Parameter-Based Data Aggregation for Statistical Information Extraction in Wireless Sensor Networks

Hongbo Jiang, *Member, IEEE*, Shudong Jin, *Member, IEEE*, and Chonggang Wang, *Senior Member, IEEE*

***Abstract*—Wireless sensor networks (WSNs) have a broad range of applications, such as battlefield surveillance, environmental monitoring, and disaster relief. These networks usually have stringent constraints on the system resources, making data-extraction and aggregation techniques critically important. However, accurate data extraction and aggregation is difficult, due to significant variations in sensor readings and frequent link and node failures. To address these challenges, we propose data-aggregation techniques based on statistical information extraction that capture the effects of aggregation over different scales. We also design, in this paper, an accurate estimation of the distribution parameters of sensory data using the expectation–maximization (EM) algorithm. We demonstrate that the proposed techniques not only greatly reduce the communication cost but also retain valuable statistical information that is otherwise lost in many existing data-aggregation approaches for sensor networks. Moreover, simulation results show that the proposed techniques are robust against link and node failures and perform consistently well in broad scenarios with various network configurations.**

***Index Terms*—Algorithm/protocol design, data aggregation, sensor networks, statistical information extraction.**

I. Introduction

A SENSOR network consists of the sink (or the base station) and a set of autonomous sensor nodes that spontaneously create impromptu communication links and then collectively perform a task without much help from any centralized servers. An individual sensor node is generally an inexpensive wireless device with limited resources, such as a power supply. Thus, energy consumption is one of the most important factors to be considered in designing protocols and algorithms for sensor networks, such as data extraction or query processing.

Data extraction copes with how the sink can efficiently and accurately obtain the sensory data from a sensor network. To that end, numerous techniques have been proposed for query processing in sensor networks. They intend to minimize energy cost and maintenance and, at the same time, obtain accurate query results under even lossy wireless communication environments. Those techniques, however, all consider accurate sensor readings, despite the fact that, for many applications and sensing modalities, such as reporting temperature readings, it is unnecessary for each sensor to report its entire data stream in full fidelity. Moreover, very often, it is unrealistic to obtain the exact query results due to the inherent unreliability of sensory readings and the lossy communication links. In addition, each message transmission involves operations that consume significant amount of energy. Therefore, *in-network* data *aggregation* as a promising technique has been introduced in recent years. A straight method for in-network aggregation is to compute such aggregates as AVERAGE, SUM, and COUNT over a routing tree, minimizing both the number and the size of messages. The Cougar [29], TinyDB, and TAG [17] architectures are based on this method. They, however, have certain limitations in search efficiency and system scalability. First, sensor networks often have a moderate (or even very high) link/node failure rate, which can result in the loss of a large amount of information. In other words, failure is inevitable in a routing tree-based aggregation. If this happens at a location close to the sink, the impact on the resulting aggregate can be significant. Second, aggregation measures such as AVERAGE, SUM, and COUNT are not sufficient in some applications. There exist situations such as declarative queries [7] and distributed message-passing algorithms [21], where it is necessary to provide the distribution of sensor readings. By having the estimation of data distribution available at the base station, users can pose more complex queries and perform more sophisticated analyses.

The focus of this work is to extract statistical information from sensory data while keeping the communication cost low. In this paper, we aim to design efficient aggregation techniques based on statistical information extraction, which can answer various kinds of queries, such as "*What is the percentage of nodes whose readings are higher than X?*", *"What is the confident interval with 90% confidence level?", and "Are there outliers that are away from a given data distribution function?"* These queries can be used in many sensor applications, such as monitoring traffic congestion [18], where the question becomes "*What is the percentage of sensor nodes detecting vehicle speed whose readings are lower than 20* mi/h?" If this percentage is larger than a given threshold, traffic congestion could exist. We argue that guaranteeing exact data extraction is generally impractical when packet losses are eminent. Therefore, we instead consider approximate methods.

Manuscript received October 17, 2009; revised March 17, 2010, May 20, 2010, and July 7, 2010; accepted July 20, 2010. Date of publication August 3, 2010; date of current version October 20, 2010. This work was supported in part by the National Natural Science Foundation of China under Grant 60803115 and Grant 60873127, by the Fundamental Research Funds for the Central Universities under Grant M2009022, by the Youth Chenguang Project of Wuhan City under Grant 201050231080, by the CHUTIAN Scholar Project of Hubei Province, and by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry. An earlier version of this work was presented at the 26th International Conference on Distributed Computing Systems, Lisboa, Portugal, July 4–7, 2006. The review of this paper was coordinated by Dr. L. Li.

H. Jiang is with the Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: hongbojiang2004@gmail.com).

S. Jin is with Case Western Reserve University, Cleveland, OH 44106-7071 USA.

C. Wang is with NEC Laboratories America, Princeton, NJ 08540 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2062547

0018-9545/$26.00 © 2010 IEEE

We make the following contributions in this paper. First, we present a scalable and robust aggregation approach based on statistical information extraction for arbitrary sensor networks. We utilize a mixture model to approximate the exact data distribution, which is the key to scaling up the system and making the aggregation more robust. In addition, extensive simulation results based on real and synthetic data demonstrate that the proposed data aggregation schemes of extracting statistical information dramatically reduce the communication cost, compared with simple centralized schemes, and provide a very accurate approximation of data distribution. It is shown that parameter-based data aggregation techniques are viable approaches to facilitating statistical information extraction in wireless sensor networks (WSNs).

The rest of this paper is organized as follows: In Section II, we discuss related work and describe the network model and the assumptions used throughout this paper. Section III introduces a statistical model for the study of decentralized aggregation in WSNs. Section IV presents theoretical analysis and discussion for the proposed schemes. Section V provides numerical results and performance comparisons with different aggregation techniques. Section VI concludes this paper.

II. Background and Related Work

A. Network Model and Assumptions

We consider a network with n sensor nodes, where all nodes are working in a common modality. The sensor nodes are powered by small batteries, and their lifetime is primarily dependent on the extent to which the battery power is conserved. This characteristic motivates the need for data aggregation techniques. Such techniques would minimize the number of message to save power. We do not assume that the sensor connectivity is uniform throughout the network. For instance, some nodes might have better connectivity (i.e., higher degrees) than others. It is even possible that some nodes are disconnected from others.

B. Statistical Information and Aggregation Techniques

To collect only statistical information rather than the whole data is practical and applicable in many cases. First, most of sensory data provides little help in improving the answer quality of queries [7]. As a result, it is not worth sending all sensory data back to the sink since it is costly in both time and energy consumption. Second, the typical aggregation strategies [17], [29] could ignore the relevance or irrelevance of the data [21]. On the other hand, Deshpande *et al.* [7] demonstrated that statistical models are capable of providing meaningful answers while being much more efficient to compute with respect to both time and energy consumption. Third, in many cases, we may want to obtain statistical information that can be examined by both a human being and a program. For example, a traffic analyst for traffic light configuration and traffic regulation may be interested in traffic distribution in the whole metropolitan area. In addition, it is beneficial to summarize the spatial pattern of sensory data at multiple resolutions so that the information can be easily queried in finer detail and can be found by drilling down to the higher level. Other examples include declarative queries [7], estimation of the distribution of the total number of targets [9], and distributed message-passing algorithms [21].

