Signature-File-Based Approach for Query Answering Over Wireless Sensor Networks

Mo Li, *Member, IEEE*, Lei Chen, *Member, IEEE*, Jizhong Zhao, *Member, IEEE*, Qian Zhang, *Senior Member, IEEE*, and Yunhao Liu, *Senior Member, IEEE*

***Abstract*—Wireless sensor networks (WSNs) are widely used in many application fields. Because sensor nodes are generally battery powered, to prolong network lifetime, energy conservation becomes a major concern in answering queries over sensor networks. In addition, a robust and fault-tolerant data-collection method is highly desirable against a lossy network with low-quality wireless communication links and unreliable sensor nodes. We propose a signature-file-based approach to approximately answer queries over WSNs. By combining the duplicate-insensitive structure of signature files and the redundant multipath routing approach, we create a robust in-network aggregation scheme, which can answer both aggregative and range queries with high accuracy while significantly reducing the cost of message transmissions. Simulations have been conducted to evaluate the performance of this approach under various network conditions. Compared with previous solutions, our signature-file-based approach achieves the highest accuracy under reasonable energy cost.**

***Index Terms*—Aggregation, range query, sensor networks, signature file.**

I. Introduction

DUE TO RECENT advances in computing and communication technologies, networked sensors are available to measure real-world phenomena. A large number of sensors, which are densely deployed in a specific region, form a wireless sensor network (WSN). These sensor networks are designed for monitoring an environment and supporting various queries. A concrete example of this type of application is the environment monitoring in underground working spaces (e.g., coal mine tunnels that are 3000 m long and tens of meters wide), which is a crucial task to preserve safe working conditions. A lot of environmental factors need to be monitored, including gas, water, dust, and so on. A precise environment overview requires a high sampling density, which results in a large number of sensing devices. Current environment monitoring is typically manually conducted and in a sparse way, due to the lack of corresponding techniques for constructing a large-scale sensing system, which conforms to the practical underground conditions and provides dense sensing points.

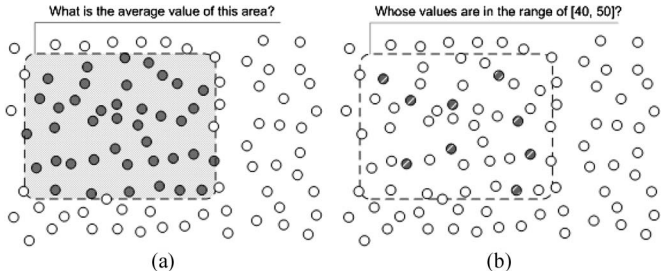


Fig. 1. Two types of queries over a WSN. (a) Aggregative queries. (b) Range queries.

In many of our coal mine monitoring system scenarios, we mainly need two types of information, as illustrated in Fig. 1.

With respect to the first type [Fig. 1(a)], a query is used to collect the average value from sensors within a rectangular area (zone) specified by the two coordinates (10, 10) and (200, 200) every 50 s for 60 min [16], an example of which is described as follows:

$$\text{Type} = \text{Avg (temperature)}$$
$$\text{Interval} = 50 \text{ s}$$
$$\text{Duration} = 60 \text{ min}$$
$$\text{Zone} = [10, 10, 200, 200].$$

The above query is called an *aggregative* query. The other examples of aggregative queries are *Min*, *Max*, and *Count*. In addition to aggregative queries, there are nonaggregative queries. A typical example of a range query is shown in Fig. 1(b), which is described as follows:

$$\text{Type} = \text{Range } (40 < \text{temperature} < 50)$$
$$\text{Interval} = 50 \text{ s}$$
$$\text{Duration} = 60 \text{ min}$$
$$\text{Zone} = [10, 10, 200, 200].$$

The above query is used to collect the IDs of sensors whose sensing temperatures are within a specified value range.

The main challenge in answering these two types of queries over WSNs is how to save sensor energy because they are

Manuscript received December 5, 2006; revised October 3, 2007 and October 9, 2007. This work was supported in part by the Hong Kong Research Grants Council under Grant HKUST6169/07E, by the National Basic Research Program of China (973 Program) under Grant 2006CB303000, by the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z180, and by the Natural Science Foundation of China Key Project under Grant 60533110. The review of this paper was coordinated by Prof. B. Li.

The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, and also with the Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: limo@cse.ust.hk; leichen@cse.ust.hk; zjz@mail.xjtu.edu.cn; qianzh@cse.ust.hk; liu@cse.ust.hk).

Digital Object Identifier 10.1109/TVT.2007.912340

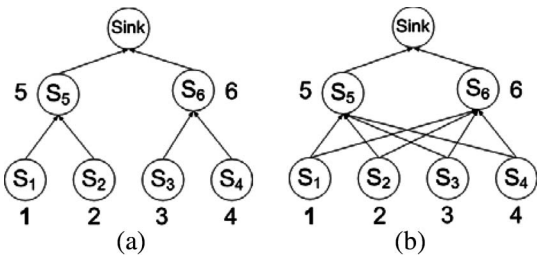


Fig. 2. Tree (TAG) versus multipath. (a) Tree (TAG). (b) Multipath.

battery powered, and changing batteries is often very difficult [17]. In addition, because packet loss often happens in WSNs due to the low quality of transmission (up to a 30% loss rate is common), it is essential to make the results robust to possible packet loss.

A simple approach to answering the aforementioned two types of queries is by asking all the sensors to send their sensing values back to the sink. However, such an approach leads to a lot of data transmissions, which will quickly deplete sensor power.

