gMAP: Efficient Construction of Global Maps for Mobility-Assisted Wireless Sensor Networks

Abdelmajid Khelil, Faisal Karim Shaikh, Azad Ali, Neeraj Suri
Technische Universität Darmstadt
Dependable, Embedded Systems and Software Group
Hochschulstr. 10, 64289 Darmstadt, Germany.
{khelil,fkarim,azad,suri}@informatik.tu-darmstadt.de

***Abstract*—Wireless Sensor Networks are seeing increasing usage in several applications such as military, rescue and surveillance scenarios. Typical for such scenarios is that mobile nodes cooperate side-by-side with stationary sensor nodes to monitor the area of interest and to support the core network operations such as data transport. Global maps of the sensor field, such as temperature and residual energy maps, are of high interest for both users and network designers. However, the map construction can become very inefficient if it requires an extensive intervention of the resource-limited sensor nodes. In this work, we present gMAP, an extremely efficient mobility-assisted approach to construct global maps. In gMAP (a) sensor nodes do not need to process readings of other nodes and (b) require to communicate a minimal number of messages compared to all existing approaches. This is achieved by opportunistically exploiting node mobility to collect data of interest, keeping sensor nodes transmit only their own readings on-demand to a mobile node in their transmission area.**

I. Introduction

Wireless Sensor Networks (WSN) represent networked autonomous embedded systems. With diversity as a key hallmark, WSNs often comprise computing nodes with heterogeneous communication, sensing, processing and storage capabilities. WSNs can be embedded in varied indoor and outdoor environments with the primary goal of sensing, monitoring and detecting phenomenons of interest in battle field, disaster area, wild- and sea-life etc. In such scenarios mobility is inherent as mobile nodes cooperate with stationary sensor nodes to support the core functionality and operations.

A *map* is an aggregated view on the spatial raw samples of a chosen attribute at a specific time. The attributes of interest are system properties (residual energy, connectivity etc.) physical world characteristics (temperature, humidity etc.) leading to network maps (nMAPs) and user maps (uMAPs) respectively. Maps transform the less comprehensible raw data into an information which is understandable for WSN users, designers and operators. Global maps are of high interest for the design, reconfiguration, deployment and maintenance of a WSN [1].

One key map is the *energy map* (E-map for short) which depicts the spatial distribution of residual energy of the WSN elements. An E-map partitions the WSN area into regions containing nodes of similar residual energy, i.e., regions of homogeneous energy density. E-maps provide elementary utilities/support for (1) network design, (2) network functionality and (3) network management. The E-map is valuable for the evaluation and optimization of the energy consumption since it can be used to compare protocols with respect to their energy efficiency/needs and to identify suspicious energy drains etc. The E-map can be utilized to enhance network functionality, e.g., to re-route data traffic to avoid energy-weak regions. An example of supporting the network management is to utilize the E-map to detect and predict important vulnerabilities such as network partitioning.

A variety of inter-node communication based approaches have been developed to build global maps (G-maps) [2]–[10]. However, all these approaches rely on multi-hop communication and/or in-network aggregation, which overstrains the stationary sensor nodes through the use of their limited energy and processing resources. In [11], the authors demonstrated that node mobility can increase the capacity of ad hoc networks, if the mobile nodes transport the message closer to the destination instead of immediately using multi-hop communication. This comes at the cost of higher end-to-end delays for communication. Several WSN applications and network management tasks can tolerate delays in the range of minutes, hours or even days [12]–[15]. Given normal operational conditions the energy distribution in the sensor field changes only on a relatively large time scale. Therefore, the data collection for constructing and updating the E-map can last longer, e.g., some hours. In this paper, we focus on attributes that do not change suddenly or radically in magnitude and develop novel delay-tolerant algorithms for collection of map data. Our approach that we refer to gMAP, opportunistically, exploits the mobility of some nodes for this collection. We show that gMAP is highly efficient compared to approaches that base on in-network aggregation. The gMAP approach presents also a novel technique to construct G-maps from the collected data. In this paper we focus on the example of E-map. However, our algorithms are tunable and can be easily adopted to create further maps such as temperature/humidity maps, provided that the considered attribute has a long time relevance, i.e., changes slowly over time.

The paper is organized as follows. After the discussion of related work in Section II, Section III presents our system model. Section IV presents our novel gMAP approach using

Research supported in part by DFG GRK 1362 (TUD GKMM), EC CoMiFin, EC INSPIRE and MUET

the example of E-maps. In Section V we evaluate our approach and compare it to related work. We conclude the paper and give some directions of future work in Section VI.

II. Related Work

In cartography [16], isoline (also isopleth or contour) and choropleth are the common types of maps. Isoline maps are based on the assumption that, the phenomenon represented has a continuous distribution and smoothly changes in value in all directions of the plane. Contrary to isoline maps, choropleth (or density shading) maps usually give a better impression of spatial distribution. Therefore, the use of choropleth maps is proposed for gMAP.

