Geoinformatica (2013) 17:567–597
DOI 10.1007/s10707-012-0170-y

Opportunistic sampling-based query processing in wireless sensor networks

Muhammad Umer · Egemen Tanin · Lars Kulik

Received: 2 November 2010 / Revised: 13 August 2012 /
Accepted: 17 September 2012 / Published online: 5 October 2012

Abstract High resolution sampling of physical phenomenon is a prime application of large scale wireless sensor networks (WSNs). With hundreds of nodes deployed over vast tracts of land, monitoring data can now be generated at unprecedented spatio-temporal scales. However, the limited battery life of individual nodes in the network mandates smart ways of collecting this data by maximizing localized processing of information at the node level. In this paper, we propose a WSN query processing method that enhances localized information processing by harnessing the two inherent aspects of WSN communication, i.e., multihop and multipath data transmission. In an active WSN where data collection queries are regularly processed, multihop and multipath routing leads to a situation where a significant proportion of nodes relay and overhear data generated by other nodes in the network. We propose that nodes opportunistically sample this data as they communicate. We model the data communication process in a WSN and show that opportunistic sampling during data communication leads to surprisingly accurate global knowledge at each node. We present an opportunistic query processing system that uses the accumulated global knowledge to limit the data collection requirements for future queries while ensuring temporal freshness of the results.

Keywords Wireless sensor networks · Query processing · Spatio-temporal modeling

M. Umer · E. Tanin · L. Kulik
National ICT Australia, Alexandria, NSW 1435, Australia

M. Umer (✉) · E. Tanin · L. Kulik
Department of Computing and Information Systems (CIS),
University of Melbourne, Victoria 3010, Australia
e-mail: mumer@csse.unimelb.edu.au

E. Tanin
e-mail: egemen@unimelb.edu.au

L. Kulik
e-mail: lkulik@unimelb.edu.au

1 Introduction

Localized information processing is a hallmark of wireless sensor network (WSN) technology. In the face of growing deployment sizes and need for higher sensing resolution, localized information processing techniques attempt to fill the gap between user expectations and technological limitations of WSN hardware. These techniques extend network life-time by energy harvesting at the node level. The primary method is to minimize the frequency and size of data transmission among distant nodes by preprocessing and data sharing among neighboring nodes. The development of localized counterparts of global information processing techniques can be traced back to the early years of WSN research. Directed Diffusion [16], one of the first data collection systems designed for WSNs, attempts to provide localized yet efficient route discovery between sources and sinks of sensed data. TinyDB [23] and Cougar [30], two early WSN database systems, provide strategies to distribute data aggregation tasks among neighboring nodes.

One of the foremost challenges in localized information processing in WSNs is the need for global knowledge at the node level. Consider, for instance, the requirement in TinyDB to maintain `MIN` and `MAX` aggregates at each node for efficient query forwarding. Similarly, localization of essential WSN operations such as event discovery, query optimization, data validation, and node utility computation require global statistical knowledge at the node level [27]. The need for global statistical knowledge to augment local information processing has motivated a number of works in WSN literature. Techniques such as multi-resolution in-network data storage improve localized range query processing by storing aggregates of multiple granularity at each node [11, 12, 27]. A common thread among current global statistics collection techniques is their specialized nature, i.e., these approaches are purpose-built for a certain type of global statistic and have to be triggered regularly to maintain accuracy of the aggregates.

In this paper, we propose *opportunistic sampling*; a novel alternative to the specialized approaches for acquiring global statistics at the node level. Opportunistic sampling is based on the key insight that an operational WSN that is actively being queried by its users already possesses a certain degree of data distributed in the network. Since WSN communication is multihop and multipath (broadcast) in nature, data from a source to a sink node is relayed and overheard by several intermediate nodes. Due to this multihop, multipath (M2) communication advantage, each user query leaves a trace of collected data among the communicating nodes. We propose that nodes opportunistically sample this data as they respond to user queries. Our experiments show that if the M2 advantage is properly harnessed, nodes accumulate surprisingly accurate global knowledge after responding to only a small number of user queries.

1.1 Motivating example

Figure 1 presents a simple example to motivate the case for opportunistic sampling based on the M2 advantage. The figure shows a WSN deployed as a grid where each node can communicate to all nodes present within one hop distance. Assume that the WSN data collection is performed using a routing tree rooted at the sink node,

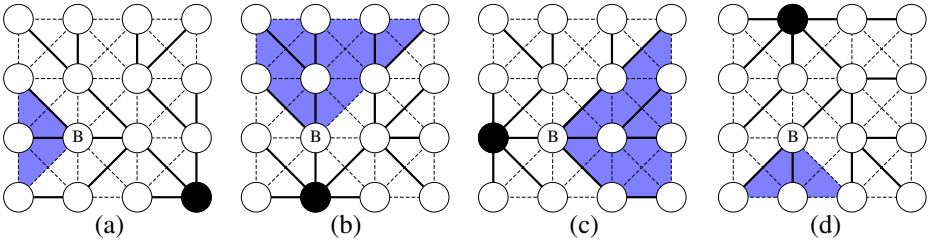


Fig. 1 A motivating example for opportunistic sampling based on the M2 advantage. The *black circle* represents the sinks in each subfigure. *Solid edges* represent the flow of data towards the sinks while *dashed edges* between two nodes show that the nodes can overhear each other

as shown. A routing tree is typically constructed by a repeated broadcast of a query by each node in the network. During the query forwarding phase, nodes synchronize their sleep, listening, and transmission periods such that each internal node wakes up in time to receive data from its child nodes. It aggregates the incoming data and transmits the resultant to its parent node [22]. In a typical routing tree constructed in a dense network, each node has more than one option for choosing its parent node and mostly it chooses its parent randomly. However, due to the synchronization of listening periods among all nodes, an internal node may still overhear the data relayed by one of its potential child nodes, even if the data is not transmitted towards it.

Figure 1a illustrates the region from which an arbitrary node (node B in the figure) in the network overhears or receives the data during data collection in response to a query issued from a sink node. We refer to this area as the M2 coverage of node B. It can be seen from Fig. 1b–d that different locations of the sink node result in a different level of M2 coverage of node B. Assume that in each case the sink node issues the following query:

```
SELECT Humidity
FROM Sensors
SAMPLE PERIOD 30 min
FOR 4 h
```

For node B, the presence of sinks at the four shown locations leads to the accumulation of humidity information from the entire network. Hence, in this example, node B can report an accurate global statistic, such as `MIN`, `MAX` or `AVG`, without issuing a single query itself.

1.2 Components of an opportunistic sampling system

The above example shows the strength of M2 coverage based opportunistic sampling in a simple setting. The example highlights that the extent of M2 coverage for a node primarily depends on its position with respect to sink nodes. For accurate aggregation however, factors such as the freshness and suitability of samples also play an important role. Figure 2 depicts our opportunistic sampling approach for in-network

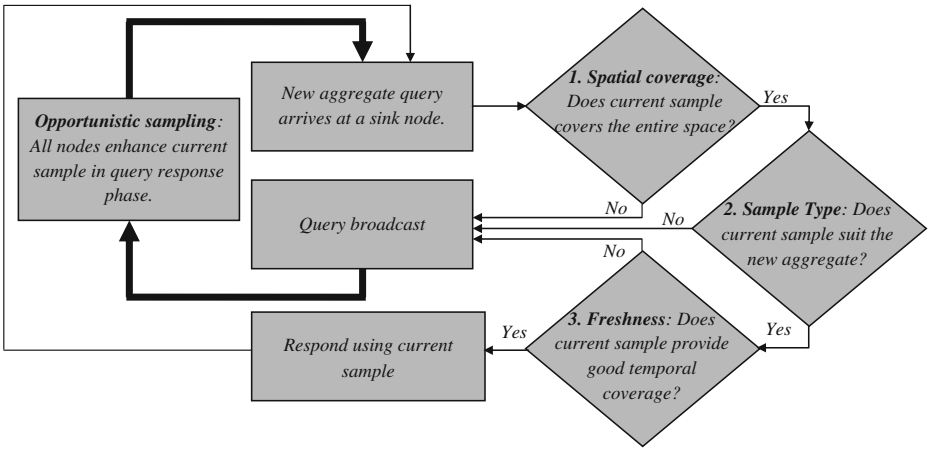


Fig. 2 Flow of the opportunistic sampling system

aggregation that integrates these key factors. The structure and main components of our approach are discussed below:

Control flow Our approach relies on a feedback loop (shown in bold lines in Fig. 2) involving new queries being broadcasted in a network and opportunistic sampling by all nodes during the query response phase. Before a node broadcasts a query in the network, it assess the validity of the data it currently holds with respect to the query. If a node determines its current sample to be suitable for the query, it evaluates the query on its current sample without further need of communication with other nodes. For certain queries, the current sample held by a given node may not suffice. In such situations, the node would proceed with regular data collection by broadcasting the query in the network. The feedback loop component of our system then ensures that this data collection helps not only in responding to the query in question but also enhances the quality of opportunistic sample held by all nodes in the network.

Spatial coverage The spatial extent of a sample held by a node is fundamental to the accuracy of query processing using an opportunistic sampling scheme. Nodes in the proposed control flow first determine the spatial coverage of the data they hold by computing their current M2 coverage. This coverage depends on the relative position of a node with respect to past sink nodes and the deployment area. For instance, consider the M2 coverage reported for a simulated WSN shown in Fig. 3. The figure presents the distribution of M2 coverage (as a fraction of deployment area) that results from the data flow generated by a sink node located at the center of the network. The figure shows that the M2 coverage of any node is dependent on its location relative to the location of the current sink and the deployment area.

Accuracy The accuracy of global statistics is affected by the spatio-temporal variation in the sensed phenomenon. Similar to the spatial coverage case, nodes measure the suitability of their sampled data in terms of its freshness and validity based on the spatio-temporal covariance in the sensed phenomenon. As a result, nodes may either proceed with using the stored sample for query evaluation or may issue a new query.

