Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 345052, 20 pages
doi:10.1155/2010/345052

Research Article

Collaborative Event-Driven Coverage and Rate Allocation for Event Miss-Ratio Assurances in Wireless Sensor Networks

H. Ozgur Sanli and Hasan Çam

Computer Science and Engineering Department, Arizona State University, Tempe, AZ 85287, USA

Correspondence should be addressed to Hasan Çam, hasan.cam@asu.edu

Received 2 November 2009; Accepted 30 March 2010

Academic Editor: Xinbing Wang

Wireless sensor networks are often required to provide event miss-ratio assurance for a given event type. To meet such assurances along with minimum energy consumption, this paper shows how a node's activation and rate assignment is dependent on its distance to event sources, and proposes a practical coverage and rate allocation (CORA) protocol to exploit this dependency in realistic environments. Both uniform event distribution and nonuniform event distribution are considered and the notion of ideal *correlation distance* around a clusterhead is introduced for on-duty node selection. In correlation distance guided CORA, rate assignment assists coverage scheduling by determining which nodes should be activated for minimizing data redundancy in transmission. Coverage scheduling assists rate assignment by controlling the amount of overlap among sensing regions of neighboring nodes, thereby providing sufficient data correlation for rate assignment. Extensive simulation results show that CORA meets the required event miss-ratios in realistic environments. CORA's joint coverage scheduling and rate allocation reduce the total energy expenditure by 85%, average battery energy consumption by 25%, and the overhead of source coding up to 90% as compared to existing rate allocation techniques.

1. Introduction

Wireless sensor networks usually consist of a large number of sensor nodes that collaboratively gather data from a target region of interest. A sensor network is expected to provide event miss-ratio guarantees, while minimizing the energy consumption of its sensor nodes. The *miss-ratio* for an event is equal to the fraction of those occurrences that are missed by wireless sensor nodes in a given time interval. While a sensor network can dynamically adapt its aggregate coverage in response to event misses, providing such assurances with minimum energy consumption is not easy due to the following problems. First, event occurrences may follow a nonuniform distribution. Consequently, two sensor nodes may detect a different number of events in a given interval even though their contribution to aggregate coverage of the network is the same. Second, minimizing the number of active nodes by reducing the overlapping areas among sensing regions of neighboring nodes often sacrifices their correlation. However, sensor data correlation can be exploited by distributed source coding for reducing the energy spent for data transmission which is a major component in overall system energy consumption [1, 2]. Inspired by these problems, this paper is the first of its kind to exploit the relationship between coverage scheduling (i.e., deciding which sensing unit(s) of each node should be activated) and source coding rate allocation (i.e., determining the number of bits that a sensor node should transmit for its sensor reading), and to show how they should cooperate. The main contributions of this research include the following.

(1) We demonstrate the existence of an ideal *correlator radius* within cluster of sensor nodes around which on-duty node selection will minimize the amount of data sensed and transmitted under both uniform and nonuniform event distributions.

(2) CORA (COverage and Rate Allocation) Protocol is proposed to exploit ideal correlator radius concept via cooperation between event-based scheduling and rate allocation. CORA schedules on-duty nodes to achieve better accuracy on the coverage of the monitored area than existing algorithms and provides

event miss-ratio assurances even when nodes have no location information and there's noise in sensing ranges of nodes.

(3) This paper develops a novel technique called *differential data transmission* to remove those data redundancies which cannot be eliminated via distributed source coding. Nonredundant data gathered by distributed source coding are compared with known reference data and then only its difference from the reference data are transmitted to the clusterhead.

We noticed that certain cluster members are critical for meeting the required event miss-ratios energy efficiently due to their following two features. First, by taking advantage of their location in cluster topology, they can apply distributed source coding for other cluster members and significantly reduce data redundancy. Second, due to their proximity to the hotspots (i.e., regions with high event occurrence probabilities), they can detect majority of the event occurrences which are condensed around hotspots. Therefore, in CORA, rate allocation requests coverage scheduling to activate such critical cluster members within the ideal proximity of the clusterhead which we refer as *correlator radius*. Similarly, coverage scheduling helps rate allocation assign various rates to sensor nodes by determining which neighboring sensor nodes of a given sensor node should be put in sleep mode.

CORA's rate allocation first identifies the uniform/ nonuniform distribution of event occurrences using a continuously updated histogram. Under nonuniform event distribution, those hotspot locations where the event occurrence probability is very high are determined. For both uniform and nonuniform event distributions, we then demonstrate the existence of an ideal radius around the clusterhead where the source coding gain is maximized. At the distance of ideal radius from clusterhead, our rate allocation algorithm forms groups of active nodes. Each group has a so-called *correlator node* that gathers data from its group members and transmits the aggregated data to its clusterhead. To further improve data reduction, we propose a new differential data transmission technique for correlator nodes. To the best of our knowledge, there is no previously known technique that assigns rates in accordance with the occurrence of the observed phenomena.

CORA's coverage scheduling is unique of its kind in guaranteeing the miss-ratio required for each event without any need for location information of nodes. While activating less number of nodes than existing coverage scheduling algorithms to provide such assurances, coverage scheduling also relieves sensor nodes computation and storage overhead. To be of practical value, the coverage scheduling algorithm takes into account the effects of realistic environments on coverage such as the changes in sensing ability of nodes and the noise in distance measurements between node pairs.

The preliminary results of this research have appeared in [3, 4]. The rest of the paper is organized as follows. The related work is presented in Section 2. In Section 3, system model and assumptions are given. The problem we tackle is formally described in Section 4. Section 5 presents the CORA protocol. The performance evaluation of CORA is discussed in Section 6. The concluding remarks are made in Section 7.

2. Related Work

Coverage scheduling algorithms and distributed source coding techniques are usually addressed as two separate research fronts in the literature. CORA distinguishes itself from the existing solutions in closing the loop between these two research fronts.

Coverage Scheduling. Coverage scheduling in wireless sensor networks has received increasing attention in the recent years [5–15]. Some research efforts have considered coverage scheduling in conjunction with connectivity requirements [16, 17], bandwidth constraints [18], and target localization [19]. But, in none of the previously known coverage scheduling techniques, the impact of coverage scheduling on distributed source coding is addressed, even though distributed source coding efficiency depends on which set of sensor nodes gather data for an event. In comparison, CORA identifies the nodes that are critical for distributed source coding by keeping track of the event distribution. Then, only such critical nodes are put on duty to provide the variable coverage needed over the monitored area for minimizing event misses. CORA also uses less number of active nodes to meet the coverage needed than existing coverage scheduling techniques [5].

Rate Allocation. Distributed source coding [20, 21] and explicit communication [22] reduce the amount of data transmitted by taking advantage of the correlation structure between the sensor measurements. Let X and Y be the data sources at two neighboring sensor nodes that have overlapping sensing areas and belong to the same cluster. In case of distributed source coding, the data of X are sent as $H(X)$ to the clusterhead which then informs Y about the correlation structure between the data of X and Y. Finally, the data of Y are encoded as $H(Y \mid X)$ with side information X in accordance with the Slepian-Wolf theorem [20]. In case of explicit communication, the node with source Y receives explicit side information $H(X)$ from the node with source X and decodes $H(X)$. Then, the data of Y is encoded as $H(Y \mid X)$ and sent to their clusterhead in addition to $H(X)$.

Recently, rate assignment algorithms are developed to determine the coding rates to be used by each sensor node [22–24]. These rate assignment algorithms usually focus on minimizing energy consumption and do not consider the overhead of distributed source coding. In CORA however, cooperative coverage scheduling and rate assignment activates and maintains correlation information only for those critical nodes that would maximize the energy saving of distributed source coding.

The main drawback of aforementioned existing rate allocation algorithms is that they usually assume uniform distribution of event occurrences. In our work, we address both uniform and nonuniform event distributions and show the ideal correlator radius for each case. In light of the ideal correlator radius, our rate allocation algorithm partitions on duty nodes into groups, each with a designated correlator node, and assigns rates within each group in response to the uniform/nonuniform distribution of event occurrences.

Rate allocation and network lifetime are very related to each other. For instance, in [25], rate allocation in wireless sensor networks is addressed under a given network lifetime requirement. This research work advocates the use of lexicographical max-min rate allocation for all sensor nodes in order to avoid a severe bias in rate allocation when the objective is to maximize the sum of rates of all nodes.

Event Miss-Ratio Assurances. The problem of event-misses have been addressed by several node level scheduling algorithms [26–29]. The objective of such research efforts is to keep a node in low power modes as much as possible while processing event occurrences within their deadlines. However, these existing techniques do not provide any event miss-ratio assurances and schedule events at a sensor node separately from its neighboring nodes. Such algorithms can not relieve a sensor node from processing those events that are also detected by its neighboring nodes. In comparison, CORA assigns each event only to some sensor nodes within a group of cluster members. Therefore, CORA significantly reduces the event processing load at each node.

3. System Model and Terminology

Network and Node Model. This paper focuses on dense wireless sensor networks which consists of multiple clusters that are formed based on node residual energy levels. We consider single hop clusters where each cluster member is within the direct transmission range of the clusterhead. The sensor nodes are static and have the same hardware capabilities. The distance between any two sensor nodes within a cluster is assumed to be known based on standard methods such as Received-Signal-Strength-Indicator (RSSI) and Time of Arrival (ToA).

We consider the recent commercial off-the-shelf sensor nodes that have multiple sensing units [30]. The sensor nodes do not have an attached GPS device but their positions within a cluster are estimated based on internode distances. The power control circuit of each sensing unit enables putting it in sleep or active mode [31]. The sensing unit consumes constant power level in active mode. The radio unit has dynamic transmission power control capability which enables sensor nodes to control their transmission range under the effect of environmental factors and multipath propagation. In order to transmit n bits between two sensor nodes separated by a distance ℓ, the required power is modeled as $n(a\ell^2 + b)$ for transmission and nb for reception where a and b are constants [32].

Coverage and Event Model. In sensor networks, different types of coverage models exist, (i) the monitored area consists of a discrete set of points and each point should be within the sensing region of at least one active node [12], (ii) the monitored area is a predefined contiguous region and the active sensor nodes should be covering each point within this region [10], and (iii) the monitored area is a contiguous region defined by the union of all individual nodes' sensing regions, and a subset of active nodes should cover the sensing regions of others [11]. In this paper, we focus on the third coverage model. We consider both uniform and nonuniform event distribution models. In uniform case, each node has the same event occurrence probability. To model nonuniform event distribution, however, random hotspot locations are created within the monitored area, and event occurrences follow the Gaussian distribution around these hotspot locations.

Data Correlation Model. For n active sensor nodes that participate in source coding within the same group, a vector μ of n entries keeps the mean values of their sensor readings and an $(n \times n)$ matrix Γ maintains their correlation. $\mu(\overrightarrow{X}) = (\mu(X_1), \mu(X_2), \dots, \mu(X_n))^T$, where the random variable X_i corresponds to the source sensor readings at ith active cluster member for $1 \leq i \leq n$. $\Gamma(i, j)$ denotes the correlation degree between ith and jth active cluster member for $1 \leq i, j \leq n$.

The existing research efforts consider only internode distances while modeling sensor data correlation [33]. Given a set of nodes $\{X_1, \dots, X_{(i-1)}\}$, their model calculate the nonredundant data produced by node X_i based on the minimum Euclidean distance between node X_i and any node X_j for $1 \leq j \leq (i-1)$. However, this distance-based approach faces problems when node X_i is very close to more than one member of the node set $\{X_1, \dots, X_{(\mathrm{i}-1)}\}$. In this case, the amount of nonredundant data generated by X_i is actually much less than the amount that is approximated by their model. To address such cases and represent correlation more accurately, this paper models the data correlation among sensor nodes based on the overlapping area of their sensing regions. Specifically, the data correlation of active cluster members i and j is expressed as $\Gamma(i, j) = \text{Area}(S_i \cap S_j)/\pi r_s^2$, where S_i and S_j denote the sensing regions of ith and jth active cluster members, respectively, for $1 \leq i, j \leq n$ and r_s is the sensing range.

The total number of bits in the sensed data of a group of n sensor nodes is expressed based on their aggregate coverage. Specifically, if the entropy [34] of the source at a single cluster member is n_B bits then the sensor nodes produce $(n_B \times (\text{Area}(\cup_{i=1}^{n} S_i)/\pi r_s^2))$ bits of nonredundant data where S_i is the sensing region of node ith sensor node for $1 \leq i \leq n$. The number of entries in μ and Γ indicate the *storage overhead* of source coding which is equal to $(n \times 1) + (n \times n)$. The *update overhead* of source coding is determined by the modifications made on μ and Γ for each new event occurrence. Table 1 lists most of the notation used in the rest of the paper.

