A Swarm Intelligence Based Protocol for Data Acquisition in Networks with Mobile Sinks

Hua Yang, Fengji Ye, and Biplab Sikdar, *Member, IEEE*

***Abstract*—This paper addresses the problem of data acquisition in ad hoc and sensor networks with mobile sinks and proposes a protocol based on swarm intelligence, SIMPLE, to route data in such environments. The proposed protocol is based on a swarm agent that integrates the residual energy of nodes into the route selection mechanism and maximizes the network's lifetime by evenly balancing the residual energy across nodes and minimizing the protocol overhead. The protocol is robust and scales well with both the network size and in the presence of multiple sinks. An information theoretic lower bound on the protocol overhead associated with the swarm agent advertisement is obtained. Simulation results are used to verify SIMPLE's performance.**

***Index Terms*—Swarm intelligence, data acquisition, mobile sink, energy awareness**

I. Introduction

Ad hoc and sensor networks may have a large number of nodes deployed over large areas and nodes typically have limited battery and computational capabilities. The introduction of mobility, either in the nodes or in the agents which collect data from them (i.e. sinks), makes the design of networking protocols more challenging and complicated. Examples of possible scenarios, for example, include sensor networks that are deployed to monitor areas with natural disasters, forests or civilian areas. Information is generated at the sensors and reported to the sinks, which could be first responders, forest rangers or policemen, respectively. In these scenarios, which reflect the scenarios of interest in this paper, most of the nodes stay static while sinks are mobile. The problem addressed in the paper is: *how should the static sources report their data to a mobile sink so that network lifetime is maximized?*

The constant and unpredictable changes in the sink's location form the main obstacle in the path of designing data acquisition protocols in the mobile sink scenario. Most of the existing proposals addressing this issue are based on the assumption that the mobile sink continuously updates all the nodes in the network with its current location information [1], [2]. However, such frequent updates lead to excessive consumption of the nodes' battery in addition to creating traffic congestion. Besides being energy unaware, the communication and state overheads associated with maintaining the routes in most existing protocols degrades their scalability and ability to maximize network lifetime.

H. Yang is with Intel Research Labs, Shanghai, China.

F. Ye is with Cisco Systems, San Jose, CA 95134 USA.

B. Sikdar is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Insitute, Troy, NY 12180 USA.

This work was supported in part by NSF grants 0347623 and 0313095.

Manuscript received December 31, 2006; revised July 25, 2007 and September 20, 2007.

With the specific goal of maximizing the network lifetime in the "data acquisition using mobile sink" scenario, this paper presents an on-line, energy aware protocol, SIMPLE, based on the concept of swarm intelligence [3]. Without requiring individual nodes to possess much intelligence, global information or cooperate with each other tightly, the protocol specifies a set of simple rules for each node and by their collective behavior, the globally optimum performance is achieved. In particular, SIMPLE achieves the following:

1) **Smart Data Delivery to the Mobile Sink:** SIMPLE has been designed to tolerate a degree of information inaccuracy regarding the sink's location. Thus frequent and expensive updates of all nodes with the sink's location information are avoided.
2) **Network Lifetime Maximization:** The protocol maximizes the network lifetime, defined as the time till the first node runs out of battery power.
3) **Robustness and Scalability:** Nodes can keep record of multiple path gradients to counteract node failure events. Also, when multiple sinks are present in the network, nodes can choose to report to the sink that maximizes the network lifetime and the protocol scales with the number of sources.

The main drawback of the proposed scheme is the energy required to transmit the swarm agent packets to set up the routes although most other protocol also incur similar overheads. Additionally, there is a small delay, in the order of few hundreds of milliseconds, associated with the setup of routes using the proposed scheme.

The rest of the paper is organized as follows: Section II presents the related work. Background information and the SIMPLE protocol are elaborated upon in Section III and Section IV respectively. Section V is devoted to the analysis of the proposed protocol. We present the simulation results in Section VI and conclude in Section VII.

II. Related Work

The problem of data acquisition in ad-hoc networks with static sinks has been extensively studied in recent years. Using "hop-count" as the metric, authors of [4], [5] propose shortest path routing without considering the energy constraints. Minimum energy routing protocols that fail to balance the energy consumption across nodes leading to shorter network lifetimes are proposed in [6], [7]. In [8] it is shown that energy aware metrics can significantly improve the performance of routing protocols in wireless ad hoc networks. In [9] the lifetime maximization problem is formulated as an offline linear programming problem that requires full knowledge of traffic demands. In [10], [11] "maximizing network lifetime" is taken as the objective and online algorithms are developed for routing data in static networks. The similar offline algorithm in [12] deals with static or slowly changing dynamic networks.

In mobile sink scenarios, frequently updating all nodes with a sink's current location leads to significant overheads. Recent literature suggests several alternative approaches. Directed diffusion [2] and its enhancements [13] route data based on data interests periodically broadcast by the sink but are incapable of accommodating high levels of network and sink dynamics. A scheme where *each* node builds a "grid" to route data to mobile sinks and thus incurs high overhead is proposed in [14]. Authors of [15] propose a set of algorithms that adaptively select a path that consists of a subset of nodes with high residual energy or a path with least total power consumption. The energy aware routing protocols proposed in [16], [17] require power control while this paper considers fixed transmission powers. Power aware routing schemes that require the nodes to be aware of their geographic location are proposed in [18], [19]. A probabilistic routing protocol is proposed in [20] where the probability of choosing a node as a forwarding node is inversely proportional to the aggregate load the node is carrying.

A cluster based architecture is considered in [21] and the authors propose power allocation strategies for cluster heads to offset the impact of skewed loads on the residual power distribution. This framework is not scalable and applicable in the non cluster based networks considered in this paper. The PANDA-RB routing algorithm proposed in [22] is capable of balancing the residual energy levels at all nodes but relies on flooding, making it resource expensive. The authors of [23] propose an energy aware routing protocol that accounts for residual energy levels at each node. The protocol however requires an elaborate route discovery phase where each node is required to reply to the sink, making it computation and energy intensive.

In contrast, the proposed protocol does not require any prior knowledge of the sink's mobility pattern, the distribution governing the data generation or traffic demands at the nodes. Also, no geographic information is needed in our scheme. In addition to being simple and distributed, SIMPLE is scalable, robust and handles node mobility, failures, insertions and deletions easily.

III. Background and Definitions

A. Assumptions and Terminology

This paper makes the following assumptions about the network: (1) no prior knowledge about the sink's mobility characteristics is available; (2) all sensors in the network are potential sources and no prior knowledge about data generation characteristics of a source is available; (3) all sensors transmit at the same power level. These assumptions reflect the conditions in most realistic deployment scenarios and are necessary to ensure that the developed protocol is practical. The paper also uses the following definitions:

- **Gradient** of a node indicates its next hop neighbor on the selected path leading to the sink.
- **Downstream and Upstream:** Downstream is defined as the "to-the-sink" direction, while upstream refers to the opposite.

B. Problem Definition

The objective of this paper is to develop a routing mechanism to allow static sources to report their data to mobile sinks while maximizing the network lifetime. As in [9], [24] we define network lifetime as the time until the first battery drains out, i.e., the minimum lifetime over all nodes. Then following the arguments of [9], [24], a routing mechanism that strives to balance the residual energy levels of all nodes and picks paths with nodes with higher residual energies maximizes the network lifetime. Energy aware schemes such as CMAX, zP_{min} and "max-min" that balance the energy consumption levels of nodes have been proposed in literature. In this paper we propose a variation of the "max-min" approach which is described in detail below. The reader is referred to [10], [11] for details on the CMAX and zP_{min} algorithms.

Suppose between a given source and destination there exist n paths, which we denote by $p_j, j \in 1, 2, \cdots, n$. The residual energy of the kth node v_j^k on path p_j is denoted by $e_j^k, k \in 1, 2, \cdots, h_j$, where h_j is the hop count on path p_j. Max-min routing chooses the path p_x where:

$$x = \arg \max_{j \in 1,2,\cdots n} \min_{k \in 1,2,\cdots,h_j} e_j^k \tag{1}$$

i.e. it chooses the path which contains the node with the highest minimum residual energy. Distributing the routing burden on nodes with higher residual energies serves to prolong the life of nodes with depleted energy levels thereby increasing the network lifetime. All three algorithms CMAX, zP_{min} and max-min take the energy balance issue into consideration. Although their performance is close to each other, in this paper, we choose the max-min algorithm for selecting routes because:

- Both CMAX (and its distributed version D-CMAX) and zP_{min} involve non-trivial parameter tuning based on specific source traffic patterns (which are usually not known beforehand) in order to achieve their best performance.
- zP_{min} requires multiple shortest path algorithm invocations to calculate one shortest path, which does not scale when the number of edges grows bigger.

If not indicated otherwise, throughout this paper the term "max-min path" will be used to refer to the path specified by Eqn. (1).

IV. The Data Acquisition Protocol: SIMPLE

In this section we develop a routing protocol, SIMPLE, to address the problem of data delivery from the nodes to the mobile sink. For ease of illustration, we first start with the case of a single sink in the network. We address the multiple sink scenario in Section V-E.

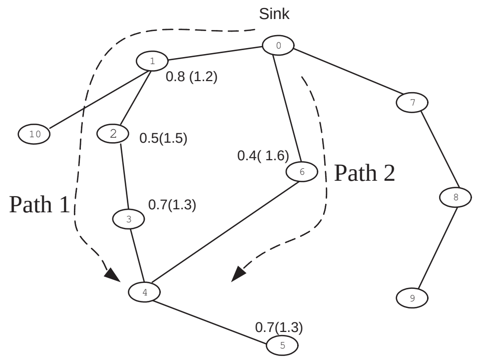

Fig. 1. Example: using a swarm agent to update the max-min path.

