{SF}$: This is same as π_{SF} except that here a relay node does not know K, but *estimates* its value as $\lfloor \lambda|\mathcal{S}_i| \rfloor$ nodes where
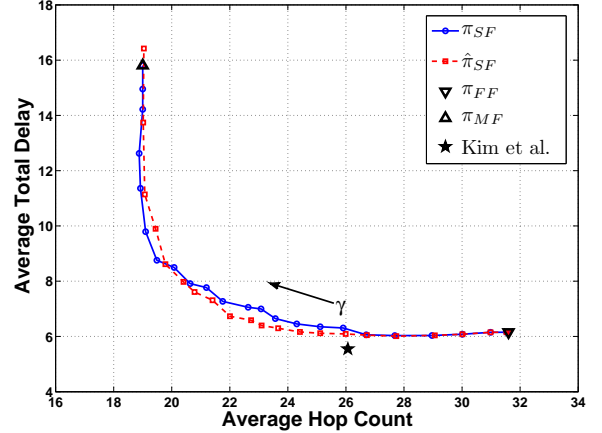

Fig. 5. End-to-end performance: Plot of average end-to-end delay vs. average end-to-end hop count when the one hop optimal policy for the progress constraint γ is used at each hop. The operating points of the policies π_{FF}, π_{MF} and Kim et al. are also shown in the figure. Each point on the curve corresponds to a different value of γ which increases along the direction shown.

$|\mathcal{S}_i|$ is the area of the region $\mathcal{S}_i$ (Equation (3)). If there is no eligible node even after the $\frac{T}{t_I} - th$ beacon signal (one case when this is possible is when the actual number of nodes K is less than $\lfloor \lambda|\mathcal{S}_i| \rfloor$ and none of the nodes make a progress of more than the threshold) then i will select one which makes the maximum progress among all nodes.

Kim et al.: We run the LOCAL-OPT algorithm [18] on the network and obtain the values $h_j^{(i)}$ for each pair (i, j) where i and j are neighbors. We use these values to route from source to sink in the presence of sleep wake cycling. Contentions, if any, are resolved (instantly, in the simulation) by selecting a node j with the highest $h_j^{(i)}$ index.

In Fig. 5 we plot average total delay vs. average hop count for different policies for fixed node placement, while the averaging is over the wake times of the nodes. Each point

on the curve is obtained by averaging over 1000 transfers of the packet from the source node to the sink. As expected, Kim et al. achieves minimum average delay. In comparision with π_{FF}, Kim et al. also achieves smaller average hop count. Notice, however that using π_{SF} policy and properly choosing γ, it is possible to obtain hop count similar to that of Kim et al., incurring only slightly higher delay.

The advantage of π_{SF} over Kim et al. is that there is *no need for a configuration phase*. Each relay node has to only compute a threshold that depends on the parameter γ which can be set as a network parameter during deployment. A more interesting approach would be to allow the source node to set γ depending on the type of application. For delay sensitive applications it is appropriate to use a smaller value of γ so that the delay is small, whereas, for energy constrained applications (where the network energy needs to conserved) it is better to use large γ so that the number of hops (and hence the number of transmissions) is reduced. For other applications, moderate values of γ can be used. γ can be a part of the ID signal so that it is made available to the next hop relay.

Another interesting observation from Fig. 5 is that the performance of $\hat{\pi}_{SF}$ is close to that of π_{SF}. In practice it might not be reasonable to expect a node to know the exact number of relays in the forwarding set. $\hat{\pi}_{SF}$ works with average number of nodes instead of the actual number. For small values of γ both the policies π_{SF} and $\hat{\pi}_{SF}$, most of the time, transmit to the first node to wake up. Hence the performance is similar for small γ. For larger γ, we observe that the delay incurred by $\hat{\pi}_{SF}$ is larger.

VIII. SUMMARY AND FUTURE WORK

The problem of optimal relay selection for geographical forwarding was formulated as one of minimizing the forwarding delay subject to a constraint on progress. The simple policy (SF) of transmitting to the first node that wakes up and makes a progress of more than a threshold was found to be close in performance to the optimal policy. We then compared the end-to-end performance (average delay and average hop count) of using SF at each relay node enroute to the sink with that of the policy proposed by Kim et al. [1], which is designed to achieve minimum average end-to-end delay. However, the delay obtained by the policy in [1] is only a little smaller than that obtained by the FF policy. Further, by using the SF policy with a appropriate γ, performance very close to that of the policy in [1] can be obtained without the need for an initial global configuration phase. We note that π_{SF} is self-configuring; each node takes decisions based only on local information. The end-to-end performance obtained can be tuned by the use of a single parameter γ. For a small γ we obtain low end-to-end delay but the number of hops is large and vice versa.