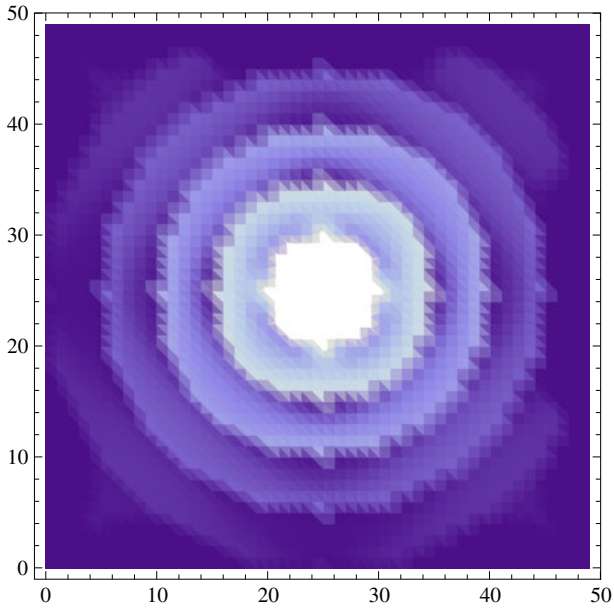


Fig. 3 A density plot of the distribution of M2 coverage due to the data flow generated by one sink present in the center of a 2500 node network. Lighter tones represent high coverage areas

If a new query is broadcasted, the feedback loop results in extending the spatio-temporal scope of the samples held by all nodes thus increasing the likelihood that future queries might not require a broadcast.

Sample type As opportunistic sensing relies on mandatory data sharing among nodes, its performance is tied to the nature of information being communicated in a network. If most user queries involve aggregates, for instance, seeking average humidity instead of all humidity values as in the motivating example, raw sensor readings are suppressed by in-network aggregation during transmission. As a result, opportunistic information sampled at each node may only suit a certain type of global statistic. To reduce this dependency, we also propose that each node piggy-backs some augmented information along with the partial query result it transmits. For example, if the global statistic `MAX` is required, each node can piggy-back augmented information in the form of a partial `MAX` aggregate based on incoming data and its own reading. Regardless of the type of query being processed, if all nodes add to and sample augmented information, a large proportion of nodes may accumulate enough local information to accurately compute the required statistic. An alternative of the piggy-backing strategy could be to exploit the partial query results computed by each node. A number of query optimization techniques (e.g., [15]) in the relational database systems literature propose to share partial query results in future queries. Such techniques can be adopted in our opportunistic sampling system.

1.3 Our contributions

Our contributions in this paper are twofold. First, we present a critical result on M2 coverage available in a network and show that only a small number of user queries lead to very high levels of M2 coverage in a network. Second, using M2 coverage for

opportunistic sampling, we present a comprehensive approach for data aggregation and show its superior performance in multiple application scenarios using real and simulated data sets.

The initial idea of our opportunistic sampling scheme was introduced in our earlier short paper [29]. In this paper, we present a formal analysis of the level of M2 coverage available in a typical WSN. We also introduce a system that exploits this feature for query processing while ensuring spatial and temporal coverage.

The rest of this paper is organized as follows. In Section 2 we present the M2 coverage model and propose an approximation method to estimate the spatial coverage available due to opportunistic sampling in a WSN. We show the strength of opportunistic sampling by comparing it to an exhaustive greedy approach in Section 3. In Section 4 we propose a method to maintain accuracy in the opportunistic sampling system. Section 5 provides the system implementation details, outlines two applications of opportunistic sampling and reports our experimental results. A further analysis of certain operational details is given in Section 6. We survey related literature in Section 7, and we conclude in Section 8 with our key findings and an outlook.

2 Spatial coverage of opportunistic sampling

In this section, we model the M2 coverage that is available to all WSN nodes due to the data flow generated by an arbitrary number of sink nodes. We present a probabilistic analysis of M2 coverage typically available in a network to establish that only a small number of uniformly distributed sink nodes lead to a high level of M2 coverage for a large part of the network.

2.1 System model

We assume a network of N nodes uniformly deployed inside a $d \times d$ square. Two nodes can communicate with each other if the distance between their locations is less than R, the transmission radius. We consider a completely decentralized architecture, i.e., there is no designated sink or base station. Upon receiving a request from a user, any node can inject a query in the network.

Query forwarding and data collection are performed using a TinyDB-based random data collection tree [22]. All nodes in the network exploit the multihop and multipath (M2) advantage during the query response phase as follows. Along with its response to a user query, each node i opportunistically piggy-backs a data item d_i where d_i comprises of node i's sensed value combined with all piggy-backed data items that node i has itself received (or overheard). For instance, in the case of collecting the global maximum of a sensed parameter, node i piggy-backs the largest sensed value known to it, i.e., by taking into consideration its own and all incoming sensed values.

2.2 Coverage model

The M2 advantage virtually extends the sensing coverage of each node much beyond its communication radius. Figure 4a shows an example where a query, $Q1$, is issued

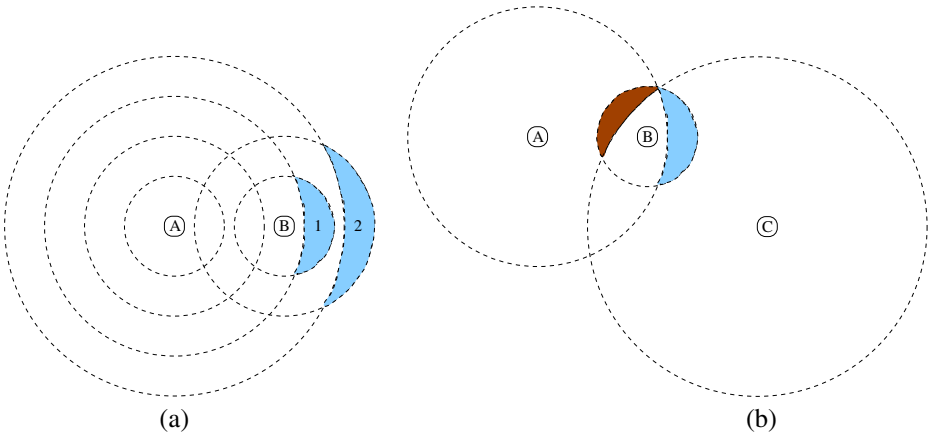


Fig. 4 Estimating the M2 coverage

by a node A. The concentric rings of increasing radii around node A depict broadcast rounds (referred to as epochs) as nodes receive and forward $Q1$.

An arbitrary node, represented as node B in this example, receives $Q1$ during the third epoch and sets its listen and transmit periods accordingly. Region 1 in Fig. 4a represents node B's first epoch M2 coverage, i.e., the area within direct transmission range of node B, from where it expects to receive (or overhear) data once the network starts responding to $Q1$. This coverage region is formed by the part of node B's transmission range that does not intersect with the query wavefront that delivers $Q1$ to node B. All nodes falling inside the *overlapping* part of node B's transmission range and the query wavefront must have received $Q1$ before node B and must have set their transmission time periods accordingly. Thus, during its listen time-period, node B cannot receive or overhear any data from these nodes.

Node B's M2 coverage increases as nodes in its first epoch M2 coverage rebroadcast $Q1$ further into the network. Analogous to the case above, the second epoch M2 coverage for node B will include the part of node B's second epoch broadcast that does not overlap with the corresponding wavefront for $Q1$. Depicted as region 2 in Fig. 4a, node B's second epoch M2 coverage is greater in area than its first epoch M2 coverage. Further re-broadcasts of $Q1$ increase node B's M2 coverage in a similar manner.

Figure 4b extends the above example by including a second query, $Q2$, issued by node C. Node B can now overhear or receive data from all nodes in the non-overlapping part of its transmission range and the query wavefront that delivers it query $Q2$. Consequently, the M2 coverage for node B is further increased. The overall M2 coverage for node B can now be computed by estimating the complement of joint intersection of node B's transmission range with wavefronts $Q1$ and $Q2$. We discuss this notion of overall M2 coverage in detail below.

2.3 Coverage computation

For a formal definition of M2 coverage of a node i, we assume a set V of WSN nodes deployed uniformly at random, and a set, $S \subset V$, $S = \{s_1, s_2, \ldots, s_K\}$, of K sink nodes.

We represent the query wavefront broadcasted by node i during epoch j, as a circle $C_{i,j}$ of radius $R \times j$, where R is the fixed radio transmission range. The M2 coverage for node i as a result of a query broadcast in its first epoch, $\lambda_{i,1}$, can be derived as:

$$\lambda_{i,1} = \|C_{i,1}\| - \|C_{i,1} \cap C_{s_1,d_1} \ldots \cap C_{s_K,d_K}\| \tag{1}$$

where, $d_k = \frac{\text{distance}(i,s_k)}{R}$, $1 \leq k \leq K$ represents the epoch number during which the query from sink s_k is delivered to node i. Figure 4b presents an intuitive explanation where query wave-fronts from nodes A and C deliver two queries to node B in broadcast rounds 3 and 4, respectively (earlier broadcast rounds not shown in the figure). M2 coverage $\lambda_{B,1}$ for node B's first broadcast epoch is simply the area of its communication range ($\|C_{B,1}\|$) subtracted by the area of intersection of its communication range with query wavefronts originating from nodes A and C ($\|C_{B,1} \cap C_{A,3} \cap C_{C,4}\|$).

The overall M2 coverage for node i, Λ_i, can be derived by adding M2 coverage for each rebroadcast epoch, until the broadcast reaches the entire deployment area. Formally,

$$\Lambda_i = \sum_{j=1}^{D} (\lambda_{i,j} - \varepsilon_{i,j}) \tag{2}$$

where D is a constant large enough to guarantee that the broadcast reaches the entire deployment area, and $\varepsilon_{i,j}$ is an error term. Since we assume a rectangular deployment and represent coverage with circles of increasing radii, it is expected that some part of a coverage circle my fall outside the deployment area. The error term $\varepsilon_{i,j}$ in Eq. 2 subtracts this area from the M2 coverage during each epoch.

2.4 Discrete coverage approximation

Equations 1 and 2 establish the M2 coverage of a given node as a function of its own location and those of K sink nodes currently active in a network. Provided with the required location information, any node can use Eqs. 1 and 2 to locally compute its M2 coverage. However, a model cannot be easily established using these equations as it would involve a close form representation of the area defined by the intersection of $K + 1$ circles of different sizes in general positions; a surprisingly hard problem [9].

In this section, we propose a discrete approximation for the M2 coverage estimation. We base our approximation on a simplifying assumption that the M2 coverage of a node extends from the nodes present on the boundary of its transmission radius. Accounting for the M2 coverage extended through the border nodes automatically includes that offered by any inner nodes. Therefore, instead of computing the exact joint intersection of all wavefronts delivering K queries to node i, we propose to estimate only the length of the boundary (the circumference) of node i's wavefront that falls under this joint intersection.