Many approaches have been proposed to address the above issue. Among these approaches, *in-network aggregation* [17] is used to reduce the number of transmissions from the sensors to the sink for aggregative queries. As shown in Fig. 2(a), a routing tree is built to answer aggregative queries. At each parent node (e.g., S_5), it aggregates (e.g., computes *sum* and *count*) the data that it receives from the child sensors and its own sensing value and sends the partial aggregative result (partial sum and count) to the sink. Thus, the total number of transmissions is reduced. However, the packet loss may render the answer inaccurate. To address this problem and make the result robust to the packet loss, a multipath routing scheme is used, as shown in Fig. 2(b). Sensors are divided into levels according to their hop count from the sink. The smaller the hop count, the higher the level will be. Each sensor sends its partial aggregation data to multiple sensors in the next upper level (one hop count less to the sink). Thus, the probability of no data packet arriving at the upper level is reduced, which makes the results robust to packet loss. However, this scheme introduces another problem, which is the possibility of the same data being counted more than once (*duplicate sensitive*). Recent works [3], [18] use approximated aggregation for answering aggregative queries. Duplicate-insensitive structures like sketch are used to carry the approximated information. Their approach, however, only works on aggregative queries, which is thus infeasible for range queries. There is also no control over the error rate of aggregation results due to the random hash function properties that sketch uses. Furthermore, the accuracy of the sketch approach is based on a large number of nodes located in the query zone. If the number of nodes is small, the possible excessive bias on the sketch insert may lead to very low accuracy in the results.

In this paper, we address the above problems by encoding the collected sensor readings using signature files, reducing the cost of transmitting data and removing possible duplicates by a simple bitwise "OR" operation. To reduce the cost of transmitting long signature files, we introduce an adaptive compression scheme to shorten the length of each signature file. The goals of this paper are summarized as follows:

1) to propose a signature-file-based approach that would fully utilize the power of in-network aggregation and answer both types of queries;
2) to utilize the properties of signature files and multipath routing schemes to make the results robust to possible packet loss and duplicate insensitive, and more importantly, to allow our approach to control the error rate, no matter what size the networks are;
3) to develop dynamic bucket allocation schemes to adapt various data distributions that would improve the accuracy of aggregative queries. We also propose an adaptive compression scheme that would reduce the energy cost of transmitting signature files.

The rest of the paper has been organized as follows. We briefly introduce the signature files and review the related work in Section II. We introduce our signature-file-based approach to approximately answer queries in Section III. The simulation studies of our proposed approach are presented in Section IV. Finally, we conclude this paper in Section V.

II. Background and Preliminaries

A. Related Work

A number of research results have been published on energy-efficient query processing for sensor networks both on the network and database domains. On the network domain, researchers propose energy-efficient routing protocols for sensor networks, such as directed diffusion [12], low-energy adaptive cluster hierarchy (LEACH) [10], and sensor protocols for information via negotiation (SPIN) [9]. Directed diffusion is a data-centric protocol in which the sink periodically floods queries into the network, and routing trees are constructed by selecting low-delay paths. The data transmission rate is controlled by the rate of data generation from each node and the available bandwidth between nodes. LEACH is a scalable adaptive clustering protocol in which nodes are organized into clusters. The system lifetime is extended by randomly choosing the cluster heads, thereby fairly spreading energy consumption over the entire network. SPIN is proposed to address the deficiency of flooding, which uses negotiation to ensure that only useful information is transferred, and employs resource adaptation to control the message passing or processing based on the current available energy. All the above approaches suffer from lossy network conditions with low-quality wireless communication links and unreliable sensor nodes.

In the database domain, Madden *et al.* [17] proposed the tiny aggregation service (TAG) to reduce the energy cost of processing aggregation queries by in-network aggregative techniques. In TAG, a routing tree is built by using flooding or wedge flooding. TAG adaptively adjusts the sampling rate of sensors based on the query constraints and energy of sensors. By reducing sampling rates, energy spent on sensing is saved. Many other works have also been proposed based on TAG to

address energy-efficient query processing, utilizing the temporal coherence among data collected from the same sensor to reduce the number of transmissions.

The techniques discussed above normally assume that sensing data are collected through routing trees. However, the routing tree scheme is not robust to communication loss and "good" links are hardly expected. Spatial correlations are also investigated for data aggregation. Deshpande *et al.* [5] used a data model to encode the relationship between sensing values at different sensors. Instead of getting values from the sensors, the data model can be used to predict the data. Gupta *et al.* [8] proposed an energy-efficient method to build a correlation graph of the sensor network so that only a small subset of sensor nodes are needed to reconstruct the data for the entire network. However, these approaches heavily depend on the correlations of different values. Recently, Considine *et al.* [3] and Nath *et al.* [18] concurrently proposed approximated approaches to answer aggregative queries using sketch. In their approach, for approximately answering aggregative queries in the multipath routing scheme, the duplicate-insensitive sketch is used to carry the *SUM* and *COUNT* information. Their approach achieves an approximate result with much reduced communication overhead. However, it only works on aggregative queries, and there is no control on the error rate of aggregation results due to the property of random hash functions that sketch uses. The accuracy of the sketch approach is based on the large number of nodes located in the query zone. If the number of nodes is small, the possible excessive bias on the sketch inserting may result in very low accuracy.

B. Signature Files

Signature files were first introduced as an indexing method for text retrieval [6]. A fixed-width signature (bitstring) m bits (length) is assigned to represent each key word (or distinct value) with w bits (weight) being set to 1. The m bits are set with a number of hashing functions. One advantage of signature files is being duplicate insensitive after superimposed coding. Furthermore, the overall false drop (alarm rate) can be controlled by carefully setting w and m. Here, we give a detailed example of how a signature file works on the multiple-path topology shown in Fig. 2(b). Assume that we tried to answer the first example aggregative query in Section I. As shown in Fig. 2(b), the distinct sensing values are 1, 2, 3, 4, 5, and 6, and they are encoded into the following six signature files, respectively, by a set of hash functions: 1—001011010101; 2—010010110001; 3—011001010101; 4—001010110100; 5—011000110001; and 6—011011100000.

