Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 10-004

DECOR: Distributed and Energy Efficient Collection of Raw Data in Sensor Networks

Sarah Sharafkandi, David Du, and Alireza Razavi

February 23, 2010

DECOR: Distributed and Energy Efficient Collection of Raw Data in Sensor Networks

Sarah Sharafkandi, David H.C Du
Department of Computer Science and Engineering
University of Minnesota
Twin cities, MN, USA
Email: {ssharaf, du}@cs.umn.edu

Alireza Razavi
Department of Computer Science and Engineering
University of Minnesota
Twin cities, MN, USA
Email: raza0007@umn.edu

***Abstract*—In wireless sensor networks, collection of raw sensor data at a base station provides the flexibility to perform offline detailed analysis on the data which may not be possible with in-network data aggregation. However, lossless data collection consumes considerable amount of energy for communication while sensors usually have limited energy. In this paper, we propose a Distributed and Energy efficient algorithm for Collection of Raw data in sensor networks called DECOR. DECOR exploits spatial correlation to reduce the communication energy in sensor networks with highly correlated data. In our approach, at each neighborhood, one sensor shares its raw data as a reference with the rest of sensors without any suppression or compression. Other sensors use this reference data to compress their observations by representing them in the forms of mutual differences. In a highly correlated network, transmission of reference data consumes significantly more energy than transmission of compressed data. Thus, we first attempt to minimize the number of reference transmissions. Then, we try to minimize the size of mutual differences. We derive analytical lower bounds for both these phases and based on our theoretical results, we propose a two-step distributed data collection algorithm which reduces the communication energy significantly compared to existing methods. In addition, we modify our algorithm for lossy communication channels and we evaluate its performance through simulation.**

I. Introduction

Collection of raw sensor data at a base station is one of the most prominent yet challenging problems in sensor networks. This is required in a variety of environmental monitoring applications where an offline ad-hoc analysis on the data is desired [1], [2]. A naive approach to this problem is to ask every sensor to transmit the raw data on the forwarding path to the base station, without any suppression. This approach consumes a significant amount of energy especially for the sensors close to the base station which should relay the raw data for the rest of the network. Energy is the major concern for sensor networks since sensors are usually battery operated, and hence energy-constrained. Yet sensors are typically deployed in an environment where they will be left unattended for months or even years and are expected to operate for such long periods. Therefore, to guarantee the desired lifetime of the network, it is essential to consume energy efficiently. Recognizing that computation is less energy consuming than communication [3], it is well known that significant energy savings can be obtained by reducing number of transmitted bits through extra computation.

If a user/application can express its interest as a query, the query will be sent into the network. Based on the query, sensors perform aggregation on their readings to reduce the amount of data that is transmitted to the base station [4], [5], [6], [7]. For example, if in a network, a user is only interested in the minimum or maximum temperature of the sensors, all the redundant sensor observations can be easily dropped on their way to the base station and only one value is reported. However, in many situations, raw data from all the sensors is required. For example, evaluation of sophisticated statistical models for tree growth requires all the temperature readings in an area [8]. Collecting all the data provides the flexibility to perform arbitrary and detailed analysis on the offline data for variety of application scenarios.

Lossless collection of data in sensor networks comes at the price of high energy consumption. To reduce this cost, different techniques have been suggested to exploit the spatial correlation among the data of sensors for data collection [8], [9], [10], [11], [12], [5], [13], [14], [15], [16]. One of these techniques is distributed source coding [9], [16] which avoids transmitting any redundant data. Thus, it is theoretically the most efficient scheme in terms of the number of bits that need to be transmitted. However, it requires perfect knowledge of the correlation among data of sensors. Such information is usually very costly to provide if possible at all. To acquire this knowledge, we need at least an expensive initialization phase during which raw data from all the sensors is collected. Techniques proposed in [13], [10] do not need perfect knowledge of the correlation but they still have a major initialization cost for collecting the raw data from sensors in order to explore spatial correlation. In [13] this data is used for partitioning sensors into clusters and in [10] it is used for selecting a subset of sensors [10] that are sufficient for reconstructing the data. The techniques proposed in [15], [17] avoid the expensive initialization phase by assuming that the entropy rate for each sensor's observation and the conditional entropy rate over all the pairs of observations is provided. This may not be the case for a lot of the data sets. Thus, an initialization phase, during which raw data is collected, will be inevitable for many sensor network application scenarios.

To reduce the energy consumption in the initialization phase, redundant data can be removed at each relaying node

on the forwarding path to the base station. This is the approach proposed in [12], [5], [11]. The technique proposed in [12], [5] is to send the data along the shortest path to the base station and the redundant data is suppressed opportunistically along this path. We call this technique Routing Driven Compression (RDC) similar to [11]. The other approach is to organize the nodes into clusters. Within each cluster, each sensor's data is routed to a cluster head where redundant data can be suppressed. This technique has been discussed in [11]. We call this technique Cluster-based Routing Driven Compression (CRDC). The drawback of these schemes is that each sensor should still perform at least one full transmission where a full transmission is a transmission of one sensor's raw data without any suppression. Full transmissions are inevitable during lossless data collection especially when no history of the data is available. In a highly correlated network, full transmissions consume considerably more energy than transmissions of compressed data. Therefore, to minimize the total number of transmitted bits, as the first step of our solution, we suggest to minimize the number of full transmissions. To the best of our knowledge, no other work has suggested reducing full transmissions during data collection which is especially important when no prior knowledge about correlation of the sensors' data is available.

In this paper, we analytically determine the minimum number of full transmissions required for lossless collection of data at the base station. We consider the communication graph of the network and introduce what we call *Minimum Semi-Connected Dominating Set* (MSCDS). The elements of this set are the only nodes that require to do full transmissions for lossless data collection at the base station. Other sensors use this reference data to compress their observations by representing them in the form of mutual differences and so are only required to transmit the difference between their data and a data that they have already received. We call these differences *delta*.

To further minimize the number of transmitted bits, the size of deltas in each neighborhood should be minimized. To this end, we analytically determine the minimum number of bits required for representing a set of data, produced by sensors, in the form of mutual differences. This lower bound is equal to the sum of edge-wights of the minimum spanning tree of a logical weighted graph whose edge-weights correspond to the size of mutual differences between each pair of data.

Based on our theoretical results, we suggest DECOR which is a distributed algorithm for collecting raw data of sensors at the base station. We suggest a two-step solution: 1) Minimize the number of full transmissions. 2) Minimize the size of deltas belonging to a set of sensors in each neighborhood. To minimize the number of full transmissions, we introduce a distributed algorithm which approximates MSCDS. We show that the cardinality of the approximated set is within a constant factor of that of MSCDS. Then, we propose a distributed algorithm based on which sensors contribute in compressing a data set that represents the observations of their neighborhood before transmitting the data set towards the base station. This algorithm can compress a set of data optimally for a set of sensors within the same communication range. We evaluated the performance of the algorithm both analytically and through numerical simulations. Using simulations, we showed that our algorithm outperforms existing approaches that exploit spatial correlation. In addition, we modify the algorithm to perform in a noisy environment with lossy communication channels and through simulations, we show that it is resilient to channel failures.

