SIMPLE: using Swarm Intelligence Methodology to design data acquisition ProtocoL in sEnsor networks with mobile sinks

H. Yang, F. Ye and B. Sikdar
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180

***Abstract*— This paper addresses the data acquisition problem in sensor networks using mobile sinks. Sensor nodes' low computational capabilities and limited energy motivate our design of a swarm intelligence based, energy aware protocol, SIMPLE, to route data to the mobile destination via the shortest paths. Using a swarm agent technique to integrate nodes' residual energy as a metric for the shortest path selection, SIMPLE maximizes the network's lifetime by evenly balancing residual energy across the network and minimizing the protocol overhead. The protocol's resilience against node failures is guaranteed by the multiple path technique. It scales well with both the network size and multiple sinks. Simulation results are presented to observe and verify SIMPLE's performance and robustness. Compared with existing algorithms, SIMPLE is shown to have superior performance. A general tradeoff model is presented to evaluate the performance tradeoffs associated with different protocol parameters.**

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

I. INTRODUCTION

THE tremendous potential of sensor networks in both military and civilian environments has been widely recognized. These networks can have thousands of sensors involved over large areas and nodes are typically limited in their battery and computational capabilities. The introduction of mobility, either in the sensors or in the agents which collect data from them, increases the applicability and possible deployment scenarios of these sensor networks. However, it also makes the design of protocols more challenging and complicated. For example, sensor networks could be deployed to monitor battlefields, forests or civilian areas. Information is generated at the sensors and reported to the sinks, which could be soldiers, forest rangers or policemen, respectively. In these scenarios, which reflect the scenarios of interest in this paper, most of the sensors stay static while sinks are mobile. The problem of interest is: *how should the static sources report their data to the mobile sink so that network and individual sensor's lifetime is maximized?* To address this issue, we present a new on-line, energy aware protocol motivated by swarm intelligence theory to carry out data acqusition even when the source nodes are not aware of the mobile-sink's current location.

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 sensors in the network with its current location information [21], [22], [23]. However, such frequent updates lead to excessive consumption of the sensors' battery in addition to creating traffic congestion. A two-tier approach to data dissemination (TTDD) is proposed in [6] wherein each source forms a grid like path to the sink. However, aside from being energy unaware, the communication and state overheads associated with maintaining these routes degrade the protocol's scalability and ability to maximize network lifetime.

With the specific goal of maximizing the network or sensor lifetime in the "data acquisition using mobile sink" scenarios, this paper presents a protocol, SIMPLE, based on the concept of swarm intelligence [20]. Without requiring individual sensors to possess much intelligence or cooperate with each other tightly, each of them follows simple rules and by their collective behavior the optimum is achieved. Using a packet-pair based swarm agent, SIMPLE ensures that messages are forwarded to the sink along the path with most remaining energy and tries to balance the energy at each sensor. In particular, SIMPLE achieves the following:

1) **Smart Data Delivery to the Mobile Sink,** designed to tolerate a degree of information inaccuracy regarding the sink's location. Thus frequent and expensive updates of all sensors with sink's location information are avoided. The protocol design ensures that a sensor's knowledge of the sink's location gets progressively more accurate as it gets nearer to the sink. Thus, even if data delivery is initiated with inaccurate sink location, the route will be rectified asymptotically and collectively.
2) **Lifetime Micro-Maximization,** defined as maximizing individual sensor's lifetime. Due to a sensor's limited energy, storage and computational capabilities, it is not feasible or efficient to require each sensor to possess network-level intelligence for determining the optimal paths. By employing a swarm intelligence approach, individual sensors are relieved from the burden of collecting, storing and processing global information while the global optimum is achieved and the system reliability and robustness is significantly improved.
3) **Lifetime Macro-Maximization,** aimed at network level lifetime maximization. SIMPLE uses a lightweight swarm agent to efficiently locate the best paths from data sources to the moving sink. The best paths are dynami-

cally updated based on nodes' residual energy, even when the sink stays static. Paths that involve nodes with less residual energy are avoided so that the whole network's residual energy distribution stays even, thereby prolonging the network lifetime.

4) **Robustness and Scalability,** with little overhead, nodes can keep record of multiple path gradients to counteract node failure events. When multiple sinks are present in the network, nodes can choose to report to the closest one to save energy. The protocol also scales to multiple sources.

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

Our approach is based on swarm intelligence, as discussed in [20]. Swarm intelligence argues that the individual agents do not have to possess much intelligence or cooperate with each other tightly to fulfill complicated tasks. Each agent follows simple rules and by their collective behavior the optimum is achieved. Swarm intelligence approach performs greatly as far as reliability and adaptability to network dynamics are concerned. It scales easily since global information is not necessary at individual agents to achieve the global optimum. Swarm intelligence based algorithms have been proposed in literature for wired telecommunication networks [1], [2], [3].

The problem of data acquisition in ad-hoc network with static sinks has been extensively studied in recent years. Using "hop-count" as the metric, [10], [11] use shortest path for routing without considering the energy constraints. In [12] it is shown that energy aware metrics can significantly improve the performance of routing protocols in wireless ad hoc network. A number of minimum energy consumption routing protocols, which choose the path that consumes the minimum energy to deliver the data are proposed in [13], [14], [15], [16]. However, this leads to a much faster power drainage of nodes on the best path and consequently causes uneven energy distribution across the network and shortens its lifetime. In [4] the lifetime maximization problem is formulated as an offline linear programming problem and flow augmentation/redirection algorithms are proposed to balance energy consumption across nodes. In [5] a technique to maximize a node's utility given a network lifetime is discussed. However, these schemes require full knowledge of traffic demands and do not handle node insertion and deletion well. In [7], [8] "maximizing network lifetime" is taken as the objective and online algorithms are developed for static networks to route the data. The similar offline algorithm of [9] deals with static or slowly changing dynamic networks.

Most of the aforementioned routing protocols assume knowledge of the destination's identifier-based address. In the mobile sink scenario, frequently updating all sensors with sink's current location leads to significant overheads. Recent literature suggests several alternative approaches. Directed diffusion [22] routes data based on data interests periodically broadcasted by the sink. The sink reinforces certain paths for a given source based on previously received data from the source. The fact that once the sink moves the reinforced paths are not valid anymore makes the scheme ineligible for accommodating high level of network and sink dynamics. The author of [17] considers a scenario where random walks are used to route data or queries in the network and the probability that they successfully reach the destination is evaluated for a number of scenarios. However these are all probabilistic strategies that do not guarantee query success. A data dissemination model for sensor networks with mobile sinks is proposed in [6] where data is sent to the sink through its primary and immediate agents. The first drawback of this scheme is that excessive state information is maintained in the network since each source sets up its own "grid" spanning the whole network. The grids have to be rebuilt once they time out. As the number of sources grows, significant overhead is introduced by grid setup and maintenance, which further damages the protocol's scalability. Second, unlike our scheme, outdated paths usually lead to data loss. Finally, all data to the sink are relayed through its primary and immediate agents, introducing central points of failure.

