
Xinxin Liu, Han Zhao
Dept. of Computer & Information Science & Engineering
University of Florida
Gainesville, FL 32611
Email:{xinxin,han}@cise.ufl.edu

Xiaolin Li
Dept. of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611
Email:andyli@ece.ufl.edu

Abstract—This paper addresses energy-efficient data gathering issues in wireless sensor networks (WSNs). Leveraging data correlation in densely-deployed sensor networks, we propose an Energy-aware Probability-based Clustering algorithm (EPC), featuring high scalability and flexibility particularly suitable for large-scale WSNs. Unlike most existing data gathering schemes that construct static routing structures or only consider spatial correlation among sensed data, EPC establishes energy-efficient routes on the fly during the data gathering process, and dynamically organizes sensor nodes into clusters based on a probability factor determined by both spatial and temporal data correlations. Redundant data transmissions are suppressed within a cluster and energy consumption is balanced to prolong network lifetime. To verify the effectiveness of EPC, extensive simulations are conducted on a network of 625 randomly deployed sensor nodes. Results show that EPC balances the energy consumption of the whole network and reduces up to 71% of the transmission costs with near negligible error rates for representative aggregation functions.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have been widely used in surveillance applications such as habitat [14] and weather condition monitoring [8]. These environmental monitoring systems are envisioned to consist of hundreds to thousands of low-cost sensor nodes, which have small on-board memories, limited computing capabilities, and restricted power supply. Intra-network communications are accomplished through short range radio transmission. To accommodate sensor failures and ensure coverage, sensor nodes are usually deployed densely in sensing fields. As sensors monitoring common environmental phenomena, such as temperature, in close regions may yield similar readings, known as spatially correlated data, dense deployment will result in collecting a significant amount of redundant data.

To reduce communication overhead in data gathering and thereby prolong sensor nodes lifetime, redundant data should be eliminated by leveraging data correlation in the network. In-network aggregation is a common way to do so. Most existing algorithms, however, rely on predefined network structure or geographic location information to reduce the amount of data transferred, this makes them unable to adapt to dynamically changing field situations. For example, if the data correlation changes among sensor nodes in the same predefined cluster, aggregation operations will be less efficient. Even worse, there are very few in-network aggregation algorithms designed for mobile data collectors. As the miniature technology makes it possible to use mobile objects, such as animals or vehicles, as data collectors, these mobile sinks may appear within the monitoring area in ad hoc fashion. Once the location of a data collector changes, the whole network need to be reconfigured, which is very energy expensive and even impossible for some aggregation algorithms.

In this paper, we propose an Energy-aware Probability-based Clustering algorithm (EPC) for correlated data gathering in large-scale WSNs with mobile sinks. EPC employs probability-based clustering method to dynamically adjust network architecture in accordance to environment situations. By choosing energy-efficient routes for data forwarding, EPC effectively balanced the energy consumption in a WSN, therefore achieves longer network lifetime. Our contributions are threefold. (1) We propose the decentralized EPC algorithm for correlated data gathering with a mobile data collector, which reduces energy consumption and prolongs network lifetime with high data accuracy. (2) We analytically demonstrate the effectiveness of EPC on transmission reduction and energy balancing. (3) We implement and verify the effectiveness and efficiency of EPC through extensive simulations on scalable environments.

The rest of the paper is organized as follows. Section II presents related works. We present the EPC algorithm in Section III. In Section IV we systematically analyze EPC and validate its effectiveness. Further we discuss potential extension of EPC to multiple-sink environment. Simulation results and discussion are presented in Section V. Finally we conclude the paper in Section VI.

II. RELATED WORKS

To reduce energy costs, in-network aggregation schemes have been extensively studied and used for data gathering in sensor networks. For aggregation with data compression, coding algorithms such as [3], [7], [17] are proposed to minimize data size without losing information of the sensing field. However, as finding optimal transmission structures is difficult and is proved to be an NP-complete problem if joint entropy-based coding model is used [4], uncompressed aggregation algorithms are also investigated.

Representative uncompressed aggregation algorithms such as LEACH [9], CCC [5], Directed Diffusion [10], CAG [19],
A wireless sensor network with data correlations. Concentric circles represent sensor nodes, and cloud shapes indicate that these nodes have correlated data.