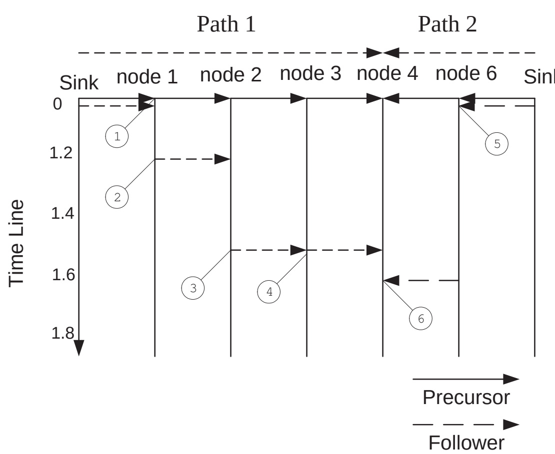

Fig. 2. Propagation of the *precursor* and *follower*.

A. Protocol Description

The proposed routing protocol is motivated by swarm intelligence and bandwidth measurement techniques in wired networks presented in [25] and references therein. Rather than storing complete path information for the max-min path, each node only maintains a "gradient" pointing to its downstream neighbor on the max-min path leading to the sink. The protocol is based on "swarm agents" which comprise of two very short packets, namely the *precursor* and *follower*. The packets comprising the swarm agent are injected into the network by the sink are sent out by the sink to mark out the max-min path gradient, as defined in Eqn. (1), at each node. Each set of swarm agent packets is stamped with an unique and increasing sequence number. These packets are then forwarded by the nodes while following a set of rules. Using the example topology shown in Figure 1, we now show how the swarm agent updates the max-min path gradients at all nodes.

The swarm agent is advertised to the network by the sink each time it wants to update the routes in the network. As shown in Figure 2, upon receiving the *precursor*, each node immediately re-advertises it to all its neighbors. It then starts a timer and waits for it to expire before forwarding the *follower*. The value of the timer, T, is a non-negative, bounded and decreasing function of the residual energy level at the node. Any function that satisfies these properties, for example

$$T = 2 - e_r \tag{2}$$

where e_r is the node's remaining energy (normalized between $[0, 1]$), may be chosen and our scheme does not depend on the specific function. Thus, nodes with higher residual energy will time out faster and this is the key to differentiating paths with different minimum residual energy levels. In the rest of the discussion, we show that the *follower* from the max-min path arrives at each node first and the node can thus simply mark the sender of the *follower* as the gradient. In Figure 1 the numbers alongside the nodes that are outside the parentheses are the nodes' normalized residual energy and numbers inside the parentheses are the nodes' corresponding timer value. Figure 2 also shows how the *follower* is propagated at each intermediate sensor. For the sake of an easily understandable protocol description, we omit the delays induced by queueing and MAC layer, which will be addressed in next section.

The sink sends out the swarm agent at time 0. Since the *precursor* simply cuts through and we are omitting the queueing and MAC layer delays, all nodes receive the *precursor* at time 0. Based on the max-min path definition, $sink \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ (path 1 in the figure) is the max-min path between the sink and node 4. Even though the other path $sink \rightarrow 6 \rightarrow 4$ (path 2 in the figure) has fewer hops than the max-min path, it is still "longer" since node 6 has only 0.4 energy left, which is the least among all nodes on paths between the sink and node 4. We use node 4 as an example to explain how the gradient is marked out at each node. Following are events occurring at the times indicated by the circled numbers in Figure 2:

Time ①: Node 1 receives the *follower*;
Time ②: Node 1's timer times out at time 1.2, and the *follower* is sent out;
Time ③: Node 2 receives the *follower* at time 1.2 and sends it out at time 1.5 when its timer times out;
Time ④: Node 3's timer has already timed out when it receives the *follower* at time 1.5 and thus forwards it immediately. The *follower* reaches node 4 at time 1.5;
Time ⑤: Node 6 receives the *follower* at time 0;
Time ⑥: Node 6 sends out the *follower* at time 1.6 when it times out. Node 4 gets a second copy of the *follower* from node 6 at time 1.6 and it is simply dropped. Node 6 can be recorded as the backup path gradient.

On path 1, the longest timer times out ahead of that from path 2 even though the latter has fewer hops. Thus in our scheme, the first copy of the *follower* will always arrive along the max-min path. In Section V we formally prove that our technique indeed selects the max-min path. Since each node can safely refrain from relaying all *followers* except the first copy, this greatly suppresses the amount of swarm agent copies that are circulating in the network. In our example, since node 4 receives the *follower* from node 3 first, it makes node 3 its "gradient" on the max-min path leading to the sink. To counteract node failures or sleeping, node 6 can be marked as the backup path gradient to the sink.

It should be pointed out that for gradient initialization immediately after the deployment of the ad hoc or sensor

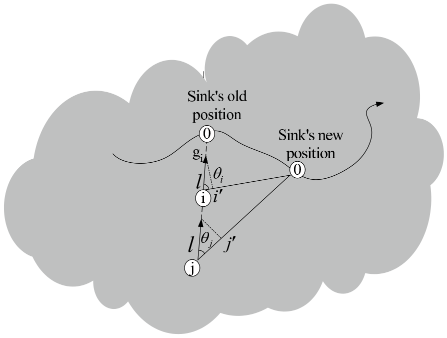

Fig. 3. Scenario 1: Sink's movement has lower effect on nodes further away.

network, since every node has same residual energy, E_{init}, our scheme actually initializes the max-min path as the path with minimum hops, which is consistent with energy efficiency rules. From an individual node's point of view, all its operations are straightforward and the globally optimal gradient is set up without involving any global information collection or complicated computation algorithms, enabling the scheme to scale. The scalability issue will be further elaborated upon in Section V-C.

A strong point of SIMPLE is that it is naturally loop free since node i will always discard any swarm agent from node j if it has already sent one to j. Also, the scheme does not rely on any assumptions regarding messages sent from the sink to the nodes, such as queries. If the nodes' report is triggered by queries from the sink to all nodes, the swarm agent can actually be integrated into the queries with little extra cost.

B. Constrained Advertisement Model

The scheme designed above updates the max-min path from each sensor to the sink using the swarm agent. However, some of these advertisements may be redundant and could be suppressed without sacrificing too much of the performance. We first identify the scenarios with redundant swarm agents. Then, a constrained advertisement model is presented to enhance the protocol efficiency.

1) Advertisement Suppression Scenarios: We first define the "utility" of a node. Node i's utility increases if any node j picks i as its next hop on the max-min path based on node i's advertisement. Otherwise the utility of node i will decrease, for example, exponentially as time lapses. Since all advertisements cost energy, a higher utility/energy consumption ratio is desired for each node. Before introducing how unnecessary advertisements are suppressed, we first identify the scenarios where they occur.

Scenario 1: In Figure 3 suppose node i is not updated with the sink's movement. Messages originating at or relayed by i will be sent along gradient g_i, set up based on the sink's old position. Denoting the progress along gradient g_i by l, we define the "effective progress" $\overline{ii'}$ as:

$$\|\overline{ii'}\| = l \times cos(\theta_i) \tag{3}$$

As we can see, when a node is further away from the sink, such as node j, θ becomes smaller and the effective progress is closer to l. This suggests that for a given sink displacement, nodes further away are less affected. Thus, gradients of nodes further away from the sink can be updated less frequently as compared to nodes nearer to the sink. This scenario is similar to the scenario considered in Section 3.3.2 of [26].

Scenario 2: Consider three nodes i, j and k on paths p_i, p_j and p_k between source s and destination d. If node i has the maximum residual energy, path p_i will be chosen as the max-min path and node j and k's advertisements are futile and only serve to lower their utility/energy ratio. Ideally, we would like to have only the nodes on the max-min path to advertise the swarm agent, while all other nodes suppress their advertisements to save energy.

Scenario 3: Consider a scenario where node i frequently relays data to the sink, while node j seldom does so. In this case, node i should advertise more frequently than node j since i's upstream neighbors are expecting i's advertisement, and the advertisement will increase node i's utility. In the extreme case, if a node is never chosen to relay data for others, its agent advertisements will not increase its utility but decrease its utility/energy consumption ratio. Thus, nodes that seldom relay data should advertise less frequently than nodes that often relay data.

2) A Probabilistic Advertising Model: A deterministic solution to suppress advertisements in the scenarios described above requires global information at each node, which makes it impractical. We thus introduce a probabilistic model for swarm agent re-advertisement. Based on the discussion above, we let a node's re-advertising probability ρ:

1) increase if it relays data for its neighbors;
2) decrease if the node does not relay any data as time elapses;
3) have a higher lower-bound when the node has more residual energy.

An algorithm based on these rules to calculate the re-advertisement probability is shown in Algorithm 1. Note that we set a lower bound so that a sensor's ρ will never reach 0 except when its energy is fully depleted. Thus, even less active nodes will advertise occasionally so that they may be selected when the energy of other nodes depletes. By applying this probabilistic model, nodes further away from the sink will have a smaller advertisement intensity as compared to nodes closer to the sink. In addition, nodes with more residual energy will have a higher probability to join the advertisement.

The probabilistic re-advertisement is essentially a trade-off between network's energy balance and overhead energy consumption. Non suppression of advertisements makes the network most balanced but incurs the highest overhead, while suppressing all advertisements causes the opposite. Suppression of the agents however does not mean that packets are not delivered to the sink as shown in Section V-E as well as through simulation results. The scheme adapts to node insertions and deaths fairly easily. When a new sensor joins the network, it forwards any received swarm agent to make its neighbors aware of its existence. When a sensor leaves or dies, its upstream neighbors will not receive any further swarm agents, which naturally removes the node from their next hop candidates list.

Algorithm 1 Algorithm for calculating re-advertisement probability

t: time at which last swarm agent was received
$\rho(t)$: re-advertisement probability at time t
$relay$: binary variable to indicate if node is a relay node
l_{bound}: lower bound on re-advertisement probability
α, β, δ, γ: positive constants, $0 < \delta, \gamma < 1$
while (1) **do**
  if `new swarm agent received` **then**
    $t + \tau$: current time
    $l_{bound} = \delta + \gamma \frac{\text{residual energy}}{\text{battery capacity}}$
    $tmp_1 = \rho(t) - \alpha e^{\beta\tau - 1}$
    $tmp_2 = \max\{l_{bound}, tmp_1\}$
    if $relay == 1$ **then**
      $\rho(t+\tau) = \frac{1+tmp_2}{2}$
    else
      $\rho(t+\tau) = tmp_2$
    end if
    $relay = 0$
  end if
  if `data packet forwarded` **then**
    $relay = 1$
  end if
end while

V. Protocol Analysis

In this section, we analytically address various aspects of the proposed protocol.