Figures 5 and 6 illustrate the main idea behind the proposed discrete approximation. In this example, a node B receives a query wavefront from a sink node $S1$ intersecting its transmission range at points P_1 and P_2. We exclude all the nodes located anti-clockwise on the arc (P_1, P_2) from node B's M2 coverage. These nodes have received the query during the same epoch as node B and hence cannot form part of B's M2 coverage. Therefore, we refer to the arc (P_1, P_2) as node B's *uncovered arc*.

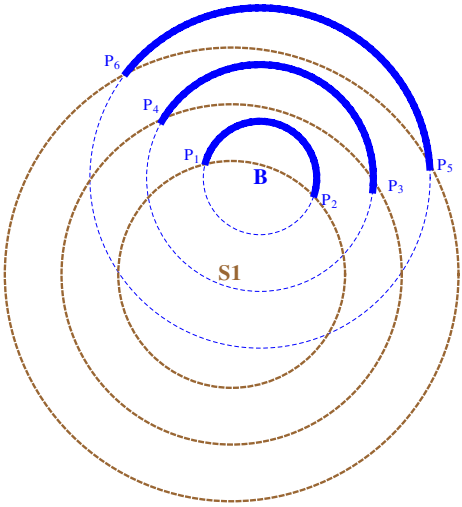
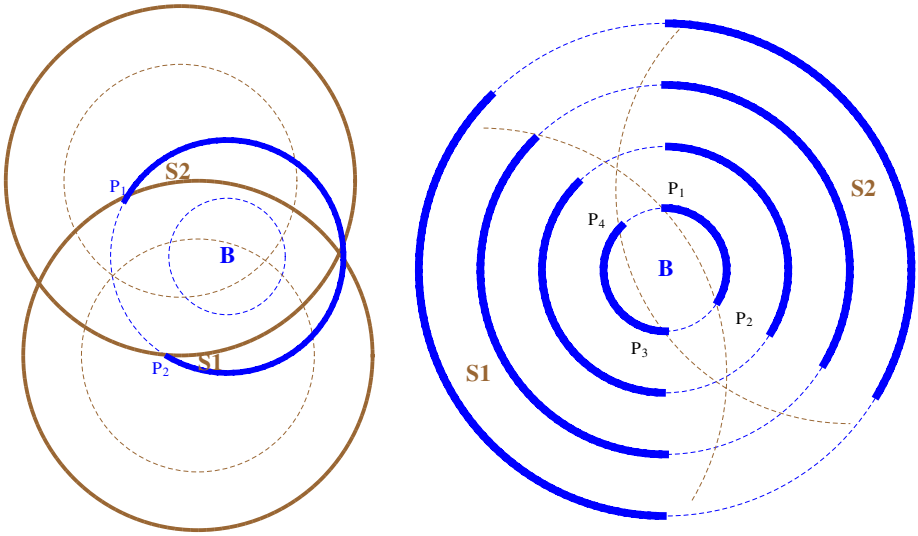


Fig. 5 Discrete approximation for the M2 coverage for a node with one uncovered arc and one sink

Based on the proposed assumption, we use the length of the arc (P_1, P_2) to estimate the M2 coverage area for node B during the first epoch. Subsequent rebroadcasts of the same query by node B and other nodes in the network, result in uncovered arcs of increasing lengths, shown as arcs (P_3, P_4) and (P_5, P_6).

In Fig. 6a, we show node B's uncovered arc in the second broadcast round after introducing a second query originating from sink $S2$. In this case, the number of nodes to be excluded from node B's M2 coverage reduces since some of the uncovered nodes from the first query would not have received the second query in



(a) A node with one uncovered arc and two sinks. (b) A node with two uncovered arcs.

Fig. 6 Discrete approximation for the M2 coverage

the same epoch as node B. This effect can be noted from the reduction in the part of node B's transmission circumference that intersects with *both* query wavefronts. In Fig. 6b, we present an alternative case where the two query wavefronts produce two uncovered arcs for node B. Formally, we define an uncovered arc as:

Definition 1 Given a clockwise sorted set of points $P = \{P_1, P_2, \ldots P_{2K}\}$, where a set of wavefronts $C = \{C_1, C_2 \ldots C_K\}$ intersect with node i's transmission border; an uncovered arc for node i is defined as an arc (P_s, P_{s+1}) where $P_s, P_{s+1} \in P$, and $P_s, P_{s+1} \in C_i, \forall C_i \in C$.

In general, given M_B uncovered arcs, the M2 coverage for node B's first epoch broadcast, $\widehat{\lambda_{B1}}$, can be approximated as follows:

$$\widehat{\lambda_{B1}} = 2\pi R - \sum_{m=1}^{M_B} R\theta_m - \epsilon_{B1} \tag{3}$$

where R is the fixed radio transmission range, θ_m is the angle (in radians) of arc m and, ϵ_{B1} is an error constant to subtract the parts of M2 coverage falling outside the deployment area.

The computation of the overall M2 coverage for a given node is greatly simplified by above approximation. As shown in Fig. 6, all uncovered arcs for node i grow linearly with a factor R (the transmission radius) during node B's broadcasts during subsequent epochs. In general, the M2 coverage for node B during epoch i, can be computed as:

$$\widehat{\lambda_{Bi}} = 2\pi R_i - \sum_{m=1}^{M_B} R_i\theta_m - \epsilon_{Bi} \tag{4}$$

The overall M2 coverage for node B can then be computed as:

$$\widehat{\Lambda_B} = \sum_{i=1}^{D} \widehat{\lambda_{Bi}} \tag{5}$$

where D is a constant large enough to guarantee that broadcast reaches the entire deployment area.

Equation 5 provides a compact and simple method to approximate the M2 coverage of a node as a function of its own location and that of all sink nodes present in a network. Figure 7 shows the accuracy of this approximation for a simulated WSN comprising of 2500 nodes (see Section 5 for details on the simulation setup). In this experiment, we randomly place 5 and 10 sink nodes in the network area and compute the M2 coverage available to each node in the network using uncovered arcs based approximation. We also calculate the exact M2 coverage of each node by computing the joint overlap of query wavefronts and a node's transmission range (Eq. 2) using a Montecarlo simulation based approach [1]. Scatter plots in Fig. 7 show high positive correlation (0.91 and 0.85 for experiments with 5 and 10 sinks, respectively) between M2 coverage approximation and its exact computation.

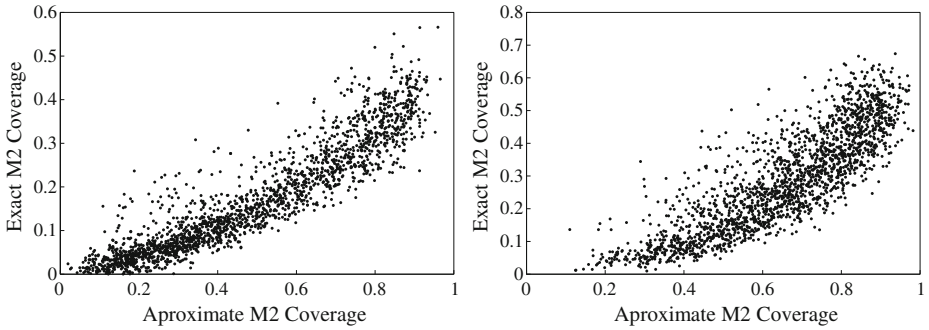


Fig. 7 Accuracy of M2 coverage approximation. Coverage values on both axis are normalized to the network area

3 Achieving high levels of spatial coverage

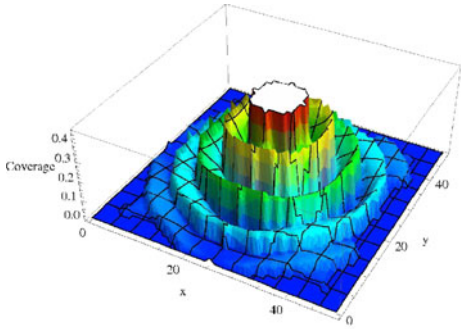
M2 coverage of a given node is a function of its location relative to the rest of the network and the locations of sink nodes. Therefore, a set of sink nodes that results in a high level of coverage for one node may not provide similar coverage for other nodes in the network. In this section, based on our experimentation, we present the critical observation that the overall M2 coverage in a large part of a network rises rapidly to high levels with a modest increase in the number of sink nodes. We then present a probabilistic analysis on the number of randomly located sink nodes that guarantees a high level of overall M2 coverage with a desired probability.

Figures 8 and 9 present the results of our experiments with a simulated WSN comprising of 2500 nodes (see Section 5 for details on the simulation setup). Figure 8a shows the approximate M2 coverage of each node in the simulated network due to a single sink placed at the center of the deployment area. As expected, the M2 coverage is highest (between 35 and 40 % of the entire network area) at the center nodes and decreases gradually with distance from the center. Figure 8b and c show the M2 coverage for 5 and 10 randomly placed sinks, respectively. In each experiment we report the mean coverage of each node after 10 runs of the experiment. We observe from the coverage surface that with an increase in the number of sinks the mean coverage increases significantly and coverage distribution becomes increasingly uniform.

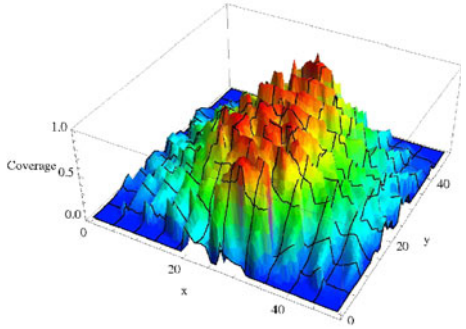
The chart in Fig. 9 reports the M2 coverage histogram as the number of sink nodes is increased up to 5 % of the network. It is important to note that for a number of sink nodes as low as 2 % of the entire network, 64 % of the nodes cover more than 80 % of the deployment area, while 87 % of the nodes cover more than half of the deployment area.Following the experimental observations, we now present a probabilistic analysis that shows that such high levels of M2 coverage are achievable in a network with only a small number of sink nodes.