Several studies have been focused on providing such statistical summarization. Deshpande *et al.* [7] proposed a statistical model to enrich interactive data querying. They designed a novel architecture for integrating a database system with a correlation-aware probabilistic model. It reduces the number of expensive sensor readings and radio transmissions that the networks must perform. However, it considers neither the routing architecture for finding the optimal query processing nor the link/node failures that are inherent properties in WSNs. Thus, it cannot efficiently handle the impact of packet losses. Prior to [7], there has been other related work on approximate probabilistic querying for sensor networks, e.g., [8] and [30]. In particular, the Gaussian Abstract Datatype [30] models the uncertainty as a continuous probability distribution function over possible measurement values.

On the contrary, Shrivastava *et al.* [23] presented a quartile-maintaining algorithm based on a q-digest structure (hereinafter, e.g., the q-digest algorithm). It preserves the information of high frequency values while compressing the information of low frequency values. Hence, it provides a good approximation when there are wide variations in the frequencies of different values. Although it performs well under the network environment with a bound-constrained readings and perfect link quality, the approach is not practical in the sense that such environment is unrealistic.

III. Data-Aggregation Scheme

As our objective is to obtain the statistical information of the sensory data, it will suffice if aggregation algorithms return the probability distribution of the sensory data. In this section, we present the theoretical foundation, describe the process of aggregation, and formulate and solve the problem of distribution parameter estimation by leveraging general mixture model techniques.

Consider a sensor network where the link between a pair of sensor nodes is lossy. Packets can be corrupted or dropped due to link (and route) failures. The routing algorithm (and protocol) is used to decide how to forward data toward the base station. With multipath routing, each node can have multiple predecessors and successors in the routing graph. Each node aggregates the results received from its predecessors and the value from its own reading and then sends the result to one or more of its successors. We ignore the details on how to generate the routing graph but point out that our aggregation techniques (which are the focus of this section) can work with any multipath routing scheme.

A. Theoretical Approximation

Consider a sequence of samples or sensor readings X. Let $f(X)$ denote its continuous probability model and $f(x)$ denote its probability density function. Theoretically, an infinite-dimension mixture model can asymptotically approach any

probability density function [19]. That is, for any continuous probability density function, if we properly select the parameters of the base distributions in the mixture model, then we have

$$f(x) = \lim_{N\to\infty} \sum_{i=1}^{N} \alpha_i B_i(x;\theta_i)$$

where $B_i(x;\theta_i)$ is the ith base distribution with parameter θ_i, and $\sum_{i=1}^{N} \alpha_i = 1$. In the rest of this paper, we use Θ to denote all the parameters on the right side (all θ_i and all α_i). Hence, this leads to an aggregation technique: Sensor nodes only need to transmit packets that contain the distribution parameters, instead of the individual values. If we can carefully choose the parameters, a good approximation is attainable. Furthermore, since the number of parameters determines the message size, we can adjust it to trade off the error rate and the communication cost. We design our aggregation algorithm based on this theoretical approximation.

B. Aggregation Process

The general process of our aggregation algorithm is given as follows: The aggregation starts with the remote nodes toward the sink. A remote node may first send packets to its successors using multipath routing. Intermediate sensor nodes, upon receiving packets from other sensors, will aggregate them with its own data and forward aggregation results contained in packets to their successors. This distributed and iterative process continues until the base station receives and aggregates the final results.

The packet sent by node v contains a 2-tuple (w_v, Θ), where w_v is the weight of the aggregate in this packet, and Θ is the set of all parameters of the specific mixture model. The first field is recursively computed as follows:

$$w_v = \frac{\sum_{u\in PRED_v} w_u + 1}{(1-p)\cdot |SUCC_v|} \tag{1}$$

where p is the packet loss rate, $PRED_v = \{u | v \in SUCC_u\}$ is the set of immediate predecessors of v in the routing graph, and, as described before, $SUCC_v$ is the set of immediate successors of v.[1] Notice that w_v approximates the weight for an aggregation value contained in the packet sent by node v. First, w_v is proportional to the sum of all weights from its predecessors plus one (for v's own data). Thus, the weight reflects the effect of aggregation operation at v. Second, w_v is inversely proportional to the number of successors. Thus, this weight also reflects the effect of splitting the results since, in the next hop, each successor will perform its own aggregation operation. Both are important since every individual value (by each node) should have an equal weight on the final outcome at the sink.

The packet loss rate p represents the approximate average loss rate.[2] The packet loss could be due to wireless interference, congestion loss resulting from buffer overflow, and/or node failure. While the accuracy of our aggregation highly depends on an accurate packet loss rate, we acknowledge that it is not easy to estimate this metric in reality. One may use historical aggregate connectivity data such as [1], or each node v can send small packets periodically to its neighbor v' (or some complicated techniques, such as [5]) such that it can estimate the average loss rate $p_{v,v'}$ over time [6]. Accordingly, the weight (1) of the aggregate in the packet can be adjusted to $w_{v,v'} = (\sum_{u\in PRED_v} w_{u,v} + 1)/((1-p_{v,v'})\cdot |SUCC_v|)$. For simplicity, however, our performance evaluation only uses a fixed packet-loss rate. The calculation of $w_{u,v}$ has also taken into consideration the effect of packet losses and therefore prevents the loss of aggregation information.

Notably, at the sink s, the expression $\sum_{u\in PRED_s} w_u + 1$ approximately represents the total number of readings. In other words, we can approximate the duplicative–insensitive COUNT query by this value at the sink. Since we have taken the packet losses into consideration, this value provides an accurate and robust approximation.

C. Estimating Distribution Parameters

The key problem in our aggregation algorithm is how to estimate the parameters of data distribution function. In this section, we formulate the problem of estimating distribution parameters Θ and describe the expectation–maximization (EM) algorithm to solve the problem. We then present a case study: EM algorithm for Gaussian mixture model (GMM) estimation.

1) Problem of Parameter Estimation: Let us now focus on the parameters Θ. At each sensor node, the input information includes both the value acquired from its own sensing and the values received from its predecessors. The precisely defined input distribution is

$$B_T(x) = \sum_{v'\in PRED_v} (\alpha_{v'} B_{v'}(x;\theta_{v'})) + \alpha_0 B_0(x;\theta_0)$$

where $\alpha_{v'} = w_{v'}/(\sum w_{v'} + 1)$ and $\alpha_0 = 1/(\sum w_{v'} + 1)$ are normalized weights. To generalize the value of the local reading of a sensor, we assume a specific single-value distribution B_0 with θ_0.