A. Validity of the Max-min Path

Claim 1: When MAC and other delays are negligible compared to the swarm agent's timer, at any arbitrary node v with n paths to the sink p_i, $i = 1, \cdots, n$, the gradient set up by the unconstrained swarm agent is given by

$$\arg\max_{1\le j\le n} \min_{1\le k\le h_j} e_j^k \tag{4}$$

conforming to the max-min path specified in Eqn. (1).

Proof: Let the sink send out a swarm agent at time t. Define a mapping function:

$$t_v = M(e_v) \tag{5}$$

where e_v is the residual energy of node v and $M(e_v)$ could be any bounded and monotonous decreasing function. t_v gives the initial value of node v's timer for the swarm agent. The swarm agent traveling along path p_j is also attached with an "agent timer", T_j, with initial value 0 when advertised from the sink.

Let v_j^k denote the k-th hop on path p_j with initial energy e_j^k and timer t_j^k. As the swarm agent passes through this node, the agent timer, denoted as T_j^k, will be updated as:

$$\begin{aligned} T_j^k &= \max\{T_j^{k-1}, t_j^k\} \\ &= \max\{T_j^{k-1}, M(e_j^k)\} \end{aligned} \tag{6}$$

The swarm agent will be re-advertised by node v_j^k when T_j^k times out. Thus, finally node v will receive a swarm agent from path p_j with attached agent timer of value:

$$T_j = \max_{1\le k\le h_j} M(e_j^k)$$

where h_j is the total hop count along path p_j. Now assume that node v receives swarm agents from n different paths. It is easy to see that an agent with a smaller "agent timer" always arrives earlier. From the monotonous decreasing nature of the mapping function (5), agent timer of the first arriving swarm agent is T_x where:

$$\begin{aligned} x &= \arg\min_{1\le j\le n} T_j \\ &= \arg\min_{1\le j\le n} \max_{1\le k\le h_j} M(e_j^k) \\ &= \arg\max_{1\le j\le n} \min_{1\le k\le h_j} e_j^k \end{aligned} \tag{7}$$

The equation above is exactly the same as Eqn. (1) which defines the max-min path thereby proving the claim. ■

Note that all the n paths leading to node v can share many intermediate nodes, but this does not affect the validity of our conclusion. Duplicates of the swarm agent simply get dropped at each node, which saves energy.

B. Errors in the Presence of MAC Layer Delays

The protocol correctness proof in the preceding section assumed that there are no MAC layer delays. In the presence of MAC layer delays, the *follower* from the max-min path may be delayed long enough so that the *follower* from a path with smaller minimum residual energy may reach a tagged node first. However, if the lower bound on the timers associated with the *followers* is is made large compared to the expected MAC layer delay at a node, the probability of such events becomes very small. In this section we evaluate the likelihood of such errors in selecting the max-min path.

Consider two paths from the sink to a tagged node, A and B, with M and N hops, respectively. Let Δ_{A-B} denote the difference in the timer values of the nodes with the minimum residual energies in paths A and B and let δ_{B-A} denote the difference in the cumulative MAC layer delays of path B and path A. An error in marking the correct gradient occurs if $\Delta_{A-B} < \delta_{B-A}$ when $\Delta_{A-B} > 0$ or if $\Delta_{A-B} > \delta_{B-A}$ when $\Delta_{A-B} < 0$. We now evaluate the likelihood of these events.

The timer at each node for delaying the *follower* is dependent on the node's residual energy. We assume that the energy levels at each node is independent and identically distributed (iid). Consequently, the timer value at each node is also iid with probability density function (pdf) $f_d(x)$ with $a \le x \le b$ where a and b are the lower and upper limits on the timer. Using standard results from order statistics, the pdf of the minimum timer value along path A is

$$f_{M.A}(x) = M(1 - F_d(x))^{M-1} f_d(x) \tag{8}$$

where $F_d(x)$ is the cumulative distribution function (CDF) of $f_d(x)$. The corresponding expression for path B is obtained by substituting N for M in Eqn. (8). The pdf of the difference in the minimum timer value of path A from path B is given by

$$f_{\Delta_{A-B}}(x) = \begin{cases} \int_{a-t}^{b} f_{M,B}(t) f_{M,A}(x+t)dt & a-b \leq t < 0 \\ \int_{a}^{b-t} f_{M,B}(t) f_{M,A}(x+t)dt & 0 \leq t \leq b-a \end{cases} \tag{9}$$

We assume the MAC layer channel access delays at each node in the network is iid with pdf $f_D(x)$, $0 \leq x < \infty$ with mean μ and variance σ^2. Also, we assume that $a \gg E[D]$, i.e. the lower bound on the timer is much greater than the expected MAC layer delay at a node. We do not account for any queueing delays since the swarm agent packets are expected to receive prioritized treatment and moved to the head of the MAC layer queue. The total time that elapses from the instant the swarm agent is sent by the sink till the follower reaches the tagged node can be broken into three parts: (i) the MAC layer delays experienced by the *precursor* while reaching the node with the minimum residual energy, (ii) the *follower* timer delay at this node and (iii) the MAC layer delays experienced by the *follower* after it leaves this node. Since the statistical distribution characterizing the MAC layer delays of *precursors* and *followers* is identical, the total MAC layer delay of path A (B) is the sum of the M (N) random variables with pdf $f_D(x)$. The cumulative MAC layer delays along a path and its variance increases with the hop count. Thus the likelihood that the difference in the MAC layer delay on the two paths exceeds the difference of the timer values of the nodes with the minimum residual energies in the two paths increases with the path length. As the path length increases, from the central limit theorem, the cumulative delay on each path follows a Gaussian distribution and for path A is given by

$$f_{D,A}(x) = \frac{1}{\sqrt{2\pi}\sigma_A} e^{-\frac{1}{2}\left(\frac{x-\mu_A}{\sigma_A}\right)^2} \tag{10}$$

with $\mu_A = M\mu$ and $\sigma_A^2 = M\sigma^2$. For path B, we have $\mu_B = N\mu$ and $\sigma_B^2 = N\sigma^2$. The difference in the delays of path B and A characterizes δ_{B-A} and its pdf is given by

$$f_{\delta_{B-A}}(x) = \frac{1}{\sqrt{2\pi(\sigma_A^2+\sigma_B^2)}} e^{-\frac{1}{2}\left(\frac{(x+\mu_A-\mu_B)^2}{\sigma_A^2+\sigma_B^2}\right)} \tag{11}$$

The probability that the follower from the non max-min path reaches the tagged node first is then

$$\begin{aligned} P[\text{error}] &= P[\Delta_{A-B} < \delta_{B-A}, \Delta_{A-B} > 0] \\ &\quad + P[\Delta_{A-B} > \delta_{B-A}, \Delta_{A-B} < 0] \end{aligned} \tag{12}$$

From [27] the average 802.11 MAC layer access delays and its variance is about a millisecond when a node competes with 10-15 nodes for the channel. Thus we may use $\mu = 1$ms and $\sigma^2 = 1\text{ms}^2$ for our example numerical evaluation. The evaluation of Eqn. (12) requires knowledge of the distribution of the residual power levels at each node and to the best of our knowledge, such distributions are not known. However, an indication of the low probability of error due to MAC layer delays can be seen from the fact that even if the difference in the timer values of the two paths is 20ms, with $M = 20$ and $N = 25$, the probability of error is less than 0.0001.

The requirement of large timer values as compared to the expected MAC layer delays at a node does not adversely affect the time required to setup the routes. This is because for the most part, the timers of different nodes overlap. As an extreme example, consider the setup of routes when the battery levels of all nodes is almost 0. If we consider that timer values are bounded in the range $[0.1, 0.2]$ seconds, then the timer value of all nodes will be close to 0.2 seconds. If the expected value of the sum of the MAC layer access delays and the swarm agent transmission time at each node is realistically assumed to be 2ms, a *precursor* from the sink reaches a node 100 hops away after 200ms. This node waits for at most 0.2 seconds before sending out the *follower*. Thus routes are setup at a node 101 hops away after approximately 0.4 seconds. The route setup delay is smaller when the network is initially set up and also if smaller values are used for the timer range.

C. Scalability of SIMPLE with Constrained Flooding

The potential scalability issue of SIMPLE concerns the advertisement scope of the swarm agent as the network size grows. A swarm agent's "advertisement scope" is defined by its radius L, indicating that at least one node advertising the swarm agent is L hops away from the sink. In order to prove the scalability of SIMPLE with probabilistic forwarding, we now show that L is bounded even when the network size is not. If we define $C(G)$ as coverage of the sensor network G, then we have:

Claim 2:

$$\lim_{C(G)\to\infty} \text{Prob}(L \to \infty) = 0 \tag{13}$$

Proof: Assume that the mobile sink is advertising the swarm agent to the network at a rate of R_s agents per second. Define R_{v_i} as the advertisement intensity of an arbitrary node v_i in the sensor network. Denote node v_i's re-advertisement probability as ρ_{v_i}, and SP_{v_i} as the set of all nodes on the max-min path from node v_i to the sink. Then at node v_i:

$$R_{v_i} = \rho_{v_i} R_s \prod_{v_j \in SP_{v_i}} \rho_{v_j}$$

where v_j are all nodes on node v_i's max-min path to the sink, and ρ_{v_j} is the re-advertisement probability of node v_j. Since

$$\rho_{v_i} \leq 1, \text{for any } v_i$$

we have

$$\lim_{h(v_i)\to\infty} \text{Prob}(R_{v_i} = 0) = 1$$

where $h(v_i)$ is node v_i's hop count to the sink and $h(v_i) \to \infty$ when $C(G) \to \infty$. The equation above indicates that a node's advertisement intensity goes to 0 as its distance from the sink increases. Thus, following equation is proved:

$$\lim_{C(G)\to\infty} \text{Prob}(L \to \infty) = 0 \tag{14}$$

■

This scalability claim does not apply to SIMPLE without constrained flooding. Eqn. (14) indicates that the "advertisement scope" is bounded even when the sensor network's size grows larger. Thus as nodes get further away from the sink, their gradient might not be updated often. In Section V-E we address the problem of how nodes far away from the sink and thus with possibly outdated location information correctly deliver their data to the sink.