3.1 Probabilistic estimation based on randomly located sink nodes

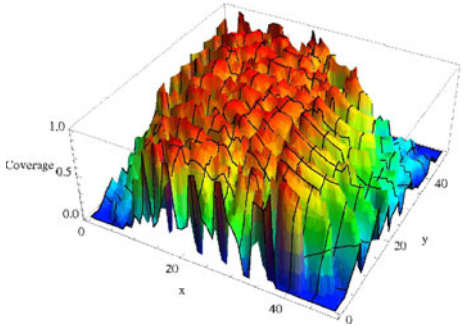
The M2 coverage of all nodes increases with the increase in the number of sink nodes but the magnitude of increase varies with each node's location. Given a nonempty set



(a) 1 sink (placed at the center of network area)



(b) 5 sinks



(c) 10 sinks

Fig. 8 The approximate M2 coverage with increasing sink nodes distributed uniformly at random

of sink nodes, a node i achieves maximum M2 coverage if it has no uncovered arcs on its transmission border. This requires sink nodes to be located such that the joint intersection of all query wavefronts reaching node i does not include any segment on its transmission border. To ensure maximum M2 coverage for any node in the network, its uncovered arcs must be eliminated. An uncovered arc (P_s, P_t) of a node i can only be fully eliminated if a query wavefront from a newly added sink reaches node i such that the wavefront does not intersect with the arc (P_s, P_t). If (P_s, P_t) is the only uncovered arc of node i, the newly added sink will result in maximum M2 coverage for node i.

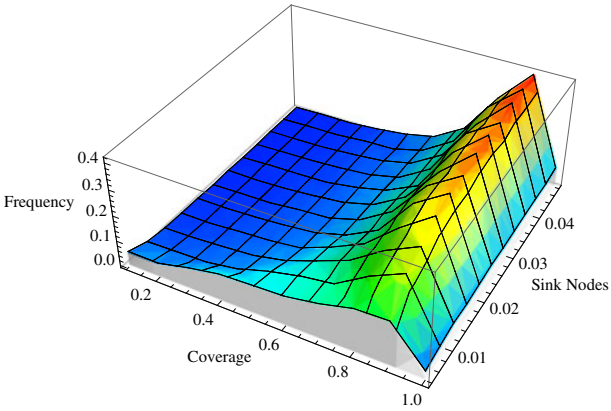


Fig. 9 The M2 coverage histogram for an increasing number of sinks for a 2500 node network. The frequency axis shows the number of nodes in a histogram bin while all values are normalized to the total number of nodes in the network

Figure 10 further explains the notion of uncovered arc elimination for maximum M2 coverage: node B's uncovered arc is bounded by points P_1 and P_2 with its center at point P_{12}. To achieve maximum M2 coverage, node B requires a sink node located such that its query wavefront reaches B and does not include points P_1 and P_2. To compute the subregion inside the network where such a node may be located, we create a Voronoi diagram using node B's location and points P_1, P_2 and P_{12} as vertices. The required sink node cannot be located inside the Voronoi cells of points P_1, P_2, and P_{12} as any node inside these cells will be closer to P_1, P_2 or P_{12}, respectively, than node i. Therefore, a wavefront from such a sink node must include these points before reaching node B, hence violating the basic criteria set for the required sink node. Observation 1 states this insight.

Observation 1 *Given a node i located at point P_i, its uncovered arc (P_s, P_t) with center at P_{st} and the Voronoi diagram V of points P_s, P_t, P_{st}, P_i bounded inside the network area. The uncovered arc (P_s, P_t) can only be eliminated by a sink node located inside the Voronoi cell $V(P_i)$.*

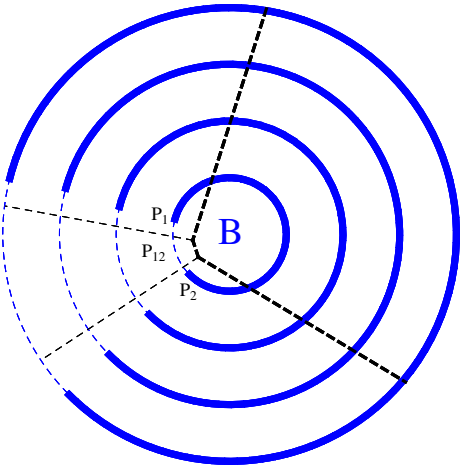


Fig. 10 Deriving the probability of exhaustive M2 coverage using the Voronoi diagram of a node's location and end and mid points of its uncovered arc

Based on the analysis of potential sink locations for maximum M2 coverage, we derive the number of uniformly distributed sink nodes that eliminate an uncovered arc of a node i with probability p. According to Observation 1, the number of sink nodes required to eliminate an uncovered arc can be related to the ratio between the area $\|V(P_i)\|$ of node i's Voronoi cell, and the network area. If the network area is denoted by A, the probability that a newly arrived sink node will fall inside node $V(P_i)$ is: $\frac{\|V(P_i)\|}{A}$. If κ_i new sink nodes arrive in the network, the probability that at least one sink node will fall inside $V(P_i)$ is:

$$p = 1 - \left(1 - \frac{\|V(P_i)\|}{A}\right)^{\kappa_i} \tag{6}$$

where $(1 - \frac{\|V(P_i)\|}{A})$ is the probability of failure in one attempt, i.e., the probability that a new sink node is located outside $V(P_i)$. To derive κ_i, i.e., the number of sink nodes that eliminate an uncovered arc of node i, we rewrite Eq. 6 in terms of the desired probability p of success as:

$$\kappa_i = \frac{\log(1-p)}{\log\left(1 - \frac{\|V(P_i)\|}{A}\right)} \tag{7}$$

If a node has more than one uncovered arc, for instance as shown in the example in Fig. 6b, it may require more sinks to eliminate all uncovered arcs. To derive κ_i for a node i with M uncovered arcs, we first create Voronoi diagrams $V_1, \ldots, V_M$ corresponding to each uncovered arc. Equation 7 shows that the number of sink nodes required to eliminate an uncovered arc depends on the area of the Voronoi cell of the node's location in the Voronoi diagram corresponding to the arc. For a node i with multiple uncovered arcs, an upper bound on the number of required sinks can be established by considering the arc, say m, that leads to the smallest Voronoi cell $V_m(P_i)$. Formally, the number of sink nodes that eliminate M uncovered arcs of node i with probability p is given as:

$$\kappa_i = \frac{\log(1-p)}{\log\left(1 - \frac{\|V_m(P_i)\|}{A}\right)} \tag{8}$$

where, $\|V_m(P_i)\| = \text{MIN}(\|V_j(P_i)\|), \quad j \in \{1, \ldots, M\}$.

The proportion $\frac{\|V_m(P_i)\|}{A}$ in Eq. 8 captures the effect of the relative position of a given node with respect to the deployment area and sink nodes already present in the network. The chart in Fig. 11 shows the number of required sinks for success probabilities, 0.99, 0.9 and 0.8, for a range of Voronoi cell proportions. We observe that the success probability does not play a major role in determining the number of sink nodes required to provide maximum M2 coverage if the Voronoi cell of a node is large. Intuitively, the more centrally a node is located, the smaller is the number of expected sink nodes required to cover it. Border nodes, on the other hand, are the most challenging to optimally cover. It is often only possible when the node itself acts as a sink node.

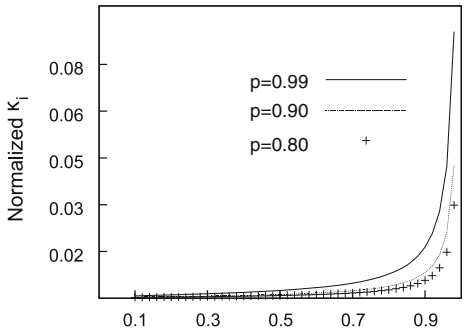


Fig. 11 Worst case bound on the number of required sink nodes (normalized to network size, i.e., 2500 nodes)

Given N nodes, the number of sink nodes required to maximize the coverage for all nodes with probability p is:

$$\kappa = \underset{i \in \{1,\ldots,N\}}{\operatorname{argmax}} \kappa_i \tag{9}$$

The above computation is based on the assumption that sink nodes are distributed uniformly at random. Since we only take the total number of nodes and not specific locations into account, a node j with $\kappa_j \leq \kappa$ is expected to be already covered as the number of sink nodes grows to κ.

3.2 Approximation accuracy

Using the probabilistic analysis in Section 3.1, one can derive the number of sink nodes required to provide maximum M2 coverage. In this section, we analyze the approximation accuracy of our analysis by comparing it to a greedy strategy.

As our analysis assumes a uniform distribution of sink nodes, it can be argued that we may miss the number of required sinks in comparison to a strategy that picks sink locations carefully. We show that the optimal sink placement to maximize M2 coverage for an entire network using a minimum number of sink nodes is an NP-hard problem. Given the approximate M2 coverage for a given node (Eq. 5), the overall node coverage for a set V of WSN nodes, with respect to a set of sink nodes S, can be formulated as:

$$\Lambda_S = \sum_{i \in V} \widehat{\Lambda}_i \tag{10}$$

For maximum M2 coverage over an area A, the M2 coverage of each node $v \in V$ must be close to A. The optimal sink placement (OSP) problem can then be defined as to find the smallest set of nodes $S^* \subset V$, that yields: $\Lambda_{S^*} \approx \|V\| A$.

Lemma 1 *The OSP problem is NP-hard.*

Proof of Lemma 1 The OSP problem can be reduced to the NP-hard set-cover problem [5] as follows: Assume the set of all WSN nodes V to be the universe for which we seek a cover. The M2 cover for each sink node $i \in V$ can be represented as a set $s_i \subset V$, i.e., a set of all nodes that fall inside the M2 coverage area of sink i. Since any WSN node can be a sink node, we obtain a family of subsets $\mathcal{U} = \{s_1,$

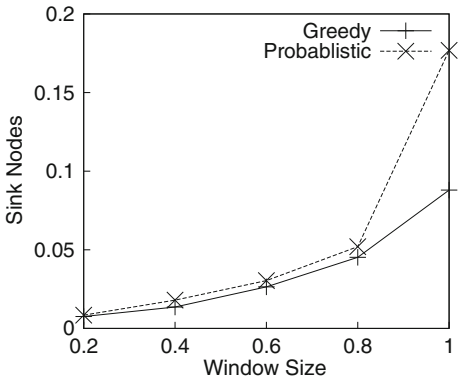


Fig. 12 Greedy vs probabilistic approximation. Number of sink nodes are normalized to the total number of nodes in the network while the window size is normalized to the network area

$s_2, \ldots, s_N\}$, $N = \|V\|$, where each subset in $\mathcal{S}$ represents the M2 cover of corresponding sink node. For complete coverage, we are required to pick a subfamily $\mathcal{S}^* \subset \mathcal{S}$ whose union is V. Finding the smallest such subfamily is equivalent to set-cover optimization. □

Experimental evaluation of approximation accuracy Typically, a greedy heuristic is adopted for set-cover optimization problem that achieves the best possible polynomial time approximation [20]. For our simulated network, Fig. 12 shows a comparison between this greedy approximation and our proposed probabilistic approximation. In each instance of this experiment, we compute the number of sink nodes required to provide maximum M2 coverage to all nodes located inside a square spatial window co-centric with the deployment area. We compute the number of sinks using the greedy heuristic and Eq. 9 with a success probability, p, of 0.99.