The naive approach to collect raw data for map construction would be if each node reports its value to the sink using multi-hop communication. This is obviously inefficient. Consequently, more efficient approaches have been developed based on techniques such as in-network aggregation [6]–[8]. Other approaches use suppression mechanisms to reduce the number of nodes reporting their raw readings to the sink [9], [10], [17].

A. Aggregation-based Approaches

eScan [6] and isobar [7] are approaches based on polygon aggregation. First, a request for energy values is flood to all network nodes. This constructs an aggregation tree that can be used to aggregate the energy values while being reported by each node. The aggregation consists in grouping sensor readings that meet a certain criteria (being geographically adjacent and in the same value range). The outcome of the aggregation is a list of (spatial) regions. A region is a polygon that is defined by the line spanning its border nodes. At the sink the aggregation results in an energy map delivered to the user. The approach assumes that all nodes know their exact position, which is a strong requirement on the WSN. Each sensor propagates its position along with the energy values for aggregation purposes. Furthermore, the sensor nodes (even those that have critical residual energy level and especially those closer to the sink) are main actors in map construction, leading to higher processing and communication activities and subsequently to a serious degradation of the network lifetime and disturbance of the core functionality.

INLR [8] is an aggregation-based approach that focuses on small scale WSNs. A sensor node sends its reading or the calculated aggregate not only to its parent in the aggregation tree but to all its neighbors that are 1-hop closer to the sink. Therefore, nodes possess a partial map and the sink the global map (choropleth). The knowledge of sensor node locations is needed at the sink. While using more than one parent increases the accuracy of the map, the efficiency is sacrificed.

B. Suppression-based Approaches

Isoline [9] is an approach based on localized isocluster aggregation. The map building is reduced to the detection of isolines. Neighboring nodes share their readings. A node compares its reading with the readings of all neighbor nodes and detects an isoline, when the readings lie in different sides of a globally defined isoline. The detection of an isoline needs to be reported to the sink by the closest neighbor to the sink. The isocluster aggregation outperforms polygon aggregation in terms of accuracy with minor energy savings. The locations of all sensor nodes are needed at the sink.

Meng et al. [10] motivate the use of contour (isoline) maps for efficient continuous monitoring in sensor networks. The main contribution of this paper is the design of a temporal and spatial local suppression mechanism that prohibits some nodes to report their readings. The number of saved reports highly depends on the spatial correlation between sensor readings. Sensor nodes report their readings using multi-hop routing without any in-network processing. The map is constructed on the sink using interpolation and smoothing techniques. The sink has knowledge about the position of all network nodes.

Iso-Map [17] also does not rely on in-network processing. It uses a suppression mechanism to reduce the number of nodes that report their readings to the sink using multi-hop communication. This approach is very similar to that of Isoline [9]. However nodes need to report the gradient direction of the isolines, which requires excessive processing on sensor nodes.

C. Our Contributions Compared to the Existing Approaches

Our gMAP approach uses a *minimal* number of messages without sacrificing the completeness of sensor information. This provides for high efficiency with respect to both energy and bandwidth consumption. The gMAP approach does not require that sensor nodes necessarily know their location. In gMAP we *decouple* the collection of the sensor values from the construction of the map, which results in minimal processing on sensor nodes reducing the energy consumption on them. Furthermore, gMAP charges all sensor nodes similarly and contributes to the desired energy balancing in WSNs. As mentioned earlier, our approach is independent from the map's attribute and can be used for further maps with minimal changes. In fact many maps can be generated simultaneously with a minor modification, i.e., nodes piggyback the different attributes values and reply with one message. This only leads to a slightly larger communication overhead. Also the gMAP's overhead can be shared between several instances interested for different maps. Our approach is *resilient to network partitioning*, which increases the dependability of the WSN, since monitoring tasks can continue reporting the health of the network even if critical failures/situations occur. gMAP *replicates* maps onto the sink and mobile nodes which increases their availability for sensor nodes. Nodes can postpone querying relevant global information till they encounter a mobile node. By this way not only the map construction is highly efficient but also its use.

III. System Model

In this work, we consider the established *mobile* Wireless Sensor Network (mWSN) model. This model is used in a variety of WSN deployments, in particular in emergency and military scenarios. The main functionality of the mWSN is implemented by a large number of stationary resource-limited *sensor nodes (SN)* that are deployed following either an arbitrary or structured spatial distribution in the area of interest. Also one dedicated stationary sink is selected as the interface to the user. Additionally, a few mobile *assist nodes (AN)* are deployed with generalized support roles such as (1) application support (e.g., additional interface to users), (2) functionality support (e.g., delay-tolerant data transport), and (3) network support (e.g., diagnosis). The mobile nodes cover a functional capability spanning robots, unmanned air vehicle (UAV), etc. In this paper, we consider a mWSN composed of $N-1$ SNs, with one sink and one mobile AN.