D. Protocol Overhead

Swarm agents are sent out by the sink occasionally to keep the other nodes abreast of its location. In this section, we obtain an information theoretic bound on the overhead incurred due to the transmission of the swarm agents. We obtain the minimum rate at which swarm agents must be sent, so that the probability that each node has the correct gradient to the sink when it has data to send, is greater than an arbitrary value $1-\epsilon$. Here we consider SIMPLE without constrained flooding in order to estimate the worst case overhead.

We formulate the problem of obtaining the protocol overhead as a rate distortion problem. For each node, the probability that it perceives the sink as its neighbor when it is not, is used as the distortion measure. We assume that an arbitrary number, n, of nodes are randomly and uniformly distributed on a two dimensional plane. The set of all nodes is denoted by $\mathcal{N}$. In addition to these nodes which are assumed to be static, a mobile sink continuously moves around the network. The movement of the sink is governed by a two dimensional random walk in continuous time. The position of node j at time t is denoted by $x_j(t), y_j(t)$ and the distance between two nodes i and j at time t is given by $\Delta_{ij}(t) = \sqrt{(x_i(t)-x_j(t))^2+(y_i(t)-y_j(t))^2}$. All nodes and the sink are assumed to have a transmission range of r.

Each node determines whether it is a neighbor of the sink or not, and also the route to the sink, based on the swarm agents it receives. We denote by $N_s(t) = \{j : \Delta_{sj}(t) \le r, j \in \mathcal{N}\}$ the set of nodes that are actually neighbors of the sink at time t and by $\hat{N}(t)$ the set of nodes that perceive themselves to be the sink's neighbor. A node may perceive itself to be the sink's neighbor when it is not, or vice versa, due to use of outdated swarm agent information. We also define

$$Z_j(t) = \begin{cases} 1 & \text{if } j \in N_s(t) \\ 0 & \text{otherwise} \end{cases} \text{ and } \hat{Z}_j(t) = \begin{cases} 1 & \text{if } j \in \hat{N}_s(t) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

as variables to indicate whether node j actually is or perceives to be the sink's neighbor or not. The difference

$$E_j(t) = Z_j(t) - \hat{Z}_j(t) \tag{16}$$

denotes the accuracy of the information about the sink's position available at node j. It is desired that $E_j(t) = 0$ at all times for all j so that accurate forwarding information may be used by the nodes while transferring packets to the sink. We now state the minimum swarm agent rate problem in terms of $E_j(t)$.

Minimum swarm agent frequency problem: What is the minimum rate at which the sink must transmit swarm agents such that

$$P[E_j(T_j^k) = 0] \ge 1-\epsilon, \quad \forall j \in \mathcal{N}, \quad 1 \le k < \infty \tag{17}$$

where T_j^k is the instance when node j generates or receives its k-th packet to be sent to the sink.

We now formulate the minimum swarm agent frequency problem as a rate distortion problem. Our analysis is based on extending the results of [28] which considers the problem of protocol overhead in location based routing. We denote by Z_j^N and $\hat{Z}_j^N$ the vectors

$$\begin{aligned} Z_j^N &= \{Z_j(T_j^1), Z_j(T_j^2), \cdots, Z_j(T_j^N)\} \\ \hat{Z}_j^N &= \{\hat{Z}_j(T_j^1), \hat{Z}_j(T_j^2), \cdots, \hat{Z}_j(T_j^N)\} \end{aligned}$$

and denote by $\mathcal{P}_N(\epsilon)$ the family of joint probability distribution functions of Z_j^N and $\hat{Z}_j^N$ such that $P[E_j(T_j^k) = 0] \ge 1-\epsilon$, $\forall j \in \mathcal{N}$ and $1 \le k < \infty$. We also denote by $R_N(\epsilon)$ the minimum swarm agent rate such that $P[E_j(T_j^k) = 0]$ and as per the discussion on rate distortion in [29], it is given by

$$R_N(\epsilon) = \min_{P_N \in \mathcal{P}_N(\epsilon)} \frac{1}{N} I_{P_N}(Z_j^N; \hat{Z}_j^N) \tag{18}$$

where $I_{P_N}(Z_j^N; \hat{Z}_j^N)$ is the mutual information between Z_j^N and $\hat{Z}_j^N$. The minimum swarm agent rate, $R(\epsilon)$, is then given by

$$R(\epsilon) = \lim_{N\to\infty} \min R_N(\epsilon) \tag{19}$$

We now obtain a bound for $R_N(\epsilon)$ and consequently $R(\epsilon)$ by evaluating a bound for $I_{P_N}(Z_j^N; \hat{Z}_j^N)$.

Claim 3: The minimum swarm agent rate $R(\epsilon)$ is greater than or equal to $R_1(\epsilon)$.

$$R(\epsilon) \ge R_1(\epsilon) \tag{20}$$

Proof: To prove the equation above, we first show that the mutual information between Z_j^N and $\hat{Z}_j^N$ satisfies the relationship

$$\inf_{P_N \in \mathcal{P}_N(\epsilon)} I_{P_N}(Z_j^N; \hat{Z}_j^N) \ge N R_1(\epsilon) \tag{21}$$

To show Eqn. (21) holds, we note that the standard definition of mutual information gives us

$$I_{P_N}(Z_j^N; \hat{Z}_j^N) = H(Z_j^N) - H(Z_j^N \mid \hat{Z}_j^N) \tag{22}$$

From Eqn. (B3) of [30]

$$\begin{aligned} H(Z_j^N \mid \hat{Z}_j^N) \quad &\le \quad H(Z_j(T_j^1) \mid \hat{Z}_j(T_j^1)) + \\ \sum_{k=2}^{N} H(Z_j(T_j^k) &- Z_j(T_j^{k-1}) \mid Z_j(T_j^{k-1}), \hat{Z}_j(T_j^k)) \end{aligned} \tag{23}$$

We define the random variable

$$\chi^k = \hat{Z}_j(T_j^k) - Z_j(T_j^{k-1}). \tag{24}$$

Then from Eqns. (B5) and (B6) of [30] we have

$$\begin{aligned} H(Z_j(T_j^k) - Z_j(T_j^{k-1}) \mid Z_j(T_j^{k-1}), \hat{Z}_j(T_j^k)) \quad &\le \quad H(Z_j(T_j^k) \\ &-Z_j(T_j^{k-1}) \mid \chi^k) \end{aligned} \tag{25}$$

$$H(Z_j^N \mid \hat{Z}_j^N) \le H(Z_j(T_j^1) \mid \hat{Z}_j(T_j^1)) + \sum_{k=2}^{N} H(Z_j(T_j^k) - Z_j(T_j^{k-1}) \mid \chi^k) \tag{26}$$

Now, $Z_j(T_j^k)$ and $Z_j(T_j^{k-1})$ are independent of each other. Thus we also have

$$H(Z_j^N) = H(Z_j(T_j^1)) + \sum_{k=2}^{N} H(Z_j(T_j^k) - Z_j(T_j^{k-1})) \tag{27}$$

Substituting Eqns. (26) and (27) in Eqn. (22) we have

$$I_{P_N}(Z_j^N;\hat{Z}_j^N) \geq I(Z_j(T_j^1);\hat{Z}_j(T_j^1)) + \sum_{k=2}^{N} I(Z_j(T_j^k)-Z_j(T_j^{k-1});\chi^k) \quad (28)$$

Now, the difference in the actual and perceived neighborhood information at time T_j^k is

$$D^k = \hat{Z}_j(T_j^k)-Z_j(T_j^k) = \chi^k-(Z_j(T_j^k)-Z_j(T_j^{k-1})) \quad (29)$$

We denote the expected value of D^k by d^k, i.e., $d^k = E[D^k]$. Consider $Z_j(T_j^1)$: $Z_j(T_j^1)$ has the same distribution as $Z_j(T_j^k)-Z_j(T_j^{k-1})$ and also satisfies Eqn. (29). Then from the definition of the rate distortion function

$$\begin{aligned} R_1(d^k) &= \min_{P_1\in\mathcal{P}_1(d^k)} \frac{1}{1} I(Z_j(T_j^k)-Z_j(T_j^{k-1});\chi^k) \\ &\leq I(Z_j(T_j^k)-Z_j(T_j^{k-1});\chi^k), \quad k\geq 2 \end{aligned} \quad (30)$$

Define $d^1 = E[\hat{Z}_j(T_j^1)-Z_j(T_j^1)]$. Then substituting Eqn. (30) in Eqn. (28) and using the convexity of the rate distortion function we have

$$I_{P_N}(Z_j^N;\hat{Z}_j^N) \geq R_1(d^1)+\sum_{k=2}^{N} R_1(d^k) \geq NR_1\left(\frac{1}{N}\sum_{k=1}^{N} d^k\right) \quad (31)$$

Now, if $P_N \in \mathcal{P}_N(\epsilon)$, we have

$$\frac{1}{N}\sum_{k=1}^{N} d^k = \frac{1}{N}\sum_{k=1}^{N} E[\hat{Z}_j(T_j^1)-Z_j(T_j^1)] \leq \frac{1}{N}\sum_{k=1}^{N}\epsilon = \epsilon \quad (32)$$

Since R_1 is a non-increasing function, combining Eqns. (31) and (32) gives us

$$I_{P_N}(Z_j^N;\hat{Z}_j^N) \geq NR_1(\epsilon) \quad (33)$$

which proves that Eqn. (21) holds. To prove the claim we note that the definition of the rate distortion function gives us

$$R_N(\epsilon) = \min_{P_N\in\mathcal{P}_N(\epsilon)} \frac{1}{N} I_{P_N}(Z_j^N;\hat{Z}_j^N) \geq R_1(\epsilon) \quad (34)$$

and thus

$$R(\epsilon) = \lim_{N\to\infty} \min R_N(\epsilon) \geq R_1(\epsilon) \quad (35)$$

which proves that Eqn. (20) holds and thus proves the claim. ■

As the next step, we find a bound on $R_1(\epsilon)$ in order to bound $R(\epsilon)$. Here we consider two cases: **(1)** $Z_j(0)=1$: node j is initially in the neighborhood of the sink and **(2)** $Z_j(0)=0$: node j is not in the sink's neighborhood initially. $R_1(\epsilon)$ is then bounded by the maximum of the rate distortion functions for these two cases.

