

# Amorphous Placement and Retrieval of Sensory Data in Sparse Mobile Ad-Hoc Networks

HANY MORCOS      AZER BESTAVROS      IBRAHIM MATTA  
hmorcos@cs.bu.edu    best@cs.bu.edu    matta@cs.bu.edu

Computer Science Department  
Boston University

Technical Report BUCS-TR-2006-005

April 14, 2006

**Abstract**—Personal communication devices are increasingly being equipped with sensors that are able to passively collect information from their surroundings – information that could be stored in fairly small local caches. We envision a system in which users of such devices use their collective sensing, storage, and communication resources to query the state of (possibly remote) neighborhoods. The goal of such a system is to achieve the highest query success ratio using the least communication overhead (power). We show that the use of Data Centric Storage (DCS), or *directed placement*, is a viable approach for achieving this goal, but only when the underlying network is well connected. Alternatively, we propose, *amorphous placement*, in which sensory samples are cached locally and *informed exchanges* of cached samples is used to diffuse the sensory data throughout the whole network. In handling queries, the local cache is searched first for potential answers. If unsuccessful, the query is forwarded to one or more direct neighbors for answers. This technique leverages node mobility and caching capabilities to avoid the multi-hop communication overhead of directed placement. Using a simplified mobility model, we provide analytical lower and upper bounds on the ability of amorphous placement to achieve uniform field coverage in one and two dimensions. We show that combining informed shuffling of cached samples upon an encounter between two nodes, with the querying of direct neighbors could lead to significant performance improvements. For instance, under realistic mobility models, our simulation experiments show that amorphous placement achieves 10% to 40% better query answering ratio at a 25% to 35% savings in consumed power over directed placement.

## I. INTRODUCTION

**Motivation:** Advances in the manufacturing and miniaturization of sensors of various modalities are making it possible for such sensors to be embedded in mobile devices such as cellular phones, handheld computers, and automotive navigational systems. Sensors are even expected to be embedded in future wearable computers to monitor vital signs [1], [21]. While sensors may be embedded into mobile devices in support of applications that are local to the devices in which they are embedded, the communication and storage capabilities of these devices open up the possibility of using a set of (possibly large number of) mobile devices in a given field as constituting a distributed repository of spatiotemporal sensory data. Prior research in which sensor networks are viewed as

“databases” that may be used to store sensory data and to answer queries thereupon was mostly concerned with issues of efficient representation [6], aggregation [5], [23], and routing [12].

An interesting motivating application comes from the military field. Military research labs are designing and producing new wearable units to enable soldiers to cover wider areas of the battlefield while maintaining a high level of efficiency in communication and maneuvering [2], [3]. Each soldier is equipped with a backpack that has multiple sensors (*e.g.*, motion sensors, acoustic sensors, infrared light emitting diodes, and pan/tilt cameras) with the goal of programming these units remotely to perform certain tracking or monitoring tasks, while the soldiers roam the field. Samples gathered by these sensors may prove very useful to other soldiers in the field. Imagine a scenario in which a soldier is interested in moving to a certain location; information about any movement detected in that location during the last five minutes would be a crucial piece of knowledge. Moreover, providing this information to that soldier *on time* will be even more important. In case a group of soldiers are temporarily out of communication with their base station, the task of communicating this information to the interested soldier becomes a distributed challenge to which the system has to respond.

Embedding sensing abilities into mobile devices allows us to remove a number of key assumptions often made in prior research. First, prior studies have mostly been concerned with fields in which sensors are densely deployed so as to leverage the limited-range radio communication abilities of the sensors in setting up the “network”. In particular, in these studies, there is an inherent assumption that once disconnected from the network, a sensor node is not useful since there is no way to access or query the sensory data collected by that node by any other part of the network. Clearly, this is not the case if sensor nodes are mobile, since it is possible for such nodes to become reconnected by virtue of their (or other nodes’) mobility. This also implies that in the presence of mobility, good spatiotemporal coverage of a field is possible with a sparse population of sensors. Second, prior studies have mostly been concerned with resource-impooverished sensor nodes, with an added emphasis on preserving battery power and on the efficient use of very limited memory as exemplified in [9]. The integration of sensors into mobile devices allows us

<sup>§</sup> This work was supported in part by a number of NSF awards, including CNS Cybertrust Award #0524477, CNS NeTS Award #0520166, CNS ITR Award #0205294, and EIA RI Award #0202067.

to loosen these constraints a bit since these devices are likely to possess richer computing and storage resources, and may allow for power restoration once a node moves to a location in which the battery could be recharged. In particular, the availability of additional memory makes it possible for such memory to be used to ferry data between two disconnected sensor network neighborhoods by virtue of the mobility of the node in which this memory resides from one neighborhood to the other.

The above arguments suggest that the mobility of sensor nodes could be leveraged to improve (possibly significantly) the query performance of such networks, especially when the sensor field is sparsely populated. In this paper, we demonstrate this by considering a protocol that allows a set of mobile nodes in a sensor field to cooperate so as to provide better storage and query performance. Our protocol adopts a fresh perspective in which the local memory of a sensor node is used to cache a spatiotemporal “sample” of the sensor field. By a spatiotemporal sample, we mean that each entry in a cache corresponds to a sensory data with spatial and temporal coordinates that identify the physical/geographical location in which the sample was taken as well as the time in which the sample was taken. Clearly, the cache memory in a sensor node must be managed in a manner that maximizes its utility. For example, upon the generation of a fresh sensory reading, a sensor node must decide whether to store this new reading in its cache memory or not, and if it decides to do so, which existing cache entry this new reading should replace.

**Directed versus Amorphous Placement:** An important question here is related to the placement and storage of spatiotemporal samples – specifically, should each sensor node be assigned a spatiotemporal subspace for which it is responsible, or should the responsibility of the entire spatiotemporal space be shared across all nodes? In the remainder of this paper, we use the term “directed placement” to refer to the former of these approaches and the term “amorphous placement” to the latter of these approaches.

Directed placement has been proposed and evaluated in a number of studies in peer-to-peer networks [33], [30] as well as in sensor networks [4], [27], [31]. When used in conjunction with sensor databases, directed placement has been termed as Data Centric Storage (DCS) [32]. In our context, using directed placement, once a sample is obtained by a mobile node, storage of this sample requires its transport to the node (or locale of nodes) responsible for the spatiotemporal subspace to which this sample belongs. Such transport could be carried using any number of multi-hop ad-hoc routing techniques [15], [16], [26].

Directed placement simplifies query processing significantly, since a well-defined “home” for a spatiotemporal subspace makes it straightforward to route future queries over that space. Moreover, partitioning the spatiotemporal space over the various nodes in the system allows the system to collectively store a larger number of samples, since ideally, a reading is stored only in a single entry at its “home” node or set of nodes. For these reasons, in an ad-hoc or sensor networks, directed placement is expected to work well, but only when the connectivity of the underlying network is rich enough to support the transport/routing of samples/queries

from one point in the network to another. In sparse, often partitioned networks such as those we envision in this paper, directed placement may not perform well.

Instead, in this paper, we propose the use of amorphous placement whereby a reading is not associated with a locale to which it must move, but rather such a reading could be stored in any one of the (and even replicated across multiple) mobile caches in the system. Clearly, this approach requires that some form of constrained flooding be used to locate samples belonging to a spatiotemporal subspace of interest. In dense, well-connected networks, flooding techniques are viewed as wasteful of bandwidth and power, but as we show in this paper, in sparse, often partitioned networks, the combination of associative (and possibly replicated) placement of samples coupled with flooding-based routing of queries tend to improve the query recall rates and/or the quality of the recalled data.

Directed and amorphous placement approaches represent two extremes. In this paper, we also consider a hybrid approach which combines benefits from both approaches. In particular, we consider a placement strategy that allows nodes to store spatiotemporal samples that belong to other nodes (and which cannot be immediately delivered to such nodes) in their caches. Such entries are held “in transit” as lower-priority entries, and are opportunistically forwarded towards their destination if and when that destination comes within communication range.

**Paper Overview and Outline:** We used both analytical and synthetic tools to evaluate various directed and amorphous placement and retrieval strategies, and identify the conditions that are most adequate for the operation of each. Our study shows that under most operational assumptions, amorphous placement achieves the highest query success ratio with the lowest communication overhead.

In Section III we give bounds on the performance of amorphous placement strategies. We do so by first characterizing the ability of a single node to uniformly sample the field, under a simple mobility model that is amenable to analysis. We, then, relax this assumption and use synthetic tools to evaluate the potential from cooperation between multiple mobile nodes using more realistic mobility models. Insights gained from this analysis are used to motivate the use of amorphous placement strategies – namely Amorphous Placement and Retrieval (APR) and Speculative Placement and Retrieval (SPR) – which we propose in Section IV. Performance evaluation based on event-driven simulations are presented in Section V. We discuss related research efforts in Section VI, and we conclude with a summary and an overview of on-going work in Section VII.

## II. QUERY AND DATA MODELS

**Query Model:** We assume that nodes are equally interested in the whole field, *i.e.*, nodes get queries whose targets are selected uniformly over the whole field. One particularly important parameter of queries over a given location or region is the tolerable inaccuracy that the query allows in the result. We assume that queries target a specific location (or region) in the field along with some desirable *precision* ( $\ell$ ), which constrains how far the readings used to answer the query could

be from the actual location of the query target. In this paper, and unless otherwise specified, we assume that all queries have the same query precision. A query is counted as a success if the inquiring node finds a sensor reading sampled at a location not further from the query target by the query precision.