III. EPC: ENERGY-AWARE PROBABILITY-BASED CLUSTERING ALGORITHM

Before we formally present our energy-aware probability-based clustering algorithm, the following assumptions are made: (1) the network is fully connected, meaning that there is no isolated partition in the network; (2) all the sensor nodes are potential data sources, on which environment information is gathered through switching alternatively between wake up mode and sleep mode; (3) sensor nodes are synchronized and the communication delay caused by mode switching is negligible. Some synchronization algorithms [6], [15] can be adopted to meet the assumption requirements; (4) data collection operation is performed by a mobile object (e.g. an animal) carrying a sink node, so its location may change at times.

A. Problem formulation

The sensor network topology considered here is represented as an undirected graph $G = (V, E)$, in which $V = \{v_1...v_n\}$ is a finite set of sensor nodes deployed over an information field, and $E$ stands for radio links among sensor nodes. A small example network is shown in Fig. 1. The transmission range of a sensor node $v_i$ is $r(e^t_i)$, where $e^t_i$ stands for transmission power for node $v_i$. The actual physical network topology, whether grid or other distributed structures, is of no significant relevance in the problem formulation. In the network there is one sink node $s$, whose location is unknown to other sensors, this corresponds to a mobile data collector carried by a moving object in the field. Data sensed by sensor nodes have certain degree of correlation, as indicated in Fig. 1 by cloud shapes. The objectives of EPC algorithm are: (1) to minimize the amount of redundant data gathered from network; (2) to maximize network lifetime through utilizing energy-efficient routes for data transmission. The following notations are used for ease of presentation.

- $v_i$: the $i$th node in network
- $e_i$: energy residual of node $i$
- $d_i$: sensed data of node $i$
- $n_{ij}$: the $j$th neighbor node of node $v_i$
- $\delta_{ij}$: data correlation factor for node $i$ and $j$ in relation $E$
- $\tau$: user defined data accuracy factor
- $P$: data reporting probability

Several different definitions have been used in literature to define the lifetime of a sensor network, such as from the time of deployment to the time that half of the nodes’ battery die out. As for most monitoring applications, network coverage is critical, here we adopt the network lifetime used in [2] and [18] to ensure that the whole area is covered all the time. The lifetime of a sensor network is defined as the time of the first node runs out of energy and dies. Therefore the aim of energy-aware routing is to balance the energy consumption of the network.

EPC algorithm consists of three phases controlled by sink node during each data gathering procedure. They are “data correlation detection phase”, “energy-aware routing phase” and “response phase”. In the following sections these three phases will be presented in the order of execution. A summary for the EPC algorithm is listed in Algorithm 1 for sink node and Algorithm 2 for sensor node respectively.

B. Data correlation detection

Each sensor node in the network maintains a data correlation table in order to detect data correlation. The data correlation table keeps records of data correlation factor for all the neighbors of the current node. Sample data correlation tables are shown in Fig. 2. At the very beginning no data correlation is recorded in the table. The entries in the data correlation table get updated with time goes, which will be elaborated later.

The data correlation detection phase of EPC starts as soon as a user issues a specific query or a data gathering request. A
short message, called start-up message, is broadcasted by the sink node. The start-up message has a form of \(< \psi, \text{Data} >\), in which \(\psi\) is a monotonically increasing sequence number that indicates the freshness of the message to prevent loops in flooding. The messages come from the sink have ‘NULL’ value in the ‘Data’ field. Due to the nature of wireless media, sensor nodes closest to sink node \(s\) will first hear the message. Assume node \(v_1\) is one of these nodes, upon receiving this new start-up message from \(s\) three things will be done by \(v_1\) before it rebroadcasts the message. First, it records the \(\psi\) for later reference. Second, it starts a timer \(T_{1, s}\) to form an energy-efficient route in the next phase of EPC. Finally, it updates its data correlation table accordingly.

Sensed data \(d_1\) and \(d_2\) are considered to be correlated when they meet the condition defined by Eq. (2).

\[
d_i = d_j \times (1 \pm \tau)
\]  

(2)

The data correlation factor gets updated each time a node overhears new data from its neighbors. \(\delta_{ij}\) is defined as a fraction number \(\delta_{ij}\), in which \(\epsilon\) is the number of times data from node \(v_i\) and \(v_j\) are correlated according to Eq. (2), and \(\eta\) is the total number of data exchanges happened between \(v_i\) and \(v_j\), which is not only confined to exchanges during data gathering process, but also includes exchanges happened in overhearing and packet forwarding. This correlation factor uses historical data information to predict future data correlations, which essentially reflects temporal correlation among sensor nodes.