Case 1: Denote by L_j the region in space of possible positions for node j at time $t=0$ such that $Z_j(0)=1$, i.e. $L_j = \{x_j, y_j : \sqrt{(x_s(0)-x_j)^2+(y_s(0)-y_j)^2} \leq r\}$.

Claim 4: The rate distortion function in this case, $R_{1,C1}(\epsilon)$ is bounded by

$$R_{1,C1}(\epsilon) \geq \max_{x_j(0),y_j(0)\in L_j} H(Z_j(T_j^1))+\epsilon\log\left(\frac{\epsilon}{2}\right)+(1-\epsilon)\log(1-\epsilon) \quad (36)$$

Proof: From the definition of mutual information

$$I_{P_1}(Z_j(T_j^1);\hat{Z}_j(T_j^1)) \geq = H(Z_j(T_j^1))-H(E_j(T_j^1)) \quad (37)$$

Since $Z_j(T_j^1)$ and $\hat{Z}_j(T_j^1)$ take on a value of either 0 or 1, its probability mass function can be written in terms of some $p1$, $p2$ and $p3$ as

$$E_j(T_j^1) = \begin{cases} -1 & \text{w.p. } p1 \\ 0 & \text{w.p. } p2 \\ 1 & \text{w.p. } p3 \end{cases} \quad (38)$$

where $P[E_j(T_j^1)=0]=p2\geq 1-\epsilon$ and $p1+p2+p3=1$. Thus we have $p1+p3\leq\epsilon$. The entropy of $E_j(T_j^1)$ is then given by $H(E_j(T_j^1)) = -p1\log p1-p2\log p2-p3\log p3$ which is maximized when $p2=1-\epsilon$ and $p1=p3=\epsilon/2$. This maximum entropy is given by

$$H(E_j(T_j^1)) = -\epsilon\log\left(\frac{\epsilon}{2}\right)-(1-\epsilon)\log(1-\epsilon) \quad (39)$$

Now $H(Z_j(T_j^1))$ depends on the position of node j at $t=0$ and the swarm agent rate should account for the initial location that results in the maximum entropy. Substituting Eqn. (39) in Eqn. (37) we then have

$$\begin{aligned} I_{P_1}(Z_j(T_j^1);\hat{Z}_j(T_j^1)) &\geq \max_{x_j(0),y_j(0)\in L_j} H(Z_j(T_j^1)) \\ &+\epsilon\log\left(\frac{\epsilon}{2}\right)+(1-\epsilon)\log(1-\epsilon) \end{aligned}$$

■

To obtain $H(Z_j(T_j^1))$ we note that if $P[Z_j(T_j^1)=1]=\delta$ then $H(Z_j(T_j^1)) = -\delta\log\delta-(1-\delta)\log(1-\delta)$. We now obtain the probability $P[Z_j(T_j^1)=1]$ by obtaining $P[Z_j(T_j^1)=1 \mid \Delta_{sj}(0)=l, T_j^1=\tau]$ with $l\leq r$ and then unconditioning on τ. Since the sink follows a two dimensional random walk with variance α, unconditioning the result of Appendix case 1, Eqn. (48), on packet interarrival times (which have a pdf $f_T(\tau)$), we have

$$\begin{aligned} P[Z_j(T_j^1)=1 \mid \Delta_{sj}(0)=l] &= \int_0^\infty\int_0^{r-l} \frac{2x}{\alpha\tau} e^{-\frac{x^2}{\alpha\tau}} f_T(\tau)\,dx\,d\tau \\ &+ \int_0^\infty\int_{r-l}^{r+l} \frac{2\cos^{-1}\left(\frac{-r^2+l^2+x^2}{2lx}\right)}{\pi\alpha\tau} x e^{-\frac{x^2}{\alpha\tau}} f_T(\tau)\,dx\,d\tau \end{aligned}$$

Note that $H((Z_j(T_j^1)) = -\delta\log\delta-(1-\delta)\log(1-\delta)$ is maximized at $\delta=0.5$ and we denote the maximum value of $H((Z_j(T_j^1))$ for this case (i.e. where $Z_j(0)=1$), achieved at $l=l^*$ (say), by $H^*_{C1}(Z_j(T_j^1))$. We then have

$$R_{1,C1}(\epsilon) \geq H^*_{C1}(Z_j(T_j^1))+\epsilon\log\left(\frac{\epsilon}{2}\right)+(1-\epsilon)\log(1-\epsilon) \quad (40)$$

Case 2: Denote by L'_j the region in space of possible positions for node j at time $t=0$ such that $Z_j(0)=0$, i.e. $L'_j = \{x_j, y_j : \sqrt{(x_s(0)-x_j)^2+(y_s(0)-y_j)^2} > r\}$.

Claim 5: The rate distortion function in this case, $R_{1,C2}(\epsilon)$ is bounded by

$$R_{1,C2}(\epsilon) \geq \max_{x_j(0),y_j(0)\in L'_j} H(Z_j(T_j^1))+\epsilon\log\left(\frac{\epsilon}{2}\right)+(1-\epsilon)\log(1-\epsilon) \quad (41)$$

Proof: The proof is identical to that for Case 1. ■

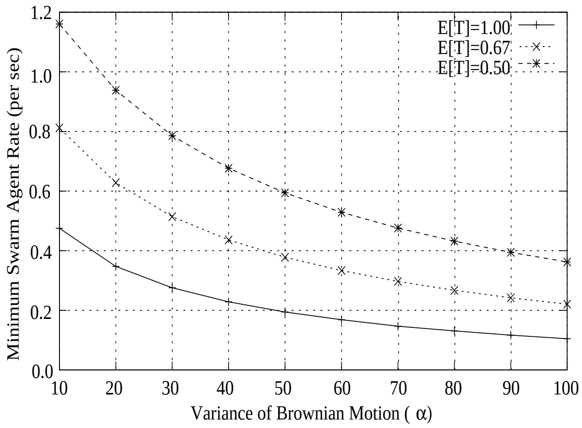

Fig. 4. Bound on swarm agent rate, $\epsilon = 0.01$.

Using the results of the Appendix Case 2, Eqn. (50), $H(Z_j(T_j^1) \mid \Delta_{sj}(0) = l)$ with $l > r$ in this case is given by

$$P[Z_j(T_j^1) = 1 \mid \Delta_{sj}(0) = l] = \int_0^{\infty} \int_{l-r}^{r+l} \frac{2\cos^{-1}\left(\frac{-r^2+l^2+x^2}{2lx}\right)}{\pi\alpha\tau} xe^{-\frac{x^2}{\alpha\tau}} f_T(\tau)\, dx\, d\tau \tag{42}$$

For this case, the maximum $H(Z_j(T_j^1))$ is achieved when $l = r$ and we denote this entropy by $H^*_{C2}(Z_j(T_j^1))$. We then have

$$R_{1,C2}(\epsilon) \geq H^*_{C2}(Z_j(T_j^1)) + \epsilon \log\left(\frac{\epsilon}{2}\right) + (1-\epsilon)\log(1-\epsilon) \tag{43}$$

The lower bound on the swarm agent rate is then

$$R(\epsilon) \geq R_1(\epsilon) \geq \max\{R_{1,C1}(\epsilon),\, R_{1,C2}(\epsilon))\} \tag{44}$$

Figure 4 compares the bound on the swarm agent rate as obtained from Eqn. (44) for different variances of the Brownian motion and packet arrival rates. The packet interarrival time is assumed to follow an exponential distribution.

E. Robustness of SIMPLE

We now consider the forwarding of data in scenarios where the gradient at a node is outdated due to the movement of the sink. For this discussion, we assume that the sink moves at speed v and sends out swarm agents at rate R. Consider a node i whose max-min path gradient is out of date and still leads to the sink's old position, A. At time t node i sends a packet to the sink, which has now moved to position B. If the gradients of the nodes on the path to A have not been updated yet, the packet moves towards A and in the worst case, reaches A where it is unable to reach the sink. The gradient at A is updated after at most a period equal to the interval between swarm agent advertisements $1/R$ and the MAC layer delays and maximum timer value associated with the *follower* on the max-min path to the sink. Once the gradient is updated at A, the packet is now forwarded towards the current location. This precess may be repeated a number of times at different locations before the packet finally reaches the destination. However, as long as the distance traversed by the sink between two updates is smaller than the distance that the packet traverses in this interval, the packet is guaranteed to catchup with the sink. As an example, for a 512B data packet sent at 1Mbps, the transmission and channel access time would typically be less than 10ms. If the average distance traveled towards the sink between two successive hops is considered to be a very small 10m, the packet progresses towards the sink at a speed of 2000m/s which is much higher than typical maximum sink speeds of 35-45m/s (130-160km/hr or 80-100mph). Thus, the information delivery path is rectified on the fly and the protocol is robust.

The above discussion also suggests that even if swarm agents are not advertised by the sink at the rate determined in Section V-D, packets are still delivered correctly. This observation may be exploited to reduce the frequency of swarm agent advertisements and the associated overhead.