In this paper, we only focus on the spatial correlation on the data of sensors and do not address the temporal correlation which can be exploited in a dynamic scenario. Exploiting temporal correlation for data collection has been addressed in [8], [18], [13]. Our work is orthogonal to these approaches and can be combined with them to further reduce number of transmitted bits in a dynamic scenario. Note that DECOR is especially useful when no history about the data is available and so no temporal correlation can be used.

The paper is organized as following: Section II describes the system model and problem formulation. In section III, we describe the theoretical foundation of our proposed solution for lossless data collection. In section IV, we explain our data collection algorithm. In sections V and VI, we evaluated our algorithm through theoretical analysis and numerical simulations, respectively. Conclusion and future works are in section VII.

II. System Model and Problem Statement

Consider a data collection problem in a sensor network where a base station collects highly correlated observations of n sensors $s_i, i \in \{1, \ldots, n\}$.

We model the communication network among the sensors and the base station with a unit-disk graph C where two nodes in the network are adjacent in the graph C if the Euclidean distance between them is less than the communication range r. We assume that C is connected otherwise some sensors can not share their data with the base station. Sensors use broaDECORst communication. As a result, when a sensor sends out a message, it is received by every sensor or base station within its communication range.

The objective is to collect the data of all the sensors at the base station with the minimum number of transmitted bits to preserve the energy and make efficient use of bandwidth.

III. Theoretical Foundation of the Proposed Solution

In this section, we introduce the theoretical foundation of our proposed solution. We first find the minimum number of full transmissions required for lossless collection of data at the base station. Then, we find the minimum number of bits required for representing a set of data in the forms of mutual differences.

A. Minimum Number of Full Transmissions

Let D be a set of sensors that perform full transmissions for lossless data collection at a base station. We call these

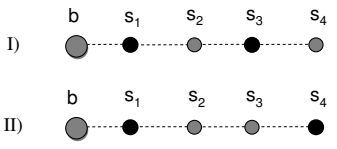

Fig. 1. Sensor network in Example 1

sensors 'reference nodes'. The rest of the sensors compute the difference between their data and that of adjacent reference nodes and so are only required to transmit their computed deltas. If a sensor is not a reference node, it should at least receive one reference data to compute its delta. Therefore, set D should be a *dominating set* of the communication graph C where set D is a dominating set, if every vertex in the graph that is not in D, has a neighbor in D. But set D should have an extra property to ensure that all references and deltas can be collected at the base station without any more full transmissions. We call this specific dominating set D a Semi-Connected Dominating Set (SCDS). We prove that the cardinality of a SCDS with minimum size (MSCDS) is equal to the minimum number of full transmissions required for collecting all the data at the base station.

We first introduce a *bridge graph* which will be used for defining a SCDS.

Definition 1 *For a graph $C = (V, E)$, its* **bridge graph** *on the set of vertices $W \subseteq V$, denoted by C_W, is a new graph such that two vertices in the set W are adjacent in C_W if and only if they are within the 2-hop distance of each other in graph C.*

Definition 2 *A dominating set D of graph C, is a* **Semi-Connected Dominating Set** *(SCDS) of graph C, if its bridge graph C_D, is connected. We refer to a SCDS of minimum size as a Minimum SCDS (MSCDS).*

Here, through a simple example, we first explain why reference nodes that perform full transmissions should be SCDS rather than just a dominating set. We then prove that minimum number of required full transmissions is equal to the size of MSCDS.

Example 1 *Consider a sensor network with the communication graph C presented in Fig. 1 where sensors $s_i, i \in \{1, \ldots, 4\}$ should transmit their data $f_i, i \in \{1, \ldots, 4\}$ to the base station b.*

In this example, set $\{s_1, s_3\}$ (black nodes in Fig.1.I) is a SCDS of the communication graph since s_1 and s_3 make a dominating set and are within the two-hop distance of each other and so their corresponding bridge graph is connected.

We show that, only the two full transmissions by sensors s_1 and s_3, are enough for data collection at the base station b. When sensors s_1 and s_3 perform full transmissions, f_1 and f_3 will be received by their neighbors. Sensor s_4 computes $\Delta_{4,3}$ and transmits it to s_3 where $\Delta_{i,j}$ is the difference of f_i with respect to f_j. Sensors s_3 transmits $\Delta_{4,3}$ to s_2. Now s_2 computes $\Delta_{3,1}$ as it already has received f_1 from s_1 and f_3 from s_3. Sensor s_2 also computes $\Delta_{2,1}$ and then forwards $\{\Delta_{2,1}, \Delta_{3,1}, \Delta_{4,3}\}$ to s_1. To collect all the data in b, it is enough that sensor s_1 retransmits these deltas to b. Since base station b already has received f_1, it can reconstruct $\{f_2, \ldots, f_4\}$ using the received deltas. Therefore, in this case, the two original full transmissions by s_1 and s_3 are enough for collecting all the data at b.

On the other hand, set $\{s_1, s_4\}$ (black nodes in Fig.1.II) is a dominating set but not a SCDS because s_1 and s_4 are not within the 2-hop neighborhood of each other and so their corresponding bridge graph is not connected.

We can easily show that the two full transmissions by $\{s_1, s_4\}$ are not enough for data collection. When sensors s_1 and s_4 perform full transmissions, f_1 and f_4 will be received by their neighbors. In this case, sensor s_3 should transmit either f_4 or f_3 to sensor s_2. Thus, at least three full transmissions are required for collecting all the data at b.

We generalize the result of Example **1**, in Theorem **1**.

Theorem 1 *The minimum number of full transmissions required for data collection at node $b \in C$, is equal to the size of MSCDS of the communication graph C.*

Proof: We first show that if D is a SCDS, full transmissions by the nodes in D, are the only full transmissions required for data collection in b. Then, we prove that the number of the full transmissions which are necessary for data collection, is not less than $|MSCDS|$ where $|MSCDS|$ is the size of MSCDS.

Consider a SCDS D in graph C. The nodes in set D, form a connected bridge graph C_D and each perform one full transmission. For collecting all the data, for each sensor s_i there should be at least one path to the base station b on which s_i's data will be transmitted. We first show that data of any node $s_i \in D$ can be transmitted to b without any more full transmission. Since set D is a SCDS, for any node $s_i \in D$, there is another node $s_j \in D$ in the 2-hop neighborhood of s_i such that s_j is on a path from s_i to b. Note that s_j is closer than s_i to b on this path. When s_i and s_j send out their reference data, there is at least one node s_k that receives both. s_k transfers s_i's data to s_j by only transmitting $\Delta_{i,j}$. Sensor s_j then forwards $\Delta_{i,j}$ on the next hop towards b.

Now, we show that the data of any node $s_r \notin D$ can also be sent to the b without further full transmissions. For s_r, there is at least one node $s_l \in D$ in its neighborhood. Since there is a path to b from s_r, node s_r can transmit $\Delta_{r,l}$ to b on this path.