We consider two major classes of mobility: Structured mobility, i.e., predictable & controllable, and unstructured mobility, i.e., unpredictable & uncontrollable. The AN possesses high processing, storage and energy capabilities compared to SNs. Furthermore, it has no energy limitations because it can recharge its batteries by means of on-board renewable energy resources [18] or through moving to recharging energy-stations. We assume that SNs use the batteries as a main energy source. These batteries continuously discharge following a long-running process in the range of months or even years. We consider that AN knows its position. For the SNs we consider both cases: Either they know their position or not. We assume that all deployed nodes are cooperating and that no misbehaving nodes may exist.

For simplicity, we consider all nodes (AN and SNs) are equipped with a conformal level of communication technology and are able to communicate if they are in each others transmission range R. We use a CSMA/CA based MAC layer, where communication links are symmetric and bidirectional, and collisions may occur. Furthermore, we assume that network can get partitioned, i.e., some SNs may not be able to communicate with the sink. We allow for the use of duty cycles for SNs. However, we assume that the magnitude of the movement distance covered by the AN during the time period of a duty cycle is negligible and that the duty cycles scheduler assures that all SNs in the AN's transmission area eventually receive the messages sent by the AN.

IV. eMAP: Mobility-Assisted Energy Map Construction

We now present our novel gMAP approach comprising new algorithms to collect samples in a mobility-assisted way and a new technique to construct maps. We use the E-maps as an example, however, our methodology is generic and can be easily adopted for other functionality maps. We refer to our gMAP approach for E-maps by eMAP.

A. Overview of Approach

The main reasoning behind the eMAP approach to construct E-maps is that battery depletion occurs over an extended period of time and it is sufficient to check the battery level at a daily or weekly basis. This shows that collection of energy-based health indications is a *delay-tolerant* process, which allows us to deploy established concepts from the delay-tolerant networking research. Accordingly, the main design principle for the eMAP approach is to exploit the mobility of nodes to transport messages and collect information in a delay-tolerant way, thus reducing the communication overhead.

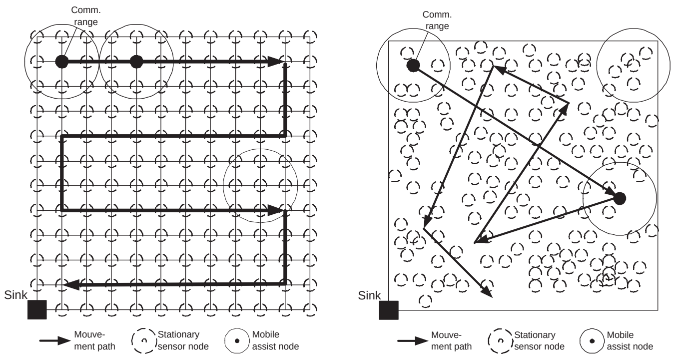
We let the mobile AN scan the sensor field and collect the energy information from each node it encounters. We are using one single mobile AN for simplicity of communicating the idea whereas a real implementation can consider multiple nodes or some primary/secondary arrangements. The AN sends a short beacon, on which nodes reply with their energy value and optionally their position. We proceed progressively, by first considering a structured scenario and then an unstructured one. For each scenario we consider both possibilities, whether stationary SNs know their position or not, and design appropriate algorithms to collect energy information. Overall, we perform the following three steps. First, we develop a sampling algorithm for a structured scenario, i.e., SNs are located according to a pre-known topology (e.g., the grid topology) and the mobility of the AN is predictable and controllable. Second, we consider the same structured scenario, however the SNs now do not know their positions and, therefore, reply only with their residual energy values. Third, we consider an unstructured scenario, where the mobility of the AN is neither controllable nor predictable, and also treat both cases concerning the position knowledge for SNs.

We also present an efficient technique for the mobile AN to locally construct an appropriate E-map from the collected energy samples. The technique is based on measuring inequalities between neighboring samples and to group similar values into a region. Therefore, we refer to our technique by regioning (Section IV-C).

B. Energy Information Collection

We first classify the topological mWSN scenarios and then present for each class a suitable information collection algorithm.

1) Scenario Classification: In this work, we focus on two extreme types of scenarios: Structured and unstructured. They provide basic features to build realistic scenarios. In the structured scenario we assume that the spatial deployment of SNs is known a priori and that the mobility of the AN is predictable and controllable (Fig. 1 (a)). Without loss of generality, we consider the grid topology with a cell-size c. We set the communication range to $R = \sqrt{2} * c$. In the unstructured scenario the topology is unknown (e.g., random) and the mobility of the AN is unpredictable and uncontrollable (Fig. 1 (b)). We selected the commonly used random waypoint mobility model as its high randomness maximizes the unpredictability. Our main driver for the scenario selection is the proof of concept in extreme scenarios. Furthermore, in a realistic scenario the spatial deployment of SNs can be partially structured and partially unstructured. The mobility of the AN can be either controllable or uncontrollable and may follow varied patterns.



(a) Structured ($\sqrt{N}$x$\sqrt{N}$ grid) (b) Unstructured

Fig. 1: Basic scenarios

2) Structured mWSN Scenarios: We now present a simple algorithm to plan the movement of the AN and two algorithms to efficiently collect the energy information of SNs.