Take node \(v_3\)’s data correlation table in Fig. 2 as an example, it records a data correlation factor \(\delta_{13} = \frac{9}{10}\) in node 3’s entry, meaning that out of 10 times of data exchange between \(v_3\) and \(v_4\), 9 of them are evaluated as correlated data.

C. Route construction

Many routing algorithms have been proposed with different criteria for evaluating whether a route is energy efficient or not. In EPC we aim at balancing the energy consumption in the network, thus an energy-efficient route should be the one with “max-min” energy residual. Suppose there are two routes \(R_1\) and \(R_2\) from a node to the sink, where \(R_1 = v_1v_2\ldots v_i\) and \(R_2 = v'_1v'_2\ldots v'_i\), if \(\min\{e_1, e_2, \ldots, e_i\}\) is greater than \(\min\{e'_1, e'_2, \ldots, e'_i\}\), then we consider \(R_1\) as a more energy efficient route than \(R_2\).

To find out the route with the “max-min” residual energy, at the start of the second phase, a short message called routing agent is flooded into the network. Whenever a sensor node

\(1\) An upper bound can be set for \(\eta\) in certain applications to allow the system to periodically refresh data correlation tables.
receives a routing agent from its neighbor nodes, it first checks timer $T^1$ started in the previous phase. If the timer has already expired, it will rebroadcast this routing agent message immediately to all its neighbors. Otherwise it will simply wait until the timer fires, and then rebroadcast the routing agent. As there may exist several routes from the sink to a sensor node, only the source of the first routing agent is recorded as the “next-hop” for reporting data to sink. All subsequent routing agent messages are discarded to ensure loop free routing.

An illustrative example is shown in Fig. 3. When two copies of routing agent trying to reach node $v_{12}$ through two different routes, these intermediate nodes decide whether to forward this message immediately or to wait until its own timer $T^1$ expires. Since transmission time of this short routing agent is negligible compared to $T^1$, total delay of routing agent on each route is the same as the maximum timer value within each routes, which is shown in Fig. 4. According to EPC, the source of the first routing agent arrives at node $v_{12}$ will be recorded as $v_{12}$’s “next hop”. Consequently route $\{v_1,v_{12},v_{10}\}$ is selected.

When a sensor node receives the first routing agent message, it will start another timer $T^2$, which is defined in Eq.(3) ($E_2$ is defined similar to $E_1$). The sensor node will wait until $T^2$ expires to start the third phase of EPC.

$$F_2(e_i) = E_2 - e_i$$ (3)

**Algorithm 1** EPC algorithm for sink node
1: $\{\text{Data correlation detection phase}\}$
2: Sink node broadcast start-up message with data field setting to $NULL$
3: Wait for a certain amount of time
4: $\{\text{Routing phase}\}$
5: Sink node broadcast routing agent message
6: $\{\text{Reporting phase}\}$
7: Selected sensors report data via the energy-efficient routes

**D. Response with probability-based clustering**

In the reporting phase of EPC, each sensor node in the network keeps listening to the channel and waits until its timer $T^2$ that started in the second phase expires. Generally two actions need to be done in this phase as described below:

**Probability updating:** For each sensor node there is an associated probability $P$ to control sensed data reporting. At the beginning of each data collection period the value of $P$ at all nodes are set to be $P_0 = 1$. Suppose at the reporting phase $v_i$ overhears a data report message from $v_j$, it refers to its data correlation table, checks entry $\delta_{ij}$, and updates its data reporting probability value to $P = P_0 \times (1 - \delta_{ij})$. As the initial value of $P_0$ is 1, if there are $q$ neighbor nodes indexed from 1 to $q$ replied before $v_i$’s timer $T^2$ expired, the updated probability for $v_i$ would be:

$$P = \prod_{m=1}^{q} (1 - \delta_{im})P_0$$ (4)
In other words, a node $v_i$ forms clusters with its neighbors based on their data correlation factors.