**Data Freshness:** In this paper, in order to be useful, the returned answer to any query should not be “stale”, *i.e.*, it should be collected relatively recently. This guarantees that each query answer is an accurate representation of the current state of the field. To that end, we assume that a well-defined mechanism exists via which nodes are able to discard obsolete samples, or otherwise assign a marginal utility to keeping one sample versus another – *i.e.*, an aging mechanism. Clearly, choosing the right parameters for aging depends on the stationarity (or time-scale of change) of the target phenomenon sampled by the sensors. In our model, we assume that any collected sample stays fresh, and so a returned answer is always fresh. This assumption is reasonable if the rate of query/response is much larger than the rate of change in the data.

### III. TOWARDS UNIFORM COVERAGE

Our goal in this section is to characterize the ability of a group of mobile sensors with limited caches to collect samples that cover a given space *most uniformly*, under the assumption that the mobility pattern of these nodes is not under our control. Queries are not considered explicitly in this uniform coverage model. Rather, we argue that covering the field most uniformly maximizes the probability of successfully answering most of the queries using only local caches (assuming a uniform model for selecting query targets).<sup>1</sup>

We start with analysis of how much coverage an individual mobile node is able to achieve on its own by periodically sampling the field. Next, we consider the case in which a set of nodes cooperate by exchanging their collected samples. We present performance bounds obtained through analysis and simulations, which motivate the main “building blocks” of the placement and retrieval approaches we consider later in the paper.

#### A. Bounds on Single Node Coverage

We present an analytical model that allows us to quantify the relationship between various characteristics of a node (*e.g.*, local cache size), its mobility parameters (*e.g.*, speed), and the length of time it takes such a node to approach an optimal coverage of the space. To be tractable, our model will make simplifying assumptions about such aspects as mobility models and cache management strategies. These assumptions are then relaxed in an event-driven simulator that we validate by showing that our simulations produce behaviors that match our analytical predictions (under identical conditions). In the next section, we use this simulator to study the various storage and retrieval techniques we discussed in Section IV, under much more realistic settings.

<sup>1</sup>The results in this section are easily extended to cases in which queries target the field according to some distribution known *a priori* (*i.e.*, not uniform).

Our analysis of field coverage by a single node is done in one dimension and then in two dimensions. We consider only discrete fields, where the state of the node is expressed in terms of its distance from the starting position. Without loss of generality, we assume that the node starts at location 0 in the field, and performs an  $m$ -step random walk in a transitionally invariant discrete field. A transitionally invariant field in one dimension is a ring of length (perimeter)  $L$ ; in two dimensions, it is an  $L$  by  $L$  torus. Assuming such a field type for the analysis spare our model handling halting and reflecting states in the field which will complicate the analysis ([36], Section 4.5). At each step of the random walk, the node samples the location in which it ends up. We assume that the node has a cache of size  $c$  and that the set of readings in that cache is  $S$ . Our analysis aims to derive bounds on the probability of this node successfully answering queries about random locations in the field. A query is successfully answered if the node has a reading that is at most  $\ell$  away from the location targeted by the query.

In an  $m$ -step random walk, the walker (node) would collect a total of  $m$  samples. Assuming  $c \ll m$ , the walker would have to perform cache replacement at some point in time. The following is a description of the cache replacement algorithm. Let's denote the set of samples kept at a node by  $S$  such that  $|S| \leq c$ . At any time the node has  $c$  samples in its cache, and one new sample ( $x$ ) is collected from the field, the node selects the two samples that are closest to each other (*i.e.*, whose inter-distance is the minimum over all samples) from the set  $S \cup \{x\}$ . We call these two points  $a$  and  $b$ . For each of these two points, the node calculates the minimum distance between this point and the set  $S/\{a, b\}$ . The point whose distance is minimum gets evicted from the cache. This cache management algorithm is the application of the algorithm given in [28] as a 0.5-approximation of  $c$ -sample of the set  $S \cup \{x\}$ . The proof is given in [34].

**One-Dimensional Analysis:** Since at steady state, we know that the node has a non-zero probability of sampling all points in the field, the optimum spacing between points, in this case, would be:  $S_{opt} = \frac{L}{c}$ . Applying the aforementioned cache replacement algorithm gives a 0.5-approximation of this optimal result. So the node's cache should not have any two samples that are closer than  $\frac{L}{2c}$ . Let's call this distance  $S_{min}$ . In such a case, coverage corresponds to the success rate of answering random queries over the field, where coverage is defined as how much of the field can the node answer queries about using its cache. We recall that coverage is a function of the query precision. The following Theorem gives us the asymptotic probability of success of the node in answering queries about random locations in the field.

*Theorem 1:* A single node whose mobility model is a random walk in a transitionally invariant one dimensional field (*i.e.*, ring) will have the following asymptotic bounds on its performance of answering random queries in the field with precision  $\ell$ :

1. if  $\ell < \frac{S_{min}}{2}$ , then success probability =  $\frac{c}{L} \times (2\ell + 1) \leq \frac{1}{2}$
2. if  $\frac{S_{opt}}{2} > \ell > \frac{S_{min}}{2}$ , then success probability =  $\frac{c}{L} \times (2\ell + 1)$  which is lower bounded by 0.5 and upper bounded by 1.

3. if  $\ell \geq \frac{S_{opt}}{2}$ , then success probability = 1

*Proof:* The proof is included in the Appendix. ■

The only point left in this analysis is to determine when the system reaches steady state. [36] (Equation (4.191)) gives the relation between the number of uniquely visited sites in the field and the number of steps needed to visit these sites. In 1-D, this equation can be written as:

$$U_L(t_x) = L(1 - [1 + \frac{1}{L\phi_L}]^{-(t_x+2)}) \quad (1)$$

where  $U_L(t_x)$  is the number of uniquely visited sites in a ring of perimeter  $L$  after time  $t_x$ .  $\Phi_L$  is given by (see Equation 4.169 in [36]):

$$\Phi_L = \frac{1}{L} \sum_{s=1}^{L-1} \frac{1}{1 - \hat{p}(\frac{2\pi s}{L})} \quad (2)$$

where  $\hat{p}(\frac{2\pi s}{L})$  is the Fourier series of the single-step displacement function,  $p(j)$ , and is given by

$$\hat{p}(\frac{2\pi s}{L}) \equiv \sum_{j=0}^{L-1} p(j) \exp(\frac{2\pi i j s}{L}) \quad (3)$$

where  $p(j)$  is the single-step displacement function. Assuming the node starts from location 0 in the field,  $p(j)$  gives the probability that in one time unit, the node moves to another location that is  $j$  units of distance away. For a periodic unidimensional field (*i.e.*, a ring) of size  $L$ , we have  $0 \leq j \leq L - 1$ .

$t_{ss}$  is the time at which the number of uniquely visited sites approaches the field size  $L$ . Hence

$$U_L(t_{ss}) \approx L \quad (4)$$

Using Equations 1 and 4, we get

$$[1 + \frac{1}{L\phi_L}]^{-(t_{ss}+2)} = \epsilon \quad (5)$$

for an arbitrarily small  $\epsilon$ . Given equation 5, we can calculate  $t_{ss}$  as:

$$t_{ss} = \frac{-\ln(\epsilon)}{\ln(1 + \frac{1}{L\phi_L})} - 2 \quad (6)$$

Once the single step displacement function of the walker is defined, we can calculate  $\hat{p}(\frac{2\pi s}{L})$ , which enables us to calculate  $\Phi_L$ , using which, with the field size  $L$ , and the desired fidelity level  $\epsilon$  we are able to calculate  $t_{ss}$ .

For the purposes of our analysis, we assume that a single step of the random walk consists of a sequence of  $d$  steps of a drunkard walk [36]. For each step of a drunkard walk, the walker flips a coin,  $p$ . If  $p < 0.5$ , the walker moves one step to the right, otherwise he/she moves one step to the left.

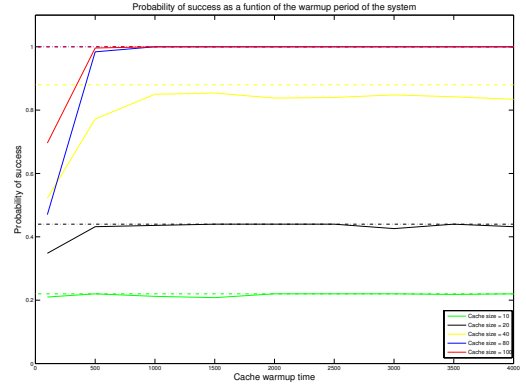


Fig. 1. Probability of success as a function of (cache warmup) time.

Let's define  $p_{dw}(d, j)$  as the probability of a drunkard walker ending up at state  $j$  after taking  $d$  steps.

**Convergence:** Figure III-A shows the upper bound on the probability of success obtained from simulation as a function of time for different cache sizes. In this simulation, the field size was taken to be  $L = 500$ , and the precision was set at  $\ell = 5$ , which is 1% of the field size. For each cache size, the upper bound on the performance calculated from the analysis is marked by the dashed line. It is clear that as the system converges to steady state, its performance approaches the upper bound calculated from the analysis.