From the view of each individual node, the input data of a set of parameters lead to an output Θ_v, which is a set of parameters with smaller size. The problem is a new distribution $B_v(x)$, which was derived from Θ_v, should constitute an approximation of the original distribution $B_T(x)$.

2) EM Algorithm for Parameter Estimation: To solve the problem of parameter estimation, we utilize the EM algorithm, which is standard for finding maximum-likelihood (ML) estimates of parameters in probabilistic models [19]. It is ideally suited since it produces ML estimates of parameters for the distribution representing the observation.

First, before the EM procedure, the measurements should be provided as the algorithm input $x_1, x_2, \ldots, x_k$. For this purpose, we generate data from each input model. Because w_v approximates the weighted value for a data set, for each $B_{v'}(x;\theta_{v'})$, it is easy to generate $w_{v'}$ independently and

[1] $SUCC_v = \emptyset$ if v is the base station. $PRED_v = \emptyset$ if v is a source node.

[2] For a sensor network, we assume that link failures and node failures have the same effect in the sense that both result in the nondelivery of aggregate values. In our later experiments, we will evaluate the impact of both link failures and node failures.

identically distributed (i.i.d.) random values with this distribution. Hence, in addition to its own reading, the node can obtain $k = \sum_{v'}(w_{v'})$ measurements. This random number generation method is efficient and accurate for the next EM algorithm.[3]

Let Z_{ij} be a random variable that is equal to 1 if and only if observation j comes from the ith Gauss. We define a complete data log likelihood function

$$Q\left(\Theta|\Theta^{(n)}\right) = \sum_j \sum_i Z_{ij}\left\{\log \alpha_i^{(n)} B_i\left(x_j; \theta_i^{(n)}\right)\right\}.$$

First, we modify the complete data log likelihood function by replacing Z_{ij} with its conditional expectation $E(Z_{ij}|x_j)$

$$\sum_j \sum_i E(Z_{ij}|x_j)\left\{\log \alpha_i^{(n)} B_i\left(x_j; \theta_i^{(n)}\right)\right\}.$$

Notice that $E(Z_{ij}|x_j) = P(j \text{ from model } i|x_j)$; then, we may estimate $E(Z_{ij}|x_j)$ if we have preliminary estimates for $\Theta^{(n)}$. Then, an individual re-estimation step that derives $\Theta^{(n+1)}$ from $\Theta^{(n)}$ takes the following form:

$$\Theta^{(n+1)} = \texttt{argmax}_\Theta Q\left(\Theta|\Theta^{(n)}\right).$$

In other words, $\Theta^{(n+1)}$ is the value that maximizes (M) the expectation (E) of the complete data log likelihood with respect to the conditional distribution of the latent data under the previous parameter value.

An additional improvement technique proposed here is that, by carefully selecting the initial value $\Theta^{(0)}$, the procedure of EM algorithm can converge fast. Fortunately, the weight for an aggregation value is contained in each node's packet. Hence, a better initial point can be provided, even via a simple linear interpolation. A straightforward method is

$$\theta_i^{(0)} = \sum_j \frac{w_j \cdot \theta_{i,j}}{\sum_k w_k}. \tag{2}$$

Surprisingly, this initial point can reach the stable point fast via just a few iterations in our experiments.

3) Special Case of GMM Estimation: In this paper, our implementation focuses on a GMM [2], [3]. Here, we emphasize, albeit representative but not exclusive, Gaussian distribution. Our framework can introduce new mixture models based on any other distribution via a seamless interface with no change. With similar analysis, one can easily develop aggregation techniques based on Poison distribution or Erlang distribution.

A standard Gaussian distribution function is given as follows:

$$G(x; \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\}.$$

Consequently, for a κ-GMM, the model contains 3κ parameters. In succession, like the usual EM algorithm, let Z_{ij} be a random variable that is equal to 1 if and only if observation j comes from the ith Gauss. In a similar procedure, let t_{ij} denote an estimate for $E(Z_{ij}|x_i)$ under the preliminary estimates of α_i, μ_i, and σ_i. We make a second modification to the complete data log likelihood function by replacing $E(Z_{ij}|x_j)$ with t_{ij}, i.e.,

$$\sum_j \sum_i t_{ij}\left\{\log \alpha_i - \frac{1}{2}\log\left(2\pi\sigma_i^2\right) - \frac{(x_j-\mu_i)^2}{2\sigma_i^2}\right\}.$$

Treating t_{ij} as fixed, we can calculate the values of α_i, μ_i, and σ_i^2 that optimize the twice-modified complete data log likelihood function. The optimal values are given as follows:

$$\alpha_i = \sum_j \frac{t_{ij}}{k}$$

$$\mu_i = \sum_j \frac{t_{ij}x_j}{\sum_j t_{ij}}$$

$$\sigma_i^2 = \sum_j \frac{t_{ij}(x_j-\mu_i)^2}{\sum_j t_{ij}}.$$

These become our new estimates for α_i, μ_i, σ_i^2. Then, we can alternate between re-estimating $E(Z_{ij}|x_j)$ (expectation step) and re-estimating α_i, μ_i, and σ_i^2 (maximization step) until the estimates become stable.

IV. Analysis and Discussion

A. Space and Time Complexity

Recall that each packet contains a 2-tuple (w_v, Θ). If we use a double-precision floating-point number (8 B) to represent each field, the message size (here, our analysis is based on GMM) would be $n_{\text{size}} = 8 + 24\kappa$ B, when we choose a κ-degree mixture model. This is still a large number. One method is to use floating-point numbers (4 B each) or even 2-B integers to approximate the parameters. Certainly, we can also pick a small value for κ. There is a tradeoff between space complexity (message size) and the accuracy of aggregation. That is, the more parameters we use, the more accurate results the system can provide. This tradeoff also provides an opportunity for the applications to choose the desired mixture model.

In addition to space complexity, time complexity is also an important factor for an efficient aggregation algorithm. This is a drawback of the traditional EM algorithm. Although the EM algorithm can theoretically guarantee convergence, its speed to converge is not stable. We overcome this limitation using the following steps: First, an appropriate initial value estimation is set up. Second, we limit the number of maximum iterations by a constant. Thus, even for the worst-case scenario, the algorithm operates within the constant time $O(1)$. Overall, the time complexity for all the nodes is $O(n)$. We find that, even if the EM algorithm is stale, the appropriate initial value estimation can still lead to a good result. This is because the initial value itself represents a good knowledge of the linear combination of foregone information.