We adopt a simple movement algorithm to control the mobility of AN (Fig. 1(a)). The AN traverses the whole area from top to bottom. It starts from top corner and continues its movement straight forward till it reaches the border of the sensor field. Then the AN moves downwards, changes the direction and moves straight forward. This movement pattern will continue till the AN encounters all SNs. After moving $2 * R$ distance the AN pauses for a small duration t_o to collect information and than moves onwards. For arbitrary deployments, we will rely on the mature discipline of motion planning and control. This discipline provides varied algorithms for different environmental and functional constraints [19] [20]. An optimized movement algorithm should generate the shortest movement path while covering all SNs and taking into consideration the size of the sensor field and the communication range R. One may also require that the movement path ends at the sink in order to minimize the communication overhead for transmitting the E-map to the sink. We note here that for a given sensor field, the movement speed of the AN directly affects the duration of the energy sampling operation. Accordingly, the speed of the AN can be adaptively fixed according to the desired sampling latency.

For the structured scenario, where SNs know their location, we present the following data collection algorithm. The AN performs a first *snapshot* by sending a REQ-beacon to all SNs in its transmission area using a MAC broadcast. A SN replies by sending a message containing its node-ID, location (loc) and energy level E_{lev}. In order to reduce collisions, nodes schedule their reply for a random time t_{rand} between 0 and a maximum value T_{max}. The AN performs the subsequent snapshot after moving for a distance of $2 * R$, or on changing the movement direction but after moving for a distance of $3*c$. The optimal result of the collection operation is a set of $N-1$ elements with the following structure: {node-ID, loc, E_{lev}}.

If SNs do not know their location, the AN can not assign for each sample its accurate location. Since it is necessary for the map construction that each sample is associated with a location, we have to approximate the sample location, i.e., interpolation is needed. One possibility is to assign the position of the mobile node at the time it initiated the snapshot to all the samples of the corresponding snapshot. This leads to a high concentration of the energy on the position of the AN. Another possibility is that the AN assumes a certain spatial distribution of nodes (e.g., uniform). Accordingly, the AN can assign the received energy samples to selected positions in its transmission area.

3) Unstructured mWSN Scenarios: In an unstructured scenario the movement of the AN is neither controllable nor predictable. We assume that the mobility of the AN eventually covers the entire sensor field.

In the following, we present our algorithm to collect energy information (Alg. 1). If the AN performs a snapshot, moves $2 * R$ away without changing the direction, and performs a second snapshot, then both snapshots are covering disjunct areas. Subsequently, we let the AN perform a second snapshot, only after moving $2 * R$ from the location of the previous snapshot. The information collection completes, when the total WSN area is covered by all snapshots. We note that if the AN changes its movement direction, then the snapshots overlap and some nodes may receive redundant REQ beacons. The major concern for SNs is to minimize the number of messages to be sent or received. The AN is powerful enough to send REQ beacons frequently. However the REQ beacons are received by energy precious SNs. Therefore, we have to minimize the number of unnecessary REQ messages sent by AN. To avoid unnecessary snapshots, the AN maintains a history of snapshots $\{snapshot_{id}, snapshot_{loc}\}$. After moving $2 * R$ from the location of the previous snapshot and before performing a second snapshot, the AN uses the history to calculate if the second snapshot has an additional coverage higher than a fixed threshold coverage $COV_{th}\%$. Only in this case the AN performs a snapshot. The value of $COV_{th}\%$ allows to investigate the trade-off between the number of redundant REQ beacons and the sampling latency. Once the AN scans the whole sensor field, the history of snapshots will be flushed and a new round will be initiated by the AN. To avoid unnecessary transmissions, SNs send information only once in a round as presented in Alg. 1.

C. E-Map Construction

The prime goal of the map construction is to identify inequalities of energy density. Expected is an E-map that divides the sensor field into regions, which are indicators of similar energy-densities. The input of the construction algorithm is the collected residual-energy information and the output is the map's regions. The E-map is a geometrical/spatial data structure (e.g., tree) which is easy to evaluate. The construction operation has to satisfy some crucial requirements. First, it should be easy to evaluate on the AN. Second, two neighboring regions should have two "sufficiently" different energy densities. The map construction process is composed of the spatial partitioning of the sensor field (*space partition*)

Algorithm 1 Collection Algorithm for Unstructured mWSN

```
/*****  On Assist Node (AN)  ******/
var HIST_snapshot
Initiate a new round round_id for sampling
AN: Do first snapshot: SEND REQ beacon with round_id
AN: STORE {snapshot_id, AN_loc} in HIST_snapshot
snapshot_id ++
If AN has moved a distance of 2*R since previous snapshot do:
AN: CHECK HIST_snapshot
AN: Compute COV_additional from current AN_loc and HIST_snapshot
if COV_additional ≥ COV_th then
    AN: SEND REQ beacon with round_id
    AN: STORE {snapshot_id, AN_loc} in HIST_snapshot
    snapshot_id ++
else
    AN: Suppress REQ beacon
end if
AN: RECEIVE E_msg
AN: GOTO 7
/*****  On Sensor Node (SN)  ******/
SN: RECEIVE REQ beacon
if SN: new round then
    SN: Schedule transmission between 0 < t_rand < T_max
    SN: SEND E_msg {ID, SN_loc (optional), E_lev}
else
    SN: Suppress SEND E_msg
end if
```

and the fusion of the regions of similar residual energy values (*regioning*).