**Effect of Speed on Convergence:** An important factor that affects convergence to the bounds we derived above is the node *speed*. More specifically, slower-moving nodes will take more time to cover the whole field, which results in longer convergence time. In our analysis above, the parameter  $d$  of the drunkard model could be used to model speed. In particular, assuming that sampling of the field is done periodically, then the number of steps in between samples (namely  $d$ ) is indicative of the node's "speed". The plots in Figure 2 show the effect of  $d$  (*i.e.*, drunkard's speed) on the performance nodes with different cache sizes. In each plot the horizontal line marks the analytically derived bound. Clearly, the walker's convergence rate to the derived bound increases with speed.

**Two-Dimensional Analysis:** The above analysis can be generalized to two dimensions. In this case the field is a torus with dimensions of  $L \times L$ , and our distance metric is the  $L_1$  (or Manhattan) distance. In this case, at each step, the drunkard walker has four choices (as opposed to two in 1-D). He/she may choose to move either up, down, right, or left with probability 0.25 for each choice. At steady state, a random walk enables the walker to sample all  $L^2$  locations in the field. We now need to find the optimum spacing between samples in the cache such that the minimum distance between any two points is maximized to get as close as possible to uniform coverage. In other words, we ask the question, given a torus of dimensions  $L \times L$ , how can we place  $c$  points on that torus, such that the minimum distance between any two points is maximized. Let's assume for now that  $c$  is a square<sup>2</sup>,

<sup>2</sup>We will relax this assumption shortly.

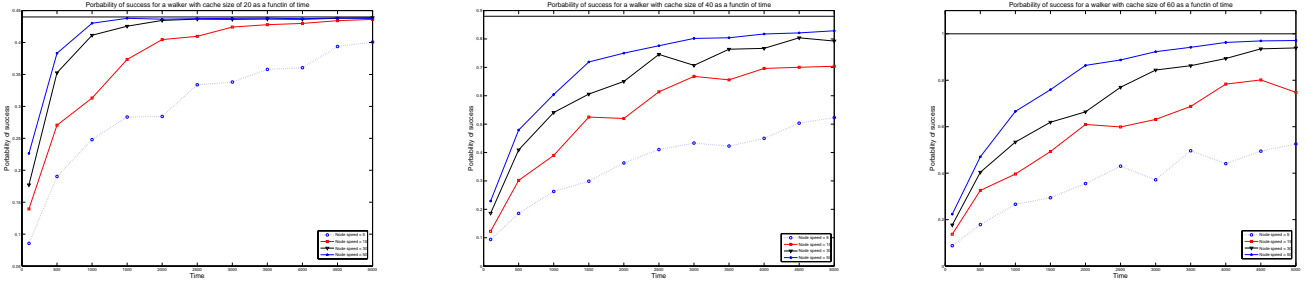


Fig. 2. Convergence to analytical bound under different speeds, when cache size = 20 (left), 60 (middle), and 60 (right).

*i.e.*,  $c = s^2$  for some integer  $s < L$  and  $s$  divides  $L$ . Then we can very easily argue that selecting  $c$  points uniformly (as in our analysis in 1-D) will maximize the minimum distance. In such a case an optimal algorithm would be one that divides the torus into  $s \times s$  squares, then places a point in each square. Selecting the corresponding points in each square will yield a minimum distance of  $L/s$ . The following set of lemmas will help us prove the bounds for 2-D random walks.

**Lemma 1:** Given an  $L \times L$  torus and a  $c$  such that  $c = s^2$ ,  $L > s$ , and  $s$  divides  $L$ , the following is true:

$$S_{opt} \geq \frac{L}{s}$$

**Lemma 2:** In a two dimensional periodic field, measuring distance using  $L_1$ , the number of neighboring points within distance  $\ell$  of any point is given by

$$R(\ell) = \sum_{i=1}^{\ell} 4i = 2\ell(\ell + 1).$$

**Lemma 3:** In a two-dimensional periodic field where distance is measured using  $L_1$ , assume we have  $G$ , an  $s \times s$  grid of points on the field, such that the minimum spacing between any two points is  $t$ . Let  $x$  be a node on the edge of this grid, then the number of neighbors of  $x$  within distance  $\ell \geq \frac{t}{2}$ , that don't belong to the grid  $G$  is lower bounded by

$$T(\ell) = \sum_{i=1}^{t/2} (2i - 1) = \left(\frac{t}{2}\right)^2$$

**Theorem 2:** A single node whose mobility model is a random walk in 2-D on an  $L \times L$  torus has the following asymptotic bounds on its ability to answer queries for random points in the field with precision  $\ell$ :

1. if  $\ell < \frac{S_{min}}{2}$  then the success probability is upper bounded by  $\frac{c}{L^2} \left[ \frac{L}{2s} \left( \frac{L}{4s} + 1 \right) + 1 \right]$
2. if  $\frac{S_{opt}}{2} > \ell > \frac{S_{min}}{2}$ , then the success probability is lower-bounded by  $\left( \frac{s-1}{4s} \right)$  and is upper-bounded by:  $\frac{c}{L^2} \left[ \frac{L}{s} \left( \frac{L}{2s} + 1 \right) + 1 \right]$
3. if  $\ell \geq \frac{S_{opt}}{2}$ , then the success probability is lower-bounded by  $\frac{s-1}{s}$

*Proof:* The proof is included in the Appendix. ■

## B. Bounds on Cooperative Coverage

We now turn our attention to what is possible when multiple nodes roam the field. Again, the problem of uniform coverage of sensory readings over a given space could be framed as follows: Given a set of readings over a multidimensional space – which without loss of generality we take to be a 2-D field – select a subset of the sample points that cover that space most uniformly.

**Optimal Selection and Approximations Thereof:** An obvious solution to the uniform coverage problem is to superpose a grid on the field, such that the number of cells in the grid equals the number of sample points we need to select. From each cell, we select a representative sample point. This solution has two major drawbacks: (1) It assumes knowledge of the field dimensions, and (2) it is not online, in the sense that a priori knowledge of the sample size is important to the correct functionality of the algorithm. Changing the sample size during the operation of the algorithm may potentially change our selection of points so far as we may need to change the dimensions of the grid cells.

To overcome these drawbacks, we formulate the problem as an optimization problem. We define a mathematical function that expresses optimal uniform coverage of space, then we select the set of points that maximizes this function. The function we propose is to maximize the minimum distance between the selected samples. More specifically, given a set of points  $G$  ( $|G| = n$ ), we propose to define  $S_{opt}$  ( $|S_{opt}| = k$  such that  $k < n$ ) as the set of samples that maximizes  $\phi(S, x)$ , the minimum distance between its selected samples. Intuitively, selecting two samples that are very close to each other is against optimal coverage, therefore maximizing the distance between every two samples in the selected set should result in better coverage of the space.

More formally, based on the formulation in [34], let the minimum distance function  $\phi : 2^G \times G \rightarrow \mathbb{R}^+$  such that for each  $S \subset G$  and  $x \in G$  we have:  $\phi(S, x) = \min_{q \in S} \|x - q\|$ , where  $\|x - q\|$  is a metric of distance between two points  $x$  and  $q$ . The subset of samples  $S \subset G$  and  $|S| = k$ , that maximize  $\phi$  over all possible subsets of points in  $G$  of size  $k$  provides the optimum space coverage.

To design an algorithm to find  $S_{opt}$  given  $G$  using this metric, we have to consider all potential subsets of size  $k$ , *i.e.*,  $\binom{n}{k}$ , which is expensive to calculate. However, Ravi

[28] provides an algorithm to find a 0.5-approximation for this problem, and Teng [34] provides a proof thereof. This algorithm runs in  $O(k^2n)$  and is guaranteed to find the 0.5-approximation of  $S_{opt}$ , *i.e.*, the minimum distance between any pair of samples in the solution of this algorithm is at least half the minimum distance between any two samples in the optimal solution. This algorithm is online in the sense that it does not need to know  $k$  a priori. The algorithm starts with an empty set  $S$ , then it chooses a random point and adds it to the set  $S$ , then calculates the minimum distance between all unselected points and the set  $S$  (the initial random point in this case), and greedily selects the point that maximizes this distance to add it to  $S$ , then repeats until the size of  $S$  is  $k$ .

**0.5-Approximation with Unbounded Storage:** In the above formulation, in order to select the set of  $k$  sample points for a 0.5-approximation, it is necessary to have potentially unlimited temporary storage in which to keep all readings before the off-line selection algorithm is applied. We refer to this approach as the “0.5-Approximation with Unbounded Storage”, which we use to obtain an upper-bound on achievable performance.

**0.5-Approximation with Bounded Storage:** Another variant of the above formulation which we describe later in this section restricts the amount of storage available for the optimal selection approximation to be equal to the total size of the caches in all nodes in the system. We refer to this variant as the “0.5-Approximation with Bounded Storage”, which we use to obtain a tighter upper-bound on achievable performance.

### C. Distributed Cooperative Approaches

The solutions we get by formulating the uniform coverage problem as an optimization problem (whether using limited or unlimited storage) is not realistic, since in a real setting, there is no guarantee that we will be able to select the sample points identified by the above approximation (since we have no control of node mobility) – not to mention that the centralized nature of the solution. Nevertheless, the solution of the above optimization problem gives us a good baseline against which we may be able to judge the performance of more “realistic” approaches. To that end, in this section, we consider a set of heuristics approaches that are not constrained by these unrealistic assumptions.