III. Background and Definitions

SIMPLE's assumptions and relevant definitions are presented in this section.

A. Assumptions and Terminology

Following are basic assumptions about the network:

- No prior knowledge about the sink's mobility characteristics is available.
- All sensors in the network are potential sources. No prior knowledge about source data generation characteristics is available.

These assumptions reflect the conditions in most realistic deployment scenarios and are necessary to ensure that the developed protocol is practical. Following are some definitions that will be used throughout this paper:

- **Lifetime** of the network is defined as the time till the first node in the network dies. Note that although the network may still function when certain nodes start to run out of energy, this network lifetime definition is an important and meaningful indicator to the protocol's efficiency.
- **Gradient** of a node indicates its next hop neighbor on the shortest path leading to the sink. In addition to the shortest path gradients, nodes may record suboptimal gradients to counteract node failures.
- **Node** and "sensor" are used interchangeably within this paper.
- **Downstream and Upstream** Downstream is defined as "to-the-sink" direction, while upstream refers to the opposite.

B. Scalability Constraint on Protocol Design

While protocols designed for the mobile sink scenario may have very different guidelines, there exists a common constraint: it is impractical to keep the whole network continuously updated with the mobile sink's location information. Thus the protocol should be able to tolerate a certain degree of inaccurate or even no information regarding the sink's current position. Otherwise the overhead incurred to keep the whole network up-to-date will grow unboundedly as the network size grows. Through this paper we will refer to this as the "scalability rule" and it will be a cornerstone of the design requirements.

C. Shortest Path Definition

The first problem in delivering data to the mobile sink is how to choose the path. Recent literature [7], [4], [8] addresses the issue related to the shortest path. It is well known that statically choosing the route which consumes the minimum energy to deliver a message will actually shorten the network's lifetime due to unbalanced energy consumption. To address this, schemes have been proposed to keep the network energy balanced by routing data through the energy-wise shortest path and dynamically updating these routes. In [7] different algorithms that serve this purpose are discussed: CMAX, zP_{min} in [8] and max-min. We refer the reader to [7], [8] for details regarding CMAX and zP_{min} and now elaborate on the max-min algorithm.

Suppose between a given source and destination there exist n paths, which we denote as $p_j, j \in 1, 2, \cdots, n$. 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. All the three algorithms mentioned above take the energy balance issue into consideration, which helps to prolong the network lifetime. 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 is 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, through this paper the term "shortest path" will be used to refer to the path specified by Eqn. (1).

IV. The Data Acquisition Protocol: SIMPLE

In the previous section we addressed the problem of in what manner should the source data be delivered to the sink, using the max-min algorithm. The sink's mobility brings up another critical issue: where should the data be delivered? In this section we address this issue and present our protocol, SIMPLE, which has been specifically designed for this purpose. 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-B.

Data delivery in the mobile sink scenario becomes complicated due to the fact that the sink keeps changing its location in an unpredictable manner. This makes the continuous possession of the accurate sink location information at each sensor rather difficult considering the "scalability rule" mentioned earlier. To ensure reliable data delivery in the presence of partial or outdated information, we develop a swarm agent based approach. The swarm agent distributively sets up and updates at each node the gradient pointing to the downstream neighbor on the shortest path leading to the sink. It is advertised by the sink only when the sink loses contact with some of its one hop neighbors. Each node relays the swarm agent based on a probabilistic model so that unnecessary relays are suppressed without sacrificing the performance. Data from the sources reache the sink by taking the path marked by the gradient at each node. Since the shortest path is setup on a max-min basis as mentioned before, data acquisition is carried out in an energy balanced manner.

A. A Lightweight Swarm Agent

In this section, we propose a swarm agent based technique to compute the max-min paths in a wireless network. Various max-min routing algorithms for sensor network with static sinks have been proposed in literature by adapting traditional routing algorithms like Dijkstra or Bellman-Ford. The major obstacle for using Dijkstra's algorithm in sensors network is that too much information has to be collected at each individual node before it can calculate the shortest path. While zone-based Dijkstra algorithm is more scalable, it suffers from the problem of inaccurate information which leads to quasi-shortest paths. Also, oscillations of Bellman-Ford algorithm before it actually converges can cause significant energy consumption in sensor networks. Aside from this, in the adapted Bellman-Ford algorithm presented in [8], one shortest path calculation has to invoke the Bellman-Ford algorithm multiple times. All these induce significant amounts of message exchanging, which greatly reduce the network lifetime.

Our shortest path protocol is motivated by swarm intelligence theory and bandwidth measurement techniques in wired networks presented in [18] and references therein. Rather than storing complete path information for the shortest path, a sensor only maintains a "gradient" pointing to its downstream neighbor on the shortest path leading to the sink. As we will show later, in the absence of accurate location information, this makes it possible for the sensors to rectify the message delivery path collectively. Based on Eqn. (1), a "swarm agent" is designed to mark out the shortest path gradient for each sensor. Each swarm agent is stamped with an unique and increasing sequence number and consists of two very short packets, namely the *precursor* and *follower*. We now show how the swarm agent probes and updates the shortest path gradients at all nodes using the example topology shown in Figure 1.

The swarm agent is initially advertised to the network by the sink each time it moves out of some neighbors' transmission range. As indicated in Figure 2, upon receiving of the *precursor*, each node immediately re-advertises it to all its neighbors and starts a timer with initial value as, for example:

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

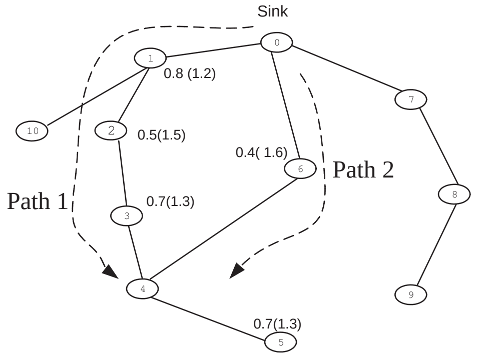

Fig. 1. Example: using swarm agent to update the shortest path.