For space partitioning, Voxel grid, triangulation (e.g., Voronoi or Delaunay), octree, k-d tree and BSP tree [21] can be used. All these schemes, except the Voxel grid are dependent from the input data. For this reason we select the simple Voxel grid for space partitioning. The primitive parameter to divide the sensor field is the size of smallest fragment of area, i.e., grid-cell size or the partitioning *resolution* (r). The energy density in the cell is the basis to form a region. Selecting r is a crucial decision for creating the E-map. Depending on this resolution a cell may contain more than one SN. We refer to the residual energy value of one cell by the sum of the energy values of all the nodes in that cell.

In order to merge the cells into regions (regioning), we need to ascertain if neighboring cells have similar values for residual energy. For this we need a technique to decide if two neighbor cells can be merged or not. A first possible technique is to use a metric to measure the inequality between two neighboring cells. In the literature we identify several inequality indices [22] that measure the inequality of a set: Variance, entropy coefficients, Hoover coefficient, Coulter coefficient, Gini coefficient etc. A second technique is to use global classes. In the eMAP approach we rely on the class-based technique for its simplicity and easy evaluation on ANs. Furthermore, we are investigating the suitability of other indices in ongoing work. The cells are classified into a fixed number of classes depending on their energy density. Neighboring cells are merged into the same region if they belong to the same class.

In Alg. 2 we propose the pseudocode of our regioning algorithm. This algorithm is based on searching and is inspired by the region growing algorithm for image segmentation [23]. We assign a region-ID to any cell to start regioning. Then we check if neighbor cells can be merged with this cell. When we merge cells we assign the same region ID to them. Once the neighbor cells are checked for merge, we will repeat the process of cell merging for the neighbors that have been successfully merged to the current region. After completing regioning for the starting cell, all the other cells (which are not assigned to a region) will form regions in the same way. Hence we complete regioning for the whole WSN field.

Algorithm 2 Regioning

```
cell := structure{cellID, neighborList[], regionID=-1, energyClass}
grid := array of all cells
var currentRegionID
var currentRegion:= array of all cells with regionID=currentRegionID
/*********    regioning ()    *********/
regioning()
for each cell_i ∈ grid do
    if cell_i.cellID > −1 then
        next iteration
    end if
    cell_i.regionID=currentRegionID;
    regionMaking(cell_i)
    for each cell_j merged with the region of cell_i do
        regionMaking(cell_j)
    end for
    currentRegionID++
end for
/*****  regioning with the 8 neighboring cells  ******/
regionMaking(myCell)
for each neighborCell do
    if (myCell.energyClass = neighborCell.energyClass) then
        neighborCell.regionID = myCell.regionID
    end if
end for
```

We observe a trade-off between the *accuracy* and comprehensibility of the constructed map. The accuracy of the E-map depends on the accuracy of energy information collection and on the accuracy of regioning. Regioning accuracy is important to comprehend node distribution. If the model provides such a map that only the neighboring cells with the same energy-level form regions, it becomes the most accurate region map. It would be a worse map if regions consist of cells with highly different energy densities.

The selection of the number of energy classes is crucial since it allows for tuning the trade-off between map accuracy and comprehensibility. It should take into account the range of possible values and the level of inequality tolerated for regioning. Selecting a higher number of classes provides for higher map accuracy on the one hand but hardens regioning and subsequently the comprehensibility of the map on the other hand. A lower number of classes sacrifices accuracy in order to provide for a better comprehensibility. However, if the number of classes is too low, we merge cells with high difference in energy densities. Thus, regioning weakly reflects the energy spatial distribution. This results in an erroneous map. Summarizing, the number of classes should be appropriately selected to provide for the required trade-off between accuracy and comprehensibility.

V. Evaluation

In this section, we compare the message complexity, processing complexity and data completeness of gMAP to that of the existing approaches. We further provide simulation results for map accuracy and comprehensibility.

A. Comparison to Related Work

We compare the performance of the gMAP approach with that of the naive (NAIVE), aggregation-based (AGG) and suppression-based (SUP) approaches. We are mainly interested in the message and processing overhead on SNs, since this determines their energy consumption concerning map construction. For the sake of simplicity we consider the structured scenario with a $\sqrt{N}$x$\sqrt{N}$ grid topology, the sink in the corner and a communication range of $R = \sqrt{2} * c$ as depicted in Fig. 1(a) to derive analytical results.