Then, instead of sending the real values, we send the signature files to the upper level node and carry out the partial aggregation by superimposing (ORing) with the received signature files. All the duplications can be removed because of the "OR" operation. For example, if a sensor node receives two "5s" from two different paths, after the "OR" operation on two signature files, we will only get one signature file of "5," and the bits that have been assigned to 1 will be only counted once. Finally, at the sink, we will get the superimposed bitstring (result signature): 011011110101.

At the sink, we compare (ANDed) the signature of each distinct value with the result signature to check whether a distinct value exists in the final result. If it does, the value will be used to compute the final aggregation result. For example, if the signature of 1 ("001011010101") matches the result signature, then it will be used to compute the aggregation result.

However, due to the "OR" operation, a value that is not sensed by the sensors within the sensor network may be identified as "existence," which is named as *false drop*. For example, if the signature of 7 is "001001110001," it also matches the result signature, but 7 does not appear in the network.

Fortunately, this false drop rate can be minimized by carefully setting the signature weight (w) and length (m). The equation of computing the w and m are proved by Davis and Kamamohanarao [4]. Assuming that each bit of a signature has the equal probability of being set to 1, the probability that y bits is set to 1 in a signature superimposed from x signatures of weight w and m is $P(x,y) = m[1-(1-w/m)^y]^x$.

In fact, the false drop rate of a signature file P^f is $P(N;w)$, where N is the number of distinct values. To minimize P^f, we can get $(1-w/m)^N = 0.5$ and $P^f = (0.5)^w$. Thus, given P^f, the w and m can be set as follows:

$$w = \left(\frac{1}{\ln 2}\right) \times \ln\left(\frac{1}{P^f}\right) \tag{1}$$

$$m = \left(\frac{1}{\ln 2}\right)^2 \times N \times \ln\left(\frac{1}{P^f}\right). \tag{2}$$

Therefore, using signature files, we can control the error (false drop) rate by setting m and w.

III. Approximately Answer Queries Over Sensor Networks

We assume that each sensor has a unique ID and knows its position in this paper. Many proposals have been made to address the sensor localization problem [2], [7], [13], which is not the focus of this paper. Without loss of generality, we assume that the sensing values range from $V_{\min}$ and $V_{\max}$. We first explain the architecture of our approach. According to Section II-B, the length of signature files may be quite long to reduce the false drop rate. We then present an adaptive compression method to compress encoded signature files to reduce transmission cost. After that, we address how to use signature files to answer aggregative queries. Finally, we show how to answer range queries with signature files.

A. System Architecture

We adopt the multipath topology for data transmission in a sensor network. The multipath topology is created in the sensor network according to the node hops to the sink. Nodes are divided into different levels according to their hop count. As Fig. 3 shows, the hop count of each node from the sink indicates its level. This could be achieved in the system initialization phase through advertising with the node hop count from the sink [17]. In the data-collection (query reply) phase, each node reports its aggregated result by local broadcasting, and all its

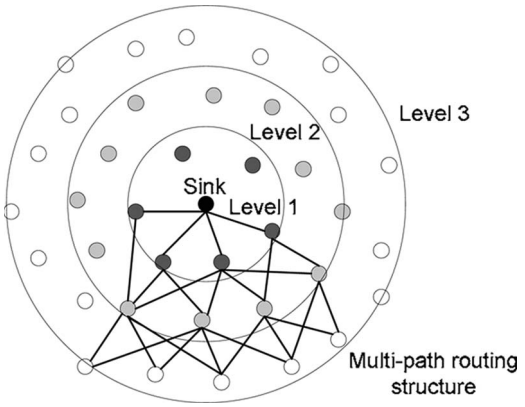


Fig. 3. System architecture.

neighbors, which are one level lower, captures the report and aggregates it. By following the above steps, a multipath routing structure is built, through which robust data collection could be achieved.

In Fig. 3, nodes at different levels aggregate data in different time epochs during the data-collection phase, as in the TAG approach [17]. Compared to higher level nodes (far from the sink), lower level nodes (closer to the sink) need to wait longer to obtain the reports from the child nodes and locally perform the aggregation. The communication epoch for each node at different levels is assigned during the system initialization phase. In each data-collection phase, nodes will receive, aggregate, and send the reports in their communication epochs. They then stay idle the rest of time to save energy.

B. Adaptive Signature File Compression

As indicated in (1) and (2), to make signature files more accurate (to lessen the error rate), the length of the signature file is usually very long. Directly transmitting raw signature files may lead to high communication cost, which is exactly the cost we want to avoid. By observing the generated signature files, we have found that even when they are very long, the number of "1s" are few. In addition, these "1s" sparsely locate in the bit string and are separated by many continuous "0s." An example of an original signature file of a distinct value is

00000000000000000001000000000000000000000000100000
00000001.

To compress the above example signature file, we can use a modified run-length encoding (RLE). The original RLE uses block representation for both "1" and "0" appearing in the string. To shorten the string representation, a k-bit block is used to represent the number continuously appearing "1" or "0." We modify the original RLE according to the feature of the signature file, which generally has long substrings of continuous "0s" but discrete "1s." For compressing the signature files, we merely substitute the continuous "0s" with the block representations (a bit "0" $+k$-bit representation of N). When using the binary value of the block to represent the length of continuous "0s," with a predefined value of k, the largest number N of continuous "0s" that could be represented in a single block is 2^k. Therefore, in our compression algorithm, we first divide the continuous "0s" into several blocks, and each block contains N "0s" except for the last one, which is normally less than N. Then, we substitute each block with our block representation such as RLE does. For example, given the above bit string, if we choose k to be 3 and N to be 8, it will first be divided into

00000000 00000000 000 1 00000000 00000000 00000000 1
00000000 0000 1.