4. Problem Statement

Let us consider an event type M to be sensed by a wireless sensor network. Let us assume that a discrete set of n sensor nodes' sensing areas define the monitored area. Also let x_i denote the point where ith sensor node is located for $1 \leq i \leq n$. Let us consider a short interval where at most

Table 1: Notation.

r_{t}	The transmission range of any given node.
r_s	The sensing range of any given node.
S	The sensing region of any given node.
M	An event to be sensed by the wireless sensor network.
K	The set of sensor nodes in any given cluster.
n_K	The number of sensor nodes in any given cluster.
K_i	ith sensor node in any given cluster where $1 \le i \le n_K$.
Area (X)	Area of any given region X.
$H(Y)$	The entropy of any given random variable Y [34].
n_B	The number of bits needed to represent a sensor reading. If the sensor source at any given node is denoted by X is then $n_B = H(X)$.

one event can occur at location x_i. We define two random variables Y and D that correspond to the total number of event *occurrences* and the total number of event *detections*, respectively, within the monitored area. When the monitored area has total k event occurrences, Y assumes the value k for $0 \le k \le n$. Similarly, D takes the value ℓ when total ℓ event occurrences are detected among k number of event occurrences for $0 \le \ell \le k \le n$. Also let us define the inputs and outputs of the problem as follows.

(i) Inputs:

(a) γ_{req}: the required event miss-ratio threshold for event M,

(b) a_i: the sensing area of sensor node at location x_i for $1 \le i \le n$,

(c) $\text{Prob}(o_i)$: probability of event occurrence at a_i for $1 \le i \le n$,

(d) $\text{Prob}(Y = k)$: probability that Y takes value k for $0 \le k \le n$.

(ii) Outputs:

(a) γ_{obs}: the observed (actual) event miss-ratio for event M,

(b) $\text{Prob}(d_i)$: probability of event detection at a_i for $1 \le i \le n$,

(c) e_i: the energy consumption of active sensor nodes for data transmission with their assigned rates at location x_i for $1 \le i \le n$,

(d) $\text{Prob}(D = \ell)$: probability that D takes value ℓ for $0 \le \ell \le k \le n$.

The problem of this paper is to show that the proposed protocol CORA guarantees that the observed event miss-ratio γ_{obs} is less than or equal to the required event miss-ratio γ_{req}

$$\begin{pmatrix}\text{Observed event}\\ \text{miss-ratio } \gamma_{\text{obs}}\end{pmatrix} = \left[1 - \frac{E(D)}{E(Y)}\right] \le \begin{pmatrix}\text{Required event}\\ \text{miss-ratio } \gamma_{\text{req}}\end{pmatrix}, \tag{1}$$

and minimize the total energy consumption of all active sensor nodes

$$\text{Minimize} \sum_{i=1}^{n} e_i, \tag{2}$$

where $E(D)$ and $E(Y)$ (i.e., the expected values of D and Y, resp.) are written as

$$\begin{aligned} E(D) &= \sum_{\ell=0}^{n} [\ell \times \text{Prob}(D = \ell)], \\ E(Y) &= \sum_{k=0}^{n} [k \times \text{Prob}(Y = k)]. \end{aligned} \tag{3}$$

There are two extreme cases for γ_{obs}. At one extreme, the observed event miss-ratio is becoming equal to one when all event occurrences are missed. The other extreme occurs when all event occurrences are detected, thereby making the observed event miss-ratio equal to zero.

Among the aforementioned probability values, $textProb(o_i)$ depends on the distribution of event occurrences. $\text{Prob}(d_i)$ is controlled by the aggregate coverage of active sensor nodes. $\text{Prob}(Y = k)$ is computed by considering all possible combinations of the n locations for having total k events in the monitored area for $0 \le k \le n$. As an example, let us derive a close expression for $\text{Prob}(Y = 1)$ where there exists a single event in the entire monitored area. The probability, say p_i, that a single event occurs at location i is equal to the probability of event occurrence at location i times the product of all those probabilities of not having event occurrences at the remaining $(n-1)$ grid locations for $1 \le i \le n$. $\text{Prob}(Y = 1)$ is equal to the sum of all p_is over n locations. Thus,

$$\text{Prob}(Y = 1) = \sum_{i=1}^{n} \left[\text{Prob}(o_i) \times \prod_{h=1, h \ne i}^{n} (1 - \text{Prob}(o_h)) \right]. \tag{4}$$

Since there is no close formula for $\text{Prob}(Y = k)$ for $2 \le k \le n$, we provide Algorithm Probability Computation for Y to express $\text{Prob}(Y = k)$. $\text{Prob}(D = \ell)$ is computed similar to $\text{Prob}(Y = k)$ with the exception that it depends on not only

```
Require: $Y = k$: The case of having total $k$ event occurrences in the monitored area where $0 \leq k \leq n$ and $n$ is the
         number of sensor locations.
Ensure: Prob$(Y = k)$ is computed.
Initialize Prob$(Y = k) = 0$.
Determine the set $A$ of all possible combinations of $k$ events that occur in any $k$ out of $n$ network locations.
while the set $A$ is not empty do
    Remove a combination from $A$ and denote it by $C$.
    Find out the product of the event occurrence probabilities at the locations selected by combination $C$ and
      probabilities of not having any event occurrences at the remaining locations of the network.
    Multiply the product from the previous step with the probability of having combination $C$ and add
      the result of multiplication to Prob$(Y = k)$.
end while
```

Algorithm 1: Probability computation for Y.

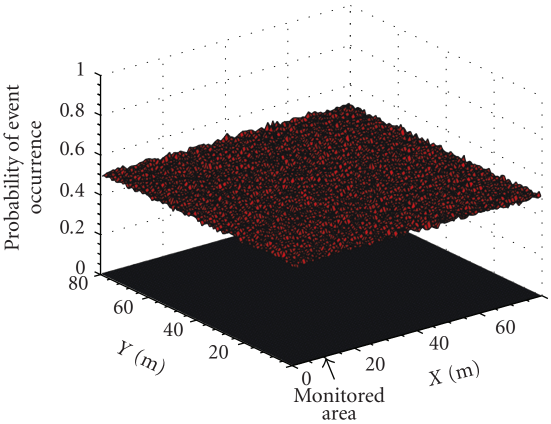


Figure 1: A uniform event distribution is shown for the black colored monitored area. A nonzero event occurrence probability at any point is indicated in red color.

the probability of event occurrences but also their probability of detection. For example, Prob$(D = 1)$ is expressed as

$$\text{Prob}(D = 1) = \sum_{i=1}^{n}\Big[(\text{Prob}(o_i) \times \text{Prob}(d_i)) \times W_i\Big], \tag{5}$$

where

$$W_i = \prod_{h=1, h \neq i}^{n} (1 - \text{Prob}(o_h)\text{Prob}(d_h)). \tag{6}$$

5. CORA Protocol for Event Miss-Ratio Assurances

We first give in this section an overview of how CORA provides event miss-ratio assurances with minimum energy consumption. The network lifetime is divided into consecutive *rounds* and CORA performs coverage scheduling and rate assignment at the beginning of each round. The round-based operation enables CORA to take the recent node residual energy information into account. At each location x_i, coverage scheduling and rate assignment control Prob(d_i) and e_i,respectively, for $1 \leq i \leq n$. Meanwhile, CORA enables event-driven cooperation of coverage scheduling and rate assignment in two phases.

In *Phase I*, event distribution is identified using a continuously updated histogram. Depending on whether the event distribution is uniform or nonuniform, the ideal correlator radius is found for each cluster. As we will show, there exists a single correlator radius within the cluster for uniform event distribution such as the one depicted in Figure 1. For nonuniform event distributions as illustrated in Figure 2, a local correlator radius exists for each hotspot. The local correlator radii of all hotspots are combined to find the cluster level global correlator radius. Under the guidance of the correlator radius, correlator nodes are selected and cluster members form groups around correlator nodes for rate assignment.

In *Phase II*, those critical nodes which significantly improve the energy consumption of the cluster are identified from their assigned rates. Coverage scheduling then activates only those critical nodes for improving energy consumption and meeting the required event miss-ratios. Since *Phase I* differs for uniform and nonuniform event distributions, the steps of *Phase I* are described next separately for uniform and nonuniform event distributions.

5.1. Phase I for Uniform Event Distribution.

The expected amount of data generated by each node is the same under uniform event distribution. Therefore, the locations of sensor nodes within the cluster topology plays the key role for overall energy consumption. In this section, we study how distance of correlator nodes to the clusterhead affects the energy expenditure. Let us refer to this distance as r_c where the subscript denotes the *correlator radius* and $0 \leq r_c \leq r_t$. Let r_c^* be the *ideal* r_c value which maximizes the energy gain of distributed source coding by correlator nodes relative to the case when cluster members transmit data directly to the clusterhead. In what follows, we will explain how r_c^* is found in Step I of Protocol CORA by formulating this energy gain as a function of r_c. We will then describe the allocation of coding rates in accordance with r_c^*.

```
Require: $M$: An event to be sensed by sensor nodes in a cluster-based wireless sensor network.
    $\gamma_{\text{req}}$: The required event miss-ratio threshold.
Ensure: i) The event miss-ratio assurance $\gamma_{\text{obs}} \leq \gamma_{\text{req}}$ is met.
    ii) $\sum_{i=1}^{n} e_i$ is minimized.
{PHASE I}
Step I: Each clusterhead finds its correlator radius.
Step II: Those cluster members that are located around the correlator radius form rate assignment groups, each consisting of a
        correlator node and a number of sensor nodes.
Step III: Each correlator node assigns the source coding rates to its group members in order to minimize $\sum_{i=1}^{n} e_i$.
{PHASE II}
Step IV: Each clusterhead identifies the critical nodes of rate assignment groups, based on their assigned rates.
Step V: The members of each cluster perform coverage scheduling by activating critical nodes, in order to meet the required event
        miss-ratio threshold $\gamma_{\text{obs}} \leq \gamma_{\text{req}}$.
```

Algorithm 2: Protocol CORA.

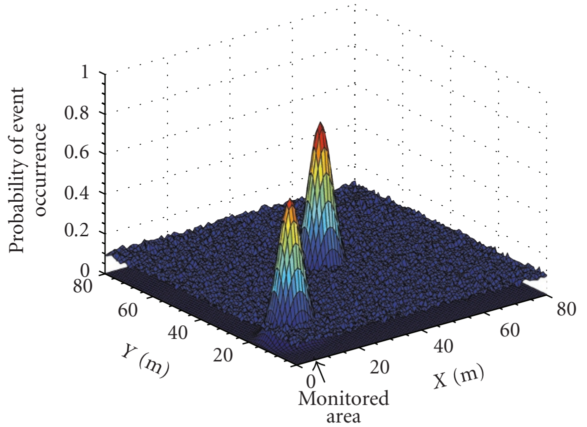


Figure 2: A nonuniform event distribution is shown for the monitored area illustrated in black color. A nonzero probability of event occurrence at any point is depicted in blue color.

We now analyze the energy consumption for a cluster of wireless sensor nodes when distributed source coding is applied at a given distance r_c from the clusterhead (i.e., when correlator nodes have distance r_c to the clusterhead). The first question we answer in this analysis is the following: For which cluster members would correlator nodes yield any improvement in energy consumption? Especially within the close proximity of the clusterhead, transmitting data directly to the clusterhead (i.e., bypassing the correlators) should be preferable over sending data through the correlator nodes.

Let us call the union of those locations from where the sensed data should be sent directly to the clusterhead as *Direct Transmission Region (DTR)*. Let r_d denote the radius of DTR where the subscript d refers to direct data transmission. The donut shaped region obtained by excluding the DTR from the clusterhead transmission region is referred to as *Distributed Source Coding Region (DSCR)*. Figure 3 illustrates DTR and $DSCR$ in blue and red colors, respectively.

To identify DTR, we find the relationship between r_d and r_c. Specifically, we consider the energy consumed for gathering data from any cluster member K_i with distance r_d to the clusterhead where $1 \leq i \leq n_K$. Note that the energy consumption of node K_i should be the same in direct data transmission and distributed source coding cases since node K_i is located on the boundary of DTR. Let us refer to the energy consumption of node K_i when it uses direct transmission to the clusterhead as e_{direct}. Let e_{indirect} be the energy consumption of K_i when its data reach the clusterhead indirectly through correlator nodes.