In the NAIVE approach every node sends one message directly to the sink through multihop *unicast*. We assume a perfect shortest path routing for simplicity. Relying on Fig. 1(a), one node that is located on the border of the WSN $k * c$ far away from the sink requires k transmissions to send one message via multihop. For the grid topology, there are $2k + 1$ nodes that need k transmissions to send a message to the sink. Therefore, the total number of message transmissions needed for the NAIVE approach is computed as follows:

$$\begin{aligned} \#MSG_{NAIVE} &= \sum_{k=1}^{\sqrt{N}-1} (2k+1)k \\ &= 2 * \sum_{k=1}^{\sqrt{N}-1} k^2 + \sum_{k=1}^{\sqrt{N}-1} k \\ &= 2\frac{(\sqrt{N}-1)(\sqrt{N})(2\sqrt{N}-1)}{6} + \frac{(\sqrt{N}-1)(\sqrt{N})}{2} \\ &= \frac{1}{3}(2N^{1.5} - N + N^{0.5}) \end{aligned}$$

Therefore

$$\#MSG_{NAIVE} = O(N^{1.5}) \tag{1}$$

The AGG approaches are based on *convergecast* communication. The message complexity of the in-network aggregation depends on the aggregation level of reported data, i.e., how many nodes aggregate the received data messages and how many nodes just forward them. According to [24] the upper and lower bound for the message complexity of aggregation, and consequently of AGG approaches, is given by:

$$N - 1 \leq \#MSG_{AGG} \leq (N-2) * d + 1 \tag{2}$$

where d is the diameter of the network graph. These bounds are independent from the topology and from the location of the source. For the structured scenario (Fig. 1(a)), obviously $d = \sqrt{N}$. Therefore, an upper-bound for the AGG message complexity is $\sqrt{N}(N-2) + 1$.

In SUP approaches nodes behave like the naive approach, however some nodes are prohibited to send their messages to the sink. Depending on the efficiency of suppression a fraction of N, say $x\%$, of nodes will not transmit their messages. Taken this into account and given Eq.(1) the optimal complexity of SUP approaches (conform to [17]) is:

$$\#MSG_{SUP} = O(N^{0.5}) \tag{3}$$

The communication model of our approach is single hop *unicast*. Assuming perfect suppression of redundant energy transmissions, each SN sends one single message for the mobile node, the message complexity of gMAP is then exactly:

$$\#MSG_{gMAP} = N - 1 \tag{4}$$

We note that if gMAP would use the spatial suppression mechanisms provided by SUP approaches the message complexity becomes the minimal among all existing approaches. However spatial suppression would sacrifice the data completeness.

The message size for the gMAP, SUP and NAIVE approaches remains constant during the collection of energy information. However the AGG messages get larger as they are transported towards the sink. The result is that the AGG approaches lead to higher energy consumption on nodes closer to the sink, which results in an unbalanced energy consumption in the network. Furthermore, the closer the nodes to the sink, the more complicated the aggregation operation becomes, which leads to even more overuse of the nodes closer to the sink. SUP approaches rely on dedicated nodes that report their readings to the sink, this also leads to unbalanced energy consumption among nodes. The gMAP approach provides however for a balanced use of resources on nodes with respect to energy and processing.

AGG has an additional overhead to construct and maintain the aggregation tree. SUP requires an additional overhead to select the reporting nodes. Also gMAP has a supplementary overhead resulting from the short beacons sent by the AN. The gMAP related overhead for the SNs consists only in receiving these beacons. However, we designed a suppression mechanism that minimizes the number of unnecessary snapshots.

We rely on the results in [17] and the above discussion to summarize in Table I the message complexity, processing complexity and the collection completeness for the gMAP as well as the existing approaches.

The investigation above proves that the gMAP approach provides for the minimal processing complexity and for a relatively low message complexity while keeping the data completeness maximal. This comes out at the cost of higher end-to-end latency, which is not a concern for the delay-tolerant energy-information collection.

B. Simulation

We first describe our simulation settings. Then we define the evaluation metrics based on which we present our results.

Approach	Message complex.	Process. complex.	Max. data completeness	Latency
isobar [7]	$O(N)$	$O(N)$	$< 100\%$	$\sim sec$
eScan [6]	$O(N^{1.5})$	$O(N^4)$	100%	$\sim sec$
INLR [8]	$O(N^{1.5})$	$\Omega(N^{1.5})$	100%	$\sim sec$
Isoline [9]	$O(N^{0.5})$	$O(N)$	<100%	$\sim sec$
Meng [10]	$O(N)$	$\Omega(Nd)$	<100%	$\sim sec$
Iso-Map [17]	$O(N^{0.5})$	$O(N)$	<100%	$\sim sec$
gMAP	$N-1$	$O(1)$	100%	$\sim hr$

TABLE I: Comparison with existing approaches

	Structured	Unstructured	
		COV_{th}=70%	COV_{th}=90%
Completeness	94.66%	88.4%	81.3%
Efficiency	1.0	1.0	1.0
#Snapshots	25	61	43
Latency [min]	5	40.71	39.25

TABLE II: Simulation results for gMAP

1) Simulation Settings: We use Tossim [25] and its Tython extension for network simulations, and Matlab for map construction. Tossim is an event-driven simulation tool widely used in the WSN community. We have used the disc radio model provided by Tossim with 5 units communication range. All nodes lying in this communication range communicate without errors and have symmetric links. Although collisions may occur. We have considered 225 SNs either generated randomly in the area of 42 unit x 42 unit or in a grid topology of 15 x 15 (cell size $c = 3$ units) including sink. The AN moves either in controlled fashion (with pause time of $t_0 = 3sec$) or according to the random-waypoint model using a constant speed of 1 $unit/sec$. SNs use $T_{max} = 500ms$ as a maximum time to schedule their replies to REQ beacons.