In this work we have assumed that each node knows the number of neighbors in its forwarding set. We had given a heuristic policy $\hat{\pi}_{SF}$ when the actual number of forwarding neighbors is not known. In future work we aim to obtain optimal forwarding policies by relaxing this assumption. Also, the use of a one-hop optimal policy for end-to-end forwarding is a heuristic. In future work we propose to directly formulate the end-to-end problem and derive optimal policies. In addition, we could also include aspects such as the relay's link quality in our formulation.

REFERENCES

[1] J. Kim, X. Lin, and N. Shroff, "Optimal Anycast Technique for Delay-Sensitive Energy-Constrained Asynchronous Sensor Networks," in *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, April 2009, pp. 612–620.

[2] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *Communications, IEEE Transactions on [legacy, pre - 1988]*, vol. 32, no. 3, pp. 246–257, 1984.

[3] T. C. Hou and V. Li, "Transmission Range Control in Multihop Packet Radio Networks," *Communications, IEEE Transactions*, vol. 34, no. 1, pp. 38–44, 1986.

[4] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 243–254.

[5] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 51–62, 2008.

[6] S. Dulman, M. Rossi, P. Havinga, and M. Zorzi, "On the Hop Count Statistics for Randomly Deployed Wireless Sensor Networks," *Int. J. Sen. Netw.*, vol. 1, no. 1/2, pp. 89–102, 2006.

[7] S. Nath and A. Kumar, "Performance Evaluation of Distance Hop Proportionality on Geometric Graph Models of Dense Sensor Networks," *Proc. 3rd International Conference on Performance Evaluation Methodologies and Tools (Valuetools '08), Athens, Greece*, October 2008.

[8] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-Based Routing in Mobile Ad-Hoc Networks," *IEEE Network*, vol. 15, pp. 30–39, 2001.

[9] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks*, vol. 3, pp. 325–349, 2005.

[10] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, and S. S. Iyengar, "Random Asynchronous Wakeup Protocol for Sensor Networks," *Broadband Networks, International Conference on*, vol. 0, pp. 710–717, 2004.

[11] S. Liu, K. W. Fan, and P. Sinha, "CMAC: An Energy Efficient MAC Layer Protocol using Convergent Packet Forwarding for Wireless Sensor Networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, June 2007, pp. 11–20.

[12] M. Zorzi, S. Member, R. R. Rao, and S. Member, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 337–348, 2003.

[13] M. Rossi, M. Zorzi, and R. R. Rao, "Statistically Assisted Routing Algorithms (SARA) for Hop Count Based Forwarding in Wireless Sensor Networks," *Wirel. Netw.*, vol. 14, no. 1, pp. 55–70, 2008.

[14] P. Chaporkar and A. Proutiere, "Optimal Joint Probing and Transmission Strategy for Maximizing Throughput in Wireless Systems," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 8, pp. 1546–1555, October 2008.

[15] H. A. David and H. N. Nagaraja, *Order Statistics (Wiley Series in Probability and Statistics)*. Wiley-Interscience, August 2003.

[16] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I.* Athena Scientific, 2005.

[17] K. P. Naveen and A. Kumar, "Tunable Locally-Optimal Geographical Forwarding in Wireless Sensor Networks with Sleep-Wake Cycling Nodes," Indian Institute of Science, Dept. of Electrical Communication Engg., IISc, Bangalore, Tech. Rep. ARXIV report number 0907.4870, 2009. [Online]. Available: http://arxiv.org/abs/0907.4870

[18] J. Kim, X. Lin, and N. B. Shroff, "Optimal Anycast Technique for Delay Sensitive Energy-Constrained Asynchronous Sensor Networks," 2008, Technical Report, Purdue University. [Online]. Available: http://web.ics.purdue.edu/~kim309/Kim08tech3.pdf

[19] D. Bertsekas and R. Gallager, *Data networks*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.