[3]In the case where the value of w_v is too small, we generate more numbers. For instance, we can generate $10 * w_v$ numbers. However, to carefully capture the influence, an additional ten duplicate values of its own readings should be severed as the input data in the mean time. (In most cases of our experiments, ten is large enough in terms of accuracy.)

B. Energy Consumption

We do not strive to analytically derive the energy consumption of our algorithm for two reasons. First, the goal of our algorithm is to reduce the number/size of packet transmissions as communication cost, which often dominates the energy consumption and is proportional to the size of the transmitted packets [15], [28]. Second, energy consumption depends on many factors, such as system configuration, which may differ for different sensor networks.

It is worth noting that there have been evidences showing that energy consumption due to computation (e.g., multiplication and other operations) is insignificant, compared with the communication cost. For example, in the Rivest, Shamir, and Adleman (RSA) encryption experiment in [27], for the same size of data, the energy consumption of computation is less than 1% of that for transmission. Note that each step in the EM algorithm should contain fewer computations, compared with RSA encryption. Therefore, even when the EM algorithm needs many steps, its energy consumption is still negligible.

C. MAX, MIN, MEDIAN, and Top-k Queries

One limitation of our schemes is that duplicate insensitive statistics such as MAX and MIN cannot be well supported. The main reason is that, as mentioned in Section I, guaranteeing exact data collection in full fidelity is generally impractical. On the other side, for applications such as outlier detection where monitoring extreme values (e.g., MAX/MIN) is a fundamental problem [24], our data-aggregation schemes are still capable of being extended to deal with this problem via multiple rounds of queries [22]. After the first round of queries described in Section III, the sink obtains the data distribution over the entire network. In the second round query, the sink diffuses the data-distribution parameters to sensor nodes. Each node then calculates the likelihood of observing its reading, given the data distribution. If this likelihood is less than a threshold (e.g., 5%), the sensor node will forward its reading back to the sink. Eventually, the sink receives a set of potential extreme values, and thus, it can identify whether each of the returned values is an outlier (or MAX/MIN values).

While the aforementioned multiround query solution for MAX/MIN query seems more complicated, compared with the typical solution, where each sensor node computes MAX/MIN aggregates over a routing tree and thus only one round query is needed, we emphasize that the multiround query solution is more flexible to be generalized to answer complex queries, such as Top-k query [28] and MEDIAN query [23]. Likewise, in the first round of queries, the sink obtains the data distribution over the network. In the second round of query, the sink diffuses the data-distribution parameters to sensor nodes. Each node first calculates the confident interval, given the confidence level k/n (or a given threshold δ for MEDIAN query), where n is the number of total readings. The node then calculates the likelihood of its readings falling into this confident interval, given the data distribution. If this likelihood is larger than a threshold (e.g., 95%), the sensor node forwards its reading back to the sink. Finally, the sink is able to answer Top-k or MEDIAN queries based on these returned values. Overall, we are not saying that our proposed algorithm is the best solution for supporting all these MAX, MIN, MEDIAN, and Top-k queries, yet our goal is to capture the distribution of sensory data; additionally, our approach can be easily extended for supporting MAX, MIN, MEDIAN, and Top-k queries.

V. Experimental Evaluation

To evaluate the performance of our algorithm, we conducted a series of experiments using the simulator developed under C++ and quantified several performance aspects of our algorithm. In this section, we first describe our experimental evaluation methodology and then present the results and analysis.

A. Experimental Methodology

1) Data Sets: Several data sets are used to evaluate our basic algorithm for data aggregation. Similar to [23], the data set is the spatially correlated elevation data released by the U.S. Geological Survey.[4] We select a 100×100 spatial subset of the original data as our data sets. This subset size can be easily scaled to cover the square where all nodes are.

2) Compared Algorithms: One of main advantages of our algorithms is that it can maintain the original data distribution as accurate as possible based on the propagation of statistical information. It is worth noting that many recent works focus on simple aggregation functions, such as SUM, COUNT, and AVERAGE. Since the q-digest [23] is the only work, by far, to provide approximation for statistical information, we will compare the performance of our proposed algorithm (labeled as "GMM" in the figures) with q-digest, although it just deals with integer and finite data input. We do not compare the efficiency of MIN/MAX queries over algorithms since our goal is to capture the distribution of sensory data. Unless otherwise stated, our implementation is based on the 2-GMM model.[5] To make a fair performance comparison among different algorithms, we first, carefully choose several representative parameters. In q-digest, we choose $k = 2$ and $\sigma = 256$, where σ is the bound value for input data and all the successive experimental sensor values must be less than this bound. With this setting, the q-digest has a comparable message size with our 2-GMM implementation. Second, we just round the floating-point numbers into q-digest in that this scheme only deals with integers. In addition, in our default sensor network configuration, the links between sensor nodes are unreliable, and packet loss rate is 5%, unless otherwise stated (labeled as "q-5%" in the figures for the q-digest algorithm). However, we also run the q-digest algorithm without packet losses just for comparison (labeled as "q-0%" in the figures).

[4]We downloaded "susanville.gif" from http://edc.usgs.gov/geodata/.

[5]Theoretically, more parameters means more accurate results. However, we find that two is big enough to get a good performance. We will also discuss the sensitivity to the number of parameters in the next section.

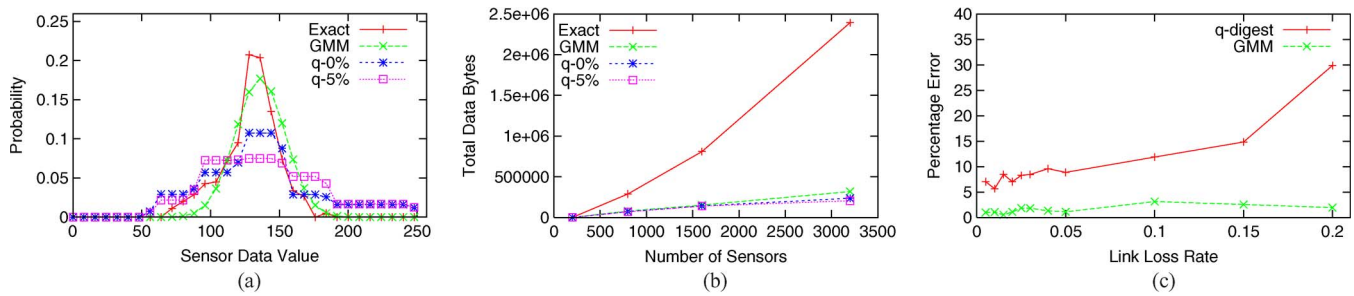

Fig. 1. Performance comparison of the algorithms in the presence of link failures. (a) Histograms (5% link failures). (b) Communication cost (5% link failures). (c) Error rates.

