Top-k Query Evaluation in Sensor Networks with the Guaranteed Accuracy of Query Results

Baichen Chen[1], Weifa Liang[1], and Geyong Min[2]

[1] Research School of Computer Science
The Australian National University
Canberra, ACT 0200, Australia
[2] School of Computing
University of Bradford
Bradford, BD7 1DP, United Kingdom

Abstract. In many applications of sensor networks including environmental monitoring and surveillance, a large volume of sensed data generated by sensors needs to be either collected at the base station or aggregated within the network to respond to user queries. However, due to the unreliable wireless communication, robust query processing in such networks becomes a great challenge in the design of query evaluation algorithms for some mission-critical tasks. In this paper we propose an adaptive, localized algorithm for robust top-k query processing in sensor networks, which trades off between the energy consumption and the accuracy of query results. In the proposed algorithm, whether a sensor is to forward the collected data to the base station is determined in accordance with the calculation of a proposed local function, which is the estimation of the probability of transmitting the data successfully. We also conduct extensive experiments by simulations on real datasets to evaluate the performance of the proposed algorithm. The experimental results demonstrate that the proposed algorithm is energy-efficient while achieving the specified accuracy of the query results.

1 Introduction

In recent years, technological advances have made it become possible to deploy large-scale sensor networks, consisting of hundreds or thousands of inexpensive sensors in an ad-hoc fashion, for environmental monitoring and security surveillance purposes [1,16]. In these applications, a large volume of sensing data generated by sensors needs to be either collected at the base station or aggregated within the network to respond to user queries. The sensor network thus can be treated as a *virtual database* by the database community [14]. The processing of queries in wireless sensor network includes the skyline query [4], top-k query [22,23,5,6,11], join query [9,20,21], and so on. Top-k query is a fundamental operator in databases that searches for very important objects according to the object rankings obtained by a variety of ranking techniques. Efficient processing of top-k query is crucial in many information systems that comprise a large amount of data [8]. Top-k query in a sensor network is to return the k

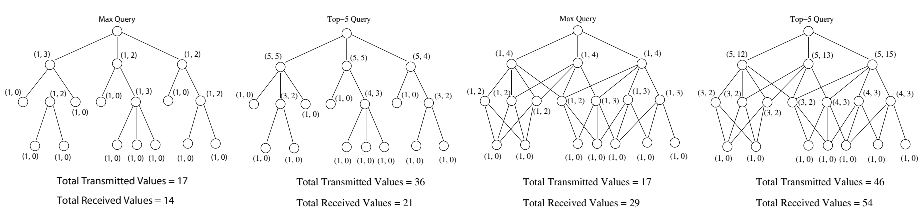
A. Hameurlain et al. (Eds.): DEXA 2011, Part I, LNCS 6860, pp. 156–171, 2011.

points with largest values to the base station, where a point is referred to as the sensed value and the ID of its generator (sensor). Wireless sensor networks that support top-k queries can be used to not only monitor the data generated by sensors in no time but also perform further data analysis for decision making. One such an application scenario is that the ornithologists who study the behaviors of various bird species in a given region forest are interested to know where the birds are most likely to gather [22]. To do so, they place the bird feeders at different locations in the monitored region and install one sensor at each feeder to count the number of birds at that feeder periodically. The result of the top-k query can assist the ornithologists to determine where the birds are likely to be attracted. For example, a top-k query can inquire which feeders attract the maximum number of birds. Thus, the ornithologists can observe the bird behaviors at a few places where the most attractive feeders are located.

A paramount concern in processing queries in energy-constrained wireless sensor networks is the energy conservation in order to prolong the network lifetime, because it usually is impractical to recharge the batteries that power the sensors. In addition, a query result with a certain degree of accuracy is acceptable, while the query results are typically computed by in-network processing. The existing in-network processing algorithms are mainly based on the tree routing structure, which include the ones in [13,15,24,25] for aggregate query, and the ones in [23,6,10] for top-k query, etc. However, the failure rates of wireless communication in wireless sensor networks are relatively high (up to 30% loss rate in common [27]), and each lost message at a sensor causes the loss of all the collected data from the subtree rooted at the sensor. As a result, it is not uncommon that 85% of sensed values are lost in a multi-hop sensor networks, causing significantly answer inaccuracy and compromising the monitoring quality [18]. To overcome the shortcoming of the tree structure in the accuracy of query results, several algorithms including algorithm `FATE-CSQ` in [12] have been proposed, which make use of the feedback-retransmission mechanism, i.e., if a transmission is failure, the parent sends the feedback messages to the children and the children retransmit their messages again. However, such algorithms result in high message complexity and long delay in message delivery. Gobriel et al [7] proposed an algorithm `RideSharing`, in which each sensor maintains two types of parents: the primary and several backup parents. If the primary parent does not receive the message from a child, it would send a vector to the other backup parents, asking for them to forward the message. To ensure that all the parents of a sensor can overhear the vector, it is required that all the parents and this sensor form a *clique*, i.e., each of them is located within the transmission range of each other. Although algorithm `RideSharing` avoids multiple retransmissions of the messages, it suffers high message complexity and long delay too, and furthermore, the clique may not exist in real networks.