After substituting all the blocks with the 4-b representations, we get the compressed representation[1]

0000 0000 0011 1 0000 0000 0000 1 0000 0100 1.

If we choose k to be 4, the compressed representation is

00000 00011 1 00000 01000 1 01100 1.

We can encode the information about k using 4 b before the compressed data so that the receiver node could decompress the compressed signature files.

In fact, the block representation size k determines the compression ratio. Thus, we can use an adaptive compression method to determine the optimized k according to the percentage of "1s" in the signature files. Specifically, for low-level sensors (far away from the sink), if the percentage of "1s" in their encoded signature files is low, a large k value is chosen to compress as many "0s" as possible. For high-level sensors (near the sink), due to the aggregated results received from low-level sensors, the encoded signature files of the high-level sensors have a higher percentage of "1s." In this case, a small k value is selected to reduce the possibility that segmented blocks have both "0" and "1." Therefore, by carefully determining an appropriate k according to the number of "1s" in the signature files, we gain a significant reduction in the transmission data length, which is tested through our simulations.

C. Answer Aggregative Queries

To answer aggregative queries, we need to encode distinct sensing values into signature files. The number of the distinct value N can be easily computed with a user- or application-specified precision ε given by

$$N = \left\lceil \frac{(V_{\max} - V_{\min})}{\varepsilon} + 0.5 \right\rceil. \tag{3}$$

In fact, we divide the range into N buckets and represent each bucket with its mean value. When the sink floods the aggregative queries, it also sends the information with respect to bucket size and hash functions. Each sensor will determine whether it needs to send its sensing data by checking its position against the query specification. If it is required to send the data, its sensing value is compared with the bucket information, and

[1]Note that we use k bits of 0 to represent N "0s" because the case of an appearing 0 "0" does not exist. In this example, we use 000 to represent 8 "0."

a bucket, which is closest to its value, is selected. Then, the selected bucket value is encoded into signature files using the hashing functions.

However, the above approach divides the range into buckets with the same size, which may introduce large errors when data distribution does not follow the uniform distribution. For example, the oxygen density data that we collected from the readings in the coal mining project described in Section I show that the data come from a Gaussian normal distribution. If we adopt the same size bucket for each data value in the data space, the probability that the values that will appear in some bucket ranges will be higher than that of the other ranges. Thus, for the buckets with a higher probability in which data will fall, multiple sensor readings may compete for the same value bucket. Once the different values from the different sensors share the same bucket, they are treated as a single value, which leads to inaccuracy in the final aggregation result.

To address this problem, we exploit the *dynamic bucket-allocation* method. In this bucket-allocation method, instead of segmenting the value space into N equal-sized buckets, we segment the value space into various-sized buckets, which conform to the data distribution curve. The boundary of the ith bucket is computed according to the following formula:

$$\int_{V_i}^{V_{(i+1)}} P(x)dx = \frac{1}{N} \int_{V_{\min}}^{V_{\max}} P(x)dx \tag{4}$$

where $P(x)$ is the normal probability distribution function, $1 \leq i \leq N$, $V_1 = V_{\min}$, and $V_{N+1} = V_{\max}$. With this type of segmentation, we assign equal probability to each bucket into which the data might fall.

To obtain the data distribution $P(x)$ of the sensor network, we can either rely on the estimation from historical data, such as the data we collected from the coal mining project [14], [15], or make an approximation by sending preliminary queries to collect sampling data and the data pattern. The initial estimation of data distribution $P(x)$ may be rough and less accurate. According to the approximated estimation, we use more buckets for a value range of higher probability and less buckets for a value range of lower probability, which forces the bucket allocation to match the data distribution. Once the bucket-allocation strategy closely reflects the real-data distribution, the consequent data aggregation based on these buckets becomes more accurate. Dynamic bucket allocation creates suitable bucket distribution according to the real-data distribution, thus improving the aggregation accuracy by reducing bucket competitions of different sensor readings. Based on this bucket allocation, we propose a data-migration method to further reduce the possibility of bucket competition. In the aggregation process, a higher level sensor node collects all the signature files from its children (lower level sensor nodes), and then, before it hashes its sensing value to some calculated bucket B, the node first checks whether bucket B has been occupied by some other node reading. If that is the case, this sensor tries to migrate (map) its sensing value to the nearby bucket, e.g., $B + 1$. If the bucket $B + 1$ is also occupied, the sensor node needs to sequentially check buckets $B - 1, B + 2, B - 2, \ldots$ until it finds an empty bucket. The probability of bucket competition exponentially drops as the migration process continues.

D. Answer Range Queries

To answer range queries, we assume that we know all the sensor IDs, and they can be represented by positive integers. As shown in Section I, a range query needs to collect at least a set of sensor IDs whose sensing values are within the query-specified value range. Because an ID is unique to a sensor and values of IDs are uniformly distributed, we can divide the value range of the IDs into N buckets, where each bucket corresponds to one unique ID. We encode the IDs as signature files from the sensors that are at the highest level (furthest from the sink). Superimposing is used on lower level sensor nodes to aggregate the signature files of the IDs. At the sink, we then check the existence of the IDs through an ANDed operation. Aside from being duplicate insensitive, another advantage of using signature files to answer range queries is that the message length is fixed. No matter how many sensors whose readings are within the query range there are, the message length will not change after the superimposing step. Compared to the simple approach that directly collects and transmits a list of IDs (*LIST* approach), whose message length increases along the path from the sensors to the sink, our signature-file-based approach significantly reduces message communication cost.

IV. Simulation

In this section, we evaluate the performance of our signature-file-based approach under large-scale sensor networks through simulation.