F. Operation under Multiple Sinks

For operation in the presence of multiple sinks, the *follower* packet contains a field, T_{max}, that contains the largest timer value that the *follower* has experienced in the path. The sink initializes its value to zero and at each node, the value in the received *follower* is compared with the node's own timer. The larger of the two values is stamped in the *follower* sent out by the node. With multiple sinks, each sink asynchronously sends out its swarm agents. Each node compares the T_{max} value for the swarm agents of each sink that it receives and selects the sink with the smallest T_{max} so as to pick the path with the highest residual energy. In addition, gradients for other sinks may be stored as backup paths in case the neighbor with the max-min path becomes unavailable. Finally, to provide resilience against sinks that may leave the network or lose connectivity, each node may maintain a keep-alive timer for each sink. If a swarm agent is not received from a sink before its keep-alive timer expires, the sink may be considered unavailable and the best route among the remaining sinks is chosen.

G. SIMPLE's Overhead Message Complexity

The algorithms presented in [10] are mainly designed for scenarios with static sinks where data could be exchanged between any arbitrary pair of nodes. These algorithms synchronize the residual energy of nodes within the same "local broadcast area". From an individual node's point of view, this makes the algorithm's message overhead complexity (i.e. the number of messages transmitted by a node for an instance of route setup) $O(n)$, where $n \geq 1$ is the number of nodes within the "local broadcast area". In addition, it is hard to adapt these algorithms for mobile sink scenarios. The TTDD protocol in [14] is even more complicated since each potential source builds a grid structure of its own spanning the whole network. The message complexity is actually $O(N)$, where N is the number of sources in the network. On the other hand, SIMPLE has an overhead message complexity $O(1)$ since each node may forward a swarm agent from the sink only once.

H. Miscellaneous Issues

1) Heterogeneity of Node Batteries: Node batteries are allowed to be heterogeneous in terms of their capacities and energy consumption rates. In SIMPLE a node's battery capacity is normalized in terms of the maximum number of messages it can forward.

2) Detecting Node Failures: When node i forwards a message to the sink via its downstream neighbor j, it can detect node j's failure by listening for the expected transmission from j. If no transmission is detected from node j within a reasonable amount of time, node i can assume node j is dead and retransmit the message via another downstream neighbor.

3) Static Sink: When a sink stays static, it may still advertise the swarm agent after it receives a given amount of data in order to avoid the use of static routes.

4) Energy Saving by Sleeping: SIMPLE allows nodes to go into the sleep mode. A node can start or stop advertising the swarm agent to switch between sleep and awake states.

VI. Simulation Results

In this section we present the simulation results to verify SIMPLE's performance and evaluate the effect of various environmental factors. We used a custom built simulator, written in MATLAB to generate the results. We compare SIMPLE with both the minimum hop count routing algorithm ([4], [5]) and the MREP protocol [15] and all three protocols were implemented in our simulator. The route discovery process for the minimum hop routing is similar to that for SIMPLE except for the fact that only one packet is sent and no timers are required. The frequency of routing updates was kept the same in all the protocols.

In the simulations, the sink's movement follows a random walk and data packets or messages are generated at each node according to either a Poisson process or at constant intervals. For each setting, the simulation was repeated with 15 random seeds and the results were averaged. The 95% confidence interval in the worst case was found to be approximately 20% of the mean. In the simulations for SIMPLE, the timer values for the *follower* were chosen in the range $[50, 100]$ms. Following the 1st order radio model of [31], the energy consumption costs for transmission ($E_{T_x}(k, d)$) and reception ($E_{R_x}(k, d)$) of a k-bit message transmitted over a distance d is assumed to be:

$$\begin{aligned} E_{T_x}(k, d) &= kE_{elec} + \epsilon_{amp}kd^2 \\ E_{R_x}(k, d) &= kE_{elec} \end{aligned} \quad (45)$$

where $E_{elec} = 62.5/bit$ is the energy dissipated to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ is for the transmitter amplifier. For our results, we normalize the energy consumed for receiving a packet to 1. The ratio of the transmission and reception energy given by the equation above is then taken to obtain the normalized transmission energy.

A. Comparison with MREP and Minimum Hop Routing Algorithms

We first compare SIMPLE with the MREP [15] and minimum hop count routing algorithms. MREP was chosen because it was shown to perform better than existing protocols including those in [22] and tries to address exactly the same max-min residual energy problem as defined in Section III-B. In our simulations, MREP performs broadcast searching for routes to each node from a moving sink. The percentage thresholds *perc* for each route discovery process are set to 20%, 10% and 0%, respectively. We tested SIMPLE with and without swarm agent suppression, denoted by SIMPLE-S and SIMPLE in the figures, respectively. All SIMPLE-S tests use the same parameter set while calculating re-advertisement probability according to Algorithm 1 with $\alpha = \beta = 0.4$, $\delta = 0.2$ and $\gamma = 0.1$.

In this set of experiments, 200 nodes were uniformly distributed in a $100 \times 100m^2$ network area. Each node's transmission range was set to either $35m$ to simulate a dense network (high connectivity) or $25m$ to simulate a sparse network (low connectivity). The precursor and follower packets were 32 bytes each and the report message was 320 bytes. Each node had 500 units of initial energy.

1) Network Lifetime vs. Sink Speed: In Figures 5 and 8 we compare the lifetime of the four protocols for various sink speeds in dense and sparse networks, respectively. For these results, data or reports are generated at each node with rate $\lambda = 0.3$ messages per second and the sink's speed is varied from 2m/s to 10m/s. The swarm agent advertisement rates were kept at 0.2, 0.4, 0.6, 0.8 and 1.0 agents per second for speeds of 2, 4, 6, 8 and 10m/s, respectively. These values are greater than the corresponding lower bounds obtained from the analysis in Section V-D for $\epsilon = 0.01$. As the sink's speed increases, SIMPLE and SIMPLE-S consume more energy with frequent path updates. However, the lifetime increases because the energy depletion rates of nodes is more balanced across the network. The sink's mobility actually helps to avoid draining the energy of the same set of nodes. This is also verified by the results of MREP and min-hop routing. The reason why SIMPLE and SIMPLE-S outperform the other two protocols is that they not only try to minimize each data report's energy consumption, they also take energy balance into consideration. We also note that the swarm agent suppression technique improves SIMPLE's performance.

2) Network Lifetime vs. Report Intensity: Figures 6 and 9 compares the performance of the four protocols when messages are generated at the nodes at different rates according to a Poisson distribution with the sink speed kept at 2m/s. Swarm agents were generated at a rate of 0.5 agents per second for these simulations. When the reporting intensity is moderate, SIMPLE and SIMPLE-S perform much better than MREP and min-hop routing. Their advantage tapers off slightly when the report intensity becomes very high. Figures 7 and 10 compare the four protocols when messages are generated at the nodes at a constant rate. The results are very similar to those for the Poisson case suggesting that SIMPLE's performance benefits do not depend on the underlying traffic model.

B. Effect of Environmental Factors

In this section we observe the effect of various environmental factors on SIMPLE's energy consumption and lifetime, starting with the sink's speed. In this section's simulations, nodes are uniformly distributed in a $100 \times 100m^2$ network area. The transmission range is 25m and nodes' initial energy

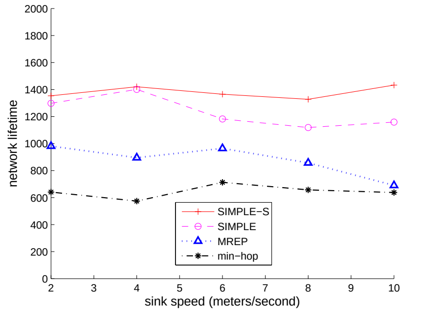

Fig. 5. Lifetime versus sink speeds in dense networks.

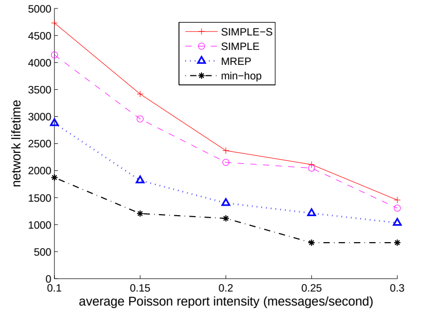

Fig. 6. Lifetime versus Poisson report intensity in dense networks.

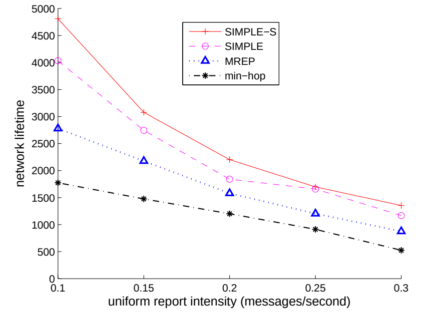

Fig. 7. Lifetime versus CBR report intensity in dense networks.

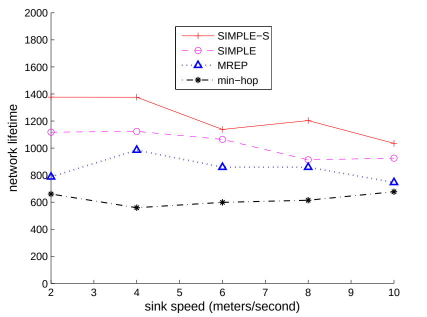

Fig. 8. Lifetime versus sink speeds in sparse networks.

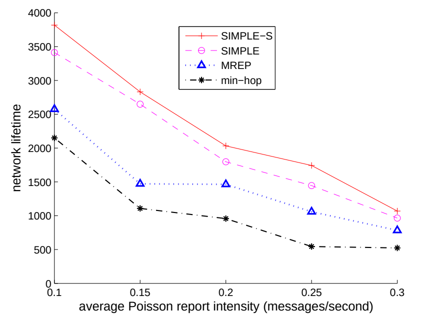

Fig. 9. Lifetime versus Poisson report intensity in sparse networks.