Now, we prove that the number of full transmissions necessary for data collection, is not less than $|MSCDS|$. Consider a data collection which requires t full transmissions to reconstruct all the data at b. We show that $|MSCDS| \leq t$. We refer to the set of nodes that transmitted the t references as

set R. Note that $|R| \leq t$ (a node may have done more than one full transmission). Set R should be a dominating set, otherwise some of the sensors do not receive any references and so they have to do full transmissions. Now, we have to show that set R is a SCDS. If R is not a SCDS, the set of nodes in R form a non-connected bridge graph C_R. That means there should be at least one node $s_i \in R$ from which there is no path in C_R to b. Thus, considering a path from sensor s_i to b, there should be at least two adjacent nodes $s_r \notin R$ and $s_k \notin R$ on the path that have no common neighbor in R. Note that if no such path exists, then R is a SCDS. Without loss of generality let's assume that s_r is closer to s_i. Nodes s_k and s_r have no common neighbor in R whose full transmissions can be heard by both. Thus, to transmit s_i's data, node s_r has no other option other than one full transmission; otherwise data of s_i can not be reconstructed at b. But s_r does not belong to R. This is in contradiction with our original assumption that t full transmissions by nodes in R is enough for data collection. Thus, set R is a SCDS and so $|MSCDS| \leq R$ which implies $|MSCDS| \leq t$. ■

As part of our two-step solution for data collection with minimum number of bits, in subsection IV-A, we suggest a distributed algorithm that approximates the MSCDS within a factor of at most 8 from the optimal solution.

B. Optimal Data Compression

One way to compress a data set, is to represent set of data as mutual differences. In this section, we determine the minimum number of bits required for representing a set of data in the forms of mutual differences. This result can be used to minimize the number of transmitted bits by each subset of sensors that transmit a data set towards the base station. In this section, we assume that $\Delta_{i,j} = \Delta_{j,i}$ for any i and j.

Definition 3 *A* **delta graph** *H defined over n sensors $s_i, i \in \{1, \ldots, n\}$ is a complete graph in which each edge-weight between two nodes s_i and s_j, is equal to $Size(\Delta_{i,j})$ which is size of $\Delta_{i,j}$ in bits.*

Any set of data can be represented as a reference plus a proper set of mutual differences. For example, $F = \{f_1, \ldots, f_3\}$ can be represented as f_1 (a reference) plus set $S = \{\Delta_{1,2}, \Delta_{2,3}\}$. We call set S a delta compression set. Note that for representing a set of n nodes, one reference plus $n-1$ deltas are enough.

Definition 4 *A* **delta compression graph** *$G \subset H$ for a delta compression set S, is a graph in which two vertices s_i and s_j are adjacent in G if and only if $\Delta_{i,j}$ or $\Delta_{j,i}$ is in the set S.*

Optimal delta compression graph is a delta compression graph corresponding to a delta compression set with minimum number of bits.

Theorem 2 *Assuming that references have equal sizes, an optimal delta compression graph G_{opt}, for a set of n sensors, is a Minimum Spanning Tree (MST) of the delta graph H.*

Proof: We first prove that G_{opt} is a spanning tree. The delta compression graph G_{opt} spans all the nodes and it should be connected, to reconstruct every data. Graph G_{opt} is a tree. Assume to the contrary that G_{opt} is not a tree and so it has a loop. Now, if we remove an edge within the loop, still every data can be represented since assuming we have one reference, $n-1$ deltas are enough for representing n nodes. Therefore, we have a new delta compression graph with a less total edge-weight which is a contradiction.

Now, it remains to prove that it is a **minimum** spanning tree. This is straight forward since the total number of bits for representing a set of data (without considering the reference data) is the sum of edge-weights of G_{opt}. Thus, G_{opt} is a MST by definition. ■

We use this result in section IV-B to reduce the number of transmitted bits during data collection.

IV. Data Collection Algorithm (DECOR)

Based on the theoretical results presented in the previous section, we introduce our proposed Data Collection Algorithm (DECOR). DECOR is distributed and and has two phases: First, the MSCDS Approximation Phase (MAP) which finds the set of reference nodes that do full transmissions. Second, the Data Collection Phase (DCP) which collects every sensor's data at the base station. These phases are distributed and run locally at each node. In this section, we first assume that the communication links are lossless and error-free. In subsection IV-C, we relax this assumption and explain how the algorithm can be modified to deal with lossy communication channels.

Through this section, we assume that every node knows its shortest distance in number of hops from the base station. We refer to these distances as *levels*. Every node also knows its *parents* which is a set of adjacent sensors which are one level closer to the base station. The process for obtaining the levels and determining parents is straight forward and can be done similar to the level calculation stage in [19].

A. MSCDS Approximation Phase (MAP)

MAP approximates MSCDS through distinguishing two sets of nodes: BLACK nodes and GRAY nodes. BLACK nodes represent the sensors which perform full transmissions, and GRAY nodes represent the rest of sensors. Initially, all the nodes are WHITE which means that it has not yet been determined whether they are BLACK or GRAY. During the algorithm run, some nodes become BLUE meaning that they may be eligible to become BLACK, later.

Fig. 2 shows the state transition diagram of MAP. BLUE state is shown with the dashed lines. The base station b colors itself BLACK and sends out a message to its neighbors. Note that b either owns a reference data or acquires one from one of its neighbors. Each node, upon receiving a message, makes a decision about its color and then announces it. The decisions are made as following:

- Each WHITE node colors itself GRAY if it receives a message from a BLACK parent.

- Each WHITE node colors itself BLUE if it receives a message from a GRAY parent.
- Each BLUE node colors itself GRAY if it receives a message from a BLACK neighbor.
- Each BLUE node colors itself BLACK if it has maximum number of WHITE neighbors among its BLUE neighbors at the same level.

If a WHITE node receives two messages at the same time from a BLACK and GRAY parent, it always gives the priority to the message of the BLACK parent and becomes GRAY.

These simple steps make sure that if a node has a reference node in its direct neighborhood, it doesn't need to do full transmission.

Notice that BLUE nodes negotiate with their BLUE neighbors to make their decision about becoming BLACK. To approximate the MSCDS, we suggest a greedy choice for BLUE nodes in the sense that a BLUE node that covers the maximum number of WHITE neighbors among its other BLUE neighbors, becomes BLACK. Each BLUE node only competes with its BLUE neighbors in its own level. This way competition is limited within each level.If two neighboring BLUE nodes, at the same level, have equal number of WHITE neighbors, one of them randomly colors itself BLACK and informs the other. Note that based on this algorithm, except for the base station, every BLACK node has been BLUE before.

To explain how MAP works on a network, consider the following example:

Example 2 *Consider the communication graph in Fig. 3 where each sensor's level is marked in the parenthesis (Fig. 3.I). MAP on this networks runs as follows:*

II) Base station b colors itself BLACK and announces its color (Fig. 3.II).

III) Sensors s_1 and s_2, at level 1, upon receiving the message from BLACK node b, color themselves GRAY and announce their colors (Fig. 3.III).

IV) Sensors s_3 and s_4, at level 2, upon receiving the message from GRAY nodes, color themselves BLUE (Fig. 3.IV, color BLUE is shown by dashed lines).

V) Sensor s_3 and s_4 have both one WHITE neighbor. Thus, one of them, say sensor s_3, randomly colors itself BLACK and sends a message to its neighbors. (Fig. 3.V).

VI) Sensors s_4 and s_5 upon receiving this message, color themselves GRAY (Fig. 3.VI).

Note that $\{b, s_3\}$ is a MSCDS of the communication graph of Fig. 3. Thus, in this example, MAP was able to determine MSCDS.

In section V, we will first prove that set of BLACK nodes identified through MAP construct a SCDS of the communication graph C. Then, we find the upper bound on the number of BLACK nodes determined through MAP. We show that the cardinality of the set of BLACK nodes found through MAP is within eight times the cardinality of the MSCDS.