The chart in Fig. 12 shows that for optimal sink placement, our probabilistic technique has an approximation accuracy close to that of the greedy heuristic. For both techniques, the number of required sink nodes increases with the increase in the window size. The greedy approach outperforms the probabilistic placement of sink nodes only when the window size is close or equal to the network area. This behavior is expected as it becomes increasingly hard to cover the border nodes. The similarity in the results of this experiment shows that a network can garner a nearly equal level of opportunistic sampling coverage if sink nodes are distributed uniformly at random or if they are *proactively* placed using a greedy strategy.

4 Maintaining accuracy in opportunistic sampling system

A node must estimate the accuracy of the data it acquires through opportunistic sampling before this data can be used in query evaluation. The probabilistic model discussed in the previous section enables the WSN nodes to monitor their spatial coverage based on the total number of sink nodes in a network. However, due to the inherent temporal variation of sensed phenomena, spatial coverage of a sample alone cannot guarantee its accuracy. Once a network reaches the number of sink nodes required for optimal coverage, new query broadcasts can be suppressed only until the samples held by all nodes remain within a desired level of accuracy. Hence,

a method is needed to *regulate* new query broadcasts such that opportunistic samples accurately reflect the global state of the phenomenon under observation.

4.1 Spatio-temporal variogram

Historic trends of a physical phenomenon learnt over past sensor readings provide the best guide for sampling accuracy. A common geostatistical tool used for this purpose is the *variogram*: a function based on of samples of a phenomenon and distance between the corresponding sampling locations [18]. The variogram model has numerous applications in spatial data analysis, including sampling accuracy estimates [3] and spatial interpolation (Kriging) [7].

Assume a random variable Z represents a spatio-temporal phenomenon observed at S sites over a time period T. Assume further that $Z(s_i; t)$ represents a point observation for sampling interval t at location s_i; $t = 1, \ldots, T$, $i = 1, \ldots, S$. For a given spatial lag h and temporal lag u, the empirical spatio-temporal variogram function can be defined as [6]:

$$\hat{\gamma}(h; u) = \frac{1}{2\|N(h; u)\|} \sum_{(i,j,t,t') \in N(h;u)} \left[Z(s_i; t) - Z\left(s_j; t'\right)\right]^2 \tag{11}$$

where, $\|N(h; u)\| \equiv \{(i, j, t, t') : s_i - s_j = h; \|t - t'\| = u, i, j = 1, \ldots, S\}$, i.e., is the number of data pairs located at distance h and sampled within u time units of each other.

To generalize the empirical variogram as a phenomenon's spatial correlation model, a surface function $f(h, u)$ is fitted onto the empirical values. The resultant variogram model can then be used to find the correlation between phenomenon values at arbitrary locations. In estimation problems, such as computing the mean value of a phenomenon using point samples, the variogram can be shown as equivalent to the variance of error [3], i.e.,

$$\sigma_e^2 = 2\gamma(h; u) \tag{12}$$

This relation greatly simplifies the calculation of a confidence interval. For instance, based on the assumption that error is normally distributed, a 95 % confidence interval for mean estimation can be derived as $CI = \pm 1.96\sqrt{2\gamma(h)}$. In certain applications, including the ones addressed by this work, point samples are aggregated prior to being used in estimation; for instance, computing the mean value of a phenomenon inside a spatial segment using the mean values of nearby segments. In such cases the variogram function is averaged over the entire area of concern before confidence intervals are derived, a process known as regularization [7]. For such situations, Eq. 12 is modified as following:

$$\sigma_e = \frac{1}{\|H\|\|U\|} \int_H \int_U 2\gamma(h; u) du\, dh \tag{13}$$

where the spatial lag, h, varies in $\|H\|$, and the temporal lag, u, varies in $\|U\|$.

4.2 An example spatio-temporal variogram model

Figure 13a shows the empirical spatio-temporal variogram computed for a dataset that we later use for experimentation. This dataset is collected by a WSN of 54

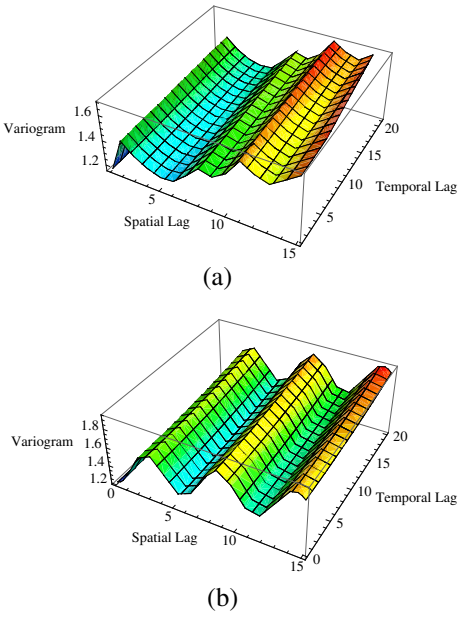


Fig. 13 Modeling the spatio-temporal variogram

nodes deployed in the Intel Berkeley Research lab [17]. Each sensor records several readings per minute including temperature, humidity, luminosity, and voltage. We preprocess the data as follows:

- first, we randomly select 10 days from the dataset;
- second, we decimate the data for each day by reducing the sampling interval to 30 min;
- third, we compute an overall mean for each sampling interval by averaging the corresponding temperature values across all selected days;
- lastly, using the spatial lag of 1 m and temporal lag of 1 sampling interval (30 min) we use Eq. 11 to compute the empirical spatio-temporal variogram for a 12-h period, from 0800 to 2000 h (Fig. 13a).

In addition to spatial and/or temporal lag variables, empirical variogram models are parameterized using the concepts of nugget, sill and range. *Nugget* is the height of the jump of the variogram at the origin, *sill* is the limit of the variogram (tending to innitely large distances) and the *range* is the distance for which the difference of the variogram from the sill is minimal. Intuitively, *nugget* reflects the correlation between samples at the closest lag, *range* is the distance after which correlation between samples stops showing any significant change while, *sill* is the value of correlation between samples at a distance equal to *range*.

As shown in Fig. 13a, the empirical variogram for Intel lab data shows a periodic trend with a sill (C) at 1.6, spatial range of influence (a) at 12 m and a nugget effect (τ) of 1.2. In the literature, an empirical variogram with a periodic trend is referred to as the *hole effect* variogram and a trigonometric function is often adopted to model this trend [21].

To model the empirical variogram, we use a linear combination of the cosine hole effect and exponential models on the spatial dimension and a purely temporal model, given as follows:

$$\gamma(s_i; t) = \tau + \alpha_1 * (1 - \cos(C_n s_i \pi)) + \alpha_2 \left(1 - \exp^{\frac{s_i}{-a}}\right) + \beta_1 t^{\beta_2} \tag{14}$$

where, τ is the nugget effect, a is the range of influence, α_1 and α_2 are scaling parameters for the linear combination of spatial models, C_n is the sill for the hole effect model, and β_1 and β_2 are parameters for the temporal model. Values for τ, a and C_n are derived from the empirical variogram itself and are found to be, 1.2, 15 and 0.35, respectively. The rest of the constants are derived using least square estimation and their values are as follows: $\alpha_1 = 0.189, \alpha_2 = 0.405, \beta_1 = 0.00027$, and $\beta_2 = 2.009$. The fitted model is shown in Fig. 13b.

4.3 Maintaining opportunistic sampling accuracy using spatio-temporal variogram

Although the variogram model building is a complex process, it is a once-off cost for one central node. Based on historic trends, a variogram characterizes the correlation among sensor nodes as a phenomenon evolves. Actual node values may vary in the future, but the correlation among nodes is expected to follow the correlation trend modeled by the variogram. Hence, it is practical to construct the variogram model for a phenomenon in a given WSN using past data and distribute the computed model in the network at the time of initialization. Therefore, we do not consider the variogram model building cost as a parameter in our analyses of the opportunistic sampling technique.

Once an appropriate variogram model is built and distributed in the network, every node can use Eq. 13 to determine the confidence interval for estimation based on its current sample. However, a node first requires the spatio-temporal extent of its current sample so that appropriate spatial and temporal lag values can be used to evaluate Eq. 13. We propose the following strategy to compute the confidence interval for a node B's current sample based on a given variogram model:

1. *Setup.* Distribute spatio-temporal variogram model (Eq. 13) to all nodes in the network;
2. For each new query:
   (a) Divide the entire network area in a regular grid G of resolution R;
   (b) For each cell $c \in G$:
      i. Find the most recent sink node S_c responsible for data flow from c to node B;
      ii. Set the temporal integral bounds (for the evaluation of Eq. 13) as the arrival time of sink S_c and the current time;
      iii. Set the spatial integral bounds as the distance between node B and the edge of cell c closest to B and the distance between node B and the edge of cell c farthest from B;
      iv. Evaluate Eq. 13

(c) Compute the mean variogram for all cells;
(d) Compute the desired confidence interval using mean variogram.

The above process is repeated by any node that requires to respond to a query. Using this computation it can determine if the resulting error is higher than a user defined confidence interval. If the error is indeed higher, the node broadcasts the query in the network. Otherwise it simply evaluates the query using its current sample.

The computational complexity of the above process is simply $|G|K$, where $|G|$ is the number of cells in grid G and K is the constant time operation cost of evaluating Eq. 13. To avoid computing the integration operation at each node, we propose to centrally compute the symbolic integral for the required variogram model parameterized in spatial and temporal lag parameters. Instead of distributing the actual variogram model, all nodes can then be initialized with the pre-computed integral. In this way, each node can simply substitute its spatial and temporal lag parameter values in the pre-computed integral to calculate a variogram value (Step 2(b)-iii).

5 System implementation and applications

In this section, we provide the implementation details for the opportunistic sampling system deployed in a simulated WSN environment. Using real and simulated datasets as seed, we present two main applications of opportunistic sampling. In the first application, we test the accuracy and communication cost of opportunistic sampling for computation of global statistics including `MIN`, `MAX` and `AVG` queries. The second application relates to the computation of same queries in a spatial range centered at each node.