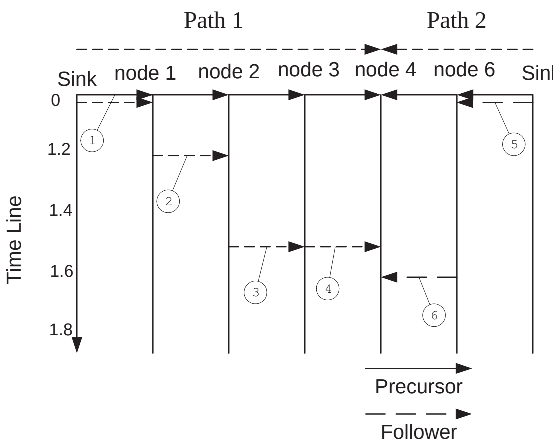

Fig. 2. Propagation of the *precursor* and *follower*.

where e_r is the node's remaining energy (normalized between [0, 1]). Thus, nodes with higher residual energy will time out faster. Note that the function above for calculating T is just an example. Realization of our scheme does not depend on any specific function as long as it is monotonously decreasing with certain bounds. When a node receives the *follower* with the same sequence number as the *precursor*, it does not re-advertise it unless its timer times out. In Figure 1 numbers out of the parentheses are nodes' normalized residual energy and numbers in the parentheses are sensors' corresponding timer value. Figure 2 also shows the manner in which the *follower* will be 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. The *follower* is critical for detecting the shortest path. Based on the definition of the previous section, $sink \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ (path 1 in the figure) is the shortest path between the sink and node 4. Note that even though the other path $sink \rightarrow 6 \rightarrow 4$ (path 2 in the figure) has less hops than the shortest 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 will take 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:

1) Node 1 receives the *follower*;
2) Node 1's timer times out at time 1.2, and the *follower* is sent out;
3) Node 2 receives the *follower* at time 1.2 and sends it out at time 1.5 when its timer times out;
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;
5) Node 6 receives the *follower* at time 0;
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 less hops. Thus in our scheme, the first copy of the *follower* will always arrive along the shortest path. In Section V we formally prove that our technique indeed selects the shortest path. Since each node can safely restrain from relaying all *followers* except the first copy, this greatly suppresses the amount of swarm agent copies that are circulating in the network. In the example, since node 4 receives the *follower* from node 3 first, it makes node 3 its "gradient" on the shortest path leading to the sink. To counteract node failures or sleeping, node 6 can be taken as the backup path gradient to the sink.

It should be pointed out that for gradient initialization immediately after the deployment of the sensor network, since every node has same residual energy, E_{init}, our scheme actually initializes the shortest path as the path with minimum hops, which is consistent with energy efficiency rules. From an individual sensor'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, which enables the scheme to scale. The scalability issue will be further elaborated upon in Section V-B.

From the whole network's point of view, sensors thus report their data in an energy-balanced manner. Note that the sink's further movement might invalidate the current shortest path and even cause loss of messages. In section V we will address this problem and show that the messages will be successfully delivered.

A strongpoint of our scheme 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 sensors, such as queries. If the sensors' report is triggered by queries from the sink to all sensors, the swarm agent can actually be integrated into the queries with little extra cost.

B. Constrained Advertisement Model

The scheme designed above updates the shortest path from each sensor to the sink using only one lightweight swarm agent. However, some of these advertisement 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 a "utility" for each node. Node i's utility increases by a unit for each node j that picks i as its next hop on the shortest 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.

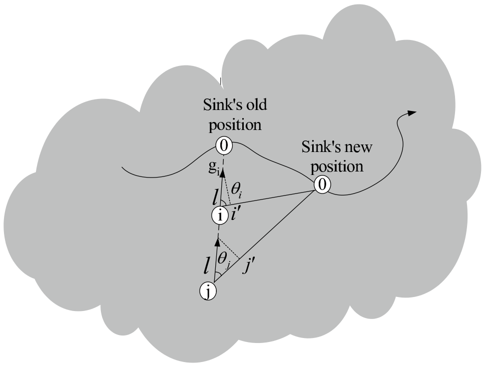

Fig. 3. Scenario 1: Sink's movement has lower effect on nodes further away.

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 with the same sink displacement, sensors further away are less affected. Thus, gradients of sensors further away from the sink can be updated less frequently as compared to sensors nearer to the sink. Note that the gradient is updated based on downstream sensors' advertisements. All these imply that the advertisement intensity can be dampened as the sensor gets further away from the sink.

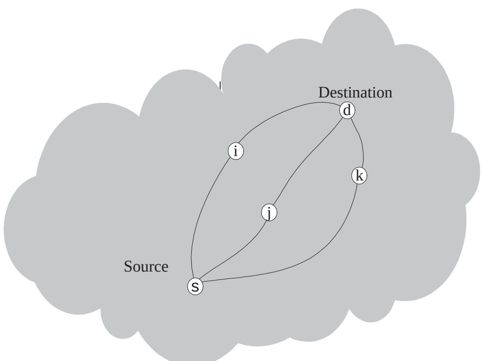

Fig. 4. Scenario 2: Sensors with more residual energy should advertise more actively.

Scenario 2: In Figure 4, i, j and k are three nodes on paths p_i, p_j and p_k between source s and destination d. If node i has the maximum residual energy, then node j and k's advertisements will lower their utility/energy ratio since finally path p_i will be chosen as the shortest path. Ideally, we would like to have only the nodes on the shortest path to advertise the swarm agent, while all other nodes suppress their advertisements to save energy.

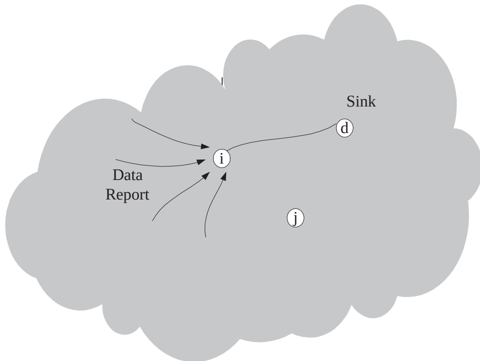

Fig. 5. Sensors relaying more data should advertise more actively

Scenario 3: In Figure 5 node i frequently relays data to the sink, while node j seldom does so. In this case, node i should advertise more 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 advertisement will not increase its utility but cause decreasing of its utility/energy consumption ratio. Thus, we should have nodes that seldom relay data to advertise less 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 introduce a probabilistic model where each node re-advertises the swarm agent based on a probability ρ. Based on the discussion above, we will have a sensor's re-advertising probability ρ:

1) increase each time 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.