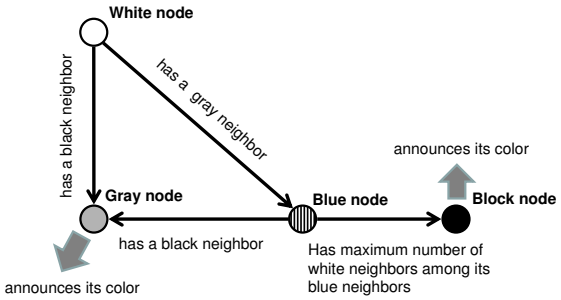

Fig. 2. State transition diagram for MSCDS Approximation Phase

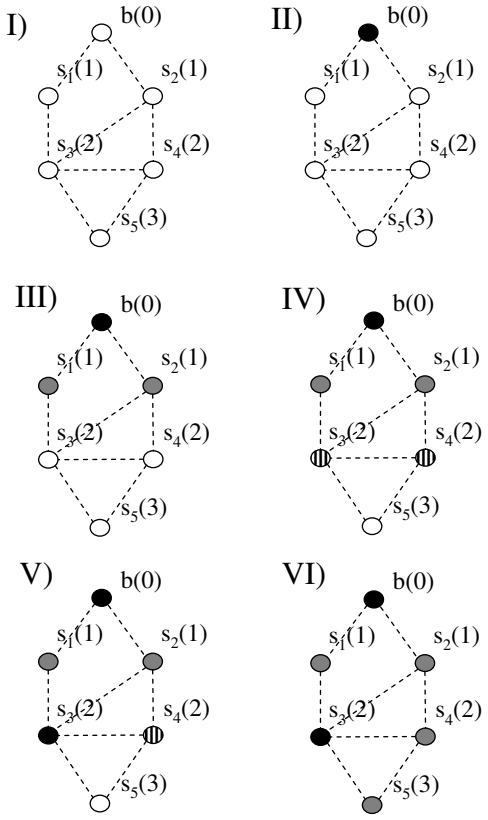

Fig. 3. Example of MSCDS approximation

B. Data Collection Phase (DCP)

So far, we explained how to minimize the number of full transmissions by approximating MSCDS through MAP. In DCP, delta and references will be collected at a base station. Our goal is to minimize the number of transmitted bits by minimizing the size of data transmitted by each set of sensors.

In DCP, each sensor tries to contribute in compressing the data set of its neighborhood optimally by finding the Minimum Spanning Tree (MST) of the corresponding delta graph in a

Algorithm 1 DCP algorithm at sensor s_i

1: **if** s_i BLACK **then** BLACKnode
2: **else** GRAYnode
3: **end if**
4: PARENTnode

Algorithm 2 BLACKnode: Algorithm at BLACK sensor s_i

1: REF $\leftarrow \{f_i,i\}$
2: Send out REF
3: Wait for SIZE from every parent
4: SELECT $\leftarrow$ ID of parent with smallest SIZE
5: Send out SELECT

distributed way.

Sensors use four types of messages to communicate during DCP.

- REF: Data message containing the reference data and the ID of the node that transmits a reference.
- DIFF: Data message containing the delta and the ID of the node to which the delta belongs and the reference ID with respect to which the delta is computed.
- SIZE: Message sent by a parent to its BLACK child containing the ID of the child and the size of the child's data represented by the parent.
- SELECT: Message sent by a child containing the ID of the *selected* parent responsible for transmitting data of the child.

Pseudocode of DCP is shown in algorithm 1. In summary, if a node is BLACK, it executes BLACKnode (algorithm 2) where it shares its reference data with its neighbors. If it is GRAY, it executes GRAYnode (algorithm 3) which attempts to minimize the size of GRAY node's delta. If a node (BLACK or GRAY) has children, it executes PARENTnode (algorithm 4) where it transmits its children's data toward the base station.

DCP starts when each BLACK node performs a full transmission by sending out a REF message that contains its reference data. The REF message is received by the set of BLACK nodes' neighbors. Every GRAY node computes its delta with respect to a transmitted reference data. Some GRAY nodes may receive multiple references and so they will be able to compute multiple deltas for their data. If a sensor has more than one delta, it can choose the smallest one.

After full transmissions, data of BLACK and GRAY nodes will be collected. The strategies for collecting the data of BLACK and GRAY nodes are different as following:

For BLACK nodes, we should make sure that every reference data is compressed with regard to another reference data of a higher level node. This avoids the loop referencing that may occur otherwise. Data of each BLACK node will be sent through one of its GRAY parents as explained in Example 1. Notice that a BLACK child has no BLACK parent since during MAP each BLACK node ensure that all its neighbors are GRAY. Similarly, a BLACK parent has no BLACK child. Every GRAY parent of a BLACK node receives at least

Algorithm 3 GRAYnode: Algorithm at GRAY sensor s_i

1: $J \leftarrow \underset{\{j|s_j \in \text{BLACK neighbor}(s_i)\}}{\text{argmin}} (\text{Size}(\Delta_{i,j}))$
2: mydelta $\leftarrow \Delta_{i,J}$
3: myRef $\leftarrow J$
4: set timer
5: **while** timer< Size(mydelta) **do**
6: **if** (s_i overhears s_k) and (s_i can reconstruct f_k) **then**
7: $\Delta_{i,k} \leftarrow \Delta(f_i, f_k)$
8: **if** Size($\Delta_{i,k}$) < mydelta **then**
9: mydelta $\leftarrow \Delta_{i,k}$
10: myRef $\leftarrow k$
11: **end if**
12: reset timer
13: **end if**
14: **end while**
15: SELECT $\leftarrow$ ID of a random parent
16: Send out SELECT
17: DIFF $\leftarrow$ {mydelta,myRef,i}
18: Send out DIFF

Algorithm 4 PARENTnode: Algorithm for parent s_i

1: **for** each BLACK child s_j **do**
2: $K \leftarrow \underset{\{k|s_k \in \text{BLACK neighbor or parent of}(s_i)\}}{\text{argmin}} (Size(\Delta_{j,k}))$
3: s_j.delta $\leftarrow \Delta_{j,K}$
4: s_j.Refid $\leftarrow K$
5: SIZE $\leftarrow Size(\Delta_{j,K})$
6: Send out SIZE to s_j
7: Wait for SELECT
8: **if** SELECT=s_i **then**
9: DIFF $\leftarrow \{s_j$.delta,s_j.Refid,$j\}$
10: Send out DIFF
11: **end if**
12: **end for**
13: **if** DIFF is received from child s_j that has selected s_i **then** forward DIFF to the selected parent of s_i
14: **end if**

another reference data from a BLACK parent or neighbor. The parent, computes the difference of these two references. Then, it sends out a SIZE message representing the size of the delta between them (If the parent has more than one delta for its BLACK child, it sends out the size of the smallest one). The BLACK child that may have received the SIZE message from multiple parents, selects a parent with the smallest size delta. Then, it notifies the parent through a SELECT message to forward its delta to its own parent. The forwarding continues until delta reaches to the base station.