When node K_i bypasses the correlators, the entropy of the source at node K_i is transmitted over distance r_d to the clusterhead. Based on the energy model given in Section 3, e_{direct} would be

$$e_{\text{direct}} = n_B\left(ar_d^2 + 2b\right). \tag{7}$$

When correlator nodes apply distributed source coding at distance r_c from the clusterhead, K_i sends only nonredundant data to the correlator node which are then forwarded to the clusterhead. Thus, we next consider the expected amount of data transmitted by node K_i when data redundancy is eliminated by distributed source coding.

The redundancy in node K_i's data is attributed to the overlap of node K_i's sensing region with the sensing regions of its neighboring nodes. Consequently, we need to find how much of the area within the sensing range of a node is covered only by the node itself. When there is coverage redundancy within the cluster, the expected size of the common area covered by node K_i with its neighboring nodes is

$$\left(\frac{n_K \pi r_s^2 - \pi (r_t + r_s)^2}{n_K}\right), \tag{8}$$

where $\pi(r_t + r_s)^2$ refers to the coverage area of the cluster and the term $n_K \pi r_s^2$ denotes the maximum area that can be covered by cluster members. If there is no coverage redundancy (i.e., no overlap among sensing regions of neighboring nodes), then the size of the common area becomes zero. Following these observations, let us define ω as the ratio of a node's sensing region that is covered exclusively by the node itself where $0 \leq \omega \leq 1$. With the assumption that

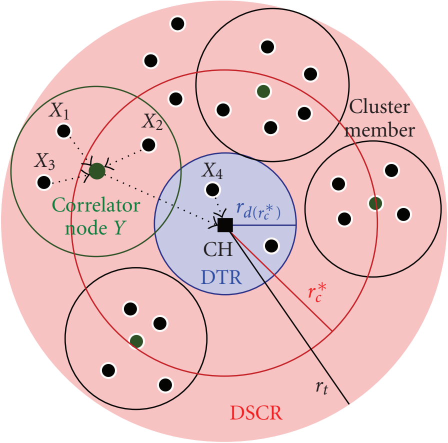


Figure 3: A correlator node Y is illustrated with its members X_1, X_2, and X_3. The ideal correlator radius r_c^* guides the formation of such rate assignment groups in Distributed Source Coding Region ($DSCR$) shaded in red color. The cluster members such as node X_4 in blue colored Direct Transmission Region (DTR) send data directly to the clusterhead.

coverage redundancy is uniformly distributed among cluster members, ω is expressed as

$$\omega = \begin{cases} \left(\dfrac{\pi r_s^2 - \left(\left(n_K \pi r_s^2 - \pi(r_t + r_s)^2\right)/n_K\right)}{\pi r_s^2}\right), \\ \qquad\qquad n_K \pi r_s^2 > \pi(r_t + r_s)^2 \\ 1, \qquad\qquad \text{otherwise.} \end{cases} \tag{9}$$

Let us also assume that the task of covering the common areas (i.e., the regions that are covered by at least two cluster members) is evenly distributed among all cluster members. We introduce the variable β as the expected ratio of the common areas covered by a sensor node to its overall sensing region where $0 \le \beta \le 1$. Since any area within the cluster that is not covered only by a single sensor node belongs to the common area, β is expressed as

$$\beta = \begin{cases} \left(\dfrac{\left(\left(\pi(r_t + r_s)^2 - n_K \omega \pi r_s^2\right)/n_K\right)}{\pi r_s^2}\right), \\ \qquad\qquad \pi(r_t + r_s)^2 > n_K \omega \pi r_s^2 \\ 0, \qquad\qquad \text{otherwise.} \end{cases}. \tag{10}$$

From the exclusive and common areas covered by node K_i, we can now determine the amount of nonredundant data transmitted by node K_i and express e_{indirect}. Specifically,

$$e_{\text{indirect}} = \begin{pmatrix} n_B(\omega + \beta + \varepsilon)\left[\left(a(r_c - r_d)^2 + 2b\right)\right] \\ + n_B(\omega + \beta)\left(a r_c^2 + 2b\right)\right] \end{pmatrix}, \tag{11}$$

where ε represents the ratio of the rate control message size to the data message size. First, rate information is transmitted from the correlator to node K_i over distance $(r_c - r_d)$. Then, node K_i codes sensor data with the assigned rate. The data received by the correlator node are transmitted to the clusterhead over distance r_c.

Let $r_{d(r_c)}$ refer to the r_d value substitution of which to (7) and (11) yields the same energy consumption for e_{direct} and e_{indirect}, respectively. The term r_c is included in $r_{d(r_c)}$ subscript to indicate that r_d is dependent on the r_c value. Since $r_{d(r_c)}$ refers to the radius of DTR, a cluster member uses direct data transmission to the clusterhead if its distance to the clusterhead is less than $r_{d(r_c)}$. Otherwise, the data of the cluster member are sent to its designated correlator node. In light of this selection criteria for data transmission, we next define the energy gain function of correlator nodes for any given r_c value.

Let e_{nsc} denote the energy consumption for gathering data from the cluster members when they bypass the correlator nodes and transmit their data directly to the clusterhead with the subscript referring to no source coding. Let e_{dsc} be the energy expenditure for the case where cluster members use correlator nodes with the subscript denoting distributed source coding. e_{dsc} is a function of r_c and is defined within the interval $[0 - r_t]$ while e_{nsc} is independent from r_c. Then, the *energy gain* due to the correlator nodes is defined as e_{gain} where

$$e_{\text{gain}} = e_{\text{nsc}} - e_{\text{dsc}}. \tag{12}$$

By incorporating each possible location of a cluster member within the cluster, e_{dsc} is expressed as

$$e_{\text{dsc}} = (A + B), \tag{13}$$

where A and B refer to the expected energy consumption of cluster members in DTR and $DSCR$, respectively. Specifically,

$$\begin{aligned} A &= \int_0^{r_{d(r_c)}} \left[P_r n_B\left(a r^2 + 2b\right)\right] dr, \\ B &= \int_{r_{d(r_c)}}^{r_c} \left[P_r \begin{pmatrix} n_B\ (\omega + \beta + \varepsilon)\left(a(r_c - r)^2 + 2b\right) \\ + n_B\ (\omega + \beta)\left(a r_c^2 + 2b\right) \end{pmatrix}\right] dr \\ &\quad + \int_{r_c}^{r_t} \left[P_r \begin{pmatrix} n_B\ (\omega + \beta + \varepsilon)\left(a(r - r_c)^2 + 2b\right) \\ + n_B\ (\omega + \beta)\left(a r_c^2 + 2b\right) \end{pmatrix}\right] dr\ , \end{aligned} \tag{14}$$

where $P_r = (2\pi r n_K / \pi r_t^2)$ that is the probability of having a sensor node with distance r to the clusterhead. The first and second terms of (14) represent the energy consumption of cluster members in $DSCR$ that are closer to and farther from the clusterhead than correlator nodes, respectively. On the other hand, e_{nsc} is defined by considering the direct transmission from each cluster member as

$$e_{\text{nsc}} = \int_0^{r_t} \left[P_r n_B\left(a r^2 + 2b\right)\right] dr. \tag{15}$$

By subtracting e_{dsc} from e_{nsc}, e_{gain} is determined and the ideal correlator radius is found as the r_c value which yields the maximum e_{gain}. Specifically, e_{gain} is a dome-shaped function of r_c with its derivative having value zero at point r_c^*. Figure 4 illustrates the behavior of e_{gain} along with the ideal correlator radius r_c^*.

```
Require: A cluster of sensor nodes that produce correlated data for a given event $M$.
Ensure: The coding rates are allocated for event $M$ to minimize the energy consumption.
The ideal correlator radius $r_c^*$ is determined by the clusterhead for uniform event distribution.
for all Active node $K_i, 1 \leq i \leq n_K$ do
    if $K_i$ is located within $DTR$ then
        Use coding rate $R_{K_i} = H(K_i)$ and transmit data directly to the clusterhead.
    else
        Check whether node $K_i$ should become a correlator node by verifying the residual energy requirement and comparing its
         distance to the clusterhead with $r_c^*$.
        if $K_i$ is selected as correlator then
            Code data at rate $R_{K_i} = H(K_i)$ and apply differential data transmission for group members.
        else
            Use coding rate $R_{K_i} = H(K_i|\Psi)$ and send data to the correlator node where the set $\Psi$ contains the group members with
             higher residual energy.
        end if
    end if
end for
```

Algorithm 3: Coding rate assignment.

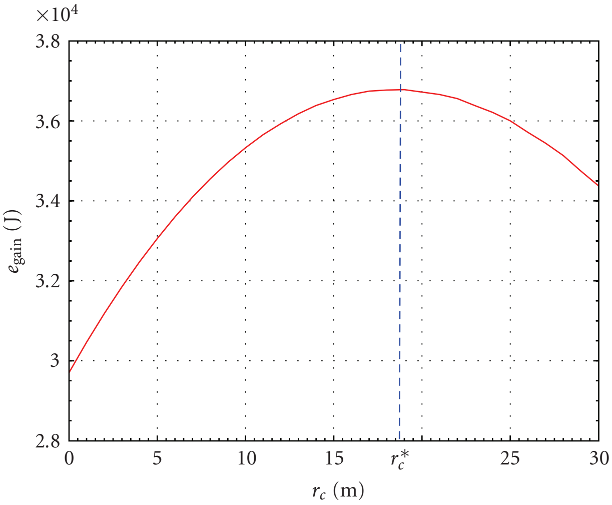


Figure 4: e_{gain} is shown for a cluster of 500 sensor nodes with settings $r_s = 10$ m., $r_t = 30$ m. (i.e., cluster range), $n_B = 16$ bits and $\varepsilon = 0.2$. The ideal correlator radius r_c^* within the interval $[0 - r_t]$ is indicated with the dashed blue line.

Algorithm Coding Rate Assignment describes how rate allocation groups are formed under guidance of r_c^* and rates are assigned to group members at Step II and Step III of Protocol CORA, respectively. To select the correlator nodes, the average residual energy of cluster members is calculated first. A cluster member within *DSCR* becomes a correlator node if (i) it has higher residual energy than the average residual energy, and (ii) its distance to the clusterhead is closer to r_c^* than any one of its neighboring nodes. The remaining active nodes join the group of the closest correlator node.

Within each group, the data transmissions from group members to the correlator node are ordered in descending order of node residual energy levels. Thus, a sensor node uses the data of other group members that have higher residual energy as side information. Note that, having a high residual energy node transmit data earlier than others does not guarantee that it will use the highest coding rate. This is especially true when such a node has highly overlapping sensing region with other high energy nodes that have already transmitted data. However, when a high energy node uses low coding rate, its residual energy would not change significantly while other nodes would be spending more energy. Consequently, CORA would schedule such group members to transmit data earlier than others in the following rounds.

The correlator node applies the proposed differential data transmission technique to further reduce the size of the data before sending it to the clusterhead. The basic motivation behind differential data transmission is that the data gathered from group members exhibit not only spatial correlation but also temporal correlation. To exploit the correlation in temporal domain, both clusterhead and correlator node maintain a common *reference data* which captures the characteristic values of last n data values transmitted from the correlator node to the clusterhead.

Let R be the reference data and W be the raw nonredundant data gathered at the correlator node. Let $\ominus$ denote the difference operator and $\oplus$ refer to its inverse. As an example, if the sensor readings are mapped to numbers than $\ominus$ and $\oplus$ may correspond to the subtraction and addition, respectively. Then, the correlator node transmits the differential data D with the value $D = (W \ominus R)$. After receiving D, the clusterhead recovers the original raw data $W = (D \oplus R)$.