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 model described above is essentially a tradeoff model between network's energy balance and overhead energy consumption. Non suppression of advertisements makes the network most balanced but causes maximum energy consumption, while suppressing all advertisements causes the opposite. The model adapts to node insertions and deaths fairly easily. When a new sensor joins the network, it can simply start to forward 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 agent, which naturally removes the node from their next hop candidates list.

C. Networks with Multiple Sinks

When multiple sinks are present in the network, two different schemes can be applied based on the network scales to avoid the energy consumption incurred by too much swarm agent broadcast:

1) In **networks of small scale**, each sink can move liberally within the whole supervised area. Its swarm agent will also traverse the whole network. Note that if a sensor receives multiple swarm agents from different sinks, it will choose to relay the one from the the closest sink. Thus, each node will report to the closest sink via the corresponding shortest path. This also effectively suppresses the amount of swarm agents circulating in the network.
2) In **networks of large scale**, the supervised area can be pre-divided into sub-regions with one sink in each of them. An example of the application scenario is a policeman patrolling in the areas that he is responsible for. In this case the swarm agent from each sink will only traverse sensors located in the area that the sink belongs to.

These two schemes can be applied collaboratively in the same network, in which case each sub-area is covered by multiple sinks. Simulation results in Section VI verify that multiple sinks do not necessary leads to shorter lifetime.

V. Analysis of SIMPLE

In this section, we analytically address various aspects of SIMPLE.

A. Validity of the Shortest Path

Theorem 1: When MAC and other delays are negligible compared to the swarm agent's timer, paths to the sink specified by Eqn. (1) are accurately marked out by the swarm agent.

Proof: Consider an arbitrary node v with n paths to the sink which form a set $P = \{p_i | i \in 1, 2 \cdots n\}$. At time t the sink sends out a swarm agent that consists of a *precursor* and a *follower*. Define a mapping function:

$$t_v = M(e_v) \tag{4}$$

where $M(e_v)$ could be any bounded and monotonous decreasing function, where e_v is the residual energy of node v. 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:

$$\begin{aligned} T_j^k &= \max\{T_j^{k-1}, t_j^k\} \\ &= \max\{T_j^{k-1}, M(e_j^k)\} \end{aligned} \tag{5}$$

The swarm agent will be re-advertised by node v_j^k when T_j^k times out. Thus, finally node v will receive from path p_j a swarm agent 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 consider that node v receives swarm agents from n different paths. It is easy to see that an agent with a shorter "agent timer" always arrives earlier. From the monotonous decreasing nature of the mapping function (4), agent timer of the first arriving swarm agent is T_x where:

$$\begin{aligned} x &= \arg\min_{1 \le j \le n} T_j \\ &= \arg\min_{1 \le j \le n} \max_{1 \le k \le h_j} M(e_j^k) \\ &= \arg\max_{1 \le j \le n} \min_{1 \le k \le h_j} e_j^k \end{aligned} \tag{6}$$

The equation above is exactly the same as Eqn. (1) which defines the shortest path thereby proving the theorem. ■

Note that all the n paths leading to node v can share many intermediate nodes among each other, 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. Scalability of SIMPLE

SIMPLE is developed based on the theory of swarm intelligence which has been proved to scale in biological areas. The first 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 define by its radius R, indicating that at least one node advertising the swarm agent is R hops away from the sink. In order to prove the scalability of the probabilistic advertising model, we now show that R is bounded even when the network size is not. If we define $C(G)$ as coverage of the sensor network G, then we have:

Theorem 2:

$$\lim_{C(G) \to \infty} \text{Prob}(R \to \infty) = 0 \tag{7}$$

Proof: Suppose the mobile sink is advertising the swarm agent with an intensity Γ_s, i.e. swarm agents are advertised to the network at a rate of Γ_s/second. Define Γ_{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 shortest path from node v_i to the sink. Then at node v_i:

$$\Gamma_{v_i} = \Gamma_s \times \rho_{v_i} \times \prod_{v_j \in SP_{v_i}} \rho_{v_j}$$

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

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

we have

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

where $h(v_i)$ is node v_i's hop count to the sink and $h(v_i) \rightarrow \infty$ when $C(G) \rightarrow \infty$. 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:

$$\lim_{C(G)\rightarrow\infty} \text{Prob}(R \rightarrow \infty) = 0 \tag{8}$$

■

Eqn. (8) indicates that the "advertisement scope" is bounded even when the sensor network's size grows larger. In the section above we assume that the delay introduced by the MAC layer can be ignored since typically they are small (queueing delay can be eliminated by assigning the swarm agent a higher priority). This is not true when hop counts from the sink goes to infinity, whence the accumulated delay might become quantitatively comparable with the timer's value. The swarm agent consists of two very short packets, whose typical one-hop transmission delay would be less than 1ms in typical MAC protocols [24]. If we bound the timer's minimum value as 1s, it will take 1000 hops for the accumulative transmission delay to reach the magnitude of the timer's value.

Due to the "scalability rule" mentioned in section III-B, as nodes get further away from the sink, its gradient might not be updated often. In Section V-D 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.

Another scalability issue concerns multiple sinks in the network. As described above, two schemes are proposed for both large and small scale networks. Simulation results in Section VI verify that SIMPLE's performance will not be deteriorated as the number of sinks increases.

C. Swarm Agent Frequency

To keep the sensors abreast of its current location, the sink occasionally sends out swarm agents. This may be done either periodically or only when the sink loses contact with some of its neighboring sensors. In this section, we determine the update frequency required to ensure that the probability that the sink loses contact with any of the sensors currently in its range after t units of time is less than an arbitrary constant β, $0 < \beta < 1$.

For our analysis, we assume that the sink's mobility is governed by a two dimensional random walk. After every τ units of time, the sink randomly chooses an angle θ, distributed uniformly over $(0, 2\pi)$ and moves a distance d along that direction. After a random amount of time t (which for ease of derivations is assumed to be an integral multiple of τ), the sink moves a distance $R(t)$. We first establish the distribution of $R(t)$.

Claim 1: If $t/\tau \geq \sqrt[3]{\frac{3r^4}{16\epsilon d^4}}$ then $\text{Prob}\{R(t) \leq r\}$ follows a Rayleigh distribution with parameter $nd^2/2$.