Nodes initially have energy values following an arbitrary distribution. In this work we use the distribution depicted in Fig. 2(a). The choice of space partitioning resolution (r) is critical. Intuitively a good choice is $D \leq r \leq R$, where D is the average distance between two neighboring SNs. For the structured as well as the unstructured scenarios (225 nodes and 42 x 42 units area), simulations with different r values showed that $r = 3$ $units$ allows for the most comprehensive E-map. This is about the average distance between two neighboring SNs in the structured scenario.

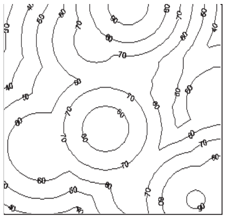
2) Evaluation Metrics: The performance of constructing global maps is commonly measured with respect to its completeness, efficiency and regioning accuracy.

- *Collection Completeness*: The ratio of nodes whose values are collected by the AN to the total number of SNs.
- *Collection Efficiency*: To measure collection efficiency we consider the number of energy messages per SN, i.e., the ratio of total number of energy messages sent by SNs to the number of SNs that received a REQ-beacon. We also consider the number of snapshots as overhead since it implies the reception of beacons by the SNs.
- *Region Accuracy*: To evaluate the regioning accuracy we compare the E-map constructed by the AN with the perfect E-map, i.e., the map constructed from complete energy information.

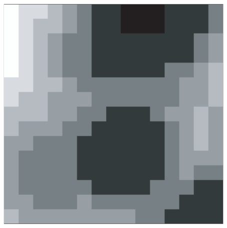
3) Results: The results of our simulations are summarized in Table II. We observe that the completeness of energy information is close to but lower than 100% in both scenarios. In structured scenarios, this is due to MAC collisions. In unstructured scenarios, besides collisions, mobility leads to fast topology changes and therefore to additional message loss. We observe that lower COV_{th} values provide for higher completeness at the cost of higher number of snapshots. This is due to the fact that if COV_{th} is increased higher overlaps between snapshots is tolerated. After sufficient number of snapshots, the additional coverage will not be able to be higher than COV_{th} and no further snapshots are possible although some nodes have not received a REQ beacon. This results in higher efficiency (limited number of snapshots) but the completeness of collected information may suffer. The efficiency is 1.0 in both scenario, given the fact that nodes that receive a REQ beacon respond with one single message per collection round, irrespective of AN received it or not. This shows the reliability of our temporal suppression mechanisms that is based on rounds. The latency of gMAP is as expected in the range of minutes to hours. The latency for data collection in unstructured scenarios is higher than that in structured scenarios. This is due to the fact the movement of the AN is random implying that more time is needed to cover the sensor field. Generally, these simulation results confirm the analytical results presented in last section and prove the gMAP efficiency.

Given that energy levels are between 0 and 100%, and as it is likely to tolerate 10% difference within the single region, we use for regioning 10 classes of energy levels. Fig. 2(a) shows the isolines of the considered energy distribution. We use this map as a reference and compare different E-maps generated by our approach. In Fig. 2(b) we show the perfect choropleth E-map of the structured scenario. Obviously choropleth is more expressive and comprehensive than the isolines. In Fig. 2(c)-(d) we show the E-maps constructed by the eMAP approach for structured scenarios. If nodes reply with their exact positions, the E-map (Fig. 2(c)) is very similar to the perfect map (Fig. 2(b)). The difference is due to the incompleteness of data which is presented by white cells in the E-map. This proves the high accuracy of our regioning algorithm. If nodes do not know their positions, the accuracy of the E-map suffers (Fig. 2(d)). Although the regioning algorithm detects the major regions which are comparable to Fig. 2(a). Overall the resolution of the E-map is low since we are using a low number of nodes.