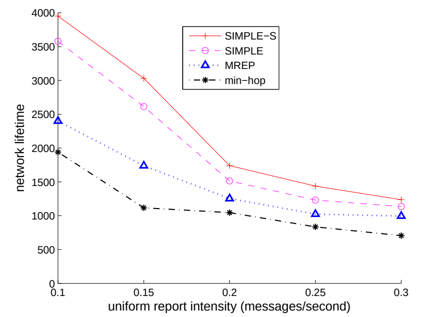

Fig. 10. Lifetime versus CBR report intensity in sparse networks.

is 500 units. Data reports are generated at each node with rate 0.05 messages per second.

1) Effect of Sink Speed and Length of Swarm Agent: Figure 11 shows the effect of the sink's speed and the ratio of data and swarm agent size on the energy consumption induced by the swarm agent (without suppression) for a network of 200 nodes. Swarm agents were generated for these results at a rate of $0.1v$ where v is the sink speed. These values were greater than the minimum swarm agent rates obtained from Section V-D for $\epsilon = 0.01$. It can be seen that for different length ratios, energy consumption induced by the swarm agent only increases slightly as the sink moves faster. This is in concert with the results in Figures 5 and 8. When the swarm agent is much smaller than the data, the energy consumption induced by the swarm agent can be as low as 1%-5%.

2) Effect of Node Density: In Figure 12 we plot the swarm agent's (without suppression) energy consumption as a function of the node density for data and swarm agent size ratios of 10:1 and 50:1 with swarm agents generated at a rate of 0.5 agents per second. When the swarm agent's lengths is small compared to the data, the energy consumption can drop to as low as 5% when the node density reaches 0.08 nodes/m^2. When node density increases, the burden of relaying data becomes less on each node. According to the constrained advertisement model in Section IV-B, nodes relaying less data will have a lower advertisement probability ρ. Thus, energy consumption induced by the swarm agent also decreases. This indirectly verifies that SIMPLE's probability model guarantees the protocol's scalability with the node density.

3) Effect of Swarm Agent Suppression: In this section, we investigate the effect of the swarm agent suppression technique proposed in Section IV-B on the protocol's performance. For these results, we consider a network of 200 nodes, swarm agent rate of 0.5 agents per second and increase the ratio of the swarm agent and data size to 2:5 to enable a more effective observation of the tradeoff between protocol overhead and the network's lifetime.

Figure 13 presents the network's lifetime (y axis) for various swarm agent suppression degrees (x axis). The degree of suppression is represented by the percentage of node energy that is consumed by swarm agents. To show the trend, a fitted curve drawn using 10th degree exponential curve fitting with error bounded within 15% is also shown in the figure. We note that both zero and full suppression lead to lowered network lifetimes. Going left to right, the two extremes in the figure are elaborated as follows:

- **No suppression, dynamic:** Protocols in this category try to continuously update the whole network with the sink's latest location. The max-min path chosen will thus be optimal and the network's residual energy is optimally balanced, which prolongs the network's lifetime. Although [2] is not energy aware, it does belong to this category as does SIMPLE without suppression. However, even though protocols in this category can find the energy-wise optimal path, the significant overhead decreases the network's lifetime.
- **Full suppression, static:** Paths to the sink are updated as infrequently as possible. Most nodes are unaware of the sink's movement and information is delivered through

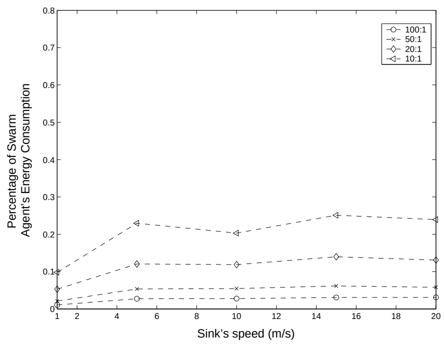

Fig. 11. Effect of Sink's Speed and swarm agent's size relative to the data report's size.

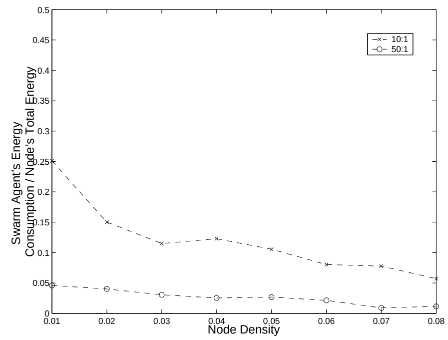

Fig. 12. Node Density vs. Swarm Agent's Energy Consumption

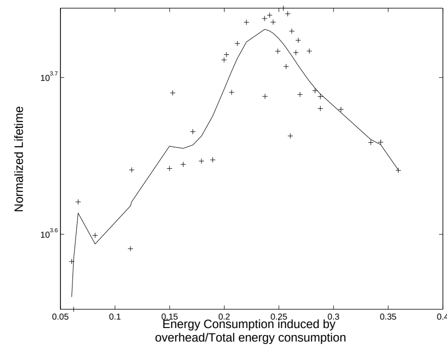

Fig. 13. Lifetime achieved with different suppression ratios.

stale and usually sub-optimal and longer routes. However, energy is conserved in the sense that protocol overhead is trivial compared to the previous case. In addition, energy of nodes on the static paths may get depleted very quickly, which shortens the network lifetime.

C. Multi-sink Scenarios

In this section we investigate the energy depletion in multi-sink scenarios. When multiple sinks are present in a small scale network, swarm agents from all sinks can traverse the whole network so that nodes can find the closest sink to deliver their information. A large scale network can be subdivided into small scale ones and sinks, with their associated swarm agents, will be confined in their respective subareas. Since the subareas in a large scale network are equivalent to small scale networks, simulations in this section focus on the energy depletion in a small scale network with multiple sinks.

In this simulation, 400 nodes, with 25m as their transmission range and 500 units of initial energy, are present in a network of $200 \times 200m^2$ area. The speed of the sinks is kept at 10m/s, data is generated at each node with rate $\lambda = 0.05$ and the swarm agent (without suppression) rate was kept at 0.5 agents per second. Figure 14 shows that for a given reporting intensity, as the number of sinks increases from 1 to 4, the time it takes for the average residual energy at a node to drop from 500 to 150 becomes longer instead of shorter. This reason is that although multiple sinks introduce greater energy consumption due to more swarm agents, it also helps decrease the average hop count between nodes and their corresponding sinks, as shown in Figure 15. The energy saved by traversing a smaller number of hops outweighs the increased energy consumption due to more swarm agents.

D. Protocol Resilience Against Node Failures

In this section we verify SIMPLE's resilience against node failures. Initially, 200 nodes, with initial energy of 500 units and transmission range 25m, are distributed in a $100 \times 100m^2$ area. One mobile sink is present in the network, with a speed of 10m/s and the swarm agent (without suppression) rate is 0.5 agents per second. Report events are generated at each node with a rate of 0.05 messages per second. In addition to the max-min path, nodes also record multiple backup paths to counteract node failures. Figure 16 shows that with only two backup paths the protocol's resilience against node failures is greatly improved.

VII. Conclusions

This paper presents an energy aware data acquisition protocol for networks with mobile sinks. The protocol design is based on techniques of swarm intelligence, energy-wise max-min path and a probabilistic model for dynamically updating the max-min paths. The swarm intelligence approach maximizes individual node's lifetime since it greatly simplifies the node's operations, keeping requirements in line with a typical sensor or node's limited computational capabilities, restricted storage and limited energy. The protocol tries to maximize the network's lifetime by dynamically choosing energy efficient paths and balancing the residual energy at each node. SIMPLE scales with multiple sinks and is robust against node failures.

VIII. Appendix

Case 1: When $x_j(0), y_j(0) \in L_j$, $Z_j(T_j^1) = 1$ and $Z_j(0) = 1$, the sink is initially within a circular region of radius r centered at $x_j(0), y_j(0)$. Then the probability $P[Z_j(T_j^1) = 1]$ depends on the likelihood that the sink is still within the circular region at time $t = T_j^1$. This probability can be evaluated by integrating the pdf of the position of the sink over the circular region. Figure 17 shows such a scenario where node j's position is marked by A and the sink's initial position is marked by B. The probability that the sink stays in node j's neighborhood can be obtained by first integrating the pdf of the sink's motion over the circle of radius $r - l$ centered at B and then over arcs subtending an angle of $2\pi - 2\theta$ at B as the radius sweeps over the range $r - l \leq x \leq r + l$. Using elementary trigonometry

$$\theta = \pi - \beta = \pi - \cos^{-1}\left(\frac{-r^2 + l^2 + x^2}{2lx}\right) \qquad (46)$$

For two dimensional Brownian motion with variance α, the distance x and angle ϕ of the sink at time τ is with respect to its origin at time 0 is given by

$$p(x, \phi) = \frac{x}{\pi \alpha \tau} e^{-\frac{x^2}{\alpha \tau}} \qquad 0 \leq \phi < 2\pi, \ \ 0 \leq x < \infty \qquad (47)$$

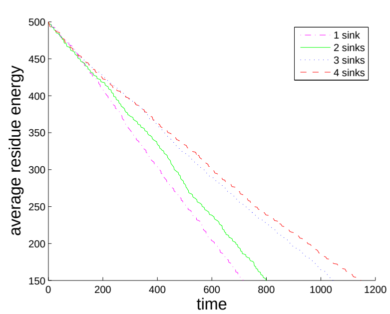

Fig. 14. Energy in multi-sink scenarios does not deplete faster than single-sink scenarios.