Proof: In an interval t, the sink changes its direction $n = t/\tau$ times and its final position is the sum of n random phasors of magnitude d. The x and y coordinates of this position are given by: $X_n = \sum_{i=1}^{n} d\cos\theta_i$ and $Y_n = \sum_{i=1}^{n} d\sin\theta_i$. As n becomes large, the use of central limit theorem implies that the distribution of X_n and Y_n become Gaussian with mean 0 and variance $nd^2/2$. Transforming the joint distribution of X_n and Y_n to polar coordinates then gives the pdf of $R(t)$. In the case where n may not be large enough to satisfy the central limit theorem, in [25] it is shown that the pdf of $R(t)$ is given by

$$p(r) = \frac{2re^{\frac{-r^2}{\alpha}}}{\alpha}\left[1+\frac{3}{8n}\left(\frac{E[d^4]}{E[d^2]^2}-2\right)\left(\frac{r^4}{2\alpha^2}-\frac{2r^2}{\alpha}+1\right)\right] \tag{9}$$

where $\alpha = nE[d^2]$. Note that the term outside the square braces is the Rayleigh distribution and thus for $p(r)$ to be within ϵ of this distribution

$$\left|\frac{3}{8n}\left(\frac{E[d^4]}{E[d^2]^2}-2\right)\left(\frac{r^4}{2\alpha^2}-\frac{2r^2}{\alpha}+1\right)\right| \leq \epsilon \tag{10}$$

For our random walk model where the step size is fixed, $E[d^4] = d^4$ and $E[d^2] = d^2$. Using these in Eqn. (10):

$$\frac{3}{8n}\left(\frac{r^4}{2n^2d^4}-\frac{2r^2}{nd^2}+1\right) \leq \epsilon \tag{11}$$

which can be simplified to

$$3r^4 \leq 16\epsilon n^3 d^4 - 6n^2d^4 + 12nr^2d^2. \tag{12}$$

When n is large, we have $n^3 \gg n^2 \gg n$ and we can approximate the equation above be neglecting the lower order terms. Then we have

$$n \geq \sqrt[3]{\frac{3r^4}{16\epsilon d^4}}. \tag{13}$$

■

Thus for large enough n the PDF of the distance traveled by the sink is Rayleigh and is given by

$$\text{Prob}\{R(t) \leq r\} = 1 - e^{-\frac{\tau r^2}{td^2}}, \qquad 0 \leq r \leq \infty \tag{14}$$

Now consider an arbitrary sensor with transmission radius R_t in range of the sink with the location of the sink being equally likely anywhere within the circle describing the sensor's transmission region. Then from the results in [26], the probability β that the sink is still within the range of the sensor after time t is given by

$$\beta = \sum_{k=1}^{\infty} \frac{(a)_k z^k}{(b)_k k!} \tag{15}$$

where $a = 1/2$, $b = 2$, $z = -4\tau R_t^2/(td^2)$ and $(a)_k$ and $(b)_k$ are Pochhammer symbols: $(a)_k = a(a+1)(a+2)\cdots(a+k-1)$ and $(a)_k = b(b+1)(b+2)\cdots(b+k-1)$. For the desired sink miss rate β after the end of t units of time, Eqn. (15) can then be solved to obtain the required update frequency $1/t$.

D. Robustness of SIMPLE

As described above, initially each sensor is installed with the shortest path gradient to the sink. Nodes far away from the sink will not be able to get their paths updated very often as the sink moves. Consider a node i whose shortest path gradient is out of date, and still leads to the sink's old position, A, as shown in Figure 6. At time t node i sends a packet to the sink, which has now moved to position B and this new location information is available to nodes in the shaded cloud in Figure 6. Since node i's gradient still leads to position A, the information will follow the solid line starting from node i to position A (note that the information's actual track is not necessarily straight). Upon the packet's arrival at A, it will follow the sink's movement track since gradients of nodes along the track have been updated in the direction of the sink's movement. The packet will be forwarded along the track until it reaches the cloud region. Nodes in this region fall within the advertisement scope and are updated with sink's movement in the manner as described before. Thus, from that point, the information delivery path will be rectified on the fly instead of having to follow the sink's movement track.

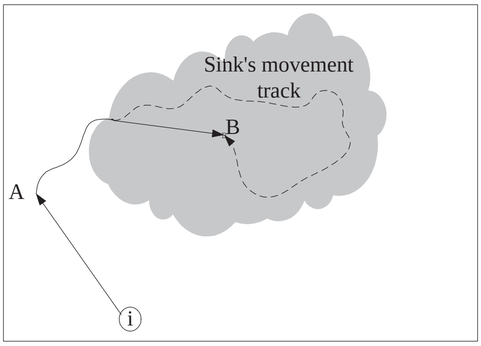

Fig. 6. The sink's track is indicated by the dashed, curved line. Nodes inside the cloud get their gradients updated based on sink's up-to-date position. Information from node i is first delivered to A, the sink's dated position. It is then forwarded along the sink's track(the solid, curved line) until it reaches the cloud, where the path starts to get rectified on the fly.

Besides registering the neighbor on the shortest path, node i can keep record of other neighbors that relay the swarm agents to it through the suboptimal paths to the sink. Thus, if the neighbor on the shortest path becomes unavailable, the report message can still be sent out through the backup paths to the sink. This greatly enhances SIMPLE's resilience against node instability due to node/link failures or node sleeping. The simulation result from Section VI shows the resilience of SIMPLE with multiple path against node failure.

E. SIMPLE's Overhead Message Complexity

The algorithms presented in [7] are mainly designed for the static sink scenario and data could be exchanged between any arbitrary pair of nodes. Residual energy of nodes within the same "local broadcast area" is synchronized. From an individual node's point of view, this makes the algorithm overhead's message complexity $O(n)$, where $n \geq 1$ is the number of nodes within the "local broadcast area". In addition, it is hard to adapt these algorithms for the mobile sink scenario. The TTDD protocol in [6] 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. Based on the description above, SIMPLE has an overhead message complexity $O(1)$, which is induced by the swarm agent from the sink. Quantitative comparison results of TTDD and SIMPLE are presented in next section.

F. Miscellaneous Issues

1) Heterogeneity of Node Batteries: The node batteries are allowed to be heterogeneous as far as their capacities and energy consumption rates are concerned. In SIMPLE a sensor's battery capacity is normalized in terms of how many 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 the channel is available but no transmission is detected from node j, node i could assume node j is dead and retransmit the message via another downstream neighbor.

3) Static Sink: When the sink stays static, the swarm agent can be advertised after receiving certain amount of data from a given source. Thus, the chosen paths will not drift far away from the optimal ones. Unaffected nodes in the network can simply suppress the advertisement.

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 sleeping and awakening.