5.2. Phase I for Nonuniform Event Distribution. CORA determines the ideal rate assignment for a nonuniform event distribution by considering all hotspots of event occurrences. For each hotspot, a *local* ideal correlator radius is computed first. Then, local ideal correlator radii for different hotspots are combined to determine the *global* ideal correlator radius for the cluster. In the following analysis, we describe the

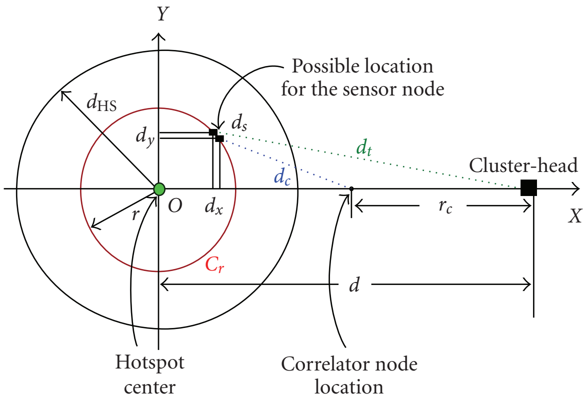


Figure 5: The analysis of energy consumption for a hotspot that is located at the origin. The hotspot has distance d to the clusterhead and correlator radius r_c is used within the cluster.

computation of the ideal correlator radius r_c^* for a hotspot with distance d to the clusterhead as shown in Figure 5.

In the analysis, we do not follow a particular event distribution but assume that event occurrence probabilities are nonzero within the local proximity of hotspots only. In particular, let $\text{Prob}(e_r)$ be the event occurrence probability at a point with distance r to the hotspot center. $\text{Prob}(e_r)$ can be calculated either by using exponential and Pareto distributions depending on the nature of the event occurrences. In exponential distribution, the probability density function is $\text{Prob}(e_r) = (ae^{-ar})$ where the event occurrence probability decreases rapidly as the distance to the hotspot center increases. For those hotspots with a slowly decreasing effect, Pareto distribution is more suitable due to its density function $\text{Prob}(e_r) = (ak^a/r^{a+1})$ where a is the shape parameter and k is the initial value.

As in uniform event distribution analysis, we will express the energy gain function e_{gain} in terms of r_c where $0 \le r_c \le d$. However, unlike the uniform event distribution case, the analysis is shaped by those locations near hotspots that have nonzero event occurrence probabilities rather than all cluster locations. Let us refer to the maximum distance between the center of the hotspot and any node within the hotspot as the *hotspot range* and denote it by d_{HS}.

In computation of both e_{dsc} and e_{nsc}, we will consider each possible distance r that a sensor node can have to the hotspot center where $0 \le r \le d_{\text{HS}}$. In Figure 5, a red circle labeled C_r illustrates the possible node locations with distance r to the hotspot center. For the upper semicircle of C_r as an example, such locations are defined by the coordinates (x, y) where $-r \le x \le r$ and $y = \sqrt{r^2 - x^2}$. Let $e_{\text{dsc}(r)}$ denote the expected energy consumption of the nodes located on C_r when they use correlator nodes. Let $e_{\text{nsc}(r)}$ refer to the expected energy expenditure of the same nodes on C_r when they apply direct data transmission to the clusterhead. We will integrate $e_{\text{dsc}(r)}$ and $e_{\text{nsc}(r)}$ over all C_r within the hotspot to find e_{dsc} and e_{nsc}, respectively.

To compute $e_{\text{dsc}(r)}$, we break up C_r into small segments ds with each segment representing a possible location of a sensor node on C_r. For each one of these segments, the event occurrence probability is the same and is equal to $\text{Prob}(e_r)$. From Figure 5, it can be observed that

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \qquad (16)$$
$$dx = \sqrt{1 + y'}\,dx.$$

Therefore,

$$e_{\text{dsc}(r)} = 2\int_{-r}^{r} B \times \left[\text{Prob}(e_r)\frac{r/\sqrt{r^2 - x^2}}{2\pi r}\right] dx, \qquad (17)$$

where

$$B = n_B \times \left[(\omega + \beta + \varepsilon)\left(ad_c^2 + 2b\right) + (\omega + \beta)\left(ar_c^2 + 2b\right)\right], \qquad (18)$$

and $d_c^2 = (d - r_c - x)^2 + (r^2 - x^2)$. In words, when there is a sensor node on C_r, the probability of event data transmission from ds is equal to the probability that the node is placed at ds (i.e., $ds/2\pi r$) *and* the probability that an event occurs (i.e., $\text{Prob}(e_r)$). The energy consumption for sending data from ds to the clusterhead consists of two components. First, the data are transmitted from ds to the correlator node over distance d_c, and then sent by correlator node to the clusterhead over distance r_c.

To determine $e_{\text{nsc}(r)}$, we follow the same segment-based division of C_r but assume that any sensor node on C_r bypasses the correlator nodes. Thus,

$$e_{\text{nsc}(r)} = 2\int_{-r}^{r} \left[n_B\left(ad_t^2 + 2b\right)\right] \times \left[\text{Prob}(e_r)\frac{r/\sqrt{r^2 - x^2}}{2\pi r}\right] dx, \qquad (19)$$

where $d_t^2 = (d - x)^2 + (r^2 - x^2)$.

To express e_{dsc} and e_{nsc}, we integrate over each r and incorporate the expected number of nodes to be located on C_r for $0 \le r \le d_{\text{HS}}$. Specifically,

$$e_{\text{dsc}} = \int_0^{d_{\text{HS}}} \left[n_K \frac{2\pi r}{\pi r_t^2} e_{\text{dsc}(r)}\right] dr,$$
$$e_{\text{nsc}} = \int_0^{d_{\text{HS}}} \left[n_K \frac{2\pi r}{\pi r_t^2} e_{\text{nsc}(r)}\right] dr. \qquad (20)$$

Finally, we take the derivative of the energy gain function e_{gain} where $e_{\text{gain}} = e_{\text{nsc}} - e_{\text{dsc}}$. The ideal correlator radius r_c^* makes the derivative function e'_{gain} equal to zero at point r_c^*.

Given a set of k hotspots with the local ideal correlator radius for the jth hotspot referred to as $r_c^*(j)$, the global ideal correlator radius is set as $r_c^* = \sum_{j=1}^{k} r_c^*(j)/k$ where $1 \le j \le k$. Once the value of r_c^* is obtained, the selection of correlator nodes and their group members are the same with the uniform event distribution case. Within a group however, the rates are allocated based on the proximity of nodes to the hotspots. Specifically, a sensor node uses the data of other group members that have larger distances to the hotspot as side information. The motivation here is that

```
Require: A cluster of wireless sensor nodes to be scheduled for sensing event $M$.
  $\gamma_{\text{req}}$: The required event miss-ratio threshold for event $M$.
  $\chi$: The scores assigned to cluster members based on coding rate estimates.
Ensure: Each cluster member is selected either as an active node or a sleeping node for event $M$.
The set $K$ of cluster members are sorted in ascending order of their scores in $\chi$.
for all Node $K_i, 1 \le i \le n_K$ do
    Prob($d_i$) is identified from $\gamma_{\text{req}}$ and event distribution for $M$.
    if Prob($d_i$) is met by neighboring nodes of node $K_i$ then
        Node $K_i$ goes to sleep mode.
    else
        Node $K_i$ is activated for $M$.
    end if
end for
```

Algorithm 4: Coverage scheduling.

those group members that are near the hotspots consume more energy due to their high event detection activity. Thus, CORA alleviates the burden on such nodes by increasing their side information in distributed source coding.

5.3. Phase II for Both Uniform and Nonuniform Event Distributions. This section presents the steps of Phase II that are implemented in the same way for both uniform and nonuniform event distributions. CORA employs a priority-based coverage scheduling technique to select the active sensor nodes for each event M. In coverage scheduling, a node that is considered earlier for activation has better chance of going to sleep mode since those nodes that are waiting to be activated can still meet the required event miss-ratio. Based on this observation, CORA implements node priorities to have critical nodes considered later in coverage scheduling, thereby forcing them to remain active. In Step IV, each cluster member K_i is associated with a score χ_i that reveals how much it improves the energy expenditure of the cluster for $1 \le i \le n_K$. χ_i is computed by estimating the rates within the neighborhood of node K_i for the following two cases:

Case I. Node K_i participates in distributed source coding.

Case II. Node K_i is excluded from distributed source coding.

The rate predictions are made according to the proposed rate assignment algorithms. Then, χ_i is set as the ratio of the expected average residual energy of the neighboring nodes for Case I to the one for Case II. Consequently, the critical nodes get assigned the highest scores.

Algorithm Coverage Scheduling implements Step V of Protocol CORA for the assignment of events to cluster members. The clusterhead first sorts the nodes in ascending order of their scores such that low score nodes are considered first. Then, each node K_i is evaluated for coverage scheduling in the order of its score where $1 \le i \le n_K$. The status of node K_i for event M is determined according to its sensing region's overlap with the sensing regions of its neighboring nodes. To control event misses, either (i) node K_i remains active and covers the physical area within its sensing region on its own, or (ii) node K_i goes to sleep mode and lets its neighboring nodes provide coverage over its sensing region.

Let U denote the set of neighboring nodes for K_i that can contribute to the coverage of K_i's sensing region where $1 \le i \le n_K$. Based on the evaluation of scores, the clusterhead creates an *eligible set* of neighboring nodes U_e for coverage of node K_i's sensing region where $U_e \subseteq U$. Each neighboring node $K_\ell \in U_e$ satisfies either one of the following conditions where $1 \le \ell \le n_K$ and $i \ne \ell$.

(i) $\chi_\ell > \chi_i$.

(ii) $\chi_\ell = \chi_i$ and node K_ℓ has already been scheduled as an active node or has not announced its status yet for M.

(iii) $\chi_\ell < \chi_i$ and node K_ℓ has already been scheduled as an active node for M.

Once its eligible set of neighboring nodes are identified, node K_i computes the uncovered part of its sensing region for event M denoted by α where $0 \le \alpha \le 1$. If $\alpha \le (1 - \text{Prob}(d_i))$, then node K_i goes to sleep mode for event M since its sensing region is covered sufficiently by neighboring nodes that would remain active. Otherwise, node K_i is scheduled active to preserve Prob(d_i) for event M.

To ensure that $\gamma_{\text{obs}} \le \gamma_{\text{req}}$, each cluster member K_i identifies its Prob(d_i) from γ_{req} and event distribution such that $0 \le (1 - \text{Prob}(d_i)) \le \gamma_{\text{req}}$. Let us define Prob($o_{\max}$) and Prob($o_{\min}$) as the maximum and minimum probability of an event occurrence, respectively, for M at any point within the monitored region. Specifically, $\text{Prob}(o_{\max}) = \max(\text{Prob}(o_i))$ and $\text{Prob}(o_{\min}) = \min(\text{Prob}(o_i))$ for $1 \le i \le n$ where n is the sensor locations in the monitored region. For node K_i located at x_i, Prob(d_i) is set as

$$\text{Prob}(d_i) = 1 - \gamma_{\text{req}}\left(1 - \frac{(\text{Prob}(o_i) - \text{Prob}(o_{\min}))}{(\text{Prob}(o_{\max}) - \text{Prob}(o_{\min}))}\right), \quad (21)$$

where $1 \le i \le n$. Consequently, CORA provides variable coverage over the monitored area with the worst and best coverage defined by the following two cases. When Prob(o_i) is equal to Prob($o_{\min}$), node K_i meets the required event miss-ratio threshold by setting $\text{Prob}(d_i) = (1 - \gamma_{\text{req}})$. On the other hand, if Prob(o_i) is equal to Prob($o_{\max}$), then node K_i detects all event occurrences by using $\text{Prob}(d_i) = 1$.

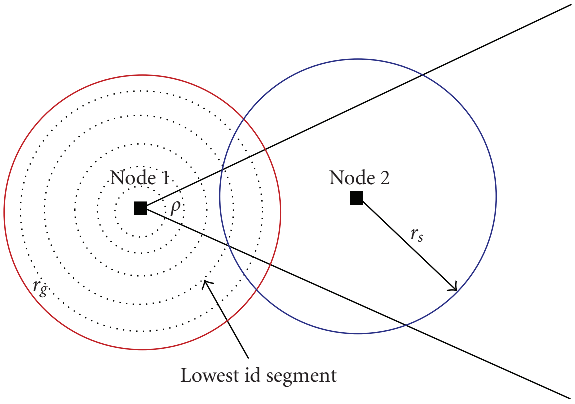


Figure 6: An example where the sensing region of nodes are divided into $n_G = 6$ segments. The covered portion of each segment by neighboring nodes is represented with an angle. In the figure, ρ corresponds to the covered portion of the lowest id segment, Sl_5, of node 1's sensing region by its neighboring node 2.