**Data Reporting:** Every node will wait until its timer $T^2$ expires before responding. The sensor node with the smallest timer value will respond first with probability $P = 1$. This ensures that sensor nodes with larger residual energy report their data with high probability. All the other sensor nodes will follow up when their own timer expires and respond with updated probability $P_a$ calculated as Eq.(4).

### IV. Analysis of EPC Algorithm

In this section we analyze the effectiveness of the proposed EPC algorithm. That is, how EPC reduces transmission overheads while maintaining data accuracy, and how EPC achieves energy balancing via energy efficient routing. Moreover we will demonstrate how to adapt EPC to multisink environment with simple extension of the existing algorithm.

#### A. Validity of transmission reduction in EPC

Transmission costs are directly proportional to the amount of sensed data sending in the network, which is determined by the number of sensor nodes reporting data. In other words, the less number of sensor nodes involved in data collection process, the less transmission overheads incurred in EPC. Therefore our target in EPC is to find near-minimum number of sensor nodes involved in data reporting with guaranteed data accuracy. In order to achieve this we first introduce the concept of correlation cluster:

**Correlation Cluster:** Two sensor nodes $v_i$ and $v_j$ are in correlation relation $C$ iff they are neighbors and their sensed data are correlated (evaluated by user-defined threshold). A correlation cluster is the set of sensor nodes $\Psi \subset \mathcal{V}$ such that $(i, j) \in C$ for all distinct $i, j \in \Psi$.

In real world applications [8], [13], [14], data gathering is often conducted in fine resolution. Typically the neighbor range is limited within one hop. In EPC node $v_j$ is in $v_i$’s data correlation table if $v_j$ is within one hop from $v_i$. As such no data forwarding is involved and less overheads is introduced in overhearing the broadcast message. The relationship of sensor node number and correlation cluster is established as follows: when correlation clusters are formed and cover the whole network, we select one node from each cluster, and these nodes are the minimum number we need to collect with guaranteed data accuracy in gathering process. This assertion is guaranteed as degree of correlation is inversely proportional to the number of correlation clusters. The following example illustrates the assertion made above: suppose node $v_1$, $v_2$ and $v_3$ are in a correlation cluster, now after $T^2$ starts the node with the highest energy residual will report its data first according to EPC. The total number of data, $d_{total}$, gathered for the cluster would be:

$$d_{total} = \sum d_1 + d_2 (1 - \delta_{21}) + d_3 (1 - \delta_{31})(1 - \delta_{32})$$

As all sensor nodes are in one correlation cluster, when they consult data correlation tables the values of $\delta$ are near to 1 when system stabilized. Therefore according to Eq.(5) only $d_1$ is replied to sink node, which verify the transmission reduction in EPC.

#### B. Validity of balanced energy consumption

Energy consumption balancing is achieved through the second and the third phase of EPC. Note that if the system starts with approximately equal energy residual, based on EPC the selection process is random and energy consumption is balanced already. Thus we only consider the scenario where energy residual left on all sensor nodes are significantly different. We will demonstrate how energy are balanced in the second and the third phase of EPC respectively.

**Energy balance in energy-aware routing:** Max-min routes will result in balanced energy consumption on all sensor nodes. Therefore in energy-aware routing if routes established are max-min routes, balanced energy consumption is accomplished. First at the start of the second phase routing agent is flooded. Since this message is small the transmission time is negligible, timer $T^1$’s start approximately at the same time on all sensor nodes. Then according to Fig. 4 message delay is only determined by the longest $T^1$ in all sensor nodes involved in the broadcasting process in EPC. Also based on Eq.(1) the longest $T^1$ corresponds to the lowest energy residual. Now if two routes are established EPC will select the route with shorter delay which in turn corresponds to the maximum of the minimal residuals. This proves that the established routes based on delays are exactly what we are looking for, the max-min routes.

**Energy balance in data reporting:** In data reporting phase suppose in one correlation cluster there are $k$ nodes from $v_1$ to $v_k$, and their energy residuals are ranked in non-decreasing order $e_1^r \leq e_2^r \leq \cdots \leq e_k^r$. Then according to Eq.(3) we have $T_1^2 \geq T_2^2 \geq \cdots \geq T_k^2$. Obviously $T_k^2$ will expire first and $v_k$ will respond to the sink node. As all other nodes $v_i$, $i \in \{1...k-1\}$ are in the same cluster with $v_k$, based on probabilities in their data correlation tables they are quite unlikely to reply. In this way the sensor node with the highest energy is selected and energies are saved for all the other sensor nodes with relatively low energy residuals.