Strategy for transmitting the data of GRAY nodes is shown in algorithm 3 and is as follows: A GRAY node, waits for a time directly proportional to the size of its delta to send out its delta through a DIFF message. This transmission will be heard by all the parents and neighbors of this node. Then, it chooses a parent randomly using a SELECT message to forward its

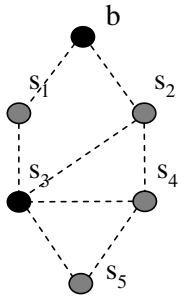

Fig. 4. Example of Data Collection Algorithm

transmitted delta. The advantage of having a waiting time proportional to the size of delta, is that sensors with the smaller deltas will transmit their deltas first. Sensors with the larger deltas have a chance to wait and overhear other transmissions to see if they can re-represent their data with smaller deltas. Every time a node receives a delta, it will reset its waiting time. Notice that a GRAY node only updates its delta if it can represent its data with a smaller delta. We will discuss in subsection V-B that in a fully-connected neighborhood of sensors, this strategy results into the optimal delta compression since it is in fact a distributed version of the Prim's algorithm [20] for computing MST.

Each parent forwards the DIFF that it has received from its GRAY children immediately by choosing a parent randomly through SELECT.

We explain our algorithm through Example 3.

Example 3 *Consider the sensor network in Fig. 4. In Example 2, we found that MSCDS for this graph is* $\{b, s_3\}$. *Now, sensors* $s_i, i \in \{1, \ldots, 5\}$ *should transmit their data to* b.

Initially nodes b *and* s_3 *transmit their reference data* f_0 *and* f_3 *through REF messages respectively. These two are the only full transmissions that are required for our data collection. Note that we assume base station* b *is a sensor too and so provides* f_0. f_0 *is received by* s_1 *and* s_2 *and* f_3 *is received by* s_1, s_2, s_4 *and* s_5.

Sensor s_1 *computes* $\Delta_{1,0}$ *and* $\Delta_{1,3}$ *and keeps the smaller one (say* $\Delta_{1,0}$*). Similarly* s_2, s_4 *and* s_5 *keep* $\Delta_{2,0}$, $\Delta_{4,3}$ *and* $\Delta_{5,3}$ *respectively.*

We first explain how data of BLACK nodes are transmitted to the base station. Node s_3 *is BLACK, so sensors* s_2 *and* s_1 *as the parents of* s_3 *compute the difference between two reference data* f_0 *and* f_3 *(*$\Delta_{3,0}$*). Sensor* s_2 *and* s_1 *both transmit size of* $\Delta_{3,0}$ *through the SIZE message. Since they both transmit the same size,* s_3 *randomly chooses one, (say* s_1*) and sends out a SELECT message containing* s_1*'s ID. Thus,* s_1 *now sends* $\Delta_{3,0}$ *to* b.

Now, we explain how data of GRAY nodes is transmitted: Assume s_4 *is the first one which transmits its delta* $\Delta_{4,3}$. *Sensor* s_4 *chooses* s_2 *as its parent through a SELECT message since it is the only parent of* s_4. *Sensor* s_2 *forwards* $\Delta_{4,3}$ *to* b *as* b *is also the only parent of* s_2. *After* $\Delta_{4,3}$ *is transmitted by* s_4*, it is received by* s_2 *and* s_5. *Sensors* s_2 *and* s_5 *use* $\Delta_{4,3}$ *to construct* f_4 *and then* $\Delta_{2,4}$ *and* $\Delta_{5,4}$ *respectively. Let's* $Size(\Delta_{5,4}) < Size(\Delta_{5,3})$. *Sensor* s_5 *updates its delta to* $\Delta_{5,4}$ *and resets its timer to start over its waiting time. But let's* $Size(\Delta_{2,0}) < Size(\Delta_{2,4})$. *Sensor* s_2 *does not update its delta but it still updates its timer. Sensor* s_5 *chooses a parent randomly (say* s_4*) and after timer of* s_5 *expires, it will transmit* $\Delta_{5,4}$ *which will be forwarded by* s_4 *to* s_2 *and then by* s_2 *to* b. *Sensor* s_2 *and* s_1 *similarly transmit* $\Delta_{2,0}$ *and* $\Delta_{1,0}$ *respectively to* b.

Having f_0, $\Delta_{2,0}$, $\Delta_{3,0}$, $\Delta_{1,0}$, $\Delta_{4,3}$ *and* $\Delta_{5,4}$, *base station* b *can reconstruct* $\{f1,\ldots,f_5\}$.

There exists a tradeoff between the size of deltas and the cost of extra storage and memory footprint of each sensor. During DCP, each GRAY node has to store data of all of its neighbors that have already transmitted their data. That's because it should be able to re-represent its delta using the data of any of its neighbors as reference. When a sensor receives a delta from a neighbor, it checks to see whether it has its reference or not and if it does, it reconstructs the neighbor's data. Thus, it has to keep an index list of the existing data in its memory. However, if each GRAY node only computes its delta with respect to the reference data provided by the BLACK neighbors, it will only need to store the smallest delta and does not need to keep any index list in memory. We call this version of the algorithm Simple DECOR (S-DECOR). Note that in S-DECOR, no attempt for MST computation is performed. We evaluate the performance of DECOR and S-DECOR in section VI and compare them with some existing state of the art solutions.

C. Data Collection Algorithm for Lossy Communication Channels

In wireless sensor networks, communication links are noisy and collision may occur especially when we use broadcast communication. Thus, designing an algorithm that is resilient to communication failures, is essential for a lossless data collection algorithm. In this section, we modify DECOR to deal with lossy networks. Our strategy is to avoid retransmissions especially for larger messages. When retransmission is inevitable, we reduce the chance of collision by introducing a random waiting time for each node after collision before retransmitting the data. We will discuss our strategy for each type of message that might be lost.

During DECOR, some control messages are sent, including the messages sent during MAP and also SIZE and SELECT during DCP. These are short messages and sensors can afford to employ simple error control techniques like ARQ (Automatic Repeat Request) methods [21] to correct the errors during transmission. Thus, we ignore the effect of noise and interference for this kind of messages.

For DIFF messages, if one parent receives the data correctly it does not need to be retransmitted. Thus, a DIFF message is lost when it is not received by any of its parents. If none of

the parents have received the DIFF message correctly, it has to be retransmitted using an ARQ type technique.

A REF message might be lost too. In this case, our strategy is to avoid retransmitting the references by the BLACK nodes as much as possible because it results in the energy exhaustion at the BLACK nodes.

In three cases, if a REF message is lost or received incorrectly, another full transmission should be performed. We explain these three cases in the following. Notice that, strategies described in 1) and 2) ensure that data of BLACK nodes will receive to the base station whereas the strategy described in 3) ensures that data of a GRAY node that has not received any reference will be received by the base station.

1) If the full transmission by a BLACK node is not received correctly by any parent, there is no other option other than BLACK node repeating its transmission. The reason is that the data of the BLACK node should reach to the base station through one of its parents and so at least one parent should receive its data. Notice that this is the only case where BLACK node has to repeat its full transmission.

2) If a GRAY parent has received an error-free full transmission from a BLACK child but none from a BLACK parent or neighbor, then the GRAY node has to do one full transmission in order to transmit its BLACK child's data. The reason is that as explained in DCP, the GRAY parent should receive at least another reference data from its parent or neighbor to compute the delta for its BLACK child. If it does not receive that, it has to to repeat the full transmission.