5.1 Simulation settings

We design two WSNs in a custom-built simulator environment using C++. WSN-1 comprises of 2500 nodes, spans an area of 50 m^2 with node density of 1 node/m^2, where all nodes are located in a grid pattern. WSN-2 is designed to be comparable to the Intel Lab data set (as described in Section 4.2), comprising of 54 nodes set in a geographical layout that replicates the original deployment. For both WSNs, the communication radius of each node is set as 5 meters. In WSN-1, each node is assigned a synthetic value determined as a function of its location and current time, while in WSN-2, each node is assigned a value according to the original dataset.

We assume a query-tree based data collection model similar to TAG [22], where each query is broadcasted to the network and data is collected through a routing tree rooted at the respective sink node. We analyze the performance of opportunistic sampling approach based on its communication cost and accuracy. For communication cost, we estimate the overall energy spent in data transmission and reception using the function $\varepsilon = \sigma_s + \delta_s x$, where ε is the total amount of energy spent in communicating a message with x bytes of content, and σ_s and δ_s represent the per-message and per-byte communication costs, respectively [28]. We use the specification of the Mica motes [8] to set the values of constants σ_s and δ_s.

5.2 Implementation

As part of the system initialization, all nodes are provided with the number of sink nodes that are required to cover the entire network with a user-defined probability (i.e., the value of κ using Eq. 9). Nodes are also provided with a variogram model, i.e., the definite integral of Eq. 13 pre-computed in terms of variables H and U. During network operation each node maintains a list of current sink nodes that it has responded to and the time of arrival of associated queries.

On receiving a new user query, each node adopts the following strategy for query evaluation:

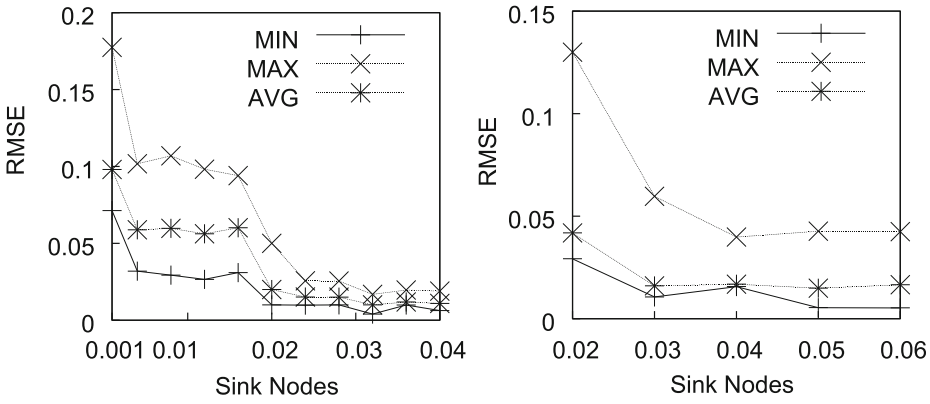
1. If the number of current sink nodes is less than κ, broadcast the query;
2. If the number of current sink nodes is greater than κ, evaluate the estimation error of the current sample using the variogram model;
3. If the estimation error is less than a predefined threshold, evaluate the query on current sample;
4. If the estimation error is greater than the predefined threshold, broadcast the query.

5.3 Gathering global statistics

In this section, we present a set of experiments where nodes opportunistically gather and transmit partial aggregates while responding to queries issued from different parts of the network. The sampling and aggregate computation strategy adopted by nodes during opportunistic sampling varies with respect to the nature of required aggregates. For instance, for the minimum (or the maximum) value of a phenomenon under observation, each node can simply maintain and transmit the minimum (or maximum) value that it receives or overhears. Other aggregates such as average can also be computed in a similar fashion with one major difference: unlike minimum or maximum, the average is not an order and duplicate insensitive (ODI) aggregate. Since the nature of opportunistic sampling is multi-path, the simplistic averaging technique of forwarding count and sum of child node values to the parent node will produce inaccuracies. Discussed in detail in [25], for aggregates, such as the average, that are not ODI, we propose to use the aggregation algorithm proposed in [25].

The aim of our experiments is to show that even a small number of queries in the network provide enough data to a large number of nodes to compute global statistics with high accuracy. We also analyze the communication cost of opportunistic sampling. We expect this cost to be minimal as opportunistic sampling is not a proactive technique, such as data collection, and works by piggy-backing partial aggregates with data flow already present in the network.

Figure 14 shows the effect of increasing number of sink nodes on estimation accuracy of `MIN`, `MAX` and `AVG` queries. In these charts, the sink node values are shown as proportion of network size, while accuracy is represented as normalized root mean squared error (RMSE). The charts show that an increase in the number of active sink nodes increases the mean M2 coverage for all nodes leading to a better estimation. For the simulated data set (Fig. 14a), a relatively small number of active sink nodes (2 %), distributed uniformly at random in the network, is enough to guarantee a mean error less than 2 % in `MIN` and `AVG` and less than 5 % in `MAX` aggregates. Experiment with real data shows similar trend (Fig. 14b).



(a) WSN-1, Simulated data (b) WSN-2, Intel lab data

Fig. 14 Effect of the number of sink nodes on estimation accuracy. Number of sink nodes normalized to the total number of nodes in the network

Figure 15 shows the error surface of AVG aggregate at each node for simulated data set with 2 % active sink nodes. Although the error at border nodes is higher, for a large number of nodes located close to the deployment center, the error is smaller than 2 %. This result further confirms the high level of M2 coverage for central nodes, even with relatively few sink nodes.

The charts in Fig. 16 show the cost comparison between computing global statistics in a network by opportunistic sampling (denoted as M2) in comparison to a proactive data aggregation technique using TAG. In our implementation of TAG, a designated base station collects aggregate data from the entire network employing in-network aggregation. It then broadcasts the final aggregates to all nodes in the network. Since during data aggregation and broadcast phases each node transmits only one message, i.e., towards its parent node, a single run of this technique is quite efficient. However, due to constant temporal variation, frequent data collection is required to maintain the aggregation accuracy.

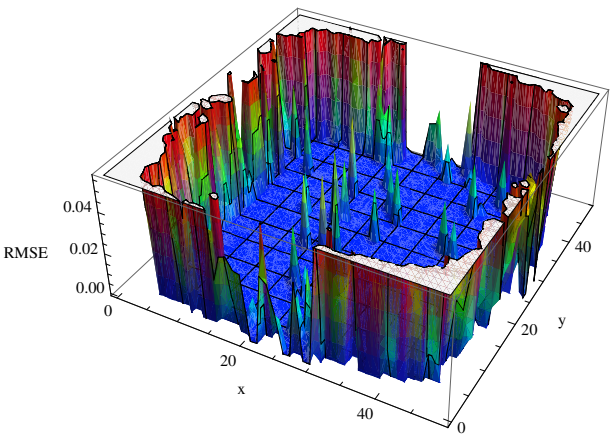
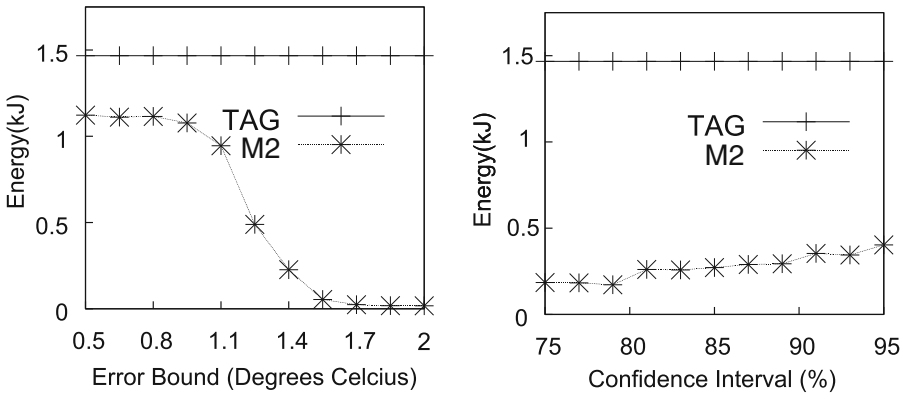


Fig. 15 Distribution of error in WSN-1 for AVG with 2 % sink nodes



(a) Effect of increasing the error bound on performance. Confidence interval fixed at 95%.

(b) Effect of increasing the confidence interval. Error bound fixed at 1.25° Celcius.

Fig. 16 Effect of temporal variation on system performance (simulations results for WSN-2, Intel lab data)

In the face of temporal variation, a simplistic technique like TAG has to resort to a constant rate of data recollection and retransmission of results to all nodes in the network. Opportunistic sampling on the other hand, makes use of historic spatio-temporal trends in the data and can adapt itself to any level of temporal variation based on a desired accuracy level. The charts in Fig. 16 highlight this strength. In this experiment, we use the Intel lab data for a 12-h period (0800–2000 h) with a requirement to maintain the mean network area temperature at all nodes in the network. We assume the presence of data flow in the network with a query arrival rate of 10 queries per hour. The experiment is repeated for different values of desired accuracy, defined in terms of: (i) acceptable absolute error (Fig. 16a), and (ii) acceptable confidence interval (Fig. 16b). Meanwhile, TAG is oblivious to these accuracy levels and always strive to compute the most accurate reading and hence having to rerun several times during the session. We repeat the experiment for five randomly selected days in the data set and report only the mean values here. Results show that the energy efficiency of opportunistic sampling system increases quadratically with increase in error threshold. This feature provides the WSN applications with a cheap yet effective method of maintaining global aggregates if a predefined level of error is acceptable.

5.4 Gathering multi-resolution statistics

In multi-sink WSNs, the queries issued by sink nodes often have a localized spatial scope, i.e., nodes are most often interested in aggregates only from the area surrounding them. Consider for instance, a WSN monitoring empty car spaces in a parking lot where nodes are queried by drivers for conditions in the space immediately surrounding a node. In such settings, multi-resolution statistics provide better estimation accuracy than global aggregates. For each node, multi-resolution statistics is computed with a bias towards the nodes located closer to it. As a result the aggregation

accuracy for the area in the immediate vicinity of a node is higher and decreases as an inverse function of distance. Typically, multi-resolution statistics for a given node in a WSN is calculated based on a sample drawn from its neighborhood. Several proactive techniques for in-network multi-resolution aggregation have been reported in the literature [11, 27].