Lemma 1. *The variable coverage provided by CORA guarantees that $\gamma_{\text{obs}} < \gamma_{\text{req}}$ for any given event M.*

Proof of Lemma 1. The proof is given in the Appendix. □

We next present an efficient algorithm to compute the area covered within a node's sensing region by its neighboring nodes. The coverage computation is used to verify that a node's sensing region has sufficient overlap with its neighboring nodes before the node goes to sleep mode. While computing coverage, a sensor node considers its neighboring nodes in ascending order of their identifiers. For each neighboring node, the area of intersection that's not covered by lower id nodes is determined. This helps the overlapping areas between the sensing regions of neighboring nodes to be considered only once.

Algorithm Coverage Computation describes the steps taken for finding the area of overlap a node has with its neighboring nodes. The sensing region of each node is represented as a set of n_G segments $\{Sl_1, Sl_2, \ldots, Sl_{n_G}\}$ as shown in Figure 6 where $n_G \geq 1$. Each segment has the same thickness r_g and the union of all segments is equal to the node's sensing region where $r_s = n_G * r_g$. The segments are numbered in increasing order from the center of the sensing region towards its boundary.

For a given neighboring node, the segment with the lowest id that intersects with the sensing region of the neighboring node is determined. Then, for each segment with a higher id than the lowest id segment, the covered part of the segment by the neighboring node is represented with an angle ρ. This angle is defined with respect to the intersection points of the segment and the neighboring node's sensing region as illustrated in Figure 6. By finding the overlap with the sensing regions of all neighboring nodes, the covered portions of any given segment j are represented with a set of angles Ω_j where $1 \leq j \leq n_G$. The covered portion of segment j is found based on the comparison of 2π and the sum of magnitudes for the angles in Ω_j. The total covered portion of the sensing region is determined by incorporating each segment in proportion with its area.

Example 1. Let us consider a particular segment j within the sensing region of node 1 for which the neighboring nodes 2, 3, and 4 can contribute to its coverage. Suppose

$$\Omega_j = \left\{< 0, \frac{\pi}{3} >, < \frac{2\pi}{3}, \pi >\right\} \tag{22}$$

initially based on the contributions of nodes 2 and 3 to the coverage of the jth segment. If the angle obtained for the coverage contribution of node 4 is found to be $< \pi/4, \pi/2 >$, then Ω_j is updated as

$$\Omega_j = \left\{< 0, \frac{\pi}{2} >, < \frac{2\pi}{3}, \pi >\right\}. \tag{23}$$

Finally, based on the magnitudes of each angle, the covered area of segment j is determined as

$$\sum(\Omega_j) = \left(\frac{5\pi/6}{2\pi}\right)\left(\frac{\pi r_g^2\left(j^2 - (j-1)^2\right)}{\pi r_s^2}\right), \tag{24}$$

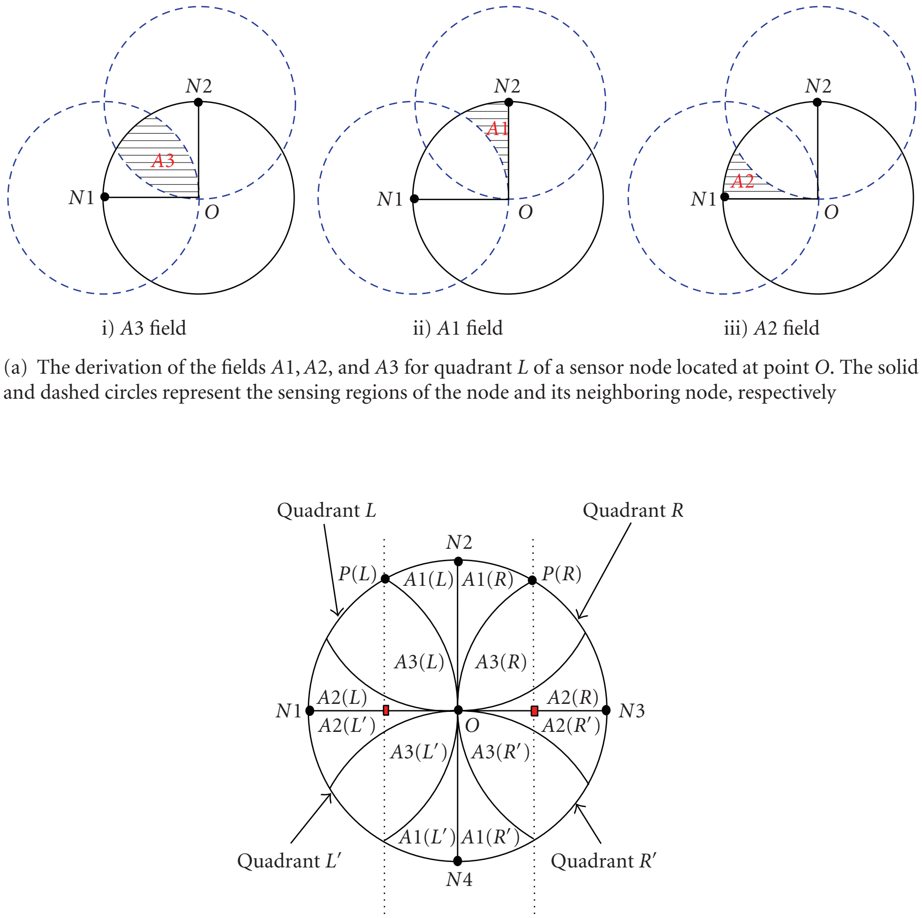
where $\sum(X)$ refers to sum of the magnitudes of angles in any given set X.

The accuracy of the proposed coverage computation technique improves with increasing number of segments. However, using a large number of segments also comes at the expense of higher computational and storage overhead. We next describe how the overhead of the segment-based representation is minimized. The key idea is to limit the application of segment-based division scheme to specific parts of the sensing region only. The remaining parts of the sensing region are divided into subregions at a higher level of granularity than segments, specifically into four quadrants. These quadrants have the special property that the full coverage of each quadrant can be verified in a very simple manner. The segment-based representation is then applied only within those quadrants for which the full coverage cannot be verified.

Let us now focus on the coverage of any given quadrant within a node's sensing region by the neighboring sensor nodes. We refer to the four quadrants within the sensing region as L (upper left), L' (down left), R (upper right), R' (down right) as shown in Figure 7(b). Without loss of generality, let us consider the full coverage of L, the upper left quadrant, shown in Figure 7(a). In order to lay down the placement constraints of neighboring sensor nodes for the full coverage of quadrant L, we first address the random placement of a single neighboring node within quadrant L. In any outcome of this experiment, the extreme points at which the neighboring node can be located are $N1$ and $N2$ shown as dots in Figure 7(a). The circles with dashed lines in the figure indicate the sensing region of the neighboring node when it would be located at $N1$ and $N2$. These boundary cases are important in the sense that they reveal the field $A3$ in which the placement of the neighboring node yields the full coverage of the quadrant.

```
Require: Any given cluster member that has overlapping sensing regions with the set neighboring nodes $Z$ where the $i$th
         neighboring node is referred to as $Z_i$.
Ensure: $\Upsilon$: The area within the cluster member's sensing region that is not covered by the neighboring nodes from the set $Z$ is
         computed.
Initialize $\Upsilon$ to zero.
Divide the node's sensing region into $n_G$ segments each of width $r_g$.
Let $\Omega_j$ be the set of angles that represent the covered portions of the $j$th segment. Initialize $\Omega_j$ to $\emptyset$ for $1 \le j \le n_G$.
Sort the set of neighboring nodes $Z$ in ascending order of node identifiers.
for all $i = 1$ to $|Z|$ do
    Identify the lowest id segment that intersects with node $Z_i$'s sensing region.
    for all $j$ = lowest id segment to $n_G$ do
        Determine the portion of segment $j$ that is covered by node $Z_i$ but not by any other node $Z_k$ and represent this covered
        portion with the angle $\rho$ where $1 \le k < i \le |Z|$.
        Add $\rho$ as an element to the set $\Omega_j$.
    end for
end for
for all $j = 1$ to $n_G$ do
    Add $(((2\pi - \sum(\Omega_j))/2\pi)(\pi r_g^2(j^2 - (j-1)^2)/\pi r_s^2))$ to $\Upsilon$ where $\sum(\Omega_j)$ denote the sum of magnitudes of the angles in $\Omega_j$.
end for
return$\Upsilon$
```

Algorithm 5: Coverage computation.



i) $A3$ field ii) $A1$ field iii) $A2$ field

(a) The derivation of the fields $A1, A2$, and $A3$ for quadrant L of a sensor node located at point O. The solid and dashed circles represent the sensing regions of the node and its neighboring node, respectively

(b) The fields in each quadrant are distinguished by appending the quadrant name $(L, L\prime, R, R\prime)$ to the end of the field name

Figure 7: The division of a node's sensing region into four quadrants to analyze its coverage.

Table 2: The conditions for node M to be placed in a specific quadrant and field over node N's sensing region.

Field	Condition for node M to be located in this field
$A2(L)$	$(\lvert(x_{N1}, y_{N1}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N2}, y_{N2}), (x_M, y_M)\rvert > r_s), (y_M \geq y_N)$
$A3(L)$	$(\lvert(x_{N1}, y_{N1}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N2}, y_{N2}), (x_M, y_M)\rvert \leq r_s)$
$A1(L)$	$(\lvert(x_{N2}, y_{N2}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N1}, y_{N1}), (x_M, y_M)\rvert > r_s), (x_M < x_N)$
$A1(R)$	$(\lvert(x_{N2}, y_{N2}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N3}, y_{N3}), (x_M, y_M)\rvert > r_s), (x_M \geq x_N)$
$A3(R)$	$(\lvert(x_{N2}, y_{N2}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N3}, y_{N3}), (x_M, y_M)\rvert \leq r_s)$
$A2(R)$	$(\lvert(x_{N3}, y_{N3}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N2}, y_{N2}), (x_M, y_M)\rvert > r_s), (y_M > y_N)$
$A2(R')$	$(\lvert(x_{N3}, y_{N3}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N4}, y_{N4}), (x_M, y_M)\rvert > r_s), (y_M \leq y_N)$
$A3(R')$	$(\lvert(x_{N3}, y_{N3}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N4}, y_{N4}), (x_M, y_M)\rvert \leq r_s)$
$A1(R')$	$(\lvert(x_{N4}, y_{N4}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N3}, y_{N3}), (x_M, y_M)\rvert > r_s), (x_M > x_N)$
$A1(L')$	$(\lvert(x_{N4}, y_{N4}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N1}, y_{N1}), (x_M, y_M)\rvert > r_s), (x_M \leq x_N)$
$A3(L')$	$(\lvert(x_{N1}, y_{N1}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N4}, y_{N4}), (x_M, y_M)\rvert \leq r_s)$
$A2(L')$	$(\lvert(x_{N1}, y_{N1}), (x_M, y_M)\rvert \leq r_s), (\lvert(x_{N4}, y_{N4}), (x_M, y_M)\rvert > r_s), (y_M < y_N)$

Figure 7(a) shows field $A3$ along with fields $A1$ and $A2$ which together form the quadrant. Specifically

$$\text{Area}(A1) = \text{Area}(A2) = \left(\frac{\pi r_s^2}{12} - \frac{(\pi r_s^2/6) - (r_s^2\sqrt{3}/4)}{2}\right),$$
$$\text{Area}(A3) = \left(\frac{\pi r_s^2}{4} - 2\text{Area}(A1)\right). \tag{25}$$

Figure 7(b) shows these three fields separately for each one of the quadrants where the quadrant name is indicated in parentheses and appended to the field name.

Let us also consider the random placement of more than one neighboring node over the quadrant. In this case, either one of the conditions below satisfies full coverage of the quadrant.

Condition 1. At least one of the neighboring nodes should be located in the field $A3$.

Condition 2. There should be at least one neighboring node in each one of the fields $A1$and $A2$.

These conditions apply to all quadrants since each one of them has its own $A1, A2$, and $A3$ fields.

We next give an important lemma which describes the coverage relationship between different quadrants of the same sensing region. The key observation here is that the existence of active neighboring nodes in a quadrant can compensate for the uncovered parts of its adjacent quadrants. Two quadrants are referred to as adjacent if they have a common boundary of length r_s and they constitute a contiguous $(S/2)$.

Example 2. The quadrant L is adjacent to L' and R but not R'.