VI. Simulation Results

In this section we present the simulation results that are used to evaluate and verify SIMPLE's performance and effect of various environmental factors. We compare SIMPLE with both minimum hop count routing algorithm ([10], [11]) and the TTDD algorithm designed for mobile sink scenarios in [6]. A critical observation showing how SIMPLE achieves a tradeoff between energy consumption and network performance is presented. We also evaluate the effect of various control and environment factors on SIMPLE's performance. Finally the network average energy depletion rates are evaluated in multiple sinks scenario to show that SIMPLE does scale to the number of sinks. SIMPLE's resilience against node failures is also verified.

Given that the receiving energy consumption is 1 unit, we normalize the transmission energy according to the 1st order radio model described in [19]. The energy consumption for transmission ($E_{T_x}(k,d)$) and reception ($E_{R_x}(k,d)$)costs for a k-bit message transmitted over a distance d is shown below:

$$\begin{aligned} E_{T_x}(k,d) &= E_{elec} \times k + \epsilon_{amp} \times k \times d^2 \\ E_{R_x}(k,d) &= E_{elec} \times k \end{aligned} \tag{16}$$

$E_{elec} = 50nJ/bit$ is the energy dissipated to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ is for the transmitter amplifier.

"Random Walk" is taken as sinks' mobility model in the simulations.

A. Comparison with Minimum Hop Count Routing Algorithm

We first compare SIMPLE with the Minimum Hop Count Routing algorithm, denoted as "min-hop algorithm" in the figures. Min-hop algorithm always uses the path with the minimum hop count from the source to the sink. Since we assume fixed transmission range, min-hop algorithm actually minimizes the energy consumption for each data report to the sink.

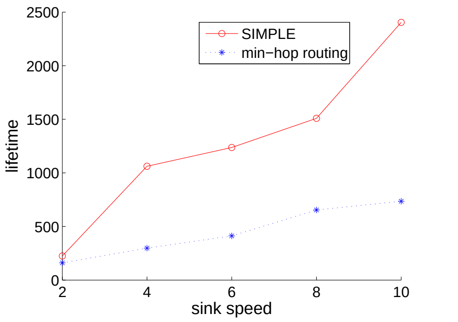

Fig. 7. The energy will not deplete faster in multi-sink scenario compared to single-sink scenario

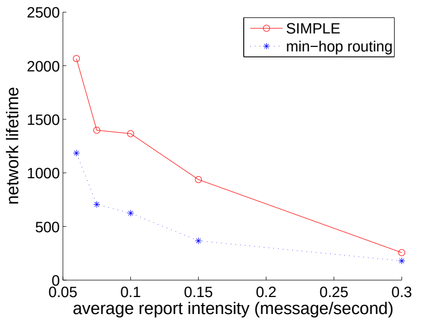

Fig. 8. The average hop count between nodes and the sink decreases as the number of sinks increases

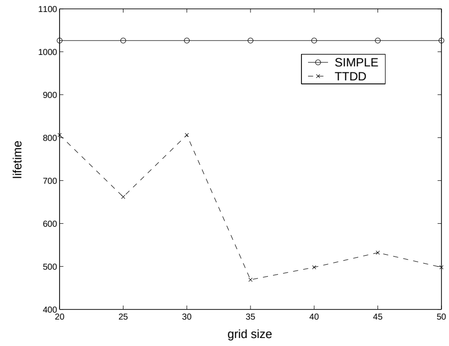

Fig. 9. Lifetime comparison between SIMPLE and TTDD with different grid sizes

In this set of experiments, 200 nodes are uniformly distributed in a $100 \times 100m^2$ network area. Node's transmission range is 25m. The swarm agent is 64 bytes and the report message is 512 bytes. Each node has 500 units of initial energy.

1) Network Lifetime vs. Sink Speeds: In Figure 7, data reports are generated at each node with the Poisson arrival rate $\lambda = 0.3$ messages per second, and the sink's speed is varied from 2m/s to 10m/s (note that this speed is relatively fast considering nodes' 25m transmission range). Both SIMPLE and min-hop's network lifetime is observed. As the sink's moving speed increases, SIMPLE introduces more energy consumption with more frequent path updates. But its lifetime increases because the network energy depletion rate is more balanced across the network. Sink's mobility actually helps avoid draining energy of the same set of nodes. This is also verified by the min-hop algorithm's result. The reason that SIMPLE outperforms min-hop algorithm with a significant margin is that SIMPLE not only tries to minimize each data report's energy consumption, but also takes energy balance into consideration.

2) Network Lifetime vs. Report Intensity: Figure 8 presents SIMPLE and min-hop's lifetime for different report intensities. When the reporting intensity is moderate, SIMPLE has a much better performance compared to min-hop algorithm because SIMPLE takes energy balance into consideration when updating the shortest path. Each update tries to avoid employing the node with the least energy to prolong the network lifetime, while min-hop algorithm sticks to the least hop path even when the path residual energy becomes very low. When the reporting rate is very high (at each node the report has a Poisson arrival rate of 0.3 message per second), SIMPLE only has a slightly longer lifetime because between two updates of the shortest path the reporting events are so frequent that a large percent of energy is already consumed. This suggests that when the sink is static or moving slowly and the data report is intense, SIMPLE can accordingly adjust to a higher path update frequency than the analytical value presented in Section V-C.

B. Comparison with TTDD

In this section we compare SIMPLE with TTDD and show that a critical drawback of TTDD is its energy unawareness, which degrades its performance even when we ignore its higher protocol overhead. We consider a grid network of 100 nodes located in a $100 \times 100m^2$ region. The area is divided into 10×10 grids and all nodes are located at cross points of grids. Nodes' transmission range is 11m. We ignore TTDD's overhead induced by each source to construct and maintain the grid.

The swarm agent is 64 byte and the average data length is 512 bytes. The sink node's movement is assumed to be a 2-dimensional random walk with speed 10m/s. Data report is generated at each node by a Poisson process with rate $\lambda = 0.05$ messages per second. In these comparisons SIMPLE does not suppress any swarm agents. Later we will show SIMPLE's performance with and without suppression of the swarm agent. Each node has an initial energy of 250 units.

Figure 9 shows the lifetime of SIMPLE compared with that of near-ideal TTDD with different grid sizes. Aside from the major drawback of energy unawareness that we mentioned earlier, another issue in TTDD is that each source has to repeatedly construct and maintain its own grid, which spans the whole network and is a major obstacle for TTDD to perform efficiently. It should be noted that although we ignore this overhead, it actually grows unboundedly with increase of the number of source nodes and decrease of grid size, which makes TTDD unscalable.