In contrast to the existing multi-resolution aggregation techniques, opportunistic sampling provides a means to exploit the data flow present in a network for collection of multi-resolution statistics at every node. Similar to the piggy-backing strategy in global statistics computation, in this application each node piggy-backs a sample of values consisting of its own and sensor readings it receives or overhears during query response phase.

Sample selection Assume each sample is a tuple $t_j = (x_j, y_j, r_j)$, where first two values are the location of a node j that creates this tuple and the last value is its reading. A node i aims to select a sample of size f, where f is expressed as a fraction of the total number of nodes in its M2 coverage. Assume d_i to be the diameter of node i's M2 coverage, we divide d_i in D discrete levels. Then, node i selects a tuple t_j originating from node j with probability:

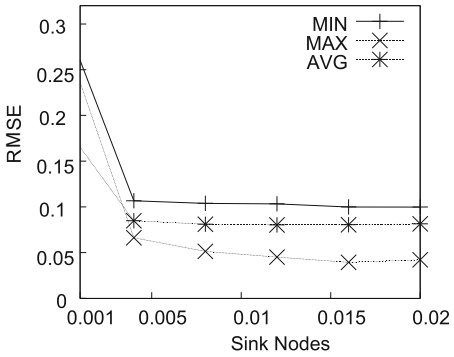
$$p_{ij} = \frac{f}{2\pi x} \qquad x \neq 0, 0 < f \leq 1 \tag{15}$$

where x is the distance between nodes i and j normalized using the range 0 to $\frac{D}{d_i}$. Given that node i is located approximately at the center of its M2 coverage, a condition true for all nodes with a high M2 coverage, the probability function in Eq. 15 ensures that node i gives preference to the samples generated by the nodes located close to it. Although the number of nodes at a distance $d \leq d_i$ from node i increases linearly with d, the function ensures that the number of selected samples remains fixed, as:

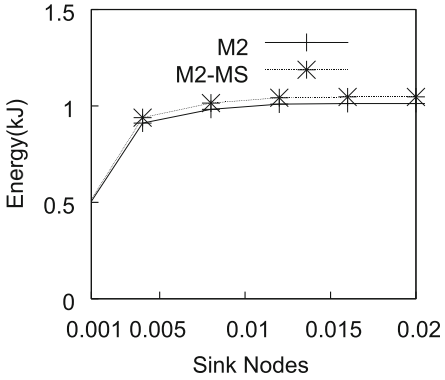
$$\int_0^1 p_{ij}\, dx \approx f. \tag{16}$$

To compute multi-resolution statistics, nodes employ the following strategy: While responding to network queries, nodes use the above sample selection technique and piggy-back the selected set of samples along with query response. Similar to global aggregate computation, nodes can terminate sampling as soon as a desired level of M2 coverage is obtained. Using the selected samples, each node can compute aggregates with various granularity levels in increasing spatial extents centered on itself.

Figure 17 shows the results of our experiments with this application domain. In these experiments, we perform `MIN`, `MAX` and `AVG` aggregates in a circular window of 15-m radius, centered at each node. Each node selects a multi-resolution sample equal to 5 % of its M2 coverage i.e., by setting f to 0.05 in Eq. 15. We only use the simulated data set in this experiment as the Intel lab data set is too small to produce meaningful results. The chart in Fig. 17a shows the accuracy of each aggregate as the coverage increases. The accuracy of opportunistic sampling approach for multi-resolution aggregation shows a pattern similar to that for global aggregation. With only 0.4 % nodes acting as sink nodes, the mean accuracy of each aggregate is above 90 %. It is important to note that the accuracy of multi-resolution statistics is lower to that we achieved for global statistics.



(a) Mean accuracy for aggregates in a window of 15m radius centered at each node.



(b) Mean cost of opportunistic sampling with and without multi-resolution sampling.

Fig. 17 Opportunistic sampling for multi-resolution statistics. Number of sink nodes normalized to the total number of nodes in the network

A major concern for adopting opportunistic sampling for multi-resolution aggregation is increase in its cost due to a larger message size. As opposed to piggy-backing just one value for each aggregate for global aggregation, multi-resolution sample requires forwarding all selected samples hence increasing the message size for node i by $O(f\Lambda_i)$ bytes, where f is the size of the selected sample and, Λ_i is the current M2 coverage for node i. As charts in Fig. 17a show that a high accuracy can be obtained for a sample size of 0.05, we do not expect the increase in transmission cost to be significant. Figure 17b presents the comparison of costs of opportunistic sampling with and without multi-resolution sampling, for increasing number of sinks and f of 0.05. As expected the difference between the costs of opportunistic sampling with and without a multi-resolution sample is not significant.

6 Further analysis

To simplify the presentation of main costs and benefits associated with opportunistic sampling system, Section 5 does not report on low level operational mechanisms such as storage and fault tolerance. Typical WSN implementations share many such implementation details and are thus affected by similar constraints. In following subsections, we provide a brief analysis of the impact of typical operational factors on our proposed system.

Sample storage An opportunistic sampling approach demands that each node maintains extra information on-board with its own sensed data. The volume of this information mainly depends on the nature of application. For instance, the storage cost of a global statistics gathering application is negligible as nodes maintain only one global statistic for each sensed parameter, i.e., in the form of a partial aggregate of all incoming data. Multi-resolution statistics applications, on the other hand, require more space as nodes maintain samples representing multiple spatial windows.

However, even in this case the storage cost is not prohibitive as the number of partial aggregates that a node maintains grows only linearly with the granularity of the multi-resolution statistic.

Communication and node failures The multi-path nature of opportunistic sampling approach makes it inherently robust against losses due to network or hardware failures. As shown in multipath routing literature [24, 25], even in moderately dense deployments, packets arising from any region may propagate through the network on many different routes. Consequently, nodes can maintain accurate global or neighborhood statistics evens in the face of communication and node failures in certain parts of the network.

Concurrent queries In an active WSN where any sink can be used to initiate a query, many queries can be injected in to the network simultaneously. In the proposed system, nodes respond to a query using data gathered during past communications and hence are not affected by any concurrent queries present in other parts of the network. Queries are only broadcasted if a node determines that it cannot accurately respond to a certain query using its local data. The worst case scenario in terms of efficiency of the proposed method occurs when similar queries are injected at the same time and all sink nodes decide to broadcast their queries. Although such a scenario is likely to be rare, the querying system could be extended by enabling nodes to suppress further broadcast of a query based on a measure of its similarity to recently relayed queries.

7 Related work

A large number of localized approaches for WSNs rely on some form of global statistics for enhancing accuracy or energy efficiency. For instance, for each node a in the network, the WSN database system TinyDB [23] proposes to maintain a copy of `MIN` and `MAX` aggregates of all direct and indirect children of node a. The `MIN` and `MAX` aggregates of sensed variables enhance the performance of selective queries based on these variables. Similarly, in the distributed model-based data collection method *Ken* [2], a copy of a global statistical model is distributed to each WSN node so that nodes send their data to the base station if it has a considerable effect on the global statistics. In distributed regression proposed in [14], neighboring nodes coordinate processing such that a global regression task is computed locally. However, this method also requires a prior distribution of global statistics in the WSN, in the form of regional correlation models (kernels).

Motivated by the need of global statistics for localized and distributed information processing, a number of methods for collecting and distributing such statistics have been proposed in the WSN literature. In the following subsections, we discusses these methods in detail.

7.1 Multi-resolution in-network data storage

Multi-resolution storage techniques propose to build an in-network storage structure such that each node possess global aggregates of multiple granularity levels.

Typically, granularity levels are defined in a hierarchical and spatially decaying fashion, i.e., the level of summarization gradually increases with the increase in area.

In [11], a multi-resolution data storage system called DIMENSIONS is proposed. In this system, the WSN area is recursively divided as grids of increasing resolution and data for each grid cell is summarized using a wavelet-based technique. The authors propose a distributed variant of the quad-tree concept to locate a data summary in the network. Although DIMENSIONS and other distributed data indexing systems [13, 26] enable fast in-network search of pre-computed aggregates, the costs involved in maintaining aggregation accuracy are often non-trivial. Therefore, it is important in such systems to strike a balance between frequency of aggregation update, the change in observed phenomenon and the volume of queries being submitted.

Other in-network storage systems rely on hash functions, such as the geographic hash table (GHT) [26], to store and retrieve aggregates at certain nodes. However, the main problem of such a technique is that since the hash function remains fixed, some nodes are queried more often than others and in turn may become a bottleneck for the entire system. Fractional cascading [12] proposes to collocate information with the sources (nodes) that generate it. As a result, the most accurate information available at the lowest cost to each node is about its own neighborhood. This approach is useful in multi-sink WSNs with no fixed base station, as users in such networks are likely to be interested in localized information and collocate with the sink node they choose to query the network.

For WSNs processing high volumes of user queries, opportunistic sampling provides a much efficient alternative to achieve the goals set out by specialized global knowledge gathering approaches. By exploiting the multihop and multipath nature of WSN communication, opportunistic sampling integrates aggregate computation with regular query processing and hence eliminates the need of specialized techniques to regularly refresh global knowledge.

7.2 Gossip-based probabilistic aggregation

Pre-computation and in-network data storage helps to improve query processing but individual nodes still require to traverse the storage structure in order to access any global aggregate. The traversal cost depends upon the distance between the node that requires the global statistic and the node hosting this information. Although some replication methods [26] have been proposed mainly to improve fault-tolerance, the traversal cost can still be quite significant. As an orthogonal approach, the *spatial gossiping* method aims to build and distribute a uniform level aggregate throughout the entire network. Typically, a gossiping message goes through several rounds of communication among randomly chosen neighbors [19]. Once the gossiping reaches an equilibrium, a large proportion of nodes is expected to accumulate the desired global aggregate.

In [27], Sarkar et al. propose the hierarchical spatial gossip (HSG) algorithm that combines in-network storage and gossiping approaches to accumulate multi-resolution aggregates at each node in the network. The HSG algorithm works in an iterative manner, where during each iteration each node in the network finds itself a gossiping partner to share data with. The distance at which nodes look for gossiping partner is increased exponentially during each iteration so that for a n node network, $O(\log n)$

iterations suffice. As it is the case in above described techniques, the communication cost of HSG is non-trivial and for large networks may become prohibitive.