Lemma 2. *Let U_1 and U_2 be the set of active nodes that are located over the adjacent quadrants Q_1 and Q_2, respectively, where $Q_1, Q_2 \in \{L, L', R, R'\}$, and $Q_1 \neq Q_2$. An element of U_1 placed in $A1(Q_1)$ field has an equal coverage effect on Q_2 as compared to an element of U_2 placed in $A1(Q_2)$. This property is also symmetric and is applicable to $A2$ fields of adjacent quadrants.*

Proof of Lemma 2. The proof is available in the Appendix. □

From Lemma 2, Condition 2 for the full coverage of each quadrant is rephrased as follows. Let the neighboring subregions of S_{sub} be denoted by $S_{\text{sub},A1}$ and $S_{\text{sub},A2}$ according to the type of adjacent field that S_{sub} has with its neighboring quadrant where $S_{\text{sub}}, S_{\text{sub},A1}, S_{\text{sub},A2} \in \{L, L', R, R'\}$ and $S_{\text{sub}} \neq S_{\text{sub},A1} \neq S_{\text{sub},A2}$.

Example 3. If S_{sub} is equal to L then $S_{\text{sub},A1} = R$ and $S_{\text{sub},A2} = L'$ since the quadrants L and R have adjacent $A1$ fields while quadrants L and L' have neighboring $A2$ fields.

Then, for the full coverage of S_{sub}, Condition 2 requires the existence of at least one neighboring node in each one of the following fields.

(i) $A2(S_{\text{sub}}), A1(S_{\text{sub}})$ or $A1(S_{\text{sub},A1})$.

(ii) $A1(S_{\text{sub}}), A2(S_{\text{sub}})$ or $A2(S_{\text{sub},A2})$.

The remaining question that needs to be answered is how to determine the field and the quadrant that a neighboring is located.

Let us consider a node N with coordinates (x_N, y_N) which computes the covered portion of its sensing region. Let node M be one of neighboring nodes of node N with coordinates (x_M, y_M) where $\lvert(x_N, y_N), (x_M, y_M)\rvert \leq r_s$. Table 2 gives a set of simple conditions that can be checked by node N to identify the particular quadrant and field at which node M is located. In these conditions, if node M is located at the boundary of two quadrants, it is assumed to belong to the right quadrant in clockwise order. The conditions are expressed in terms of distance of node M to the points $N1, N2, N3$, and $N4$ that are illustrated in Figure 7(b).

Once the exact quadrant and field information for each neighboring node is determined, the sensor node verifies whether each one of its quadrants is covered completely by checking Condition 1 and Condition 2.

We next address the effect of environmental factors on coverage computation. The changes in weather and ground status result in differences between the *effective* sensing ranges of nodes and their *nominal* sensing ranges. To incorporate effective sensing ranges into coverage computation, CORA considers both *single-node detectable* and *multi-node detectable* environmental factors such as humidity level and rain, respectively. Let β denote the set of all such environmental factors taken into account by the sensor nodes. For each sensing unit, a *correction constant* C_i is associated with ith environmental factor for $i = 1 \cdots |\beta|$ where $-1 \leq C_i \leq 1$. The magnitude and sign of each correction constant is determined by experimental evaluation with sensor hardware and different settings for the environmental factor it is associated with. The negative and positive sign for C_i represents the degradation and improvement in the sensing ability of the sensing unit due to the ith environmental factor, respectively, [35].

The effective sensing range of a sensing unit r'_s is obtained by adjusting its nominal sensing range r_s as

$$r'_s = \left(1 + \sum_{i=1}^{|\beta|} C_i\right) \times r_s. \tag{26}$$

Once the effective sensing range is determined, it is broadcast to the neighboring nodes at the beginning of each round before the coverage scheduling takes place. In order to check the full coverage of its quadrants, a sensor node compares its distance to each one of its neighboring nodes with its effective sensing range. Then, among those neighboring nodes within the sensing region, the sensor node considers only the ones which have larger effective sensing ranges than itself while checking Condition 1 and Condition 2. For the quadrants that are not fully covered, the coverage of the segments are determined based on the intersection of effective sensing ranges of the node itself and its neighboring node.

5.3.1. Noise in Internode Distance Measurements. The clusterhead uses the distances between pairs of cluster members to create a local coordinate system. The coordinate assigned to each cluster member is not exact and has an error margin [36]. This section considers how such errors in node location estimates can be incorporated into coverage computation. In the absence of exact node coordinate information, we express coverage and event miss-ratio in terms of *expected* values rather than absolute values.

Let us represent the maximum deviation on the x and y coordinates of sensor nodes as Δx and Δy, respectively. A cluster member with coordinate estimates (x, y) may be located at any point within the ellipsis bounded by a box of width $(2 \times \Delta x)$ and of height $(2 \times \Delta y)$. In what follows, we denote the set of all candidate points for the *exact* location of the node in this ellipsis by $\Re(x, y)$.

Let us now consider the coverage of any given cluster member K_i's sensing area by its neighboring nodes where $1 \leq i \leq n_K$. We refer to node K_i's location as (x, y) and the neighboring nodes that can contribute to the coverage of its sensing region as the set Z. In the case of location errors, we define two sets of angles for the coverage of each segment within K_i's sensing region. These sets of angles, namely, Ω'_k and Ω^*_k, represent the worst and best possible cases for the coverage of the kth segment by the neighboring sensor nodes in Z, respectively, where $1 \leq k \leq n_G$.

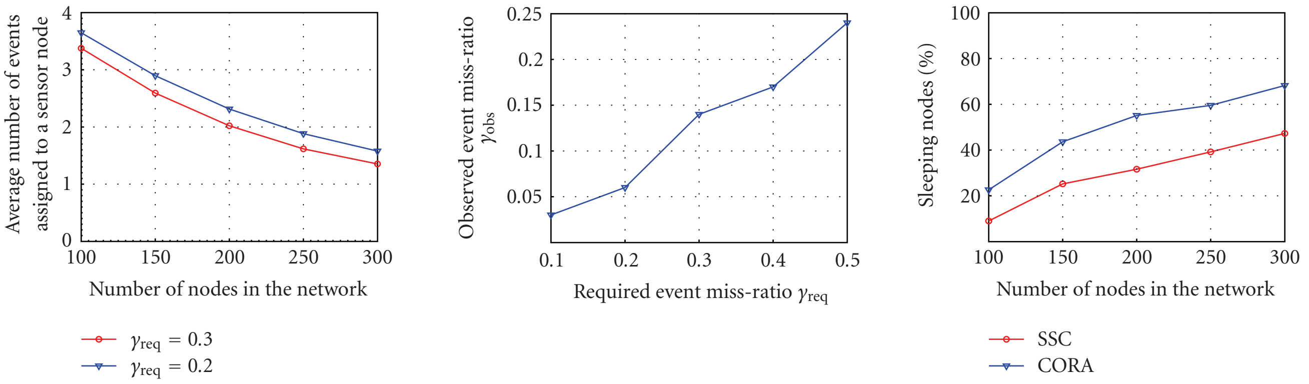
Let K_j represent one of the neighboring nodes in Z with coordinates (x_j, y_j) that covers a part of kth segment within node K_i's sensing region where $1 \leq j \leq n_K$ and $i \neq j$. Now consider the possible cases for the actual locations of nodes K_i and K_j that minimizes and maximizes the covered part of the node K_i's sensing region by node K_j, respectively. For the worst case placement in terms of coverage, let us assume that node K_j's coordinates are (x'_j, y'_j) and node K_i is located at (x', y') where $(x'_j, y'_j) \in \Re(x_j, y_j)$ and $(x', y') \in \Re(x, y)$. Similarly, also assume that when node K_j is positioned at (x^*_j, y^*_j) and K_i has coordinates (x^*, y^*), the best case for the coverage of the kth segment is achieved where $(x^*_j, y^*_j) \in \Re(x_j, y_j)$ and $(x^*, y^*) \in \Re(x, y)$. For all $(x_a, y_a) \in \Re(x_j, y_j)$ and for all $(x_b, y_b) \in \Re(x, y)$, the following condition holds

$$\begin{aligned} &\sqrt{\left(x'_j - x'\right)^2 + \left(y'_j - y'\right)^2} \\ &\qquad = \max\left(\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}\right), \\ &\sqrt{\left(x^*_j - x^*\right)^2 + \left(y^*_j - y^*\right)^2} \\ &\qquad = \min\left(\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}\right). \end{aligned} \tag{27}$$

Let the pair of angles (ρ', ρ^*) represent the worst and best case contribution of node K_j for the coverage of the kth segment, respectively. Ω'_k and Ω^*_k are updated with the addition of ρ' and ρ^*, respectively. The same coverage computation is applied for each neighboring node in set Z in addition to node K_j. Once the coverage contribution of all neighboring nodes is determined, we define $E(C_k)$ as the *expected* covered portion of kth segment where

$$E(C_k) = \begin{cases} \dfrac{\int_{\Sigma(\Omega'_k)}^{\Sigma(\Omega^*_k)} (\theta/2\pi)d\theta}{\Sigma\left(\Omega^*_k\right) - \Sigma\left(\Omega'_k\right)}, & \Sigma\left(\Omega^*_k\right) \leq 2\pi \\[2ex] \dfrac{\int_{\Sigma(\Omega'_k)}^{2\pi} (\theta/2\pi)d\theta + \int_{2\pi}^{\Sigma(\Omega^*_k)} 1d\theta}{\Sigma\left(\Omega^*_k\right) - \Sigma\left(\Omega'_k\right)}, & \\[2ex] & \Sigma\left(\Omega^*_k\right) > 2\pi, \end{cases} \tag{28}$$

and $\Sigma(\Omega)$ represents the sum of the magnitudes of angles in a set Ω. From the coverage of each segment k for $1 \leq k \leq n_G$, the uncovered portion of node K_i's sensing region



(a) CORA takes advantage of the increase in network population and the required event miss-ratio thresholds to reduce the number of events to be sensed by each node

(b) The experimental validation of the event miss-ratio assurances

(c) CORA uses less number of active nodes to meet a given coverage requirement than the sponsored sector based coverage (*SSC*) scheduling technique proposed in [5]

Figure 8: Performance evaluation of coverage scheduling.

is determined as

$$\sum_{k=1}^{n_G}\left((1-E(C_k))\frac{\pi r_g^2\left(k^2-(k-1)^2\right)}{\pi r_s^2}\right). \tag{29}$$

6. Performance Evaluation

CORA is evaluated in terms of event miss-ratios, total energy expenditure, average node battery consumption, and the storage/update overhead for the application of source coding. In the simulations, we consider sensor networks with multiple clusters in which the clusters are formed based on residual energy of sensor nodes. Each network is simulated for a single round with 100 data gatherings. For uniform event distribution, every sensor node within a cluster is assumed to detect an event per data gathering. For nonuniform event distribution, we have randomly selected hotspot centers. At each data gathering, the event occurrences are modeled around these hotspot centers following the Gaussian distribution.

In the simulation graphs, the results for uniform and nonuniform event distributions are illustrated with solid and dashed lines, respectively. Each point shown in the graphs is obtained by averaging the simulation outcomes from a large number sensor networks. The sensor networks are deployed randomly over a target area of side length 100 m. The number of sensor nodes in each network is selected from the interval [100–300] to control node/coverage redundancy. As network population increases, the sensing ranges of nodes overlap more, thereby increasing both the number of nodes redundant for coverage and the data correlation among neighboring sensor nodes.

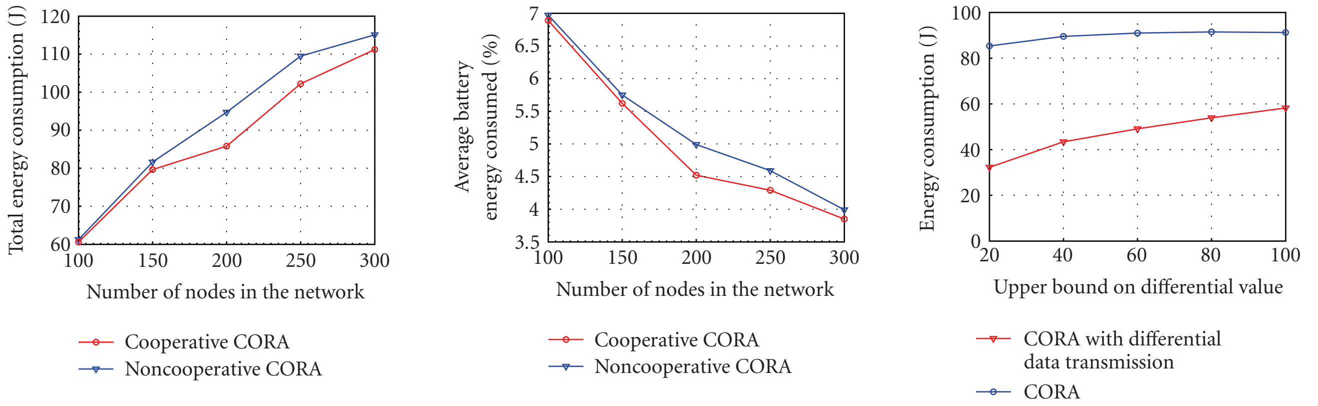
The sensing range and transmission range of the nodes are set to 10 m and 30 m, respectively. The initial residual energy level of each node is assigned as 10 Joules. To calculate the energy expenditure, we have used the radio power consumption model described in Section 3 with settings $a = 0.01$ and $b = 0.03$. The entropy of the source at a single cluster member is modeled as $n_B = 16$ bits. The ratio of the control message size to the data message size is set to $\varepsilon = 0.2$. Unless specified otherwise, the required event miss-ratio threshold is equal to 0.2.