We create a randomized network topology, including 1000 sensor nodes as our basic simulation topology. We vary the network density by changing the average node degree from 10 to 40 (the number of neighboring nodes per sensor) to examine the performance under different degrees of network connectivity. The links between nodes are assigned a probabilistic delivery ratio. Our simulator models packet-level loss rates that range from 0% to 35%. We use a 48-B message length as what TinyDB [11] uses. In the simulation, the delivered information may exceed the 48-B limitation so that a sensor node may transmit multiple packets for one round of information delivery. At the time of delivery failures, no packet retransmission or failure recovery scheme is assumed. To further test the robustness and fault tolerance of the approaches, we introduce a random node failure rate, which ranges from 0% to 35%. Finally, we vary the network size from 200 to 2000 (the number of nodes in total) to investigate the scalability of the different approaches. For each simulation, we take 100 runs and report the average.

In the simulation, each node is assigned a level according to its hop count from the sink. Any neighbor of a lower level node is considered to be its candidate parent. Each node aggregates results from its children nodes with its own readings and forwards the result to one or multiple parents. We compare the following four different approaches in our simulation:

1) TAG [17]: an approach in which each sensor node sends its aggregated result to its parent;

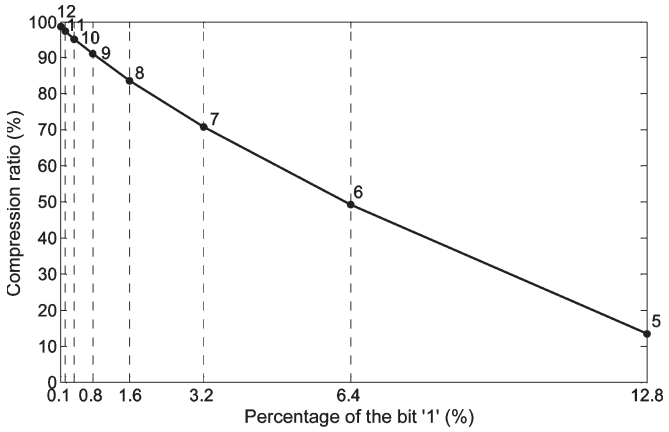


Fig. 4. Compression ratio versus the percentage of "1" bits.

2) LIST: a multipath approach in which each sensor node aggregates all received items in a list and removes duplicates. This item list is further forwarded to all parents;
3) SKETCH [3], [18]: also a multipath approach in which each sensor node aggregates the statistical data into sketches and forwards them to all the parents. The sink extracts the results from the final aggregated sketches. Because SKETCH cannot answer range queries, we only evaluate it for aggregative queries;
4) SIG: our signature-file-based multipath approach described in Section III.

Simulation 1—Effect of Block Representation Size k: As we discussed in Section III-B, the compression ratio of our compression method is affected by the number of bits "1" and the block representation size. In this simulation, we test the effect of block representation size k on the compression ratio. Indeed, the block representation size should be determined based on the number of bits "1" in the signature files. With the increase in the number of "1s" in the signature file, the block size should be reduced to avoid using more information to represent separated "0s" in each block. Fig. 4 plots the compression ratio under a different percentage of "1" bits in the signature files. The number above each data point denotes the optimal k value selected. Fig. 4 confirms our expectation that k will decrease with the increase in the percentage of "1" bits in the signature files to maintain an optimal compression ratio.

Simulation 2—Comparison of Traffic Overhead Among the Four Approaches for Answering Aggregative Queries: We compare the traffic overhead of the four approaches for answering aggregative queries. The sensing values for each node are selected uniformly random from [0, 10 000]. The traffic overhead is measured by the number of packets because in real WSN systems, in terms of power consumption, the traffic overhead is largely determined by the number of packets transmitted rather than the actual number of bytes during the communication [14]. As previously described, a TinyDB packet (up to 48 B) is assumed as the basic transmitting carrier. For the TAG approach, an aggregated data value contained in one packet is transmitted from each sensor node to its parent. The LIST approach aggregates (ID, value) items into a list; thus, its message length is 8 B long for a single item (4 B each). The SKETCH approach hires 20 sketches together to improve the estimation accuracy. The length of each sketch is set to 4 B due to $1.5 \log n$, as specified in [3], where n is the maximum possible number of aggregative results. In our SIG approach, a compressed signature file is included in the transmitted packet. The total number of transmitted packets of TAG, SIG, SKETCH, and LIST are 1000, 1886, 2000, and 7534, respectively. As we expected, the TAG approach achieves the lowest traffic because the tree topology communication strategy of TAG incurs no extra cost in reducing duplicates. The LIST approach's performance is the worst, and the SKETCH and SIG approaches lie in between TAG and LIST. However, as shown in later simulations, TAG achieves very low accuracy when introducing link loss.

Simulation 3—Comparison of Accuracy Among the Three Approaches in Answering Range Queries: Because SKETCH cannot answer range queries, in this simulation, we only compare the accuracy of the TAG, LIST, and SIG approaches in answering range queries under various network statuses. We use two metrics, namely *precision* and *recall*, which have been widely used in the information retrieval domain [1]. Assume T is the actual set of node IDs satisfying the range query, and Q is the actual returned set of IDs. The query precision $p = |T \cap Q|/|Q|$, where $|\bullet|$ gives the number of elements in the set, and p represents how many effective IDs have been collected among the returned results. A higher value of p indicates a precise collection approach. The query recall $r = |T \cap Q|/|T|$, where r tells us how many effective IDs are returned to the sink among the genuine satisfied nodes. A higher value of r represents an effective collection method. Therefore, to effectively answer range queries, high values of both metrics should be achieved. In fact, the LIST and TAG approaches always maintain the query precision $p = 100\%$ because these two approaches never aggregate the IDs of unsatisfied nodes in the collection phase. For our SIG approach, query precision p can be constrained by the false drop rate of the signature files. By tuning the signature file length m and weight w, we can achieve a bounded false drop rate. As a consequence, the query precision p can be raised to a high value (up to 99.9%) by paying little bearable traffic overhead. Because the query precisions of all the three approaches are high and similar, the query recall r dominates the overall accuracy. Thus, we run the simulations to test the query accuracy metric recall r. The results are reported in Fig. 5.