Let  $\Theta = \{\theta_1, \theta_2, \dots, \theta_w\}$  be the set of mobile sensors in the field such that  $|\Theta| = w$ . Each sensor samples the space randomly while moving within the target field. Let  $s_i$  denote the set of samples gathered by sensor  $i$  such that  $|s_i| = \beta$ , for  $1 \leq i \leq w$ . Note that  $\bigcup_{1 \leq i \leq w} s_i = G$ . Let  $\kappa$  be the cache size of each sensor, and  $c_i$  be the set of samples held at the cache of sensor  $i$ . Note that since  $\kappa \ll \beta$ ,  $c_i \subset s_i$ .

Our problem is to design a distributed algorithm to be run on the nodes such that after some bounded time of starting the algorithm  $\hat{S}$  is close to  $S_{opt}$  where  $\hat{S} = \bigcup_{1 \leq i \leq w} c_i$ . In other words, after some bounded time of running the distributed algorithm, the total set of samples retained by all sensors provides optimal or near optimal uniform space coverage. We now consider three candidate approaches.

**Random Selection:** Without any coordination or cooperation, nodes simply sample the field and store the sampled points in their caches. Once the cache is filled, “random” cache replacement is used whereby a randomly selected entry in the cache is evicted. In the remainder of this section, we use this approach to provide a lower bound on achievable coverage. Clearly this approach is fairly straightforward to implement and requires no overhead, since it incurs no overhead for cooperation or coordination among the various nodes in the system.

**Shuffle Exchange:** Using this approach, each node attempts to keep as much of a uniform coverage of the entire field as possible. This is achieved by having each node use the optimization approach (presented in Subsection III-B) to rank the set of readings in its cache, and by allowing nodes that come in contact with one another to engage in a “shuffle exchange” of the readings in their caches. In particular, when two nodes A and B come into contact with one another, both nodes engage in a “shuffle exchange”, whereby a subset of the readings in node A’s cache are sent to node B (and vice versa). The particular subset of readings exchanged between could be selected in a number of ways – *e.g.*, by selecting a random subset of the sample points in the cache, as opposed to selecting the worst (best) valued set of sample points in the cache, where the value of a sample point is determined based on the optimization function presented in Subsection III-B.

**Targeted Exchange:** Using this approach, each node is made responsible for a distinct region of the field – *e.g.*, using a hash function on the node ID. As such, each node has an incentive to keep in its cache readings from the region to which it is assigned. In particular, when two nodes A and B come into contact with one another, readings collected by node A that are in the region assigned to B are sent to B (and vice versa). Upon receipt of any such new readings, a node (say B) uses the minimum distance optimization (described in Subsection III-B) to select the set of readings that cover its assigned region as uniformly as possible.

### D. Performance Potential and Limits

To get a feel for the promise of the “Shuffle Exchange” and “Targeted Exchange” techniques, and to assess how closely they approach the baseline bounds (however unrealistic) established by the centralized approach, we conducted a set of simulation experiments, in which we used identical mobility and sampling scenarios for the various distributed approaches (namely, “Random Selection”, “Shuffle Exchange”, and “Targeted Exchange”), comparing the results obtained under these approaches to the theoretical baseline obtained using centralized optimization.

Mobility scenarios for our experiments were generated off line using different mobility models, including the corrected version of the Random Waypoint mobility model [19], the Random Direction model [29] and the Boundless Simulation Area model [10]. Since we got similar results using these mobility models, we only report results of the corrected Random Waypoint model. In this model, each node randomly selects a destination point in the field and starts moving from its current position to the destination in a straight line with

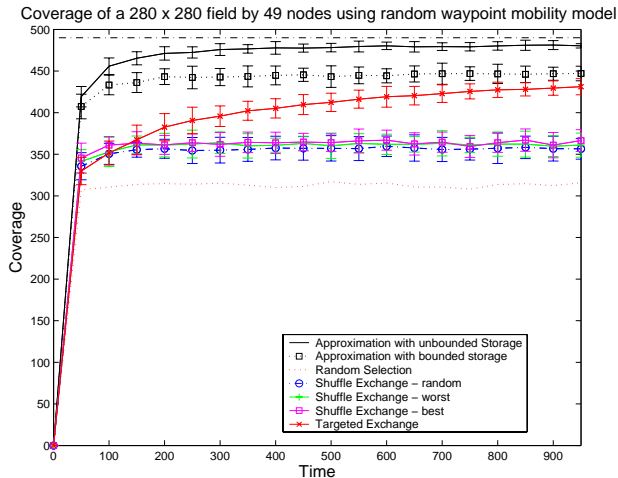


Fig. 3. Simulation results for how different heuristics perform over time and how they relate to various bounds.

speed that has the distribution function:  $f_V(y) = ky$ , where  $y \in [MinSpeed, MaxSpeed]$ ,  $k = \frac{2}{MaxSpeed^2 - MinSpeed^2}$ . In [19], Lin *et al.* have shown that assigning this distribution function to speeds guarantees the speed of all mobile nodes to be uniformly distributed within  $[MinSpeed, MaxSpeed]$  at any time during the simulation. Upon reaching the destination, the node pauses for some time and then selects a new destination and repeats this pattern. In our simulations, we set  $MinSpeed$  to zero,  $MaxSpeed$  to 20m/sec, and the pause time was set to zero.

The sampling scenarios for our experiments were generated according to a Poisson process with exponential inter-arrival time of two seconds, whereby a sample at time  $t$  constitutes the sensed value of the (static) field at the current location of the node.

We report results of simulating 49 mobile nodes moving in a field of  $140m \times 140m$ . The communication range is set to 40m. The cache size of each sensor  $\kappa$  is set to 10. The simulation runs for 5,000 seconds, with each sensor getting a new sample roughly every two seconds for an average number of samples per sensor  $\beta$  of 2500. The period after which nodes declare their presence  $\tau$  is 10 secs. The maximum number of packets nodes exchange when they are within communication range  $r$  is set to 4. In Figure 3, each point is the average of 20 simulation runs, with the 90% confidence intervals shown.

In Figure 3, the x-axis is time and the y-axis is the coverage provided by the respective technique measured by dividing the field into square cells of  $5 \times 5$  and counting the number of cells that are covered by samples in the various node caches. In this setup we have a total of 784 square cells. Since the total cache size of all the sensors =  $49 \times 10 = 490$ , the maximum number of cells that can be covered is 490. The various curves in Figure 3 show how the different techniques approach this ideal coverage over time.

The results in Figure 3 reveal some interesting observations. First, as one would expect, the performance of Random Selection is worse than all other techniques; its performance

does not improve over time. Second, we note that the 0.5-approximation algorithm with unbounded storage approach reaches optimal coverage very fast, which tells us that, in practice, the algorithm described in [28] gives a much better approximation than 0.5. Third, we note that there is some difference between the performance of the bounded and unbounded storage variants of the 0.5-approximation algorithm described in [28]. This difference quantifies the effect of limiting the total storage in the system. Fourth, we note that Shuffle Exchange (with its different variants, using random, best, and worst shuffles) achieves notably better performance over Random Selection, but is still far from the upper-bound established by the 0.5-approximation algorithm. Finally, we note that Targeted Exchange provides the best coverage across all distributed algorithms because it reduces the redundancy in samples stored in the system. Indeed, over time, it approaches the performance of the 0.5-approximation algorithm with unbounded storage. Surprisingly, it outperforms the 0.5-approximation algorithm with bounded storage. This could be explained by noting that the hashing used in conjunction with Targeted Exchange results in balancing the coverage of the various regions of the field, whereas the 0.5-approximation algorithm with bounded storage maximizes a global function over the whole field, not assuming any a priori knowledge of the field dimensions and not using field partitioning techniques.

#### IV. A SPECTRUM OF STRATEGIES

The results of the previous section suggest that opportunistically exchanging samples based on space partitioning, *i.e.*, using Targeted Exchange, achieves superior uniform space coverage compared to Shuffle Exchange. In this section we leverage this observation in a more specific setting – namely (1) a setting in which there is an explicit notion of queries, (2) a setting that supports either single-hop or multi-hop communication between nodes, and (3) a setting in which sensory readings from the field expire or become less valuable with passage of time.

##### A. Query Coverage and Precision

In the previous section, the use of space coverage as a performance metric assumes that users at the mobile nodes are equally interested in the entire field. As such, in the previous section, the notion of “queries” was implicit. In this section, we examine an explicit notion of queries, whereby queries generated at a node are defined over a specific location or region in the field and not over the entire field.

One particularly important parameter of queries over a given location or region is the tolerable inaccuracies that the query allows in the result. As discussed in the previous section, we assume that queries target a specific location (or region) in the field along with some desirable *precision* ( $\ell$ ), which constrains how far the readings used to answer the query could be from the actual location or region specified in the query. In this paper, we assume that query precision is the same for all queries submitted to the system.

## B. Single versus Multi-Hop Communication

In the distributed approaches considered in the previous section, communication between nodes was possible only when nodes came into direct contact with one another – *i.e.*, over a single hop communication protocol. In this section, we also examine the possibility that nodes are able to exchange readings and queries/responses using an underlying multi-hop communication protocol.