Figure 8(a) shows the results of the experiments in which we've considered sensor nodes equipped with four sensing units for event types $M_j, 1 \leq j \leq 4$. CORA exploits the increase in node redundancy and required event miss-ratio thresholds to minimize the average number of events scheduled to each sensor node. As simulation results indicate, CORA enables a node to keep only one of its sensing units active at any time with sufficient node redundancy. It is possible that a node keeps its sensing unit active for event type M_i while its neighboring node might activate the sensing unit for a different event type $M_k, 1 \leq i, k \leq 4, i \neq k$. Meanwhile, CORA's distributed coverage scheduling ensures that event miss-ratio assurances are met for every event types M_i and M_k.

Figure 8(b) illustrates the empirical verification of the assurances provided by CORA under different event miss-ratio requirements. For a particular γ_{req} value, a large number of simulations are run while keeping the network population constant. The observed event miss-ratios γ_{obs} (shown in y axis of Figure 8(b)) always remain below the required event miss-ratios γ_{req} (illustrated in x axis of Figure 8(b)).

CORA assigns rates to the active nodes only and therefore is highly scalable. Figure 8(c) shows the results of the simulations that are run with the same required event miss-ratio value but with varying number of nodes. Even though the network population increases, CORA limits the number of nodes that are scheduled active and get assigned rates. Having only a smaller set of nodes active prolong the time event miss-ratio assurances can be met as those sensor nodes which are put in off-duty mode can save their energy to go active in the subsequent scheduling phases.

The simulations in Figure 8(c) also evaluate the accuracy of Algorithm Coverage Computation. For comparison, we have used the sponsored sector-based coverage computation technique proposed in [5] which we refer to as *SSC* in



(a) The contribution of the collaboration between the proposed coverage scheduling and rate assignment techniques to the total energy consumption of the sensor nodes

(b) The improvement in average battery consumption of sensor nodes due to the collaboration between the proposed coverage scheduling and rate assignment techniques

(c) The energy savings due to differential data transmission between correlator nodes and their clusterhead

Figure 9: Performance evaluation of the collaboration between the proposed coverage scheduling and rate assignment techniques and the energy gains obtained from the differential data transmission technique.

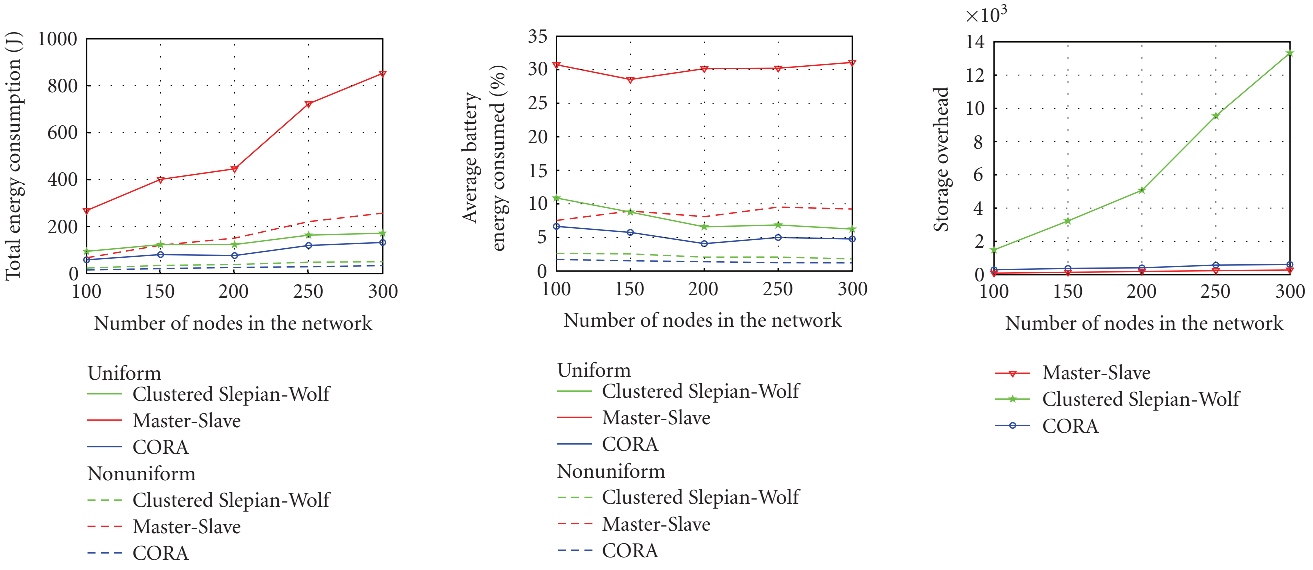
the results. *SSC* considers those neighboring nodes that are located within the sensing region of a node while checking whether the node's sensing region is covered by its neighboring nodes. If the node's sensing region overlaps sufficiently by its neighboring nodes, then it becomes eligible to go off-duty. In particular, for a neighboring node with distance closer than the sensing range r_s, *SSC* represents the region covered by the neighboring sensor node with a central angle θ which is referred to as the *sponsored sector* where $2\pi/3 \leq \theta < \pi$. If the union of sponsored sectors by all neighboring nodes is equal to 2π, then the sensing region of the node is covered completely by its neighboring nodes and the node becomes eligible for going to sleep mode.

Figure 8(c) shows that CORA approximates the coverage of the monitored area with better accuracy than *SSC* and hence activates less number of nodes to meet the same coverage requirement. This is attributed to the fact that, CORA can consider *all* neighboring nodes of a sensor node during coverage computation, including those with a distance to the sensor node larger than the sensing range r_s. While achieving more accurate coverage computation, CORA introduces little overhead with the help of its quadrant-based and segment-based coverage computation techniques described in Section 5.3.

Figure 9(a) and Figure 9(b) show that CORA is a good cross-layer combination. In these simulations, Cooperative CORA refers to the proposed collaboration where rate assignment helps coverage scheduling by identifying which critical nodes should be activated for minimizing data redundancy in transmission. On the other hand, Noncooperative CORA describes the case when coverage scheduling is applied before the rate assignment, however without any interaction between the two. In noncooperative case, as coverage scheduling does not have feedback on those critical nodes for reducing data redundancy, it is possible that such critical nodes are put in sleep mode just by checking their residual energy and coverage of their sensing area by their neighbor sensor nodes. As simulation results indicate, the increase in node density improves the amount of side data available to a sensor node and decreases the average battery consumption. However, by identifying the critical nodes and only letting them participate in source coding, Cooperative CORA yields better energy consumption and battery usage than Noncooperative CORA.

To evaluate the performance of differential data transmission, we consider the energy spent for gathering data from the correlator nodes to their clusterhead. We model the sensor data sent by correlator nodes as numbers, and define arithmetic subtraction $(-)$ as the difference operator and arithmetic addition $(+)$ as its reverse. The reference data used between the clusterhead and a particular correlator node is calculated based on the arithmetic average of the data values transmitted by that correlator node. The data value to be transmitted from a correlator node is selected as 100 plus a random differential data value for each data gathering. The upper bound used in each experiment for the differential data values are shown in the x axis of Figure 9(c). When upper bound value is 40 as an example, the sensor data sent by correlator nodes are selected randomly from the range $[100, 140]$.

Let us consider a scenario where the reference data value used by a particular correlator node is 110, and the correlator node has sensor data value of 130. The correlator node would apply the difference operator (subtraction) with respect to the reference data and would send 20 as the differential data value. Clusterhead would then recover the original data 130 using the same reference data value by applying the inverse operator (addition) to the reference data value 110 and the differential data 20 received from the correlator node. CORA in Figure 9(c) refers to the energy consumption when original data values, such as 130 in the example above, are sent by the correlator nodes without any differential data transmission. As simulation results show, differential data transmission takes advantage of the temporal correlation of



(a) The total energy consumed by sensor nodes when coding rates are assigned by CORA, Master-Slave and Clustered Slepian-Wolf rate assignment schemes

(b) The gains in average battery consumption of the nodes achieved by CORA as compared to the existing rate assignment techniques

(c) While achieving better energy savings than Clustered Slepian-Wolf rate assignment scheme, CORA introduces very low overhead that is close to the overhead of Master-Slave rate assignment technique

FIGURE 10: Performance evaluation of CORA with respect to existing rate assignment schemes.

data sent by correlator nodes to the clusterhead for reducing energy consumption.

The performance of CORA is also evaluated with respect to the existing rate assignment schemes. For comparison, we have chosen Master-Slave and Clustered Slepian-Wolf rate assignment techniques [24] due to their following special feature. These rate assignment schemes represent the two extreme cases for the tradeoff between the efficiency of rate allocation and overhead of source coding. Following the notation in Table 1, let us consider a cluster of n_K active sensor nodes with K_i referring to ith active sensor node.

(i) *Master-Slave Rate Assignment.* This rate allocation represents the low efficiency, low-overhead end of all possible rate assignments to sensor nodes. The cluster members code their data according to their correlation with the data of the clusterhead. For clarity, let us label active node K_1 as the clusterhead. Then, master-slave rate assignment allocates rate $R_{K_i} = H(K_i \mid K_1)$ to active node K_i for $2 \leq i \leq n_K$. This rate allocation has low storage overhead as only the correlation between the data of each cluster member and the clusterhead needs to be maintained. However, while being scalable, Master-Slave scheme sacrifices the potential gains achievable from the correlation of a cluster member K_i with other cluster members K_j for $2 \leq i, j \leq n_K, 1 \neq i \neq j$.

(ii) *Clustered Slepian-Wolf Rate Assignment.* This rate assignment has the characteristic of efficient rate allocation at the cost of high storage overhead of correlation information. The cluster members code their data based on their correlation with the cluster members that have higher residual energy. Let us consider active nodes as labeled in descending order of their residual energy where active node K_i has more remaining energy than K_j for $i < j$ with clusterhead K_1 having the highest residual energy. Slepian-Wolf Rate Assignment then assigns rate $R_{K_i} = H(K_i \mid K_1, \ldots, K_{(i-1)})$ to active node K_i. To achieve this rate allocation, data correlation need to be maintained between node K_i and node(s) $K_1, K_2, \ldots, K_{(i-1)}$. Consequently, clustered Slepian-Wolf brings significant storage/update overhead in order to increase the side data available to sensor nodes.

As Figures 10(a)–10(c) show, CORA possesses the desirable features of both Clustered Slepian-Wolf (efficiency) and Master-Slave (low-overhead) rate assignment schemes. In particular, CORA would first identify critical nodes within the cluster for source coding gain and coverage scheduling then activates only those critical nodes for ensuring that the required event miss-ratios are met. For clarity of discussion, let us assume there are n_M such critical nodes which are scheduled active and label them as K_y where $1 \leq y \leq n_M \leq n_K$. CORA puts noncritical nodes $K_{(n_M+1),\ldots,K_{n_K}}$ in sleep mode, thereby assigning rates to and maintaining correlation between node K_y and other critical nodes only.

Figure 10(a) shows the total energy consumption of sensor nodes when CORA Protocol, Master-Slave, and Clustered Slepian-Wolf schemes are applied on the same network instances. As node redundancy increases, CORA's ability to identify noncritical sensor nodes and to put them off duty yields better energy consumption than Clustered

Slepian-Wolf. As expected from the increase in side data, Figure 10(b) shows that node redundancy improves the average battery consumption for both CORA and Clustered Slepian-Wolf. For Master-Slave, the battery consumption of nodes remains almost the same since data is coded at each cluster member without considering its correlation with other cluster members. In terms of storage/update overhead, CORA gets close to Master-Slave with the increase in node density as illustrated in Figure 10(c).