The results show that under various conditions, our SIG approach consistently achieves nearly the same performance as that of LIST, whereas TAG performs variously, particularly under tough conditions. Fig. 5(a) shows the effects of link losses on their performance. With the increase in link loss rate, the query recall of TAG rapidly drops due to the lack of redundancy. However, our SIG maintains a high recall as LIST. In the next simulation, we introduce random node failures into the network to test the performance when the network becomes instable. Similar performance results in Fig. 5(a) are shown in Fig. 5(b). Compared with Fig. 5(a), the recalls of all the three approaches diminish more rapidly because the node failures exert a larger effect on the query accuracy, although SIG outperforms TAG.

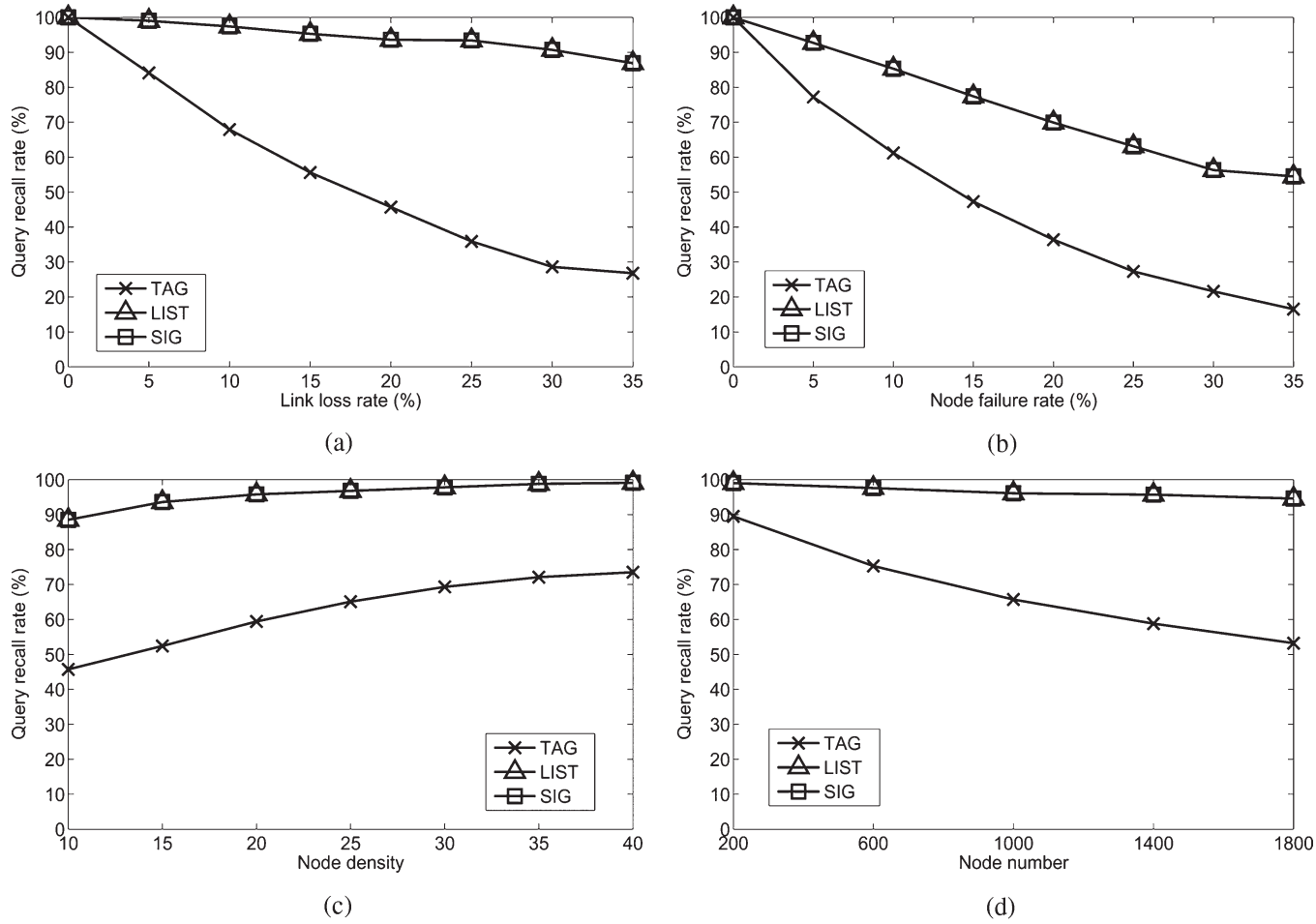


Fig. 5. Query recall.

Fig. 5(c) shows the performance of the three approaches over various node densities from 10 to 40 (neighbors per node). A higher node density supports better network connectivity and query accuracy. For various node densities, our SIG approach outperforms TAG more than 40 percentiles. Last, we scale the network and explore the scalability. The link loss rate is set to 10%. As shown in Fig. 5(d), with the increase in the number of sensor nodes, the recall of the TAG approach drops to nearly half of that achieved by the SIG and LIST approaches.