**Direct Placement and Retrieval (DPR):** At one end of the scale we have directed placement which relies on multi-hop communication to both insert samples in the system and query them. Using hashing techniques, data objects are hashed to certain node(s) or location in the field. Then using either ad-hoc routing or geographic routing (depending on whether we hashed data objects to nodes or locations), data items can be forwarded to the hashed node or the node closest to the hashed position. This setting resembles previous work [27] [31], and depends on hashing and explicitly forwarding both samples and queries to their hashed nodes. We call such node the *hosting* nodes. This idea, in effect, partitions the set of data objects, based on the hashing function, and stores each partition on a separate node or group of hosting nodes. When a node gets a query, it hashes the query target to get the hosting node or group of nodes to such a query. Successful handling of queries is conditioned on successfully reaching one of the hosting nodes. As a result of this design, each node gets a very narrow but detailed view of the field. Most samples kept by a node are concentrated in one small area of the field, which represents the *Responsibility Region* of the node.

**Amorphous Placement and Retrieval (APR):** At other end of the scale we have amorphous placement which makes no use of explicit multi-hop communication, but relies on node mobility to diffuse data around. In this paper, we adopt a setting in which nodes keep samples that are gathered locally, and upon encountering another nodes, use one of the shuffle exchange techniques discussed in the previous section to diffuse the readings throughout the system. Unless otherwise stated, we use the “Best Shuffle” approach as our shuffle exchange technique. With this approach, nodes have a broader view of the field (since shuffling aims at giving each node a uniform sample of the entire field), which enables nodes to answer queries by querying their local cache or the cache of their direct (single-hop) neighbors – *i.e.*, without having to forward queries over multi-hop routes, or without having to delay answering queries until encountering a node that hosts the region targeted by the query.

**Speculative Placement and Retrieval (SPR):** As we observed in the previous section, the targeted exchange of sample points between nodes is desirable, as it allows for readings in a given region to migrate (thanks to the mobility of nodes taking these readings) towards a “home” node (or set of nodes), which become the natural targets for queries over that region. This gives rise to our third approach for placement and retrieval, which “speculatively” routes queries over a region to the node(s) assigned to that region, in anticipation that the targeted exchange of readings between nodes has resulted in such readings reaching their assigned “home”. Thus, using SPR, we use targeted exchange as a mechanism

to ferry samples to their “home” node(s), which is determined using hashing (as in DPR). In particular, when meeting a neighbor, a node checks if any of the samples in its cache hashes to this neighbor. If any such sample is found, it is forwarded to this neighbor. Otherwise, nothing is exchanged. Under SPR, queries are speculatively carried over a multi-hop communication infrastructure to their “home” node for answers.

## V. PERFORMANCE EVALUATION

In this section, we present simulation results that provide a comparative evaluation of the various approaches we have identified for placement and retrieval of sensory data in a mobile ad-hoc network.

**Simulation Model and Baseline Parameters:** We use the simulator described in Section III-D subject to the following parametrization. We set the field size to be  $L \times L$ , where  $L = 1400$  meters. We assume that a total of 100 nodes are roaming this field, each with a cache that holds up to 49 readings – *i.e.*, node caches hold a very small fraction (0.0025%) of all the possible readings in the field. Unless stated otherwise, we use a precision  $\ell = 140$  meters – *i.e.*, a reading is suitable as an answer to a query if it is within 140 meters of the target of the query (10% of  $L$ ). The mobility model we employ is the corrected Random Waypoint mobility model[19] described earlier in Section III-D. We assume that both field sampling and query arrivals are Poisson processes with rates of 2 seconds and 10 seconds, respectively.

In order to evaluate techniques that require multi-hop routing (namely, DPR), we implemented multi-hop routing based on Dijkstra’s shortest path algorithm. Namely, when a node needs to send a packet to another node that is not within communication range, we use the shortest path routing algorithm to figure out the shortest path between the two nodes, if any. Since Dijkstra’s shortest path algorithm requires global knowledge – knowledge that any distributed mobile ad-hoc routing protocol would be lacking – any implementation of DPR will be inferior to that shown here in terms of query success as well as communication overhead. We note that for the purposes of this paper, this is prudent since our aim is to show that amorphous approaches (such as APR and SPR) outperform DPR, especially in sparse networks.

**Performance Metrics:** The performance metrics we use to evaluate the different algorithms are: the query success ratio and the energy consumption. Query success ratio is the ratio between the number of queries that could be successfully handled and the total number of queries to the system during the simulation time. Energy consumption is the total consumed energy in sending, receiving and forwarding packets. Our energy model is based on the model presented in [11].

In the remainder of this section, we study the effect of the (1) cache size, (2) query precision, and (3) communication range on the above metrics. All reported results represent the average of 20 independent runs, with the 90<sup>th</sup> percentile confidence intervals shown.

### A. Effect of Communication Range

**Impact on Success Ratio:** Figure 4 (left) shows the query success ratio as a function of the communication range. Our intuition is that the performance of DPR and SPR is conditioned on the ability of the querying node to successfully reach the “home” node of the query target, *i.e.*, that the underlying network is very well connected. We varied the communication range from 10m to 320m to quantify how sensitive each protocol is to the connectivity of the underlying network. Figure 4 (left) shows that our intuition is indeed correct; DPR and SPR are very sensitive to communication range (and consequently the connectivity of the network), while APR shows higher resilience. For small communication ranges, the network is loosely connected or even disconnected. The performance of all algorithms is poor, since most of the queries that are successfully answered are answered from the local cache. As soon as we increase the communication range, the performance of APR improves significantly compared to that of DPR and SPR. Indeed, not until the communication range reaches 200m, do we see a real improvement in their performance. Notice that at first DPR’s performance is the worst, but that eventually (*i.e.*, when the network becomes well connected) it goes on to surpass both APR and SPR, with SPR reaching a plateau with performance inferior to that of DPR and APR.

The difference in performance between APR and DPR is intuitive, since using DPR, the success of a single query requires *multiple* links to exist (to deliver both samples and queries to their hosting nodes). This requires that the communication range be increased to high-enough values to connect the whole network, and only then do we see noticeable improvement in performance. As for APR, query success is conditioned on reaching more direct neighbors (to answer queries that could not be locally answered). As soon as we increase the communication range, nodes reach more neighbors and the success ratio increases noticeably.

An interesting point to note is that the performance of SPR is always inferior to that of DPR. The reason is that, unlike DPR, SPR does not explicitly forward field samples to their hosting nodes, but depends on node mobility to deliver these samples. While increasing the communication range increases the probability of meeting *some* node, it only very slightly increases the probability of meeting *the right* node. This is why SPR reaches its plateau earlier.

To verify this intuition we plot the ratio of self-answered queries to the total number of queries asked in Figure 4 (middle). By comparing Figures 4 (left) and 4 (middle), it is easy to verify that at smaller communication ranges, most of the successful queries are locally answered. As we increase the communication range, the three protocols exhibit different behavior.

APR answers even more queries locally. This is due to having larger communication range and being able to exchange samples with more neighbors, hence, improving each node’s local view of the whole field. An interesting point to notice is that, for relatively large communication ranges, APR answers more than 50% of the queries locally.

DPR answers less queries locally. This is due to the fact that the network connectivity is improved, hence field samples can be successfully forwarded to their hosting nodes. Therefore the contents of each local cache reflects a better view of a *small* area of the field (*i.e.*, the region assigned to that node), and with high probability queries arriving at any node will not target be for the region assigned to that node, and thus will require forwarding to distant hosting nodes.

For SPR, the number of queries answered locally decreases with communication range. As we commented above, this behavior hints to the slight increase in the probability of meeting *the right* node as a result of increasing the communication range.

**Impact on Energy Consumption:** Figure 4 (right) shows the energy consumption as a function of communication range for DPR, APR, and SPR. It is clear that as we increase the communication range beyond a certain limit, the communication energy increases considerably since the relation between power and distance  $d$  becomes  $O(d^4)$  as opposed to  $O(d^2)$  for smaller ranges. Figure 4 (middle) explains why the energy consumption of APR is much less than that of DPR and SPR.

**Summary of findings:** The above experiments confirm that if the underlying network is not very well connected, APR is much more efficient in terms of query success ratio and consumed energy. On the other hand, if the underlying network is very well connected then DPR will deliver superior performance to that of APR but at higher communication cost. Hence APR provides the best query-energy performance tradeoff. Since the different protocols deliver acceptable performance with communication range of 160m, in the remainder of this section, we use this value in all remaining experiments.

### B. Effect of Cache Size

**Impact on Success Ratio:** Figure 5 (left) shows the query success ratio for the three algorithms as a function of the cache size. Cache sizes range from 5 to 99 data objects. The figure shows that for very small cache sizes, DPR performs better than APR and SPR. As we increase the size of the cache, the performance of APR improves steadily, while the performance of DPR improves only slightly. The performance of SPR is upper-bounded by that of DPR with a noticeable persistent gap.

For both DPR and SPR, more cache per node means better coverage of the same region assigned to that node, and hence coverage of the same area is improved. The performance of DPR and SPR hinges on successful communication between the inquiring node and the host of the query target, which is *not* a function of the cache size. Hence the performance of DPR is almost constant. The performance of SPR improves as more queries can be answered locally because a node is able to keep samples for other nodes locally until they encounter each other.

For APR, increasing the cache size improves the local view of each node in the field. Therefore, increasing cache size increases the probability that nodes can answer queries locally and the probability that neighboring nodes will have an answer to a query whose answer is not locally available.