3) A GRAY node that has not received any error-free references need to do a full transmission itself, so its data will be received by its parents.

V. Theoretical Performance Analysis

In this section, we evaluate the performance of DECOR. To this end, we first evaluate performance of MAP and then evaluate DCP. Finally, we determine the time complexity, message overhead, storage and memory requirement of DECOR.

A. Performance Analysis of MAP

In this subsection, we evaluate the performance of MAP. We first prove that set of BLACK nodes identified through MAP is a SCDS. Then, we derive an upper bound for the number of BLACK nodes found through MAP based on the size of MSCDS.

Theorem 3 *Set of BLACK nodes identified through MAP construct a SCDS of the communication graph C.*

Proof: Set of BLACK nodes D are a dominating set since each GRAY node is adjacent to at least one BLACK node.

Now, to show that set of BLACK nodes construct a SCDS, we need to show that bridge graph C_D is connected. For that it is enough to show that there is a path to b in graph C_D for any BLACK node and so all the BLACK nodes are connected in C_D through b. To this end, we show that for every BLACK node at level l_i, there is another BLACK node at level $l_i - 1$ or $l_i - 2$ within the two-hop neighborhood. Assume to the contrary that node s_i is a BLACK node at level $l_i \geq 2$ which does not have any BLACK node within its two-hop neighborhood at levels $l_i - 1$ or $l_i - 2$. Node s_i has been BLUE before. Thus, s_i has received a message from its GRAY parent s_j at level $l_i - 1$. But s_j has also received a message either from a BLACK parent s_k at level $l_i - 2$ or a BLACK neighbor at the same level $l_i - 1$. This contradicts the assumption that s_i has no BLACK node within its two-hop neighborhood at level $l_i - 2$ or $l_i - 1$. ■

Now, we show that the cardinality of the set of BLACK nodes found through MAP is within eight times the cardinality of the MSCDS. In other words, MAP has an approximation factor of at most 8.

The sketch of the proof is similar to [19]. We denote the size of MSCDS by *opt*. We first show that the set of BLACK nodes in MAP is an *independent* set where a set of nodes are *independent* if they are pairwise non-adjacent. Then, we find an upper bound for any independent set of a connected graph, namely set of BLACK nodes found through MAP, based on *opt*.

Lemma 1 *Set of BLACK nodes determined through MAP is an independent set.*

Proof: Assume to the contrary that MAP has determined two adjacent BLACK nodes s_i and s_j. Neither one of s_i and s_j can be the base station since all the neighbors of base station become GRAY, after base station announces its color. Thus, s_i and s_j have both been BLUE before. But a BLUE node turns GRAY if it has a BLACK neighbor. Therefore, set of BLACK nodes determined through MAP are pairwise non-adjacent and so are independent. ■

Now, we bound the size of any independent set in terms of *opt*. For our proof, we use this fact that in a unit-disk graph, each node is adjacent to at most five independent nodes [22].

Lemma 2 *The size of any independent set in a unit-disk graph $C = (V, E)$ is at most $8opt + 1$.*

Proof: Consider a spanning tree T_{MSCDS} of the bridge graph C_{MSCDS} (Note that C_{MSCDS} is connected, so T_{MSCDS} exists). We construct a subgraph $T \subset C$ as follows: 1) T includes every vertex in T_{MSCDS}. 2) For any v_i and v_j that are adjacent in T_{MSCDS}, we add exactly one vertex $v_k \in C$ to T such that v_k is adjacent to both v_i and v_j in C (by definition of bridge graph such v_k certainly exists). We also add the two edges that connect v_k to v_i and v_k to v_j to graph T. Graph T is a tree ;otherwise it has a loop. Now, we delete every vertex that is not in MSCDS from the loop in T, along with their corresponding edges. Since all deleted vertices were adjacent to exactly two vertices of MSCDS, the remaining graph which is T_{MSCDS} has also a loop. This is a contradiction since T_{MSCDS} is a tree. The size of T is at most $2opt$ since for each vertex $v_i \in MSCDS$ at most one vertex $v_k \notin MSCDS$ is added to T.

From here the proof is identical to the proof in [19].

Let U be any independent set of V. Consider any arbitrary preorder traversal of T given by $v_1, v_2, ..., v_{2opt}$. Let U_1 be the set of nodes in U that are adjacent to v_1. For any $2 \leq i \leq 2opt$, let U_i be the set of nodes in U that are adjacent to v_i but none of $v_1, v_2, ...v_{i-1}$. Then U_1, U_2,.., U_{2opt} form a partition of U. In other words, each element of U appears in exactly one subset of the list. As v_1 can be adjacent to at most five independent nodes, $|U_1| \leq 5$. But for any $2 \leq i \leq 2opt$, at least one node in $v_1, v_2, ...v_{i-1}$ is adjacent to v_i. Thus, $|U_i| \leq 4$. Thus the size of any independent set is equal to:

$$|U| = \sum_{i=0}^{2opt} |U_i| \leq 5 + 4(2opt - 1) = 8opt + 1$$

■

Theorem 4 *The approximation factor of MAP is at most 8.*

Proof: Lemma 1 shows that the set of BLACK nodes is an independent set and based on Lemma 2, the size of any independent set is bounded by $8opt + 1$. Thus, the the total number of BLACK nodes found through MAP is at most $8opt+1$. Therefore, the approximation factor of MAP is equal to 8. ■

B. Performance of DCP

In this section, we show that DCP can minimize the number of transmitted bits by each sensor in a fully-connected neighborhood.

In a fully connected network, each data transmission is received by every sensor. Since every sensor resets its waiting time after receiving a delta, at each step always the sensor with the smallest delta transmits its data. Thus, our algorithm is in fact a distributed version of the Prim's algorithm [20] for finding the MST where at each step a new edge with the minimum size will be added to the constructed tree.

In general, a communication graph C of the sensors is not fully connected. However, a set of sensors in the same communication range r are certainly fully connected. During DCP we take advantage of this property to minimize the total number of bits in each neighborhood. Therefore, although DECOR is not guaranteed to be optimal, it can minimize the total number of transmitted bits in each neighborhood of nodes within the same communication range r.

C. DECOR in Practice

Sensors are cheap devices that usually have limited memory and storage space. Thus, in order to see how practical DECOR and S-DECOR are, in this subsection, we determine the memory footprint and storage requirement of these algorithms at each node . In addition, since substantial amount of sensors' energy is consumed for extra communication for controlling messages, we also compute the message overhead of these algorithms. We also determine the time complexity of DECOR and S-DECOR in order to obtain the overall time that it takes for collecting all the data at the base station. Notice that DECOR and S-DECOR have the same first phase which is MAP. However, the second phase (DCP) is different for these algorithms.

We first determine the time complexity of DECOR and S-DECOR. The time complexity of these algorithms are both $O(n)$. The first phase for these algorithms is MAP which has $O(n)$ time complexity. The worst case scenario in MAP is when a BLUE node has to wait for the decision of all the BLUE nodes in its level. This case occurs when a BLUE node has more WHITE neighbors than one neighbor but less WHITE neighbors than the other one and there is a chain of these BLUE nodes in one level. In this case, the BLUE node waits until one BLUE node (probably at the end of the chain) decides to become BLACK. Thus, at each level, a BLUE node, at the worst case, has to wait for n_i nodes to make decision where n_i is the number of BLUE nodes at level i. Hence, in total a BLUE node at the last level can wait for a total time of $\sum_{i=0}^{L} n_i = O(n)$ where L is the number of levels.