Simulation 4—Comparison of Accuracy Among the Four Approaches in Answering Aggregative Queries: In the last simulation, we compare the accuracy of all the four approaches in answering aggregative queries under various network statuses. Fig. 6 shows the performance in data aggregation. Although the sensing values are generated in a normal distribution, dynamic bucket-allocation technology could uniformly map these values into fixed-sized data buckets. We test the aggregative query *SUM* using the four query approaches, and the relative error is measured as $|(y - \hat{y})/\hat{y}|$, where y is the calculated value, and $\hat{y}$ is the real value. As shown in Fig. 6(a) and (b), again, SIG and LIST outperform the TAG approach, whereas SKETCH lies in between with a faster error increase against the rise of link loss rate and node failure rate. This shows the sensitivity of the TAG and SKETCH approaches against the link loss and node failure. Fig. 6(c) shows that with the increase in node density, the relative errors of all the four approaches are reduced. However, TAG still introduces more than four times the relative error that SIG and LIST approaches achieve. SKETCH shows similar sensitivity to the variation in node density as SIG and LIST do. However, a larger error rate is introduced when the node density is lower, which means SKETCH does not perform well for sparse sensor networks. Fig. 6(d) shows the error rates of the four approaches against the network size. TAG introduces a large relative error that is nearly proportional to the network size. As the SIG and LIST approaches maintain a slightly increasing relative error, SKETCH performs worse, particularly under a small network size.

To summarize, the simulations on range and aggregative queries show that our SIG approach can achieve the same accuracy as the LIST approach but only consumes one quarter of the network traffic. Furthermore, compared to TAG, the SIG approach is much more robust under various network status with less communication cost, and most importantly, compared to SKETCH, SIG can answer both types of queries and be applied to various sizes of WSNs. Therefore, we conclude that the SIG approach achieves higher accuracy with less communication cost among the four methods.

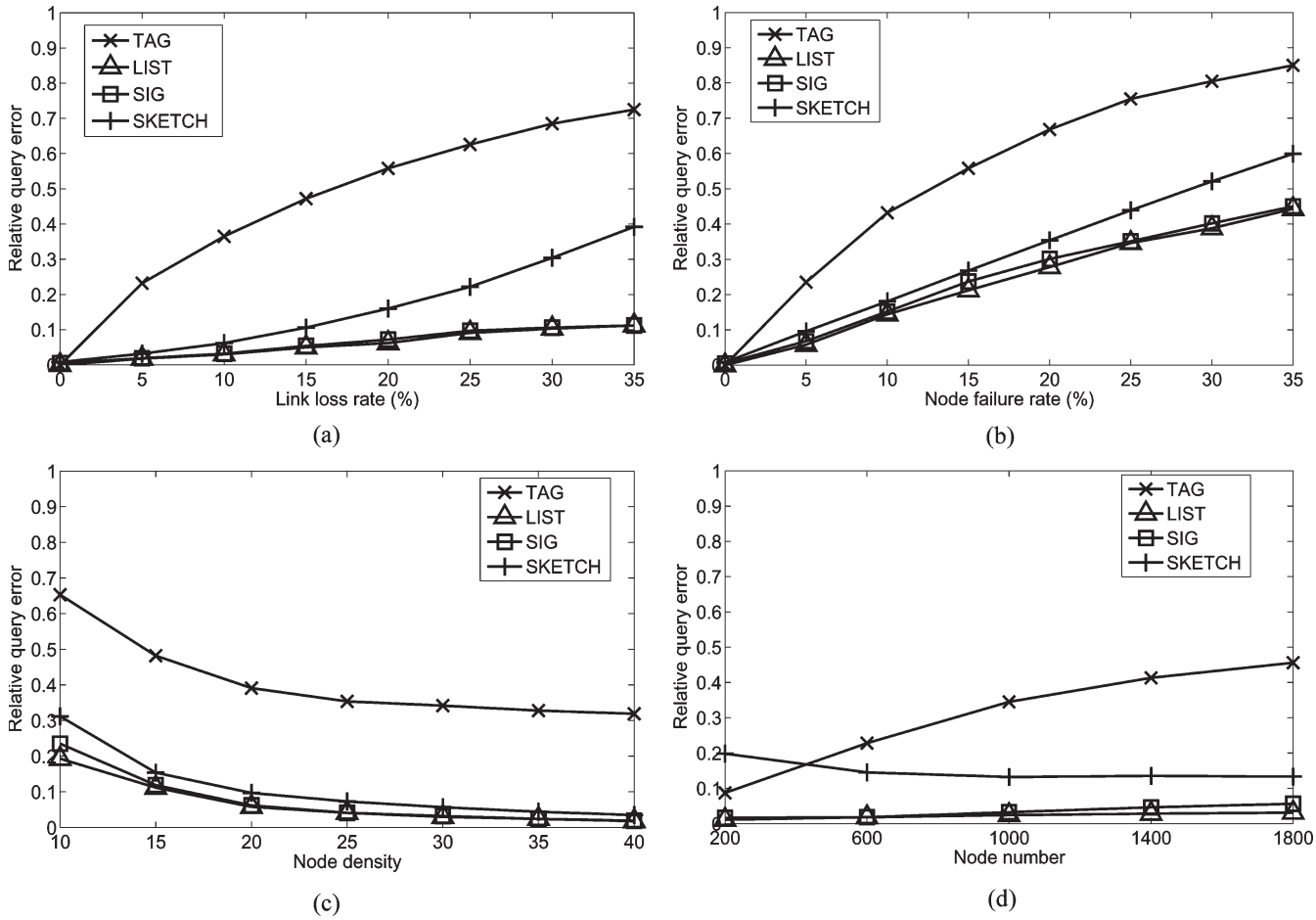


Fig. 6. Query accuracy.

V. Conclusion

In this paper, we have proposed a signature-file-based approach to answering two types of queries, namely range query and aggregative query. By using multipath as the communication scheme, our design increases the robustness of the network. Furthermore, using signature files solves the problem of message duplication, as well as reduces the energy cost of transmission. Our signature-file-based approach also offers users the option to control the error rate of the results by carefully setting the bit length and weight. The simulation results have shown that, compared to other approaches, our approach significantly reduces the communication cost and provides higher accuracy.