7. Conclusions

In this paper, we have addressed the problem of meeting event miss-ratio requirements with minimum energy expenditure. We identify a graceful cooperation between coverage scheduling and rate allocation techniques, which are addressed separately in the literature. For both uniform/nonuniform (hotspot) event distributions, we show the existence of a radius within the cluster for ideal application of distributed source coding. With the help of this radius, rate allocation guides coverage scheduling to activate only those cluster members that are critical for minimizing energy consumption. The coverage scheduling technique of the proposed protocol CORA assists rate assignment by enabling overlapping sensing regions and correlation among neighboring nodes while minimizing the overhead of distributed source coding. In comparison with existing rate allocation algorithms, CORA significantly improves energy expenditure by 85%, node battery energy consumption up to 25% and source coding overhead by 90%.

This paper has addressed the impact of environmental factors on the size of circular-shaped sensing regions. As for future work, identifying the irregular-shaped sensing regions due to environmental factors and considering their effect on event miss-ratio assurances remain interesting and challenging problems.

Appendix

Proof of Lemma 1. Considering $\gamma_{\text{obs}} = [1-(E(D)/E(Y))]$, we need to prove that $(1-\gamma_{\text{req}})E(Y) < E(D)$, specifically

$$\left(1-\gamma_{\text{req}}\right)\sum_{k=0}^{n}[k \times \text{Prob}(Y=k)] < \sum_{\ell=0}^{n}[\ell \times \text{Prob}(D=\ell)]. \tag{A.1}$$

Without loss of generality, let us consider the terms $(m \times \text{Prob}(Y = m))$ and $(m \times \text{Prob}(D = m))$ in expansion of $E(Y)$ and $E(D)$, respectively, where $0 \le m \le n$. To identify the value of $\text{Prob}(Y = m)$ and $\text{Prob}(D = m)$, we consider every possible division of n network locations into two subsets, namely, A and B, such that $|A| = m$ and $|B| = (n-m)$. If $x_i \in A$, then we assume that an event occurs at x_i and an event gets detected at x_i. Otherwise, it is assumed that there is not any event occurrence or event detection at x_i. For a *particular* combination among all such possible divisions, the probability of having m event *occurrences* over the monitored area is calculated as

$$\left[\prod_{i=1,x_i\in A}^{n}\text{Prob}(o_i)\right]\times\left[\prod_{k=1,x_k\in B}^{n}(1-\text{Prob}(o_k))\right]. \tag{A.2}$$

On the other hand, the probability of having m event *detections* is expressed as

$$\begin{aligned}&\left[\prod_{i=1,x_i\in A}^{n}\text{Prob}(o_i)\ \text{Prob}(d_i)\right]\\&\quad\times\left[\prod_{k=1,x_k\in B}^{n}(1-\text{Prob}(o_k)\text{Prob}(d_k))\right].\end{aligned} \tag{A.3}$$

By replacing the probability of event detections in accordance with the variable coverage provided by CORA, (A.3) becomes

$$\begin{aligned}&\left[\prod_{i=1,x_i\in A}^{n}\text{Prob}(o_i)\left[1-\gamma_{\text{req}}\left(\frac{\text{Prob}(o_i)-\text{Prob}(o_{\min})}{\text{Prob}(o_{\max})-\text{Prob}(o_{\min})}\right)\right]\right]\\&\times\left[\prod_{k=1,x_k\in B}^{n}\left(1-\text{Prob}(o_k)\right.\right.\\&\qquad\left.\left.\times\left[1-\gamma_{\text{req}}\left(\frac{\text{Prob}(o_k)-\text{Prob}(o_{\min})}{\text{Prob}(o_{\max})-\text{Prob}(o_{\min})}\right)\right]\right)\right]\end{aligned} \tag{A.4}$$

Note that our objective is to show that $(1-\gamma_{\text{req}})E(Y) < E(D)$, thus we multiply $(m \times \text{Prob}(Y = m))$ and every other term in $E(Y)$ with the constant $(1-\gamma_{\text{req}})$ and get

$$\left(1-\gamma_{\text{req}}\right)\times\left[\prod_{i=1,x_i\in A}^{n}\text{Prob}(o_i)\right]\times\left[\prod_{k=1,x_k\in B}^{n}(1-\text{Prob}(o_k))\right]. \tag{A.5}$$

Let us now compare the coefficients of the first term $\prod_{i=1,x_i\in A}^{n}\text{Prob}(o_i)$ for $E(Y)$ from (A.5), specifically

$$\left(1-\gamma_{\text{req}}\right) \tag{A.6}$$

and for $E(D)$ from (A.4), that is equal to

$$\prod_{i=1,x_i\in A}\left[1-\gamma_{\text{req}}\left(\frac{\text{Prob}(o_i)-\text{Prob}(o_{\min})}{\text{Prob}(o_{\max})-\text{Prob}(o_{\min})}\right)\right]. \tag{A.7}$$

Even in the case of single event occurrence and single event detection (i.e., $|A| = 1$), the coefficient in (A.6) is less than the coefficient in (A.7) since the largest value the expression $((\text{Prob}(o_i) - \text{Prob}(o_{\min}))/(\text{Prob}(o_{\max}) - \text{Prob}(o_{\min})))$ can assume is 1. Further, it is easy to see that the second term

$$\prod_{k=1,x_k\in B}^{n}(1-\text{Prob}(o_k)) \tag{A.8}$$

in (A.5) also has a smaller value than the second term

$$\prod_{k=1, x_k \in B}^{n} (1 - \text{Prob}(o_k)\ \text{Prob}(d_k)) \tag{A.9}$$

in (A.4) since $0 \leq \text{Prob}(d_k) \leq 1$. Considering that the selection of m and the combination for m event occurrences/detections were arbitrary, one can conclude that $(1 - \gamma_{\text{req}})E(Y) < E(D)$. □

Proof of Lemma 2. Without loss of generality, let us focus on quadrants L and R in Figure 7(b). We refer to the set of active nodes covering the upper left (L) and upper right (R) quadrant as U_L and U_R, respectively. For the *worst-case* analysis let us make the following assumptions.

(i) U_L does not satisfy Condition 2 for the quadrant L with the active nodes placed in $A2(L)$ only.

(ii) All active nodes from the set U_L are placed at point $N1$ within $A2(L)$.

(iii) All active nodes from the set U_R are placed at point $P(R)$ within $A1(R)$.

In order to prove that active nodes at $A1(R)$ can compensate for the uncovered portion of the subregion L, that is $A1(L)$, it is necessary to show that

(i) $|P(R), O| \leq r_s$,

(ii) $|P(L), P(\text{R})| \leq r_s$.

The condition (i) is trivial based on the disk equation itself. The condition (ii) can be proved by placing an imaginary active node at each one of the points $N1$ and $N3$. The points $P(L), P(R)$ can then be defined as the intersection of the node's sensing region centered at O with the imaginary active nodes centered at $N1, N3$, respectively.

Let $P(L)'$ be the projection point of $P(L)$ online segment $[N1, O]$ which is indicated as a red dot in Figure 7(b). The equality $|P(L)', O| = (r_s/2)$ holds due to the fact that the sensing regions are congruent. Similarly, the equation $|O, P(R)'| = (r_s/2)$ also holds from the symmetry where $P(R)'$ is the projection point of $P(R)$ on line segment $[O, N3]$ shown as a red point in Figure 7(b).

It can be concluded from these equalities that $|P(L)', P(R)'| = |P(L)', O| + |O, P(R)'| = |P(L), P(R)| = r_s$. This proves that $A1(L)$ can be covered by active nodes placed at $A1(R)$ even in worst-case placement of active nodes. Similarly, the members of U_L placed at $A1(L)$ can cover the field $A1(R)$ completely. □

References

[1] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1548–1557, April 2001.

[2] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

[3] H. O. Sanli and H. Cam, "Event-driven coverage and rate allocation for providing miss-ratio assurances in wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1134–1138, November 2007.

[4] H. O. Sanli and H. Cam, "Adaptive task scheduling for providing event miss-ratio statistical assurances in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, pp. 445–450, April 2006.

[5] D. Tian and N. D. Georganas, "A node scheduling scheme for energy conservation in large wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271–290, 2003.

[6] W. Tian and J. Liu, "A novel approach for the maximum coverage sets of WSN based on immune clone selection algorithm," in *Proceedings of the 2nd International Symposium on Electronic Commerce and Security (ISECS '09)*, vol. 1, pp. 275–279, Nanchang, China, May 2009.

[7] U. R. Chen, B. S. Chiou, J. M. Chen, and W. Lin, "An adjustable target coverage method in directional sensor networks," in *Proceedings of the 3rd IEEE Asia-Pacific Services Computing Conference (APSCC '08)*, pp. 174–180, Pisa, Italy, December 2008.

[8] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 28–39, November 2003.

[9] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 51–62, November 2003.

[10] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pp. 189–200, June 2003.

[11] J. Carle and D. Simplot-Ryl, "Energy-efficient area monitoring for sensor networks," *IEEE Computer*, vol. 37, no. 2, pp. 40–46.

[12] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 413–420, 2006.

[13] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of International Conference on Communications (ICC '01)*, pp. 472–476, June 2001.

[14] H. O. Sanli, R. Poornachandran, and H. Çam, "Collaborative two-level task scheduling for wireless sensor nodes with multiple sensing units," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 350–361, September 2005.

[15] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar, "Redundancy and coverage detection in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 94–128, 2006.

[16] C. Liu, K. Wu, Y. Xiao, and B. Sun, "Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 6, pp. 562–575, 2006.

[17] C. F. Huang, Y. C. Tseng, and H. L. Wu, "Distributed protocols for ensuring both coverage and connectivity of a wireless

sensor network," *ACM Transactions on Sensor Networks*, vol. 3, no. 1, artile 5, 2007.

[18] M. X. Cheng, L. Ruan, and W. Wu, "Coverage breach problems in bandwidth-constrained sensor networks," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, article 12, 2007.

[19] W. Wang, V. Srinivasan, B. Wang, and K. C. Chua, "Coverage for target localization in wireless sensor networks," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 118–125, April 2006.

[20] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[21] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUSS): design and construction," in *Proceedings of the Data Compression Conference (DCC '99)*, pp. 158–167, March 1999.

[22] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2571–2582, March 2004.

[23] S. J. Baek, G. de Veciana, and X. Su, "Minimizing energy consumption in large-scale sensor networks through distributed data compression and hierarchical aggregation," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1130–1140, 2004.

[24] D. Marco and D. L. Neuhoff, "Reliability vs. efficiency in distributed source coding for field-gathering sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 161–168, April 2004.

[25] Y. T. Hou, Y. Shi, and H. D. Sherali, "Rate allocation and network lifetime problems for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 321–334, 2008.

[26] R. Poornachandran, H. Ahmad, and H. Çam, "Energy-efficient task scheduling for wireless sensor nodes with multiple sensing units," in *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC '05)*, pp. 409–414, April 2005.

[27] A. Sinha and A. Chandrakasan, "Operating system and algorithmic techniques for energy scalable wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Data Management (MDM ' 05)*, vol. 1987, pp. 199–209, January 2001.

[28] G. A. McIntyre and K. J. Hintz, "Sensor measurement scheduling: an enhanced dynamic, preemptive algorithm," *Optical Engineering*, vol. 37, no. 2, pp. 517–523, 1998.

[29] S. Goel and T. Imielinski, "Prediction-based monitoring in sensor networks: taking lessons from MPEG," *Computer Communication Review*, vol. 31, no. 5, pp. 82–98, 2001.

[30] Mica Motes, http://www.xbow.com/.

[31] "MTS400CA and MTS420CA (with GPS module) Environmental Sensor Boards," http://www.xbow.com/.

[32] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33 '00)*, p. 8020, Maui, Hawaii, USA, January 2000.

[33] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 28–35, April 2004.

[34] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, John Wiley & Sons, New York, NY, USA, 1991.

[35] J. W. Casalegno, "All-weather ground sensor system with possible law enforcement applications," in *Enabling Technologies for Law Enforcement and Security*, vol. 4232 of *Proceedings of SPIE*, pp. 359–366, February 2001.

[36] H. A. B. F. Oliveira, E. F. Nakamura, A. A. F. Loureiro, and A. Boukerche, "Error analysis of localization systems for sensor networks," in *Proceedings of the 13th ACM International Workshop on Geographic Information Systems (GIS '05)*, pp. 71–78, November 2005.