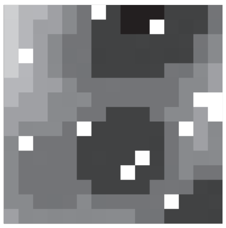
For unstructured scenario the perfect E-map is shown in Fig. 3(a). This perfect map looks different from Fig. 2(b) due to the fact that in this scenario the nodes are not uniformly distributed, which results in holes in the perfect map. Fig. 3(b)-(c) represent the E-maps constructed by our approach for $COV_{th} = 70\%$ and for the considered unstructured scenario



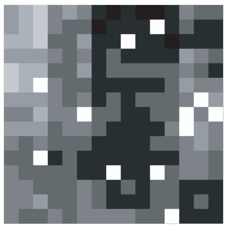
(a) Perfect isoline E-map



(b) Perfect Choropleth E-map



(c) Structured with location



(d) Structured without location

Fig. 2: Accuracy of eMAP (structured scenario)

(a) Perfect E-map

(b) Unstructured with location

(c) Unstructured without location

Fig. 3: Accuracy of eMAP (unstructured scenario)

(worst case). If nodes reply with their exact locations the constructed E-map is depicted in Fig. 3(b). Despite node density variations the regioning algorithm detects the major regions which are comparable to Fig. 3 (a). The difference is due to the incompleteness (88.4%) of energy information. If nodes do not know their locations (Fig. 3 (c)), the accuracy of the E-map decreases. The magnitude of this degradation is a function of the ratio of communication range to the sensor field size. As G-maps are more suitable for large scale WSNs, we are convinced that gMAP will provide for high accuracy also in the case where nodes do not know their locations.

VI. Conclusion and Future Work

We have presented gMAP, an extremely energy-efficient methodology that collects data of interest from the WSN and presents its geographical distribution as a map. Our approach is opportunistic as it exploits existing node mobility to collect data. Being mobility-assisted the collection process lasts for the time that mobile entities need to scan the whole sensor field. Therefore, data should be of high time relevance, i.e., do not change suddenly or radically in magnitude. As an example we focussed on the map of residual energy (E-map) since the battery depletion is a long-running process. Considering two extreme scenarios, i.e., structured and unstructured, we showed the efficiency and the accuracy of our gMAP approach.

In ongoing work, we are investigating update strategies to keep the map consistent over the network lifetime. For a generalized WSN scenario, we are convinced that aggregation-based, suppression-based and mobility-assisted strategies must co-exist to provide for high efficient and accurate global maps. We also want to use the global maps to enhance functionality and dependability of WSN. In particular we are investigating the use of energy maps to predict network partitioning.

References

[1] A. Khelil et al. MWM: A Map-based World Model for Wireless Sensor Networks. In *ACM AUTONOMICS*, 2008.

[2] Y.J. Zhao et al. Sensor Network Tomography: Monitoring Wireless Sensor Networks. In *Student Research Poster, ACM SIGCOMM*, 2001.

[3] R.A.F. Mini et al. The distinctive design characteristic of a wireless sensor network: the energy map. *Computer Communications*, 27(10), June 2004.

[4] R.A.F. Mini et al. Prediction-based energy map for wireless sensor networks. *Elsevier Ad-hoc Networks Journal*, March 2005.

[5] E. Souto et al. Sampling Energy Consumption in Wireless Sensor Networks. In *IEEE SUTC*, June 2006.

[6] Y. Zhao et al. Residual energy scan for monitoring sensor networks. In *IEEE WCNC*, 2002.

[7] J.M. Hellerstein et al. Beyond Average: Toward Sophisticated Sensing with Queries. In *IPSN*, 2003.

[8] W. Xue et al. Contour Map Matching For Event Detection in Sensor Networks. In *ACM SIGMOD*, June 2006.

[9] I. Solis and K. Obraczka. Isolines: Energy-efficient Mapping in Sensor Networks. In *IEEE ISCC*, June 2005.

[10] X. Meng et al. Contour Maps: Monitoring and Diagnosis in Sensor Networks. *Computer Networks*, 50(15), 2006.

[11] M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.

[12] Y. Wang et al. A survey on analytic studies of delay-tolerant mobile sensor networks. *Journal of Wireless Communications and Mobile Computing (WCMC)*, 7(10), 2007.

[13] R.C. Shah et al. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In *SNPA*, 2003.

[14] W. Zhao et al. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *ACM MOBIHOC*, May 2004.

[15] H. Jun et al. Power Management in Delay Tolerant Networks: A Framework and Knowledge-Based Mechanisms. In *SECON*, 2005.

[16] A.H. Robinson et al. *Elements of Cartography*. John Wiley & Sons, New York, 1995. 6th Edition.

[17] Y. Liu and M. Li. Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks. In *ICDCS*, 2007.

[18] P. Dutta et al. Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments. In *IPSN*, 2006.

[19] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[20] L. Lima and J. Barros. Random Walks on Sensor Networks. In *WiOpt*, 2007.

[21] J. Nievergelt and P. Widmayer. Spatial data structures: Concepts and design choices. In *Handbook of Computational Geometry, Elsevier Science Publishers*, 2000.

[22] B.A. Portnov and D. Felsenstein. *Regional Disparities in Small Countries*. Springer, Berlin Heidelberg, 2005. ISBN 978-3-540-24303-8.

[23] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice-Hall, Upper Saddle River, NJ, 2001. 978-0130307965.

[24] B. Krishnamachari et al. Modelling Data-Centric Routing in Wireless Sensor Networks. In *USC Computer Engineering Technical Report CENG 02-14*, 2002.

[25] P. Levis et al. Tossim: Accurate and Scalable Simulation of Entire TinyOS Applications. In *SENSYS*, 2003.