B. Experimental Results of the Proposed Data-Aggregation Scheme

In this section, we compare the performance of our proposed data-aggregation scheme and the q-digest algorithm in the presence of link failures. In the q-digest algorithm, we divide the data values into 32 equiwidth buckets and query all summaries to find the number of values in each bucket. For fair comparison, in our model-based approximation, we use the cumulative probability distribution to compute the value in each bucket (it is also a histogramlike form), e.g., if a bucket range is x_0 and x_1, then the value at this bucket is $F(x_1) - F(x_0)$. To the best of our knowledge, so far, no aggregation algorithms can provide approximation in continuous form. We believe this form is a suitable way to represent data distribution and compare the performance of different algorithms.

The network topology is generated as follows: We assume that all sensor nodes have a fixed radio range of 60 units. If two sensor nodes are within the range of each other, they are considered neighbors. We assume that this range does not change over time, and it is not affected by interference during communication. We have found that, on average, each node can directly communicate with about six neighbors at any point of time [20]. This guarantees that most nodes are connected to the network. For some experiments, we need to vary the number of sensor nodes in the field. In such cases, we also vary the size of the square area in which the nodes are distributed to keep the node density constant. In addition, our proposed framework is inserted on top of the data-routing layer.

We randomly choose a node as the sink in each experiment. The sink initiates a query by sending a query packet to all its neighbors, which forwards this query to their neighbors, and so on. This way, we construct a routing graph. Then, each node works as we described in Section III-B: each node aggregates the results received from its predecessors and the value from its own reading and then sends the result to one or more of its successors.

1) Impact of Link Failures: Fig. 1(a) shows the histogram results from our algorithm and the q-digest algorithm. In the figure, we plot the point at the beginning of each interval. As we can see, even a relatively low link failure rate (5%) causes a large error in the results of the q-digest algorithm. Under the same condition, our algorithm provides a much more accurate approximation. We also observe that, even with the 0% link failure rate, the q-digest approximation is not very good. That is because a traditional approximation technique like q-digest

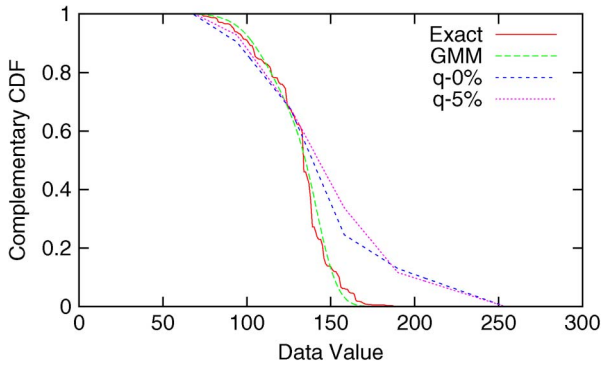

Fig. 2. Exact and approximate query results in the presence of 5% link failures.

is based on the assumption that the data value is uniformly distributed. For instance, in the low level close to the root in the q-digest structure, the information obtained by a node is assumed to have uniformly come from all the leave nodes. That is why the curve of the q-digest algorithm looks *flat*, even though the input data do not follow the uniform distribution at all. We suspect that this claim holds for any other traditional methods that assume uniform distribution of the data values.

We also compare the communication cost of the algorithms. Fig. 1(b) shows the total communication cost. The communication cost is in the unit cost of transmitting 1 B. For example, a double-precision floating number requires 8 B. While significantly improving the accuracy of approximation in the lossy environment, our multipath GMM aggregation technique inevitably introduces extra message overhead. It can be seen that the communication cost noticeably increases in our algorithm, although it only linearly increases with the number of sensor nodes. Overall, our duplicate-insensitive fault-tolerant algorithm only results in a little more message overhead, compared with the q-digest algorithm. However, the communication cost of GMM is much lower than that of Exact query, where a full list of sensor values must be sent back to the sink.

In addition, we compare the error rates of the algorithms when the link failure rate increases. Fig. 1(c) shows the effect of link failure rate on the performance (accuracy) of each algorithm under study. The error is defined as the ratio of the difference (between a sample and the true value) to the true value. That is, for a sample value x and correct value $\bar{x}$, the error is defined as $(|x - \bar{x}|)/\bar{x}$. The figure shows that, while computing the AVG aggregate, the error rate of q-digest quickly increases as the link failure rate increases. On the other hand,

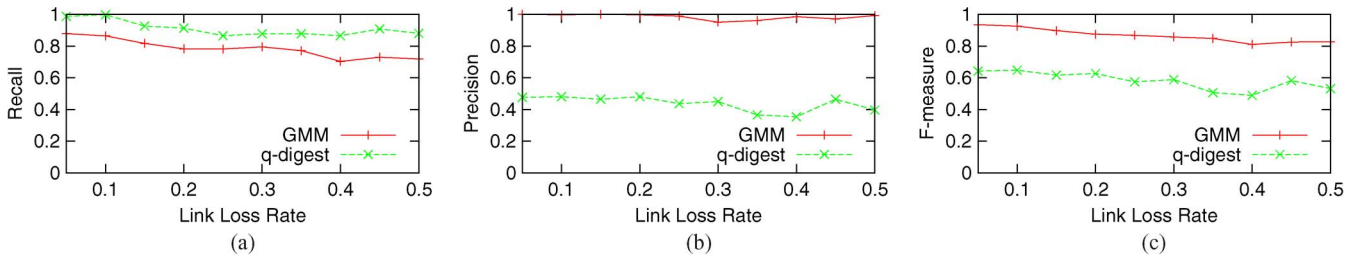

Fig. 3. Performance comparison of the algorithms with a variety of link loss rates.

the error rate of GMM still remains low as it takes the link loss rate into consideration. Another observation is that the q-digest algorithm has a relatively high error rate, even when the link failure rate is very low.

Fig. 2 shows the results for various kinds of queries that we mentioned in Section I: *What is the percentage of nodes whose readings are higher than* X? In fact, this query is equivalent to extracting the complementary cumulative distribution function (CCDF) $F(x) = 1 - Prob(X < x)$, given a threshold x. Even if there exists packet loss over the networks, due to the accurate approximation of statistical information in terms of probability function, using our proposed algorithm in this paper, the proposed algorithm is capable of providing a highly accurate answer to this query. For instance, in the case of $x = 170$, the percentage of nodes whose readings are higher than this threshold is close to zero. Using the q-digest algorithm, the estimated percentage could be close to 20%, which is far away from the exact result. Again, this is because the q-digest algorithm is based on the assumption of the uniform distribution of the data values. Mixture models, however, can be used for arbitrary distribution since, in reality, we can determine the base distribution function based on history information (i.e., training data).