#### C. Extension to multiple-sink scenario

In Section III, the data gathering scenario with only one sink node in a network is considered. However, our EPC algorithm can also be applied to a network model with multiple sink nodes. To distinguish these sink nodes, we assign each sink node a unique ID. This ID information is attached into various facilities such as routing messages and timers. Specifically, a startup message from a sink node now has the format of $<\psi$,'Data', Sink_ID$.> When timer $T^1$ launches at each sensor node it corresponds to the start request from one specific sink node, represented as $T^1$(Sink_ID). Now each sensor node might maintain multiple $T^1$s at the same time. Upon receiving a routing message from sink $s_k$, a sensor node will check if its $T^1(k)$ expires then make forwarding decision. Another difference exists in the route establishment process. Instead
of recording only the next hop information, now each sensor node will also record the destination sink ID as part of the routing information. When each sensor node reports data it will send to its next hop along with that final destination to discriminate routes for different sinks. Timer $T^2$ operates similar as described in Section III. The modified EPC will suits multiple-sink scenario well as it is expected to achieve the same effects for transmission costs reduction and balanced energy consumption.

V. PERFORMANCE EVALUATION

We evaluate the performance of EPC algorithm through extensive simulations on TOSSIM [12]. To inspect the performance we use three major metrics: (1) Total transmission cost in data gathering processes; (2) Data accuracy with respect to user-defined thresholds; (3) Balance of energy consumption in a network.

A. Simulation setup

We generate 25 sets of synthetic data for a $100m \times 100m$ two-dimensional grid environment, using the algorithms provided in [11]. Sample correlated data distribution and their variograms are shown in Fig. 5. Each value in a data set corresponds to a specific point in the grid environment. 625 sensor nodes are uniformly distributed into the simulated environment. Every sensor node has a randomly generated residual energy value $\epsilon$. A data collector/sink appears at a random location in the grid environment to simulate a mobile sink. To evaluate EPC’s performance given different data accuracy requirements, we adopt 7 sets of user-defined thresholds from $\tau = 0.5\%$ to $\tau = 11\%$ for Eq. (2). During each simulation, sensor nodes in the network fetch their data from a synthetic data set, and determine their data correlation based on a predefined $\tau$ value. Since a pseudo random number generator is used in implementing probability-based clustering, we run each simulation for 20 rounds to get average results.

B. Simulation results

Reduced transmission costs: In the first set of simulations we investigate how much transmission cost is reduced using EPC. To measure the transmission cost, we define that every time a sensor node sends or receives a data package, its energy residual is decreased by $\xi$; Whenever a control message, such as startup or routing agent message, is broadcasted or received by a node, the node’s energy is also decreased, but by half of $\xi$. This is a reasonable assumption since it is well known that transmission cost is proportional to package size. For better illustration we use the SIMPLE algorithm [18] as baseline for comparison. We test SIMPLE using the same simulation setup as EPC, and calculate the transmission reduction percentage accordingly. For a total of 175 test runs (7 thresholds on 25 data sets), EPC out performs SIMPLE with a maximum energy consumption reduction of 71% when $\tau$ is 11%. Fig. 6 plots the percentages of reduced transmission costs. We observe that as the value of $H$ increases less data packets are transmitted during data gathering, which is reflected in the trend of increased reduction percentage. This is because the higher the degree of correlation, the less correlation clusters exist in the network. Since sensor nodes in a correlation cluster are unlikely to report sensed data repeatedly, a large portion of sensor nodes do not participate in data transmission. Another interesting observation is that when the value of $H$ is fixed, transmission cost reduction becomes higher as $\tau$ increases. This is reasonable since higher error tolerance results in less sensor nodes participating in data transmission and thereby saves more energy. Therefore finding the tradeoff between
accuracy and energy efficiency is important.