7.3 Multipath advantage in WSNs

The multipath nature of WSN communication has been exploited before for a variety of goals. One of the main uses of multipath communication in WSN is fault tolerant in-network data aggregation [4, 24, 25]. Single path data aggregation, such as TinyDB, is highly susceptible to link and node failure. Multi-path aggregation enhances robustness by exploiting the fact that that all nodes within a communication range can overhear a message transmitted towards a sink a node. Instead of relaying a message using a single node as TinyDB, all neighboring nodes relay the same message toward the sink. As a consequence, multipath aggregation becomes more robust for node failures or communication losses. To deal with the resulting redundancy, multipath schemes integrate probabilistic order and duplicate insensitive (ODI) methods of the sketch theory [25]. Multipath communication has also been exploited for spatial suppression in data collection [22]. In this approach, a node suppresses its response to a query if it overhears a similar response from another node in its neighborhood. Since, our proposed opportunistic sampling method is based on a multi-path routing paradigm, its coverage properties complements the fault tolerant behavior of multipath routing.

Gandhi et al. propose a spatial sampling approach that exploits the statistical theory of the VC-dimension of a geometric shape to detect physical phenomena characterized by their spatial layout [10]. Since most simple shapes (such as circles, rectangle, and ellipse) have small VC-dimension, their detection also requires relatively less number of samples. Based on this observation, the authors show a remarkable reduction in the number of sensing nodes required to detect events occurring in large parts of a network. In contrast to opportunistic sampling, this spatial sampling approach is limited to networks with a single sink or central point with complete knowledge of exact node locations. Opportunistic sampling is oblivious to individual node positions as long as nodes are distributed uniformly at random within the network area and the sensed phenomena can be described by a spatio-temporal correlation model.

8 Conclusions and future work

The scalability of a WSN is largely defined by its communication efficiency. It is therefore natural in WSNs to discourage global data collection and dissemination operations as they often involve expensive communication among distant nodes. The localized counterparts of traditionally global network operations often require global statistics to guide and optimize their working. In this paper, we propose a novel sampling method for accumulating global statistics at the node level. Based on the multi-hop and multi-path (M2) advantage of the WSN communication paradigm, we observe that user queries being processed in an active WSN can be exploited to gather global statistics at individual nodes. We model the M2 advantage for multi-sink WSNs and show that only a relatively small number of queries are enough to guarantee the collection of accurate global statistics at individual nodes. This

hypothesis is confirmed by the simulation results with opportunistic sampling for global and multi-resolution statistics computation.

This work and our observations from the experiments, presents us with several avenues of future research. First, we observe that the temporal stability of aggregates plays a crucial rule in the accuracy of opportunistic sampling approach. A proactive aggregation scheme, such as TAG in our experiments, can cope with decreasing temporal stability by frequently refreshing the aggregate. Opportunistic sampling, on the other hand, depends on the user queries in this regard. If queries are not issued at a rate high enough to cope with low temporal stability of an aggregate, the accuracy may suffer drastically. In such scenarios, a hybrid approach could opportunistically sample aggregates whose temporal stability matches the expected rate of query arrival; for other aggregates a proactive scheme may be used.

Apart from the rate of query arrival, query (or sink) locations also play a crucial role in maintaining the accuracy of aggregates in opportunistic sampling. These parameters, however, cannot be controlled and may affect the quality of aggregates adversely. In situations where it is crucial to maintain the quality of aggregates for an essential WSN operation, for instance data validation, an artificial sink placement strategy may be employed. In such an application, as soon as M2 coverage for a set of nodes goes below a certain threshold, nodes request one of the allocated sinks to issue an artificial query. If the sink locations are chosen carefully, a few artificial queries can result in increasing the overall coverage of all nodes. The greedy heuristic discussed in Section 3.2 may be used for locating such artificial sinks.

References

1. Bourke P (2012) Area of multiple intersecting circles. http://paulbourke.net/geometry/circlearea/. Accessed Sept 2012
2. Chu D, Deshpande A, Hellerstein JM, Hong W (2006) Approximate data collection in sensor networks using probabilistic models. In: Proceedings of ICDE, p 48
3. Clark I, Harper WV (2000) Practical geostatistics 2000. Ecosse North America, Waterlooville
4. Considine J, Li F, Kollios G, Byers J (2004) Approximate aggregation techniques for sensor databases. In: Proceedings of ICDE, pp 449–460
5. Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction to algorithms. MIT Press, Cambridge
6. Cressie N, Huang HC (1999) Classes of nonseparable, spatio-temporal stationary covariance functions. J Am Stat Assoc 94(448):1330–1340
7. Cressie NA (1993) Statistics for spatial data. Wiley, New York
8. Crossbow Technologies (2009) http://www.xbow.com. Accessed Sept 2012
9. Fewell MP (2006) Area of common overlap of three circles. Technical Note DSTO-TN-0722, Defence Science and Technology Organization, Australia. http://handle.dtic.mil/100.2/ADA463920. Accessed Sept 2012
10. Gandhi S, Suri S, Welzl E (2010) Catching elephants with mice: Sparse sampling for monitoring sensor networks. ACM Trans Sen Netw 1(6):1–27
11. Ganesan D, Greenstein B, Estrin D, Heidemann J, Govindan R (2005) Multiresolution storage and search in sensor networks. ACM Trans Storage 1(3):277–315
12. Gao J, Guibas LJ, Hershberger J, Zhang L (2004) Fractionally cascaded information in a sensor network. In: Proceedings of IPSN, pp 311–319
13. Greenstein B, Estrin D, Govindan R, Ratnasamy S, Shenker S (2003) DIFS: a distributed index for features in sensor networks. J Ad Hoc Netw 1(2–3):333–349
14. Guestrin C, Bodik P, Thibaux R, Paskin M, Madden S (2004) Distributed regression: an efficient framework for modeling sensor network data. In: Proceedings of IPSN, pp 1–10
15. Huebsch R, Garofalakis M, Hellerstein JM, Stoica I (2007) Sharing aggregate computation for distributed queries. In: Proceedings of SIGMOD, pp 485–496

16. Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F (2003) Directed diffusion for wireless sensor networking. IEEE/ACM Trans Netw 11(1):2–16
17. Intel Lab Data (2010) http://berkeley.intel-research.net/labdata/. Accessed Sept 2012
18. Isaaks E, Srivatava RM (1989) An introduction to applied geostatistics. Oxford University Press, New York
19. Kempe D, Kleinberg J, Demers A (2001) Spatial gossip and resource location protocols. In: Proceedings of STOC, pp 163–172
20. Lund C, Yannakakis M (1994) On the hardness of approximating minimization problems. J ACM 41(5):960–981
21. Ma YZ, Jones TA (2001) Teacher's aide: Modeling hole-effect variograms of lithology-indicator variables. Math Geol 33(5):631–648
22. Madden S, Franklin MJ, Hellerstein JM, Hong W (2002) TAG: a tiny aggregation service for ad-hoc sensor networks. SIGOPS Oper Syst Rev 36(SI):131–146
23. Madden S, Franklin MJ, Hellerstein JM, Hong W (2005) TinyDB: an acquisitional query processing system for sensor networks. ACM Trans Database Syst 30(1):122–173
24. Manjhi A, Nath S, Gibbons PB (2005) Tributaries and deltas: efficient and robust aggregation in sensor network streams. In: Proceedings of SIGMOD, pp 287–298
25. Nath S, Gibbons PB, Seshan S, Anderson ZR (2004) Synopsis diffusion for robust aggregation in sensor networks. In: Proceedings of SenSys, pp 250–262
26. Ratnasamy S, Karp B, Shenker S, Estrin D, Govindan R, Yin L, Yu F (2003) Data-centric storage in sensornets with GHT, a geographic hash table. Mob Netw Appl 8(4):427–442
27. Sarkar R, Zhu X, Gao J (2007) Hierarchical spatial gossip for multi-resolution representations in sensor networks. In: Proceedings of IPSN, pp 420–429
28. Silberstein A, Braynard R, Yang J (2006) Constraint chaining: on energy-efficient continuous monitoring in sensor networks. In: Proceedings of SIGMOD, pp 157–168
29. Umer M, Tanin E, Kulik L (2009) Opportunistic sampling in wireless sensor networks. In: Proceedings of ACM SIGSPATIAL GIS 2009, pp 492–495
30. Yao Y, Gehrke J (2002) The Cougar approach to in-network query processing in sensor networks. ACM SIGMOD Record 31(2):9–18

Muhammad Umer is a PhD graduate from the Department of Computing and Information Systems (CIS) at the University of Melbourne. His research is mainly focused on developing localized and adaptive algorithms for query processing in sensor networks using methods from spatial statistics domain. He holds industrial experience in wearable wireless sensor technologies for medical applications and mobile application development ranging from asset management and tracking to distributed enterprize systems. His work at Melbourne University was partly funded by Victoria Research Lab of the National ICT Australia (NICTA). He was also associated with Sensing, Ubiquity, Mobility (SUM) research lab at the CIS department. He has published several papers in refereed conferences and regularly serves as external reviewer for international conferences including ACM GIS, Database Systems for Advanced Applications (DASFAA) and Australasian Database Conference (ADC).

Egemen Tanin research areas include spatial databases and distributed data management. He has finished his PhD at the University of Maryland at College Park on accessing and browsing large databases over the Internet in 2001. His work to access large data such as satellite images was later used by the Global Change Master Directory of NASA. During this time, he has also worked as a software engineer developing one of the first distributed object infrastructures, Cybele. Afterwards, he has focused on spatial data and accessing large spatial databases over the Internet. He has developed the APPOINT (Approach for Peer-to-Peer Offloading the INTernet) system to help users of the Internet efficiently access large spatial data in a distributed fashion. He has joined the University of Melbourne in 2003, where he was awarded two Early Career Researcher Grants for his work on developing indices and algorithms for accessing spatial data over distributed environments in a decentralized manner. He developed the first P2P spatial index in collaboration with researchers at the University of Melbourne and at the University of Maryland. He is currently on the program and organizing committees of multiple international conferences and workshops and has presented various invited talks in the area of spatial and distributed data management. Dr Tanin is a Senior Lecturer in the Department of Computing and Information Systems (CIS) at the University of Melbourne.

Lars Kulik overall research goal is to develop a theory of spatial information for building pervasive computing systems that anticipate, adapt and respond to the needs of users, and provide services based on the users' location and context. Specifically, his research focuses on efficient algorithms for moving objects, information dissemination algorithms in sensor networks, and spatial algorithms in pervasive computing environments. He also researches negotiation-based models for location privacy, and robust algorithms that cope with imperfection, especially in the context of mobile and location-aware computing. Dr Kulik is an Associate Professor in the Department of Computing and Information Systems (CIS) at the University of Melbourne.