We also quantify the efficacy of answering the query mentioned in Section I (*What is the confident interval with 90% confidence level*?) using standard metrics `precision`, `recall`, and $\mathtt{F-measure}$. We use CI to denote the exact confident interval and CI' to denote the estimated confident interval using varying algorithms. The `recall` denotes the probability of identifying the true confident interval, and `precision` is inversely related to the size of the estimated confident interval. $\mathtt{F-measure}$ (which is computed as the harmonic mean of recall and precision scores) is commonly used as a single metric to measure the effectiveness of the algorithms [26], i.e.,

$$\begin{aligned}
\mathtt{recall} &= Prob(x \in CI' | x \in CI) \\
&= |CI' \cap CI| / |CI| \\
\mathtt{precision} &= Prob(x \in CI | x \in CI') \\
&= |CI' \cap CI| / |CI'| \\
\mathtt{F-measure} &= \frac{2 * \mathtt{recall} * \mathtt{precision}}{\mathtt{recall} + \mathtt{precision}}.
\end{aligned}$$

Fig. 3 compares, with 90% confidence level, the q-digest algorithm and our proposed algorithm in terms of these three metrics. Fig. 3(a) shows that the probabilities of identifying the true confident interval are high using both algorithms.

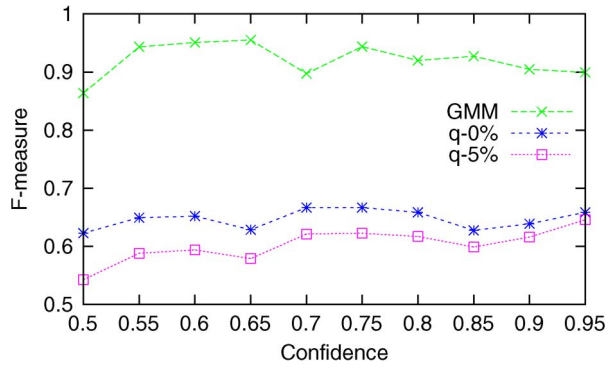

Fig. 4. $\mathtt{F-measure}$ with a variety of confidences.

However, in Fig. 3(b), we observe that the proposed algorithm in this paper is about two times more precise than the q-digest algorithm. The reason is that our algorithm more accurately captures the data distribution, whereas the distribution obtained using q-digest looks *flat*, as shown in Fig. 1(a), and hence, it demands a larger interval. Fig. 3(c) compares the $\mathtt{F-measure}$ for the two algorithms. GMM improves the answer efficacy by 40%–50%.

Fig. 4 compares the $\mathtt{F-measure}$ for the q-digest algorithm and our proposed algorithm with varying confidence level. We observe that, for all confidence levels, our algorithm is sufficiently efficient (all results are higher than 0.85) and outperforms q-digest (with the $\mathtt{F-measure}$ values of about 0.55–0.65). The improved performance is attributed to the fact that GMM more accurately captures the data distribution.

2) Impact of Quasi-UDG Model: We also conduct experiments under a different network model: the Quasi-UDG model [4] to the network we used. The Q-UDG model is characterized by a simple parameter α. That is, the packet loss rate is zero between two nodes if the distance between the two nodes is smaller than $(1 - \alpha)$ times the radio range; a link does not exist between two nodes when the distance between two nodes is greater than $(1 + \alpha)$ times the radio range, and the packet loss rate at a link between two nodes is promotional to the distance between nodes: $(d/r - 1 + \alpha)/(2\alpha)$, where d is the distant between two nodes and r represents the radio range. The statistical information extracted by our algorithm is shown in Fig. 5. We can see that, in spite of varying α values, our algorithm captures the statistical information well.

3) Statistical-Information-Assisted Top-k Extraction: We mentioned in Section IV-C that our extracted statistical information, based on multiround queries, can be used for top-k extraction. In this experiment, we implemented a

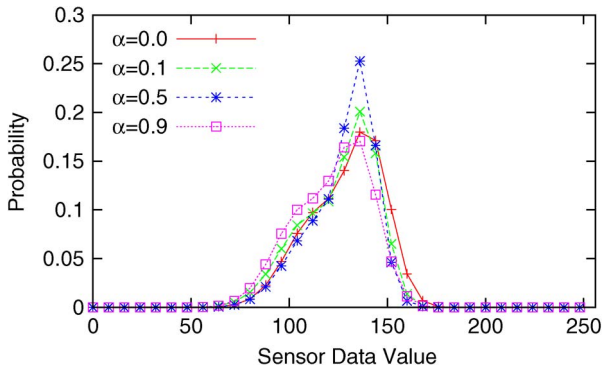

Fig. 5. Approximate histograms on the Quasi-UDG model.

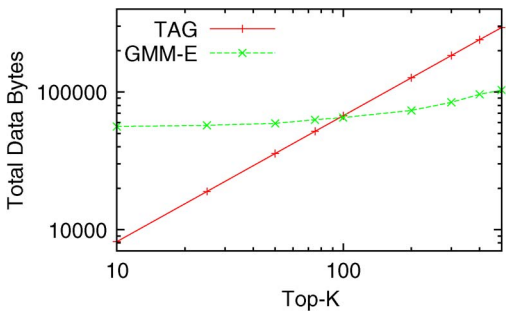

Fig. 6. Communication cost of top-k extraction with different schemes.

statistical-information-assisted top-k extraction scheme. After obtaining the first-round statistical information (i.e., the data distribution), we have knowledge of the CCDF of the data set, which is denoted by $P(x)$, as shown in Fig. 2. As such, we can set a threshold value x such that, for any value of x, if $x \geq \mathsf{x}$, $P(x) > n/k$. In other words, each node only forwards the reading when it is larger than x. It is worth noting that the number of returned top-k results is possibly less than k due to the approximation of this statistical information extraction. To address this problem, we consider the search process a sequence of Bernoulli trials, similar to [10] and [14], since $P(\mathsf{x})$ represents the probability of one data point being larger than x. Briefly speaking, using Pearson's confidence interval (more technical details can be found in [10] and [14]), the expected number of returned top-k results is at least k with a probability of 95% when we strive to retrieve top-k' ($k' = k + 2 + 2\sqrt{k+1}$), i.e., for a top-100 extraction, we strive to retrieve the 122 largest values of sensory data. By doing so, we found that, in our experiments, the number of returned top-k results is always at least k.