C. Effect of the Environmental Factors

In this section we observe the effect of various environmental factors on SIMPLE's energy consumption or lifetime. The faster the sink moves, the more swarm agents it generates. Thus the sink's speed and the swarm agent length directly affect SIMPLE's energy consumption due to its overhead. In addition, we are also interested in the node density's effect to SIMPLE's lifetime. Finally we present SIMPLE's performance model with different suppression ratios. 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 is 500 units.

1) Effect of the Sink's Speed and Length of the Swarm Agent: We consider 200 nodes to be deployed in the network. Figure 10 shows the effect of the sink's speed and the ratio of data length and swarm agent length on the energy consumption induced by the swarm agent. 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 Figure 7. 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%. This suggests that data

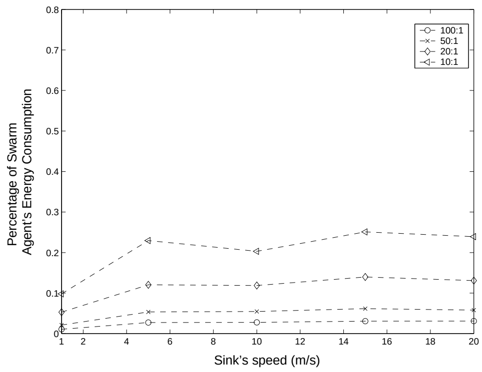

Fig. 10. Effect of Sink's Speed and swarm agent's length relative to the data report's length.

aggregation at the source area could be employed to decrease SIMPLE's overhead.

2) Effect of Node Density: In Figure 11 we plot the swarm agent's energy consumption as a function of the node density for data and swarm agent length ratios of 10:1 and 50:1. When swarm agent lengths are small compared to the data, the energy consumption can drop to as low as 5% when the node density reaches $\lambda = 0.08$ nodes/m^2. When node density increases, the burden of relaying data becomes less on each node. According to our constrained model in 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.

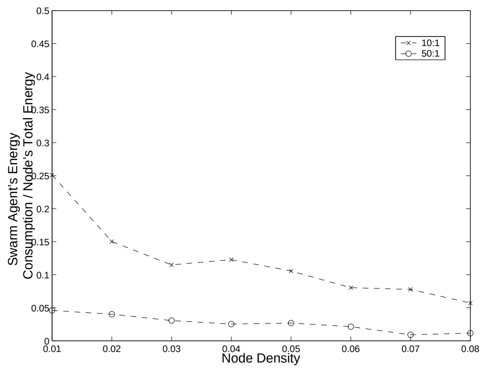

Fig. 11. Node Density vs. Swarm Agent's Energy Consumption

3) Suppression Degree versus Network Lifetime: In this section we use a similar simulation configuration as the previous one, except for the node density and the average data length. Here 200 nodes are uniformly distributed in the area. We increase the ratio of swarm agent length and average data length to $2:5$. These changes are to enable a more effective observation of the tradeoff between protocol overhead and network residual energy balance.

Swarm agents are advertised in the network to dynamically update the shortest paths to the mobile sink, and represent the overhead in SIMPLE. In order to save energy, a swarm agent suppression technique is introduced in IV-B. Figure 12 presents the sensor network's lifetime (y axis) under different swarm agent suppression degrees (x axis), which is represented by the percentage of node energy consumed by swarm agents. The curve is drawn by 10th degree exponential curve fitting, with error bounded within 15%. Note that y axis only indicates relative lifetime.

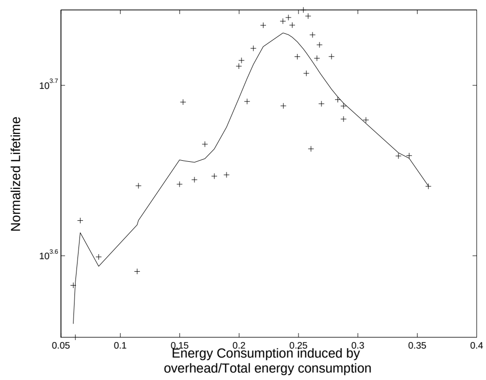

Fig. 12. Lifetime achieved with different suppression ratios.

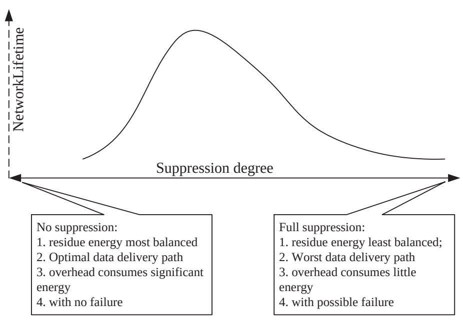

Fig. 13. The system performance's tradeoff model: protocol overhead's energy consumption and balance of network residual energy.

Figures 12 and 13 show that the network survives the longest with neither zero nor full suppression of the swarm agent. Figure 13 is a general evaluation model regarding the tradeoff between the network's residual energy distribution and the overhead of any possible protocol designed for mobile sink scenarios. Going left to right, the two extremes in the figure are elaborated as follows:

- **Dynamic:** Protocols in this category try to continuously update the whole network with sink's latest location. The shortest path chosen will thus be optimal and the network's residual energy is optimally balanced, which prolongs the network's lifetime. Information delivery failure is fully avoided. Although [22] is not energy aware, it does belong to this category as does SIMPLE without suppression. Note that although protocols in this category can find the energy-wise optimal path, the significant overhead induced thereby actually decreases the network's lifetime.
- **Static:** Paths to the sink are updated as infrequently as possible. Most nodes are unaware of sink's movement and information is delivered through stale and usually suboptimal 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 soon, which actually contributes to shortening of the network lifetime.