References

[1] R. Baeza-Yates and R. N. Berthier, *Modern Information Retrieval*. Reading, MA: Addison-Wesley, 1999.

[2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Pers. Commun. Mag.*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[3] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," in *Proc. ICDE*, 2004, pp. 449–460.

[4] R. S. Davis and K. Kamamohanarao, "A two-level superimposed coding scheme for partial match retrieval," *Inf. Syst.*, vol. 8, no. 4, pp. 273–280, 1983.

[5] A. Deshpande *et al.*, "Model-driven data acquisition in sensor networks," in *Proc. VLDB*, 2004, pp. 588–599.

[6] C. Faloutsos, "Access methods for text," *ACM Comput. Surv.*, vol. 17, no. 1, pp. 49–74, Mar. 1985.

[7] D. Goldenberg *et al.*, "Localization in sparse networks using sweeps," in *Proc. MobiCom*, 2006, pp. 110–121.

[8] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *Proc. MobiHoc*, 2005, pp. 402–413.

[9] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in *Proc. MobiCom*, 1999, pp. 174–185.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. HICSS*, 2000, p. 8020.

[11] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond average: Toward sophisticated sensing with queries," in *Proc. IPSN*, 2003, pp. 63–79.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. MobiCom*, 2000, pp. 56–67.

[13] M. Li and Y. Liu, "Rendered path: Range-free localization in anisotropic sensor networks with holes," in *Proc. ACM MobiCom*, 2007, pp. 51–62.

[14] M. Li and Y. Liu, "Underground structure monitoring with wireless sensor networks," in *Proc. IPSN*, 2007, pp. 69–78.

[15] M. Li, Y. Liu, and L. Chen, "Non-threshold based event detection for 3D environment monitoring in sensor networks," in *Proc. ICDCS*, 2007, p. 9.

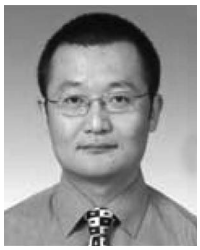
[16] J. Lian, L. Chen, K. Naik, M. T. Ozsu, and G. Agnew, "Localized routing trees for query processing in sensor networks," in *Proc. CIKM*, 2005, pp. 259–260.

[17] S. Madden, M. J. Franklin, and J. M. Hellerstein, "TAG: A tiny aggregation service for ad-hoc sensor networks," in *Proc. OSDI*, 2002, pp. 131–146.

[18] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in *Proc. SenSys*, 2004, pp. 250–262.

Mo Li (M'06) received the B.S. degree from Tsinghua University, Beijing, China, in 2004. He is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong.

His research interests include wireless sensor networks, pervasive computing, network security, and peer-to-peer computing.



Lei Chen (M'02) received the B.S. degree in computer science and engineering from Tianjin University, Tianjin, China, in 1994, the M.S. degree from the Asian Institute of Technology, Pathumthani, Thailand, in 1997, and the Ph.D. degree in computer science from the University of Waterloo, Waterloo, ON, Canada, in 2005.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. His research interests include multimedia and time series databases, sensor and peer-to-peer databases, and stream and probabilistic databases.

Jizhong Zhao (M'02) received the B.S., M.S., and Ph.D. degrees in computer science in 1992, 1995, and 2001, respectively, all from Xi'an Jiaotong University, Xi'an, China.

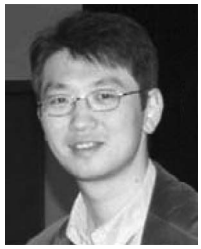
He is a currently a Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. His research interests include computer software, pervasive computing, distributed systems, and network security.

Dr. Zhao is a member of the IEEE Computer Society and of the Association for Computing Machinery.

Qian Zhang (M'00–SM'04) received the B.S., M.S., and Ph.D. degrees from Wuhan University, Wuhan, China, in 1994, 1996, and 1999, respectively, all in computer science.

In July 1999, she was the Research Manager of the Wireless and Networking Group, Microsoft Research Asia. Since September 2005, she has been an Associate Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. She has also participated in many activities with the Internet Engineering Task Force Robust Header Compression Working Group for TCP/IP header compression. She is the Associate Editor for *Elsevier Computer Networks* and *Elsevier Computer Communications*. She has also served as Guest Editor for special issues of *ACM/Springer Mobile Networks and Applications* and *Elsevier Computer Networks*. Her current research interests are in the areas of wireless communications, IP networking, multimedia, P2P overlay, and wireless security. She is the inventor of about 30 pending patents. She is the author of more than 150 refereed papers in leading international journals and key conference proceedings.

Dr. Zhang is the Vice Chair of the Multimedia Communication Technical Committee (MMTC) of the IEEE Communications Society. She is an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the IEEE TRANSACTIONS ON MULTIMEDIA. She has also served as Guest Editor for special issues of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Wireless Communications Magazine*, and *IEEE Communications Magazine*. She received the Massachusetts Institute of Technology Technology Review's TR 100 World's Top Young Innovator Award in 2004, the Best Asia-Pacific Young Researcher Award from the IEEE Communication Society in 2004, the Best Paper Award at MMTC of the IEEE Communication Society, the Best Paper Award at the Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks in 2006, and the Overseas Young Investigator Award from the Natural Science Foundation of China in 2006.



Yunhao Liu (M'02–SM'06) received the B.S. degree from Tsinghua University, Beijing, China, in 1995, the M.A. degree from Beijing Foreign Studies University, Beijing, in 1997, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, East Lansing, in 2003 and 2004, respectively.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. He is also an Adjunct Professor with Xi'an Jiaotong University, Xi'an, China, and with the Ocean University of China, Qingdao, China. His research interests include peer-to-peer computing, pervasive computing, and sensor networks.

Dr. Liu is a member of the Association for Computing Machinery.