In DECOR, the second phase which is DCP has $O(n)$ time complexity. In this case, a node with the largest delta waits for the transmission of all its neighbors and overhears them all to update its delta. Thus, assuming the maximum number of neighbors for one node is n_d, the complexity of this algorithm is $\sum_{d=0}^{L} n_d = O(n)$. Thus, the time complexity of DECOR is $O(n)$. In S-DECOR, deltas are only computed with respect to the data of BLACK nodes. Thus, the time complexity of DCP in this case is $O(L)$. Therefore time complexity of DECOR and S-DECOR are both $O(n)$.

Now, we determine the message overhead of DECOR and S-DECOR per node. In MAP each node only sends a constant number of overhead messages for determining the colors. The message overhead of DCP is also constant for both DECOR and S-DECOR because each node only sends a constant number of SELECT and SIZE messages.

The memory footprint of MAP is negligible. Memory footprint of DCP for DECOR is linear in the degree of the node. The reason is as explained in section IV-B, each node has to keep an index list of the existing data in its memory so that when it receives a delta from a neighbor, it can check to see whether it has its reference or not. DCP for S-DECOR has a negligible memory footprint.

Finally Storage requirement for MAP is negligible. However, during DCP of DECOR each node has to store data of all of its neighbors that have already transmitted their data. Thus, the storage requirement is a function of the degree of the node. Storage requirement for DCP of DECOR is also negligible.

VI. SIMULATION

In this section, we evaluate the performance of DECOR through simulations. In the following subsections, first, we evaluate the performance of MAP in estimating MSCDS by comparing it with the actual MSCDS of the network and the upper bound we achieved. Then, we evaluate the performance of DECOR by comparing it with RDC and CRDC. As explained in section I, similar to DECOR, these algorithms use spatial correlation for collecting raw data and can be used

in the scenarios where no history of the data is available. However, they do not minimize full transmissions. We also evaluate the performance of S-DECOR which is simplified version of DECOR. As explained in section IV-B, each GRAY node only computes its delta with respect to the data of its BLACK neighbors. S-DECOR is more suitable when the sensors have limited memory.

For simulation, we use MATLAB and we randomly place sensors in a two-dimensional 1×1 space based on the uniform distribution.

A. Performance Evaluation of MAP

We evaluate the performance of MAP in approximating MSCDS for a network of 100 sensors (Fig. 5). The goal is to compare the number of black nodes found through MAP with the actual size of MSCDS and the analytical upper bound we obtained in section V for MAP. We run the simulations for different communication graphs of our network. To change the communication graph, we change the communication range of the sensors. As shown in Fig. 6, the cardinality of MSCDS (minimum number of full transmissions), decreases when the communication range increases. This result is expected since the connectivity of the communication graph grows as the communication range of sensors increases. Thus, the size of MSCDS decreases. This figure shows that MAP performs much closer to the optimal answer than the analytical upper bound we derived in section V for MAP.

B. Performance Evaluation of DECOR

We evaluate the performance of DECOR by comparing it with S-DECOR, RDC and CRDC through investigating the effect of the following three parameters on the number of transmitted bits during data collection:

- Correlation coefficient
- Number of sensors
- Probability of channel failure

In our simulation, each sensor has a 10 kbit data which should be transmitted to the base station located at the center of the area (Figure 5).

1) Effect of Correlation Coefficient: Correlation Coefficient is used as measure of correlation among data of sensors. We model the relation between correlation coefficient, delta and the distance between nodes s_i and s_j as following:

$$\text{Size}(\Delta_{i,j}) = \left(1 - \exp\left(-\beta \left\|s_i - s_j\right\|_2^2\right)\right) F, \qquad (1)$$

where $\|s_i - s_j\|_2^2$ is the distance between sensors s_i and s_j, and F is the size of data (10 kbit) and β is correlation coefficient which is a constant factor that controls the correlation among data of sensors. This model is similar to the model proposed in [14], [23] for correlation among data of sensors. Notice that $\Delta_{i,j}$ has limiting values of F when distance between s_i and s_j is infinite and zero when the distance between s_i s_j is zero.

Fig. 7 shows the number of transmitted bits normalized with respect to the size of data versus correlation coefficient β for

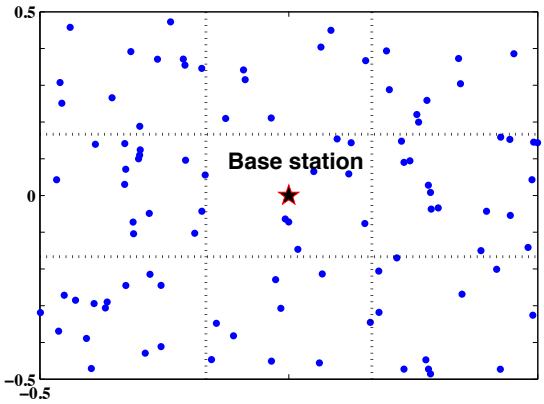

Fig. 5. Distribution of the sensors and the base station in the area

the four algorithms DECOR, S-DECOR, RDC, CRDC. The number of sensors is fixed in this simulation and is equal to 100 and the communication range is also fixed. Both RDC and CRDC algorithms require n number of full transmissions since every sensor should do one full transmission. Therefore, for high correlation (small β), the number of transmitted bits in RDC and CRDC is significantly larger than the number of transmitted bits in DECOR and S-DECOR which require only transmissions of reference data. As the correlation among the sensors data decreases (large β), the size of delta converges to the size of reference data. Therefore, The normalized number of transmitted bits in both DECOR and S-DECOR increases and converges to $\sum_{i=1}^{n} l_i$ where l_i is level of sensor i. As shown in Fig. 7, the normalized number of bits sent in both DECOR and S-DECOR are equal to that of RDC in low correlation. The reason is that at low correlation, what these algorithms do is similar which is only relaying the data along the shortest path to the base station.

We haven't shown the overhead of DECOR and S-DECOR in this simulation. As explained in section **??**, the message overhead is constant and is a function of the size of SELECT and SIZE messages which can be ignored if the reference data is much larger.