To compare with the statistical-information-assisted top-k extraction scheme (which is denoted by "GMM-E" in Fig. 6), we also implement a naive data-aggregation scheme based on TAG [16]. In this scheme, each node, after receiving the top-k candidates from all its children, aggregates these candidates (with its own readings) and then sends the k largest values to its parent. We denote this scheme as "TAG." Fig. 6 shows the comparison of the total communication cost using our proposed scheme and TAG scheme, where communication cost is in the unit cost of transmitting 1 B as in Fig. 1(b). First, when the k value is small (e.g., less than 100), we found that TAG

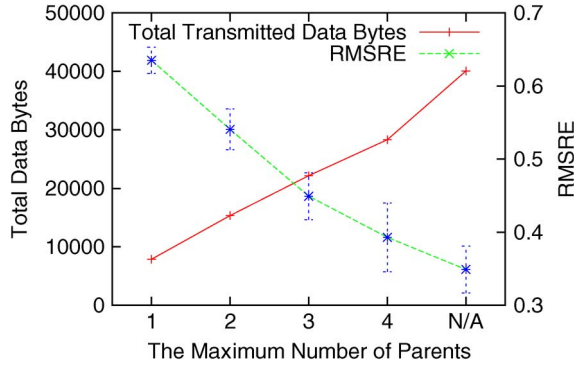

Fig. 7. Tradeoff between the accuracy and the message cost.

outperforms our proposed scheme. This is due to the fact that the first round of "GMM" for retrieving statistical information will introduce considerable communication cost, even if the k value is very small. Second, when the k value is large, the performance using our proposed scheme is better than using "TAG." This is because, for a large value of k, TAG performs almost the same as the naive centralized scheme, which asks all nodes to forward their readings. Overall, we conclude that our proposed scheme can efficiently facilitate top-k extraction in the case of a large value of k.

4) Effectiveness of Multipath Scheme: It is worth noting that, to achieve high accuracy, our algorithm utilizes the multipath routing scheme. This could be more expensive compared with a single-path routing in terms of energy efficiency as it introduces additional redundancies. To evaluate the tradeoff between accuracy and message cost, we purposely limit the number of parents of individual nodes in Fig. 7 during the routing path construction. In the case where each node identifies only one parent, our algorithm utilizes the single-path routing scheme. "N/A" in Fig. 7 represents the case where the number of parents is unlimited, i.e., our algorithm implementation described in Section V-A. Here, to evaluate the accuracy of our GMM-based approximation to the original data distribution, we use *root mean square relative error*: $RMSRE = \sqrt{1/m \cdot \sum_{i=1}^{m} E_i^2}$, where E_i is the relative error at each bucket. We take the average by running the simulation ten times and show the standard deviation of the $RMSRE$ values in Fig. 7. (We do not show the standard deviation of the massage cost as it is trivial in our experiments.) We can see that, with more and more parents being selected in the multipath-routing scheme, the message cost is increased, and the accuracy is decreased at the same time. Compared with the single-path routing scheme, the multipath-routing scheme used in our algorithm can achieve about 55% improvement in terms of accuracy while resulting in three times more cost.

In addition to the tradeoff between the accuracy and the message cost, we turn to evaluating how the multipath scheme affects the network lifetime, which is defined as the lifetime of the first sensor node that runs out of its power [25], [28]. Similar to [28], we set the energy consumption for receiving to be 50 nJ/b. For simplicity, let the energy consumption for sending packets be 100 nJ/b. We assume that the sensor nodes continuously perform the operations of sensoring, aggregation, and reporting at each time unit. The initial energy of each node

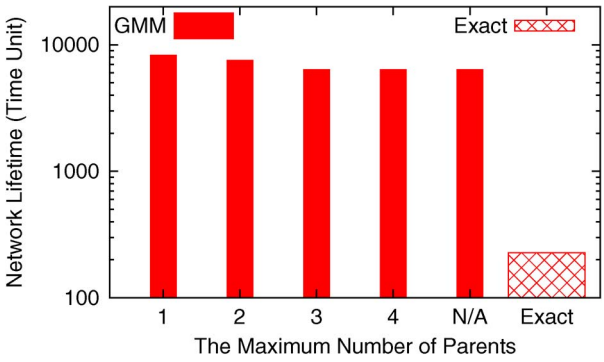

Fig. 8. Network lifetime comparison.

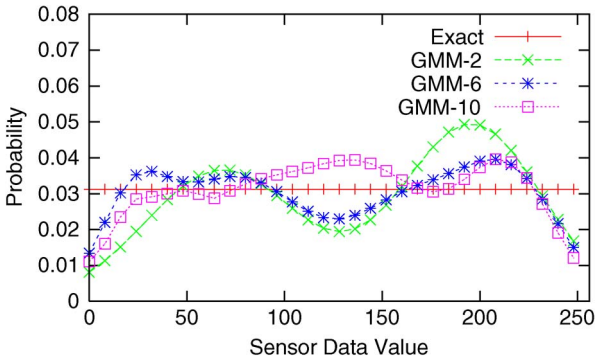

Fig. 9. Sensitivity to unknown distributions.

is set to be 100 J. First, as shown in Fig. 8, our data-aggregation-based algorithm achieves a much longer (about two orders of magnitude) network lifetime, compared with the centralized scheme, which is denoted by "Exact." The reason is that, in the centralized scheme, those nodes close to the sink are prone to running out of energy as they have to forward all the messages from their predecessors. Second, the multipath scheme, while leading to the error that is 50% lower than that of the single-path scheme, results in a comparable network lifetime.

5) Sensitivity to Unknown Distributions: One question remains to be answered. For an arbitrary data distribution, a low-dimension mixture model may not be accurate enough. This is true, particularly for some extreme distributions, e.g., uniform distribution, where no value is *typical*. However, the mixture model can approximate any distribution if the number of parameters is large enough. To demonstrate this, we run simulations using different GMMs, including 2-GMM, 6-GMM, and 10-GMM. The results are shown in Fig. 9. We observed that, with more parameters, the mixture model approximation can approach the exact data distribution better. We conclude that, to avoid a high number of mixture components in our mixture models, having the training phase understand the data pattern is necessary.

6) Sensitivity to Iterations of the EM Algorithm: As mentioned in Section IV-A, we limit the number of maximum iterations by a constant. In this part, we conduct experiments to evaluate the sensitivity of our algorithm to the number of iterations when the EM algorithm is performed. Fig. 10 shows the approximation error based on our proposed algorithm in terms of the root mean square relative error. It can be found that,

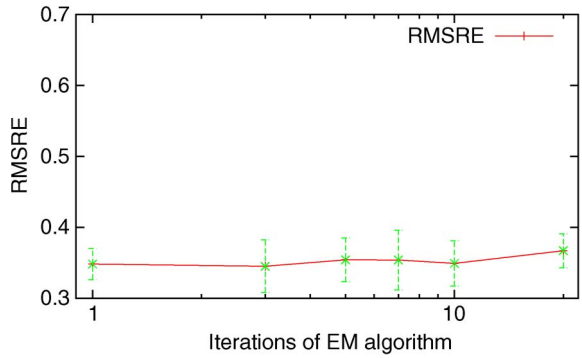

Fig. 10. Sensitivity to iterations of the EM algorithm.

often, the increased number of iterations will not considerably reduce the error. That is, our step of limiting the number of the maximum iterations is reasonable.