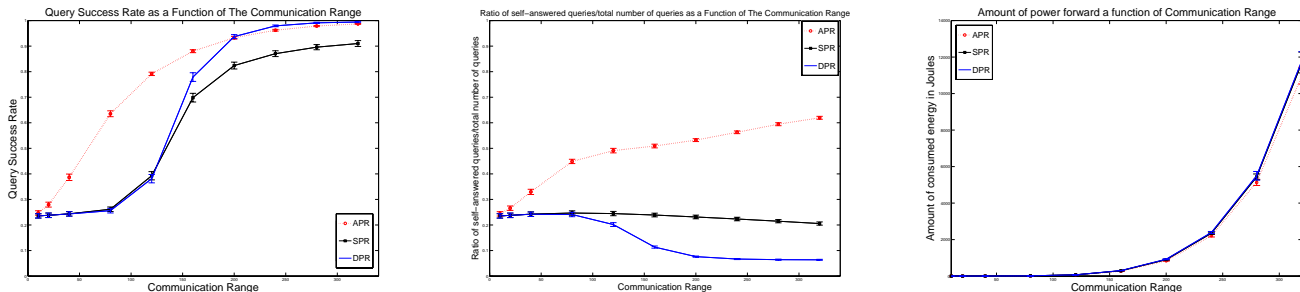


Fig. 4. Effect of communication range on query success ratio (left), ratio of locally answerable queries (middle), and consumed power (right).

This is clearly confirmed by the results shown in Figure 5 (left).

To verify this explanation, Figure 5 (middle) shows the ratio of queries answered locally to the total number of queries. The number of locally-answered queries using APR increases considerably as we increase the cache size causing the performance of APR to eventually surpass that of DPR. For both DPR and SPR, as the cache size is increased, each node has more space than it needs to keep samples from the region to which it is assigned. This enables nodes to keep samples from other regions (acquired through sampling by the node or exchanges with other nodes). This, in turn, results in more queries being answered locally.

**Impact on Energy Consumption:** Figure 5 (right) shows the consumed energy to deliver the performance given in Figure 5 (left). Power consumed by all three protocols is almost constant with slight decrease as cache sizes are increased. The reason can be clearly attributed to the higher likelihood for queries to be answered locally. Note that in the case of APR, more queries are locally answered, but more samples are exchanged between neighboring nodes, hence the total consumed energy remains fairly constant (not affected by cache size). Apart from the slight decrease, the average consumed energy for APR is about 25% to 35% less than that consumed by SPR and DPR, respectively. This confirms our intuition that explicitly forwarding field samples is a costly process in terms of energy that can be eliminated by leveraging caching and mobility.

**Summary of Findings:** The above set of experiments tell us that using APR with very small caches results in fairly poor performance, but that increasing caches even slightly pays off handsomely in terms of performance, but that this improvement reaches a plateau (saturation) fairly quickly. Increasing caches beyond that saturation point has a diminishing return on performance. It also tells us if the available cache is really small, it is worth using DPR and paying the extra communication overhead to get decent performance. If we have a little more cache, we can save up to 35% of the communication power by using a very simple protocol like APR.

### C. Effect of Query Precision

**Impact on Success Ratio:** So far we have not addressed the *quality* of the query answer returned by each protocol.

A tentative measure of quality is how spatially close/far the returned answer is to each query. To quantify this we measure performance as a function of query precision  $\ell$ . Figure 6 (left) illustrates this tradeoff by showing the query success ratio as a function of the query precision, which we vary between  $\ell = 10$  and  $\ell = 240$  meters. For higher precision (*i.e.*, smaller  $\ell$ ), both DPR and SPR perform better than APR. As we relax precision constraint, then APR's performance improves and eventually surpasses that of DPR and SPR.

The explanation of this result is straightforward. APR's premise is to allow a set of nodes within communication range from one another to get a local view (*i.e.*, coverage) of the whole field. Since nodes' caches are limited, the quality of this view is also limited, which means poor responses to queries that require high accuracy. As we relax this constraint, the samples held by APR qualify as answers to the queries and performance improves. As for DPR and SPR, the core factor of their performance is the successful routing from the inquiring nodes to the hosting nodes, which is not a function of query precision. As evident in Figure 6 (middle), as we relax the precision constraint, more local samples qualify as answers to queries without consulting the remote caches of hosting nodes. This is most pronounced for APR compared to DPR and SPR.

**Impact on Energy Consumption:** Figure 6 (right) shows the energy consumption of all three protocols. As was noticed before, APR consumes from 25% to 35% less energy than SPR and DPR, respectively. The energy consumption of DPR and SPR drops faster than that of APR as a result of relaxing the precision constraint.

**Summary of Findings:** The above set of experiments tell us that for queries with tight precision requirements, DPR and SPR outperform APR, but that this advantage disappears quite fast as we relax our precision constraints, making the use of APR more justified as it provides the best query-energy performance tradeoff.

## VI. RELATED WORK

Data storage and dissemination in sensor networks have been an active field of research for quite some time.

**Flooding-Based Approaches:** In stationary sensor networks, techniques that depend on Data Centric Routing (DCR) have been suggested for long running queries. In these techniques, sensory data is locally stored by sensors. An interested party

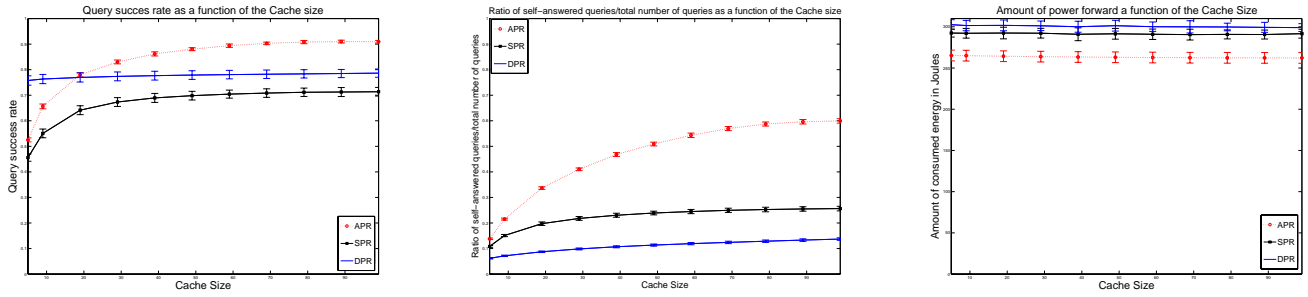


Fig. 5. Effect of cache size on query success ratio (left), ratio of locally answerable queries (middle), and consumed power (right).

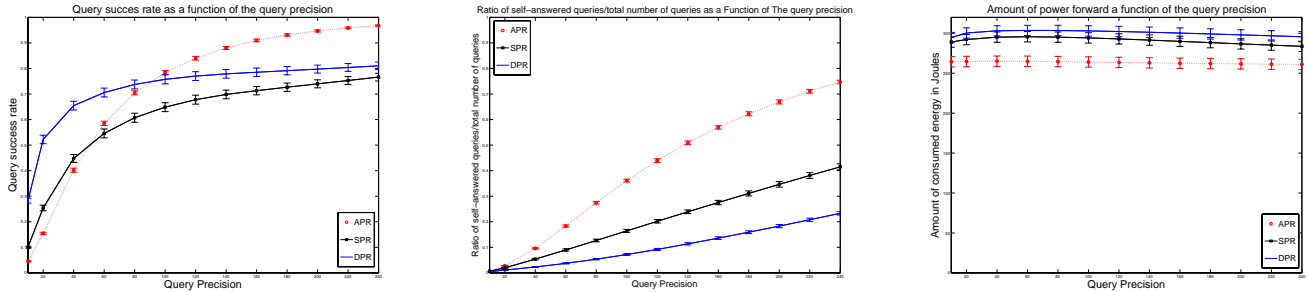


Fig. 6. Effect of query precision on query success ratio (left), ratio of locally answerable queries (middle), and consumed power (right).

then floods the network with their queries in order to reach the sensors with the data of interest. Query answers are then sent from the hosting sensors to the inquirer over paths that have been identified as “good” in the flooding phase of the protocol. Multiple measures have been suggested for the “goodness” of paths, *e.g.*, low packet-loss-probability path, low-power path, and path with fewer hops. On the way to the inquirer, data from multiple sources may be aggregated to minimize communication overhead. Pioneered by Intanagonwiwat *et al.* [12], Directed Diffusion is an example of DCR techniques. In Directed Diffusion, sensors keep their samples locally. An inquirer broadcasts some *interest* (*i.e.*, query) to the rest of the network. Initially, the inquirer requests a lower data-reporting rate, then it *reinforces* high-quality paths to get data at higher rates. Lower quality paths are also kept alive as backup for higher quality paths. TinyDB [24] and Cougar [5] are two other examples of DCR, in which the sensory data is aggregated on the way back from the sensors to the inquirer. The aggregation paths are built up during the query flooding phase of the protocol. DCR systems target long running queries in relatively immobile networks. These assumptions allow DCR systems to validate the large communication overhead in the query dissemination phase since the resulting setup will be used repeatedly in the query answering phase.