![Graph showing accuracy of average function on gathered data]

**Fig. 7.** Accuracy of average function on gathered data

**TABLE I**

<table>
<thead>
<tr>
<th>H</th>
<th>τ=0.1%</th>
<th>τ=0.3%</th>
<th>τ=0.5%</th>
<th>τ=0.7%</th>
<th>τ=0.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>3</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>1.33%</td>
<td>0.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>5</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>6</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>6.3%</td>
<td>5.7%</td>
</tr>
<tr>
<td>7</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>6.3%</td>
<td>6.2%</td>
</tr>
<tr>
<td>8</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>9</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>1.3%</td>
<td>1.35%</td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>H</th>
<th>τ=0.1%</th>
<th>τ=0.3%</th>
<th>τ=0.5%</th>
<th>τ=0.7%</th>
<th>τ=0.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>2.19%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>3</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>4</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>5.46%</td>
<td>0.0%</td>
</tr>
<tr>
<td>5</td>
<td>0.0%</td>
<td>1.58%</td>
<td>0.02%</td>
<td>0.0%</td>
<td>1.6%</td>
</tr>
<tr>
<td>6</td>
<td>0.0%</td>
<td>1.59%</td>
<td>0.001%</td>
<td>2.44%</td>
<td>0.0%</td>
</tr>
<tr>
<td>7</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>8</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>3.3%</td>
<td>5.21%</td>
</tr>
<tr>
<td>9</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

**Data accuracy:** We examine the relative errors in collected data when only partial sensor nodes participate in data reporting. To ease comparison we perform three data aggregation functions on collected data, AVG, MAX and MIN. The results are exhibited in Fig. 7, Table I, and Table II respectively. All results are compared to data collected by SIMPLE, using the same setups as in Fig. 6. From the comparison we observe that EPC provides very accurate data aggregation results that are always under the user defined thresholds. In Fig. 7 where AVG function is used, as $H$ increases the relative error slightly goes up. This can be explained as larger value of $H$ will lead to fewer sensor nodes reply. Further, across multiple curves if $H$ is kept fixed, the relative error rises with larger $\tau$. This is natural as larger $\tau$ means higher degree of error tolerance. However, all the relative errors are much less than corresponding $\tau$, indicating that EPC achieves reduced transmission without affecting the overall accuracy too much. For each set of sensed data, we also randomly select partial sensor nodes’ data to get the AVG result, as plotted in Fig. 7. This is used to rule out the possibility that any subset of the simulation data set has the similar AVG result as the whole data set. As to MAX and MIN functions, we listed error percentages under different correlation degrees and different $\tau$ settings in Table I and Table II, we observe that most of the entries in these tables are zeros, which means negligible relative errors in most of the times. This can be explained as that a sensor node sensed extreme data (max or min) is quite likely to report its data, since there is little correlation amongst its neighbors.

**Energy balancing:** We also investigate EPC’s performance on balancing energy consumption. In order to discriminate the curves we generate random energy values in different ranges for different data sets. One appropriate metric to evaluate the balancing performance for randomly generated energy values would be the standard deviation $\sigma$, as it reflects the deviation from the average value of residual energy. Hence smaller $\sigma$ in general indicates more balanced energy consumption. We run EPC algorithm in 10 continuous rounds for the purpose of checking the variation of balancing indicator $\sigma$. Fig. 8 exhibits the simulation results. We observe that for each curve, $\sigma$ is continuously decreasing as simulation goes. This can be explained that, in EPC sensor nodes with larger residual energy value will report sensed data with higher probability and participate in routing. As a result the energy is consumed at nodes with larger deviation from the balanced situation. The trend exhibited in the figure demonstrates that EPC is quite effective in balancing energy consumption in the whole network.

**VI. Conclusion**

In this paper we propose EPC, a novel probability-based correlated data gathering algorithm for energy-efficient data collection in large-scale WSNs. EPC consists of three phases. After a startup message is flooded in the first phase, EPC establishes the most energy-efficient route through energy-aware routing, and excessive transmissions are eliminated through probability-based clustering. To evaluate the performance of
EPC, we conduct extensive simulations using synthetic data sets. The simulation results demonstrate that the transmission costs can be reduced up to 71% in data gathering. The results of data accuracy for data aggregation are also satisfactory, with averaging measurements of 98.1% accuracy on AVG, and near 100% accuracy on MAX and MIN. Moreover, EPC achieves good energy balancing among all sensor nodes in a network. Our ongoing work will further extend the EPC algorithm to address data gathering issues with multiple mobile sinks and implement our algorithms on a testbed with hundreds of motes and a dozen of robots as mobile sinks.

REFERENCES