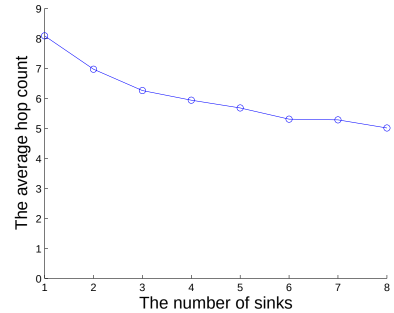

Fig. 15. The average hop count decreases as the number of sinks increases

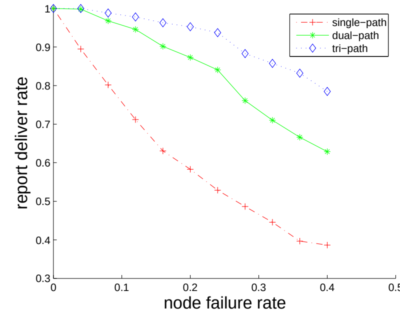

Fig. 16. Multiple paths improve the protocol's resilience against node failures

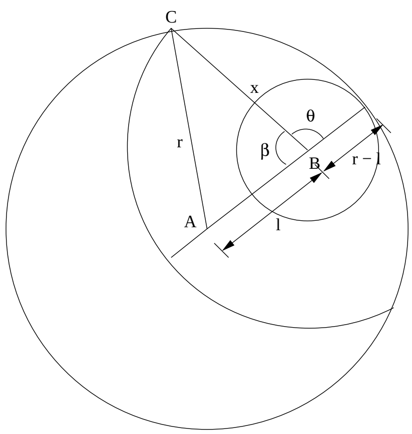

Fig. 17. Sink movement with $Z_j(0) = 1$.

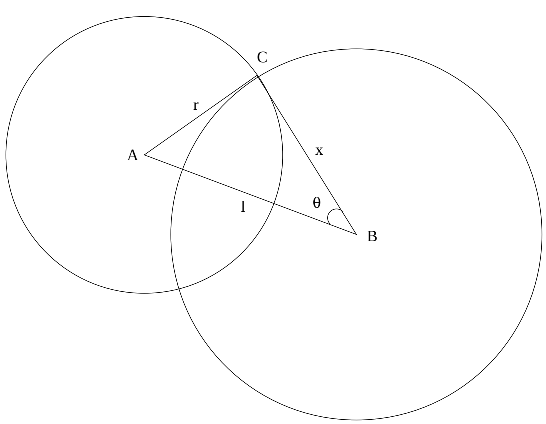

Fig. 18. Sink movement with $Z_j(0) = 0$.

Then

$$P[Z_j(T_j^1) = 1 \mid \Delta_{sj}(0) = l,\, T_j^1 = \tau] = \int_0^{r-l} \int_0^{2\pi} \frac{x}{\pi\alpha\tau} e^{-\frac{x^2}{\alpha\tau}} d\theta' dx + \int_{r-l}^{r+l} \int_0^{2\pi-2\theta} \frac{x}{\pi\alpha\tau} e^{-\frac{x^2}{\alpha\tau}} d\theta' dx \tag{48}$$

Case 2: When $x_j(0), y_j(0) \in L'_j$, $Z_j(T_j^1) = 1$ and $Z_j(0) = 0$, the sink is initially outside the circular region of radius r centered at node j's location but moves inside the circle at time T_j^1. Figure 18 shows such a scenario using the same notation as in the previous case. In this case we integrate for arcs subtending an angle 2θ as x varies from $l - r$ to $l + r$. The angle θ is given by

$$\theta = \cos^{-1}\left(\frac{-r^2 + l^2 + x^2}{2lx}\right) \tag{49}$$

and thus the probability that the sink, starting at a distance of l, $l > r$, from node j at $t = 0$ becomes its neighbor at $t = T_j^1$ is given by

$$P[Z_j(T_j^1) = 1 \mid \Delta_{sj}(0) = l,\, T_j^1 = \tau] = \int_{l-r}^{r+l} \int_0^{2\theta} \frac{x}{\pi\alpha\tau} e^{-\frac{x^2}{\alpha\tau}} d\theta' dx \tag{50}$$

REFERENCES

[1] D. Coffin, D. Van Hook, S. McGarry and S. Kolek, "Declarative ad hoc sensor networking," *Proceedings of SPIE Integrated Command Environments Conference*, San Diego, CA, July 2000.
[2] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *Proceedings of IEEE/ACM MOBICOM*, pp. 56-67, Boston, MA, August 2000.
[3] A. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, Wiley, 2006.
[4] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing, Imielinski and korth, Eds.*, vol. 353, pp. 153-181. Kluwer Academic Publishers, 1996
[5] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.
[6] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *Proceedings of IEEE ICC*, pp. 1633-1639, June 1998,
[7] J. Zhu, C. Qiao and X. Wang, "A comprehensive Minimum Energy Routing Scheme for Wireless Ad hoc Networks," *Proc. of IEEE INFOCOM*, March 2004.
[8] S. Singh, M. Wu, and C. S. Raghavendra, "Power-aware routing in mobile ad-hoc networks," *Proc. of ACM Mobicom*, October 1998.
[9] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Adhoc Networks," *Proc. of IEEE INFOCOM*, March 2002.
[10] K. Kar, M. Kodialam, T.V. Lakshman, and L. Tassiulas, "Routing for Network Capacity Maximiztion in Energy-constrained Ad-hoc Networks," *Proc. of IEEE INFOCOM*, April 2003.
[11] Q. Li, J. Aslam, and D. Rus, "On-line power-aware routing in wireless ad-hoc networks," *Proc. of IEEE/ACM MOBICOM*, July 2001.
[12] A. Sankar and Z. Liu, "Maximum Lifetime Routing in Wireless Ad-hoc Networks," *Proc. of IEEE INFOCOM*, March 2004.
[13] M. Chen, T. Kwon and Y. Choi, "Energy-efficient differentiated directed diffusion (EDDD) in wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 231-245, January 2006.
[14] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," *Proc. of IEEE/ACM Mobicom*, September 2002.
[15] B. Zhang and H. Mouftah, "Energy-aware on-demand routing protocols for wireless ad hoc networks," *Wireless Networks*, vol. 12, No. 4i, pp. 481-494, August 2006.

[16] Z. Liu, D. Xiu and W. Guo, "An energy-balanced model for data transmission in sensor networks," *Proc. of IEEE VTC*, 2005.
[17] A. Boukerche, I. Chatzigiannakis and S. Nikoletseas, "A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range," *Computer Communications*, vol. 29, no. 4, pp. 477-489, February 2006.
[18] I. Haque and C. Assi, "OLEAR: Optimal Localized Energy Aware Routing in Mobile Ad Hoc Networks," *Proceedings of IEEE ICC*, pp. 3548-3553, Istanbul, Turkey, June 2006.
[19] S. Wu and K. Candan, "Power-aware single- and multipath geographic routing in sensor networks," *Ad Hoc Networks*, vol. 5, no. 7, pp. 974-997, September 2007.
[20] N. Wang and C. Chang, "Probabilistic-Based Energy Aware Routing for Network Lifetime and Source Level Fairness in Wireless Sensor Networks," *Proceedings of ICSNC*, 2006.
[21] T. Shu, M. Krunz and S. Vrudhula, "Power balanced coverage-time optimization for clustered wireless sensor networks," *Proceedings ACM MOBIHOC*, pp. 111-120, Urbana-Champaign, IL, May 2005.
[22] J. Li and P. Mohapatra, "PANDA: A novel mechanism for flooding based route discovery in ad hoc networks," *Wireless Networks*, vol. 12, no. 6, pp. 771-787, December 2006.
[23] X. Zhang, X. Gao, D. Shi and D. Sung, "Lifetime-Aware Leisure Degree Adaptive Routing Protocol for Mobile Ad Hoc Networks," *Proceedings of ICWMC*, 2007.
[24] G. Zussman and A. Segall, "Energy Efficient Routing in Ad Hoc Disaster Recovery Networks," *Proc. of IEEE INFOCOM*, April 2003.
[25] K. Lai and M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," *Proc. of ACM SIGCOMM*, 2000.
[26] S. Tilak, A. Murphy and W. Heinzelman, "Non-uniform information dissemination for sensor networks," *Proceedings of ICNP*, 2003.
[27] O. Tickoo and B. Sikdar, "On the impact of IEEE 802.11 MAC on traffic characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 189-203, February 2003.
[28] N. Bisnik and A. Abouzeid, "Rate-distortion bounds on location-based routing protocol overheads in mobile ad hoc networks," *Proceedings of Allerton conference on Communication Control and Communications*, Urbana-Champaign, IL, September 2006.
[29] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley and Sons, New York, 2nd Edition, 2006.
[30] R. Gallager, "Basic limit on protocol information in data communication networks," *IEEE Transactions on Information Theory*, vol. 22, no. 4, pp. 385-398, July 1976.
[31] W. Heinzelman, A. Chandrakasan and H. Balakrishnam, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of Hawaii Conf. System Sciences*, Jan. 2000

PLACE PHOTO HERE

Hua Yang Hua Yang is a senior researcher at Intel Corporate Technology Labs, Shanghai. She is currently working on interference mitigation techniques for next generation WiMAX networks and is also involved in IEEE's standardization effort for next generation WiMAX system and 4G networks. Ms. Yang obtained her bachelor's and master's and Ph.D from Xi'an Jiaotong University (P. R. China), Beijing University of Posts and Telecommunications (P. R. China) and Rensselaer Polytechnic Institute, Troy NY, respectively. After her master's, she worked for Siemens Beijing. Her current research interests include mobile broadband network interference mitigation, wireless mesh and ad-hoc networks, broadband networks and next generation network integration.

PLACE PHOTO HERE

Fengji Ye Fengji Ye received his PhD degree in Electrical Engineering from Rensselaer Polytechnic Institute in 2005. His research interests include wireless MAC protocols, wireless sensor networks, and performance evaluation of wireless ad-hoc networks. He has published various papers in leading IEEE/ACM conferences and journals. After graduation he joined Cisco Systems, Inc.

PLACE PHOTO HERE

Biplab Sikdar (S'98, M'02) received the B. Tech degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, the M. Tech degree in electrical engineering from Indian Institute of Technology, Kanpur and Ph.D in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA in 1996, 1998 and 2001, respectively. He is currently an Associate Professor in the Department of Electrical, Computer and Systems Engineering of Rensselaer Polytechnic Institute, Troy, NY, USA. His research interests include wireless MAC protocols, network routing and multicast protocols, network security and queueing theory. Dr. Sikdar is a member of IEEE, Eta Kappa Nu and Tau Beta Pi.