VI. Conclusion and Future Work

We have presented novel techniques for in-network aggregation in sensor networks based on statistical information extraction. The techniques are scalable: They propagate statistical information rather than the individual values in the network, and hence, the network communication cost, even in large-scale networks is reduced. The proposed scheme exploits an unbiased loss-tolerant multipath routing for data aggregation. It strives to extract the statistical information of the original data distribution but preserve the accuracy of estimation and avoid the loss of valuable statistical information. It is the first study on using mixture models to approximate sensory data distributions. We have demonstrated that the parameter-based technique is a viable approach of providing more accurate results. The proposed aggregation techniques have many desirable properties, such as highly improved accuracy, bounded message overhead, and its robustness against link failures. Simulation-based performance evaluation demonstrates that they outperform previous approximation algorithms in most configurations.

There are several directions in the future. First, we are planning to design a more efficient algorithm by using prediction to exploit the temporal correlation of sensory data. Second, while we proposed a heuristic to control the growth of mixture model when the data distribution is unknown, more general solutions, such as unsupervised methods, will be studied. Third, we are planning to use some geometry algorithms [12], [13] to facilitate our information-extraction algorithm.

References

[1] Intel lab data. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
[2] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Autom. Control*, vol. AC-17, no. 4, pp. 439–448, Aug. 1972.
[3] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
[4] J. Bruck, J. Gao, and A. Jiang, "Map: Medial axis based geometric routing in sensor networks," *Wirel. Netw.*, vol. 13, no. 6, pp. 835–853, Dec. 2007.
[5] R. Caceres, N. Duffield, J. Horowitz, D. Towsley, and T. Bu, "Multicast-based inference of network-internal characteristics: Accuracy of packet loss estimation," in *Proc. IEEE INFOCOM*, 1999, pp. 371–379.

[6] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. ACM MOBICOM*, 2003, pp. 134–146.
[7] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. Conf. VLDB*, 2004, pp. 588–599.
[8] A. FaraDian, J. Gehrke, and P. Bonnet, "GADT: A probability space ADT for representing and querying the physical world," in *Proc. IEEE ICDE*, 2002, pp. 201–211.
[9] S. Guo, T. He, M. F. Mokbel, J. A. Stankovic, and T. F. Abdelzaher, "On accurate and efficient statistical counting in sensor-based surveillance systems," in *Proc. IEEE MASS*, 2008, pp. 24–35.
[10] H. Jiang and S. Jin, "Exploiting dynamic querying like flooding protocols in unstructured peer-to-peer networks," in *Proc. IEEE ICNP*, 2005, pp. 122–131.
[11] H. Jiang and S. Jin, "Efficient extraction of statistical information with robust aggregation in sensor networks," in *Proc. IEEE ICDCS*, 2006.
[12] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, Y. Wu, and W. Liu, "Case: Connectivity-based skeleton extraction in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009, pp. 2916–2920.
[13] H. Jiang, W. Liu, D. Wang, C. Tian, X. Bai, X. Liu, Y. Wu, and W. Liu, "Connectivity-based skeleton extraction in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 710–721, May 2010.
[14] S. Jin and H. Jiang, "Novel approaches to efficient flooding search in peer-to-peer networks," *Comput. Netw.*, vol. 51, no. 10, pp. 2818–2832, Jul. 2007.
[15] Y. Kotidis, "Snapshot queries: Towards data-centric sensor networks," in *Proc. IEEE ICDE*, 2005, pp. 131–142.
[16] S. Madden and M. J. Franklin, "Fjording the stream: An architecture for queries over streaming sensor data," in *Proc. IEEE ICDE*, 2002, p. 555.
[17] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: A tiny agregation service for adhoc sensor networks," in *Proc. OSDI*, 2002, pp. 131–146.
[18] R. A. Olson, R. L. Gustavson, R. J. Wangler, and R. E. McConnell, "Active-infrared overhead vehicle sensor," *IEEE Trans. Veh. Technol.*, vol. 43, no. 1, pp. 79–85, Feb. 1994.
[19] Y. Pawitan, *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford, U.K.: Clarendon, 2001.
[20] E. M. Royer, P. M. Melliar-Smitht, and L. E. Mosert, "An analysis of the optimum node density for ad hoc mobile networks," in *Proc. IEEE Int. Conf. Commun.*, 2001, pp. 857–861.
[21] J. Schiff, D. Antonelli, A. G. Dimakis, D. Chu, and M. J. Wainwright, "Robust messagepassing for statistical inference in sensor networks," in *Proc. IPSN*, 2007, pp. 109–118.
[22] B. Sheng, Q. Li, W. Mao, and W. Jin, "Outlier detection in sensor networks," in *Proc. ACM MOBIHOC*, 2007, pp. 219–228.
[23] N. Shrivastava, C. Buragohain, and D. Agrawal, "Medians and beyond: New aggregation techniques for sensor networks," in *Proc. ACM Conf. Embedded Netw. SenSys*, 2004, pp. 239–249.
[24] A. Silberstein, K. Munagala, and J. Yang, "Energy-efficient monitoring of extreme values in sensor networks," in *Proc. ACM SIGMOD*, 2006, pp. 169–180.
[25] X. Tang and J. Xu, "Extending network lifetime for precision-constraned data aggregation in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.
[26] C. J. Van-Rijsbergen, *Information Retrieval*, 2nd ed. London, U.K.: Butterworth, 1979.
[27] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proc. IEEE Int. Conf. PerCom*, 2005, pp. 324–328.
[28] M. Wu, J. Xu, X. Tang, and W. Lee, "Top-k monitoring in wireless sensor networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 7, pp. 962–976, Jul. 2007.
[29] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Record*, vol. 31, no. 3, pp. 9–18, 2002.
[30] Y. Yao and J. Gehrke, "Query processing for sensor networks," in *Proc. CIDR*, 2003.

Hongbo Jiang (M'08) received the B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from Case Western Reserve University, Cleveland, OH, in 2008.

He then joined the faculty of Huazhong University of Science and Technology as an Associate Professor. His research interests include computer networking, particularly algorithms and architectures for high-performance networks and wireless networks.

Shudong Jin (M'00) received the B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from Boston University, Boston, MA.

He is currently an Assistant Professor of computer science with Case Western Reserve University, Cleveland, OH. His research interests include network protocols and algorithms, network modeling and performance evaluation, multimedia streaming, and pervasive computing.

Chonggang Wang (SM'09) received the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China.

He is currently with NEC Laboratories America, Princeton, NJ. His research interests include hybrid optical and wireless networks, sensor networks and applications, cognitive radio networks, ubiquitous and distributed computing, and data centers. He is an Editor for the *ACM/Springer Journal of Wireless Networks*.

Dr. Wang was the recipient of the National Award for Science and Technology Progress in Telecommunications. He is an Associate Technical Editor for *IEEE Communications Magazine*.