**Hashing-Based Approaches:** Data Centric Storage (DCS) [32] along with Geographic Hash Tables (GHT) [27] have been proposed as an alternate solution to avoid the huge communication overhead associated with the query flooding phase of DCR. Here, sensory data is hashed to a geographic location in the field. Using GPSR [16], this data is then forwarded to the closest node to the hash for storage. Queries to similar data are hashed using the same technique and are

forwarded to the same node where the answer is supposed to reside. DCS is not primarily intended for system with high mobility; as mobility will change the hashing of data items to nodes in the field, which requires a large communication overhead to restore the correct state of the system. In an effort to alleviate this problem, Seada *et al.* [31] relax the hashing function from mapping data to a fixed *point* in the field, to mapping it to a fixed *region* in the field. Data replication within the same region is used to minimize the communication overhead incurred when a node leaves the region. In this paper, we used an “ideal” version of DCS – namely Directed Storage and Retrieval (DPR) – as a representative of DCS approaches. We say ideal, because we assumed that routing of data to region for storage and routing of queries and responses are done using global shortest-path information, which is not possible in a real setting.

**Dealing with Inquirer Mobility:** Some research efforts [37], [18], [20], have concentrated on sensor networks with mobile sinks, *i.e.*, the sensory infrastructure is stationary, but the sink (*i.e.*, the inquirer) is mobile.

In [37], Ye *et al.* propose the TTDD system for mobile sinks in sensor networks. The main idea is to avoid flooding the network with location updates for the mobile sensors, and instead constructing a grid overlay in the sensor field. Sources (*i.e.*, sensors which detect some stimulus of interest) pro-actively forward their measurements to sensors nodes comprising the grid. A mobile sink floods only the grid cell in which it resides, grid nodes in this cell can forward the sink’s query to other nodes on the grid that have the requested data. Hence network wide flooding of location updates of the mobile sinks is avoided.

Kim *et al.* [18] propose SEAD, an asynchronous dissemination protocol. In SEAD, mobile sinks are associated with the physically closest stationary sensor in the network, which acts as a representative of this sink. Data is reported back to this sensor over a dissemination tree that minimizes energy consumption in the network. The reporting tree is a minimum-cost weighted Steiner tree. If the sink moves away from its stationary representative, the dissemination tree is not changed until the distance between the sink and the representative exceeds a number of hops.

In [20], Lu *et al.* assume that the scope of the query is associated with the current sink location. They propose that the system estimate the mobility pattern of the sink and uses this estimation to predict the future mobility model of the sink. Sensor nodes that are co-located with a future location of the sink (called *collector nodes*) start forming query trees over the queried area and rooted at these nodes. Data from the queried area is collected at the collector nodes and delivered to the sink when it passes through this node. In doing so, the authors assume that queries have temporal constraint in the form of when the data is delivered and the age of the delivered data. The proposed protocol handles these constraints.

The work presented in this paper differs from these efforts in that we assume that all nodes are mobile; sensor nodes collecting samples are mobile and the sink can be any of these nodes. Only [31] share a similar model to ours.

It is important to note that previous research efforts concluded that mobility increases the capacity of ad-hoc networks [22]. The results we presented in this paper not only support this observation, but also leverage it.

**Dissemination and Epidemic Approaches:** Multiple information dissemination protocols have been proposed for wireless, ad-hoc sensor networks. Kulik *et al.* [13] proposed the SPIN protocol as a smart alternative to flooding. SPIN disseminates information taking into consideration the resources available to nodes. It defines *meta-data*, descriptors for available information in the system. Using meta-data, nodes negotiate their content and their needs for data and then exchange data based on availability of resources. Our work differs from SPIN in that SPIN does not have the notion of queries and locating their answers which is at heart of the design of our protocol. Also since we do not use forwarding, no node is charged (in terms of energy consumption) for forwarding other nodes' data, therefore we don't take available resources into consideration.

Using gossiping or epidemic models to disseminate data has been extensively studied in the literature [7], [35], [17]. In [7] for example, the authors propose the *en-passant* model, and they use it to develop UbiQuiz, an application for ad-hoc networks. This application assumes that users of similar interest are mobile within some field. Users may need to exchange data with each other to improve the state of one or both users, but data exchange is limited to users that are one-hop away to save communication overhead. This work however does not consider the limited cache size at each node and hence does not study cache replacement algorithms that maximize the utility of each cache.

The amorphous placement approach we considered in this paper (used in both APR and SPR) resembles gossiping and epidemic techniques in that field samples are only exchanged opportunistically "on contact", *i.e.*, when nodes are within communication range from each other, and multi-hop communication is totally avoided. Our protocol is different from epidemic techniques in that the exchanged data expires as time passes by, hence there is an inherent need of a continuous exchange process to disseminate fresh samples of the field in the caches of all nodes.

**On Localization:** The various protocols we considered in this paper (namely, APR, DPR, and SPR) require nodes to know their locale to be able to attach some location information to every field sample they collect. Many protocols have been developed for localization in sensor networks [25], [14], [8]. Any of these techniques could be used with our protocols to provide the needed location information. Of course, our protocol could employ also global positioning system (GPS), especially since the personal communication devices, which we envision as hosting the sensors, are likely to have such GPS capabilities.

## VII. SUMMARY AND CONCLUSION

In this paper, we investigated the potential use of local caches in mobile, ad-hoc networks for the collective storage and querying of sensory data collected by mobile nodes. We showed that the use of a Data Centric Storage (DCS) approach that uses direct placement and retrieval (DPR) is viable, but only when the underlying network is very well connected. Alternatively, we proposed an amorphous data placement and retrieval approach (APR), in which sensory samples are cached locally and *shuffling* of cached samples is used to diffuse the sensory data throughout the whole network. APR leverages node mobility and caching capabilities to avoid the multi-hop communication overhead of DPR-like approaches. Moreover, it is more resilient to wireless losses and node failures due to the inherent replication of sampled sensory data across caches. Using analytical and synthetic tools we showed that, in most cases, APR delivers high query success ratios with lower energy consumption compared to DCS-like protocols. In particular, in many realistic settings, our proposed APR protocol delivers query success ratios that are similar to those achievable using DCS-like protocols, using 30% less energy.

Our current research is focusing on cache management techniques that allow nodes to leverage their knowledge of underlying mobility models (*e.g.*, locality characteristics), as well as the spatiotemporal characteristics of the underlying phenomena being sensed (*e.g.*, using summaries for a more effective exchange of readings). Also, we are investigating the implementation of the techniques presented in this paper (and variants thereof) in real personal communication devices to answer queries related to field conditions (*e.g.*, "what is the network coverage or signal strength in location  $x$ " or "how many different people are observed in location  $y$ ").

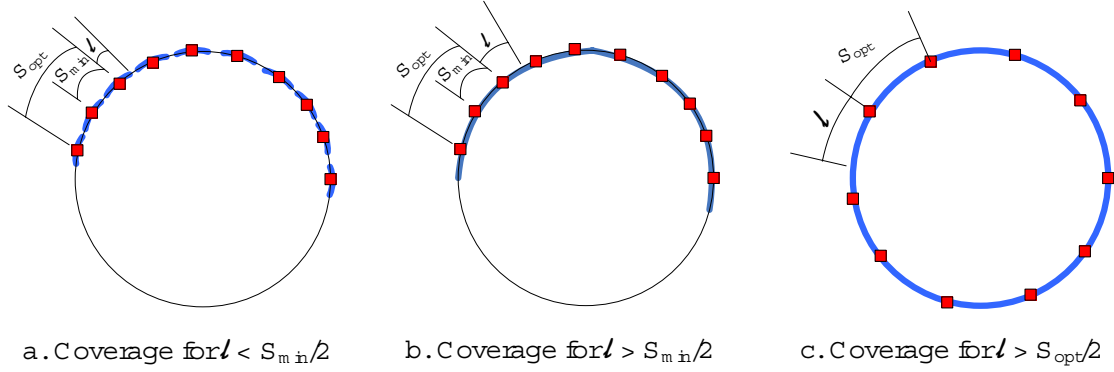


Fig. 7. Different Cases of Theorem 1

## APPENDIX

In this appendix we provide proofs for Theorems 1,2 in order.

*Proof:* Theorem 1: We have three cases depending on value of  $\ell$ .

**Case 1:** If  $\ell < \frac{S_{min}}{2}$ , then From Figure 7.a, the success probability is given by:

$$\begin{aligned} Pr[success] &= \frac{c}{L} \times (2\ell + 1) \\ &< \frac{c}{L} \times S_{min} \\ &< \frac{c}{L} \times \frac{L}{2c} = \frac{1}{2} \end{aligned} \quad (7)$$

Thus, in such a case the success probability is upper-bounded by 0.5.

**Case 2:** If  $\ell \geq \frac{S_{min}}{2}$ , then the success probability is given by:

$$\begin{aligned} Pr[success] &= \frac{c}{L} \times (2\ell + 1) \\ &> \frac{c}{L} \times S_{min} \\ &= \frac{c}{L} \times \frac{L}{2c} \\ &= \frac{1}{2} \end{aligned} \quad (8)$$

Which means that the performance in such a case is lower-bounded by 0.5. Figure 7.b illustrates this case. To obtain an upper bound in this case, we assume that minimum spacing between obtained samples will match the optimum spacing which is  $S_{opt} = \frac{L}{c}$ , then the success probability will be given by

$$\begin{aligned} Pr[success] &= \frac{c}{L} \times (2\ell + 1) \\ &< \frac{c}{L} \times S_{opt} \\ &< \frac{c}{L} \times \frac{L}{c} \\ &= 1 \end{aligned} \quad (9)$$

**Case 3:** If  $\ell \geq \frac{S_{opt}}{2}$ , then the success probability will be exactly given by

$$\begin{aligned} Pr[success] &= 2\frac{c}{L} \times \ell \\ &\geq \frac{c}{L} \times S_{opt} \\ &= \frac{c}{L} \times \frac{L}{c} = 1 \end{aligned} \quad (10)$$

Figure 7 illustrates this case. This concludes the proof of the theorem.  $\blacksquare$

*Proof:* Theorem 2: We have three cases depending on the value of  $\ell$ .