2) Effect of Number of sensors: In this section, we study the number of transmitted bits as a function of the number of sensors for fixed communication range and correlation coefficient β. We choose β=1 which enforces a large correlation among data of sensors. The results in Fig. 8 show that as the number of sensors increases, the number of transmitted bits for RDC and CRDC increases linearly. This is because in these algorithms, number of full transmissions increases with the number of sensors. Notice that RDC and CRDC show a similar performance for the selected β. However, number of transmitted bits in DECOR remains almost constant when number of sensors increases. The reason is that when number of sensors grows beyond a certain point, their data becomes highly correlated and most of it becomes redundant. DECOR successfully captures this and does not transmit the redundant

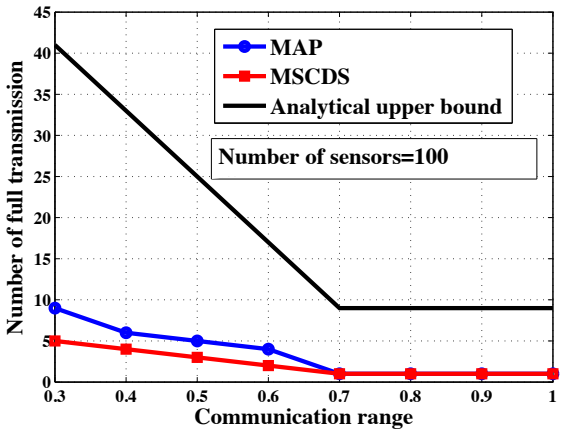

Fig. 6. Number of full transmission versus communication range

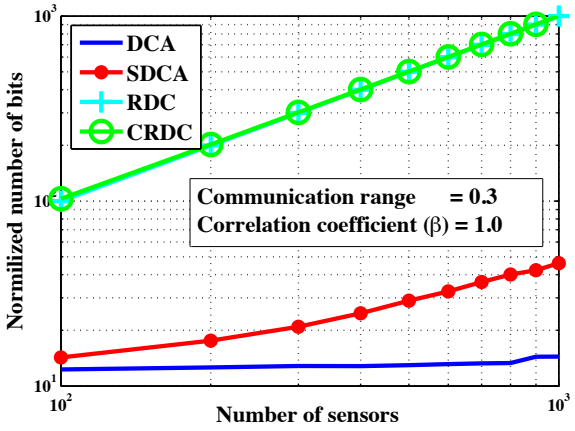

Fig. 8. The number of transmitted bits (normalized with data size) versus the number of sensors

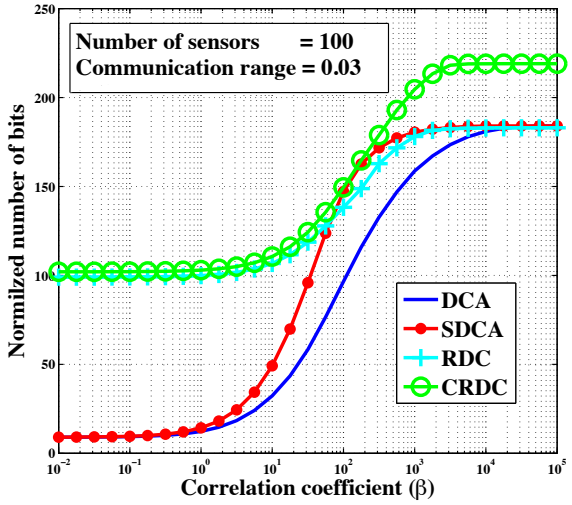

Fig. 7. The number of transmitted bits (normalized with data size) versus correlation coefficient β

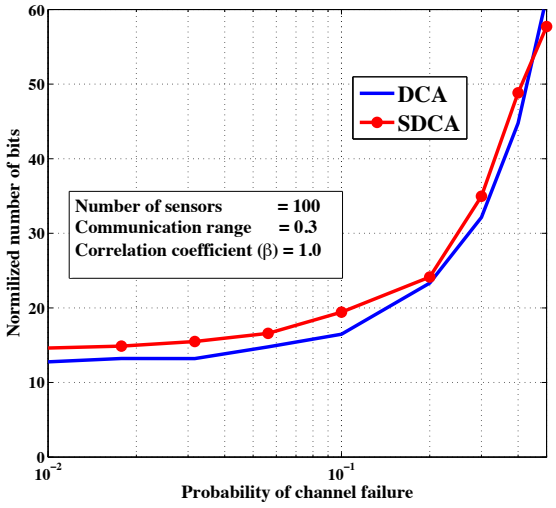

Fig. 9. The number of transmitted bits (normalized with data size) versus the number of sensors

data.

3) Effect of Channel Failure: Finally, we evaluate the performance of the algorithm presented in subsection IV-C for lossy communication channels. We consider a network of 100 sensors with fixed communication range and correlation coefficient β. Moreover, we assume that the probability of channel failure is constant over the channels. Figure 9 shows the number of transmitted bits for both DECOR and S-DECOR normalized with respect to the number of transmitted bit for error-free communication scenario averaged over 20 runs. As it is expected, the number of transmitted bits increases with probability of channel failure. Notice that DECOR is more robust to communication failure than S-DECOR. For example, if the probability of channel failure is equal to 0.5, DECOR requires only 3 times more transmitted bits comparing to the error-free case where S-DECOR requires 4.5 time more transmitted bits.

VII. Conclusion

In this paper, we exploited the spatial correlation on the data of sensors to reduce the communication energy during data collection. We suppress the redundant data sent through data collection using a two-step strategy. First minimizing the number of full transmission and second minimizing the size of data produced by set of sensors at each neighborhood. We found a lower bound on the minimum number of full transmissions required for lossless data collection. We also found the minimum number of bits required for compressing a data set using the mutual differences. Based on our theoretical results, we introduced a practical algorithm for data collection which reduces the overall energy consumption. Finally, we presented the numerical simulations to verify our theoretical results.

REFERENCES

[1] P. Tan, "Knowledge Discovery from Sensor Data," *SENSORS PETERBOROUGH*, vol. 23, no. 3, p. 14, 2006.

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM New York, NY, USA, 2002, pp. 88–97.

[3] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, and C. MIT, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10.

[4] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling Data-Centric Routing in Wireless Sensor Networks," *IEEE INFOCOM*, pp. 1–11, 2002.

[5] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pp. 575–578, 2002.

[6] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks," *Computer Networks*, vol. 42, no. 6, pp. 697–716, 2003.

[7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[8] A. Silberstein, R. Braynard, and J. Yang, "Constraint chaining: on energy-efficient continuous monitoring in sensor networks," in *Proceedings of the 2006 ACM SIGMOD international conference on management of data*. ACM New York, NY, USA, 2006, pp. 157–168.

[9] J. Chou, D. Petrovic, and K. Ramchandran, "A distribued and adaptive signal processing approach to reducing energy consumption in sensor networks," in *Proceedings of IEEE INFOCOM*, San Fransisco, CA, USA, March, April 2003, pp. 1054–1062.

[10] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 1, p. 4, 2008.

[11] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4, p. 24, 2008.

[12] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks," in *International conference on distributed computing systems*, vol. 22. IEEE Computer Society; 1999, 2002, pp. 457–458.

[13] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, p. 1010, 2007.

[14] K. Yuen, B. Liang, and B. Li, "A distributed framework for correlated data gathering in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 578–593, 2008.

[15] J. Li, A. Deshpande, and S. Khuller, "On Computing Compression Trees for Data Collection in Sensor Networks," *Arxiv preprint arXiv:0907.5442*, 2009.

[16] X. Su, "A combinatorial algorithmic approach to energy efficient information collection in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, 2007.

[17] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 1, p. 54, 2006.

[18] M. Vuran, Ö. Akan, and I. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[19] P.-J. Wan, K. M.Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings of IEEE INFOCOM 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, June 2002, pp. 1597–1604.

[20] R. Prim, "Shortest connection networks and some generalizations," *Bell Systems Technical journal*, vol. 36, pp. 1389–1401, 1957.

[21] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 3rd ed. Morgan Kaufmann, 2003.

[22] M. Marathe, H. Breu, H. Hunt III, S. Ravi, and D. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol. 25, no. 2, pp. 59–68, 1995.

[23] M. Vuran and I. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 2, p. 329, 2006.