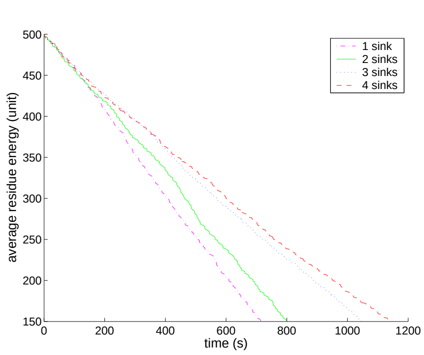

Fig. 14. The energy will not deplete faster in multi-sink scenario compared to single-sink scenario

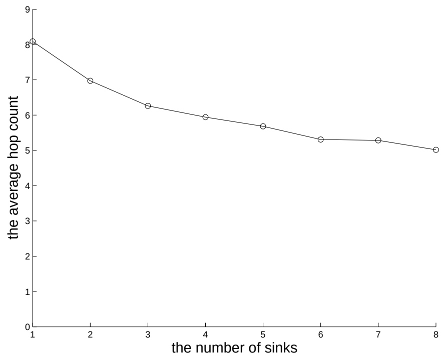

Fig. 15. The average hop count decreases as the number of sinks increases

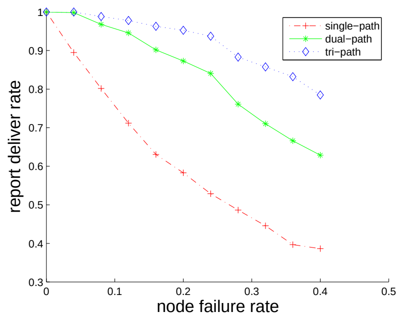

Fig. 16. Multiple paths improve the protocol's resilience against node failures

D. Multi-sink Scenarios

In this section we investigate the energy depletion issue in multi-sink scenarios. When multiple sink 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 into their belonged subarea. 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. Sinks are moving in the network with a speed of 10m/s. Reporting traffic is generated at each node with the Poisson arrival rate $\lambda = 0.05$. Each reporting message is 512 bytes and the swarm agent is 64 bytes. Figure 14 shows that with the same reporting intensity, as the number of sinks increases from 1 to 4, the time it takes the average node residual energy to drop from 500 to 150 becomes longer instead of shorter. The reason is that although multiple sinks introduce more energy consumption due to more swarm agents, it also helps decrease the average hop distance between nodes and their corresponding sinks, as shown in Figure 15. Energy saving due to lower hop counts from reporting nodes to the sinks actually outweigh energy consumption due to more swarm agents.

E. Protocol Resilience Against Node Failures

We verify SIMPLE's resilience against node failures and the results are shown in Figure 16. Initially, 200 nodes, with their initial energy 500 units and the transmission range 25m, are uniformly distributed in a $100 \times 100m^2$ area. One mobile sink is present in the network with a moving speed of 10m/s. Each node has a failure probability, as indicated by the x-axis in Figure 16. Report events are generated at each node with a Poisson arrival rate of 0.05 messages per second. In addition to the shortest path, nodes can keep record of suboptimal 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 the mobile sink scenario: SIMPLE. It is designed based on techniques of swarm intelligence, energy-wise shortest path and a probabilistic model for dynamically updating the shortest paths. The swarm intelligence approach maximizes individual node's lifetime since it greatly simplifies sensors' operations, keeping requirements in line with a typical sensor's low computational capabilities, restricted storage and limited energy. The protocol tries to maximize the network's lifetime by dynamically choosing the energy efficient paths and balancing the residual energy at each node. SIMPLE scales to multiple sinks and is robust against node failures. We analytically verify the correctness and scalability of SIMPLE. Extensive simulations are also reported to demonstrate its robustness and superior performance as compared to existing protocols.

References

[1] E. Bonabeau, F. Henaux, S. Guerin, D. Snyers, P. Kuntz, and G. Theraulaz, *Routing in telecommunications network with "smart" ant-like agents*, Proc. Of Intelligent Agents for Telecommunications Applications '98.

[2] G. Di Caro and M. Dorigo, *AntNet: A Mobile Agents Approach to Adaptive Routing in Communication Network*, 9th Dutch Conf. on Artificial Intelligence (NAIC '97) November 1997.

[3] G. Di Caro and M. Dorigo, *Ant colonies fro adaptive routing in packet-switched communications networks*, Proc. Of PPSN V-5th Intl. Conf. on Parallel Problem Solving from Nature, September 1998

[4] J. Chang and L. Tassiulas, *Energy Conserving Routing in Wireless Ad-hoc Networks*, Proc. Of Infocom 2000, March 2002.

[5] V. Srinivasan, C. Chiasserini, P. Nuggehalli, and R. Rao, *Optimal Rate Allocation and Traffic Splits for Energy Efficient Routing in Ad Hoc Networks*, Proc. Of Infocom 2002.

[6] 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 Mobicom'02 September 2002.

[7] K. Kar, M. Kodialam, T.V. Lakshman, and L. Tassiulas, *Routing for Network Capacity Maximiztion in Energy-constrained Ad-hoc Networks*, Proc. Of Infocom 2003, April 2003.

[8] Q. Li, J. Aslam, and D. Rus, *On-line power-aware routing in wireless ad-hoc networks*, Proc. Of Mobicom'01, July 2001.

[9] A. Sankar and Z. Liu, *Maximum Lifetime Routing in Wireless Ad-hoc Networks*, Proc. Of Infocom 2004, March 2004.

[10] 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

[11] C. E. Perkins and E. M. Royer, *Ad hoc on-demand distance vector routing*, Proc. Of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90-100.

[12] S. Singh, M. Wu, and C. S. Raghavendra, *Power-aware routing in mobile ad-hoc networks*, Proc. Of Mobicom'98, October 1998.

[13] V. Rodoplu and T. H. Meng, *Minimum energy mobile wireless networks*, In Proceedings of the IEEE Int. Conf. on Communications, June 1998, vol. 3, pp. 1633-1639.
[14] R. Ramanathan and R. Hain, *Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment,* Proc. Of Infocom 2000.
[15] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, *Distributed Topology Control for Wireless Multihop Ad-hoc Networks,* Proc. Of Infocom 2001.
[16] J. Zhu, C. Qiao and X. Wang, *A comprehensive Minimum Energy Routing Scheme for Wireless Ad hoc Network*, Proc. Of Infocom 2004, March 2004.
[17] S. Shakkottai, *Asymptotics of Query Strategies over a Sensor Network,* Proc. Of Infocom 2004
[18] K. Lai and M. Baker, *Measuring Link Bandwidths Using a Deterministic Model of Packet Delay,* Proc. Of the ACM SIGCOMM 2000 Conference, August 2000.
[19] W.Heinzelman, A. Chandrakasan and H. Balakrishnam, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks,* Proc. Of Hawaii Conf. System Sciences, Jan. 2000
[20] *http://dsp.jpl.nasa.gov/members/payman/swarm/*
[21] 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.
[22] 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.
[23] F. Ye, S. Lu and L. Zhang, "GRAdient Broadcast: A Robust, Long-lived Large Sensor Network," Technical Report, University of California, Los Angeles, CA, 2001.
[24] 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.
[25] P. Beckmann, *Probability in Communication Engineering,* Harcourt, Brace and World Inc., New York, 1967.
[26] A. McDonald and T. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 8, pp. 1466-1487, August 1999.