**Case 1:** If  $\ell < \frac{S_{min}}{2}$ , then since  $\ell < \frac{S_{min}}{2}$  and the minimum distance between any two node  $= S_{min}$ , then probability of success of each node = the coverage of each node times the number of nodes. From Lemma 2, the coverage of each node  $= R(\ell) = 2\ell(\ell + 1)$  then the probability of success is  $\frac{c}{L^2} \times (R(\ell) + 1)$  where the added one is for the sample position itself. The upper bounded can be derived as follows:

$$\begin{aligned} Pr[success] &= \frac{c}{L^2} \times (R(\ell) + 1) \\ &= \frac{c}{L^2} \times (2\ell(\ell + 1) + 1) \\ &< \frac{c}{L^2} \times (S_{min}(\frac{S_{min}}{2} + 1) + 1) \\ &< \frac{c}{L^2} \times (\frac{L}{2s}(\frac{L}{4s} + 1) + 1) \end{aligned} \quad (11)$$

**Case 2:** If  $\frac{S_{opt}}{2} > \ell > \frac{S_{min}}{2}$ , then we can lower-bound the probability of success by assuming that the samples are arranged in a square  $s \times s$ , and the distance between every two neighboring samples  $= S_{min}$ . In this case, the coverage of the cache equals the square of  $s \times s$  samples, plus the points within precision  $\ell$  of this square but outside it. So if we consider only neighboring points within  $\frac{S_{min}}{2}$  (as opposed to  $\ell \geq \frac{S_{min}}{2}$ ), we can lower bound the probability of success as follows:

$$Pr[success] \geq \frac{1}{L^2} \left( ((s-1)S_{min} + 1)^2 + 4(s-1) \times T(\frac{S_{min}}{2}) \right) \quad (12)$$

We can upper-bound the probability of success in this case by assuming that minimum spacing between samples equals the optimum spacing  $S_{opt}$ , in which case the upper bound can

be given by:

$$Pr[\text{success}] = \frac{c}{L^2} \times (R(\ell) + 1) \quad (13)$$

**Case 3:**  $\ell \geq \frac{S_{opt}}{2}$ , again we can lower bound the probability of success like we did in case 2.

$$\begin{aligned} Pr[\text{success}] &\geq \frac{1}{L^2} \left( ((s-1)S_{opt} + 1)^2 + 4(s-1) \times T\left(\frac{S_{opt}}{2}\right) \right) \\ &= \frac{1}{L^2} \left[ ((s-1)S_{opt} + 1)^2 + 4(s-1) \left(\frac{S_{opt}}{2}\right)^2 \right] \\ &> \frac{1}{L^2} \left[ ((s-1)S_{opt})^2 + (s-1)(S_{opt})^2 \right] \\ &> \frac{(s-1)(S_{opt})^2}{L^2} ((s-1) + 1) \\ &> \frac{s(s-1)(S_{opt})^2}{L^2} \\ &> \frac{s(s-1)}{L^2} \left(\frac{L}{s}\right)^2 \\ &> \frac{(s-1)}{s} \end{aligned} \quad (14)$$

This concludes the proof of the theorem. ■

## REFERENCES

- [1] Lifeguard, wearable wireless physiological monitor. <http://lifeguard.stanford.edu/>.
- [2] Man-Portable Networked Sensor System. <http://www.spawar.navy.mil/depts/d30/d37/d371/mpnss/mpnss.html>.
- [3] Portable Sensors Extend Warriors's Reach. [http://www.afcea.org/signal/articles/templates/SIGNAL\\_article\\_template.asp?articleid=1114&zoneid=181](http://www.afcea.org/signal/articles/templates/SIGNAL_article_template.asp?articleid=1114&zoneid=181).
- [4] M. Ali and Z. A. Uzmi. CSN: A network protocol for serving dynamic queries in large-scale wireless sensor networks. In *2nd Annual Conference on Communication Networks and Services Research (CNSR'04)*, pages 165–174, Fredericton, N.B, Canada, May 2004.
- [5] Philippe Bonnet, Johannes Gehrke, and Praveen Seshadri. Towards sensor database systems. In *MDM '01: Proceedings of the Second International Conference on Mobile Data Management*, pages 3–14, London, UK, 2001. Springer-Verlag.
- [6] R. Cristescu, B. Beferull-Lozano, M. Vetterli, D. Ganesan, and J. Acimovic. On the interaction of data representation and routing in sensor networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Pennsylvania, USA, 2005.
- [7] Hannes Frey Daniel Grgen and Christian Hutter. Information dissemination based on the en-passant communication pattern. In *Fachtagung Kommunikation in Verteilten Systemen (KiVS)*, Kaiserslautern, 2005.
- [8] L. Doherty, K. S. J. pister, and L. E. Ghaoui. "convex position estimation in wireless sensor networks". In *IEEE INFOCOM*, volume 3, pages 1655–1663, March 2001.
- [9] Deepak Ganesan, Deborah Estrin, and John Heidemann. Dimensions: why do we need a new data handling architecture for sensor networks? *SIGCOMM Comput. Commun. Rev.*, 33(1):143–148, 2003.
- [10] Z. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proc. of the IEEE Int. Conf. on Universal Personal Communications (ICUPC)*, pages 562–565, October 1997.
- [11] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.
- [12] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom: Mobile Computing and Networking*, pages 56–67, 2000.
- [13] W. Heinzelman J. Kulik and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *ACM MOBICOM*, Seattle WA, August 1999.
- [14] X. Ji and H. Zha. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In *IEEE INFOCOM*, March 2004.
- [15] David B. Johnson, David A. Maltz, and Josh Broch. Dsr: the dynamic source routing protocol for multihop wireless ad hoc networks. pages 139–172, 2001.
- [16] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.
- [17] Anne-Marie Kermarrec, Laurent Massouli, and Ayalvadi J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. Parallel Distrib. Syst.*, 14(3):248–258, 2003.
- [18] Hyung Seok Kim, Tarek F. Abdelzaher, and Wook Hyun Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 193–204, New York, NY, USA, 2003. ACM Press.
- [19] G. Lin, G. Noubir, and R. Rajaraman. Mobility models for ad-hoc network simulation. In *INFOCOM*, 2004.
- [20] Chenyang Lu, Guoliang Xing, Octav Chipara, Chien-Liang Fok, and Sangeeta Bhattacharya. A spatiotemporal query service for mobile users in sensor networks. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 381–390, Washington, DC, USA, 2005. IEEE Computer Society.
- [21] P. Lukowicz, U. Anliker, J. Ward, G. Trster, E. Hirt, , and C. Neufelt. Amon: A wearable medical computer for high risk patients. In *ISWC, the 6th International Symposium on Wearable Computers*, pages 133–134, October 2002.
- [22] Grossglauser M. and Tse D.N.C. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.
- [23] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(SI):131–146, 2002.
- [24] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. The design of an acquisitional query processor for sensor networks. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 491–502, New York, NY, USA, 2003. ACM Press.
- [25] David Moore, John Leonard, Daniela Rus, and Seth Teller. Robust distributed network localization with noisy range measurements. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61, New York, NY, USA, 2004. ACM Press.
- [26] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. 2003.
- [27] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. Ght: a geographic hash table for data-centric storage. In *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 78–87, New York, NY, USA, 2002. ACM Press.
- [28] S. S. Ravi, Daniel J. Rosenkrantz, and Giri Kumar Tayi. Facility dispersion problems: Heuristics and special cases (extended abstract). In *WADS*, pages 355–366, 1991.
- [29] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An analysis of the optimum node density for ad hoc mobile networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 857–861, Helsinki, Finland, June 2001.
- [30] Ratnasamy S., Francis P., Handley M., Karp R., and Shenker S. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM Press.
- [31] K. Seada and A. Helmy. Refereed poster: Rendezvous regions: A scalable architecture for service provisioning in large-scale mobile ad hoc networks. In *ACM SIGCOMM*, 2003.
- [32] Scott Shenker, Sylvia Ratnasamy, Brad Karp, Ramesh Govindan, and Deborah Estrin. Data-centric storage in sensornets. *SIGCOMM Comput. Commun. Rev.*, 33(1):137–142, 2003.
- [33] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM Press.
- [34] Shang-Hua Teng. Low energy and mutually distant sampling. *J. Algorithms*, 30(1):52–67, Jan 1999.
- [35] S. Voulgaris, M. Jelasity, and M. van Steen. A robust and scalable peer-to-peer gossiping protocol. In *2nd Int'l Workshop Agents and Peer-to-Peer Computing, LNCS 2872*, Springer, 2003.
- [36] G. H. Weiss. *Aspects and Applications of the Random Walk*. Elsevier Science B.V., North-Holland, 1994.
- [37] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang. A twotier data dissemination model for large-scale wireless sensor networks. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 148–159, New York, NY, USA, 2002. ACM Press.