Besides the mentioned tree-based algorithms, several researchers have proposed multi path-based and hybrid-based algorithms to deal with the aggregation queries in wireless sensor networks with high failure rates [3,2,18,17,26]. Considine et al [3,2] and Nath et al [18] proposed the multi path-based methods, based on the multi-ring routing structure for aggregation queries, in which the sensors are partitioned into different levels according to the number of hops between them and the base station. Therefore, the data transmission is performed level by level towards the base station. In the transmission by using the multi-ring structure, each sensor sends its messages to all of its neighbors at the level closer to the base station, rather than the single parent in the tree routing structure. In this paper, this approach is referred to as algorithm SD (Synopsis Diffusion) from [18]. The multi-ring structure is efficient in terms of energy consumption for some aggregation queries such as *MAX, MIN* and *SUM*, because each sensor aggregates the received data and broadcasts the aggregate result of the same size as each received data to its neighbors. Therefore, the transmission energy consumption on the multi-ring structure is almost the same as that on the tree structure. However, in dealing with complicated queries like top-k query and skyline query, a number of points rather than a partial result need to be sent to the base station as part of the query result, which means that the duplication of points will significantly increase the transmission and reception energy consumptions. This leads to that the sensors run out of their energy quickly, thereby reducing the lifetime of the sensor network. We here use an example to illustrate this. Fig. 1(a)-(d) show the number of sent and received points by the sensors when the tree based approach and algorithm SD are applied to answering a *MAX* query and a top-5 query, respectively. Each sensor has a labeled tuple, in which the first component is the number of points sent by the sensor, and the second one is the number of points received at the sensor. Initially, each sensor contains one point. To answer the *MAX* query, each sensor broadcasts a point with largest value among the received and its own points to its neighbors, while each sensor broadcasts 5 points with largest values among the received and its own points to its neighbors to answer the top-5 query. It can be seen that for the *MAX* query, algorithm SD has the same number of sent points and a few more received points, while for top-5 query, the number of sent and received points of algorithm SD is much more than those on tree topology.

To utilize the advantage brought by the tree and ring routing structures, Manjhi et al [17] proposed an algorithm TD (Tributary-Delta) for aggregation query processing, which is a hybrid approach, i.e., it adopts the routing structure for data aggregation to achieve the optimal performance. Algorithm TD tries to



(a) Max Query (Tree) (b) Top-5 Query (Tree) (c) Max Query (Ring) (d) Top-5 Query (Ring)

Fig. 1. Answering a Top-k Query and a Max Query on Tree and Ring Topologies

overcome the problems of tree and ring topologies by combining the best features of both topologies and tradeoffing the energy consumption and robustness of query processing. The core idea of algorithm `TD` is to divide the sensors into two categories. The sensors far away from the base station are called T sensors, which send their data to a single parent, while the sensors near to the base station are called M sensors, which send their data to multiple neighbors closer to the base station. The network thus is organized in regions to implement one of the two structures respectively. The region consists of the T sensors is called *Tributary* region, while the region consists of M sensors and the base station is called *Delta* region. The base station maintains the percentage of the sensors that contribute their data to the final result and decides whether to shrink or expand the *Delta* region for the future queries. If the percentage is below the user given threshold, the *Delta* region is expanded to improve the robustness; otherwise, the *Delta* region is shrunk to save energy. The adaptation of the routing structure is executed by changing the label of a sensor (T or M). However, algorithm `TD` seems not to be applicable to top-k query due to the following concerns. Firstly, the points are only sent to a parent at the T sensors, therefore the accuracy of query results will become low if the link failure rate is high. Secondly, the re-classification of sensors is purely derived from the statistics of the failure links in the transmission of current query evaluation. If the data distribution and the unpredictable status of the wireless links in the next period are different from the previous ones, such a re-classification may not improve the accuracy of the results at all. Thirdly, the sensors near to the base station usually consume more energy than the sensors far away from the base station. In algorithm `TD`, the M sensors send and receive as many points as the sensors in algorithm `SD`, which makes the M sensors run out of their energy quickly and disconnection between the base station and other sensors. Thus the sensor network is no more functioning. Lastly, the maintenance of the network structure may be expensive in energy consumption by broadcasting the messages for adapting the structure.

In this paper we deal with top-k query evaluation in wireless sensor networks efficiently and effectively. Our main contributions are as follows. We first analyze the drawbacks of applying existing algorithms for answering top-k queries, followed by giving a new definition of the accuracy of top-k query results. We then propose a localized, energy-efficient query evaluation algorithm tradeoffing between the energy consumption and the accuracy of query results. We finally conduct extensive experiments by simulations on real sensing datasets to evaluate the performance of the proposed algorithm. The experimental results show that the proposed algorithm outperforms existing ones in terms of energy consumption under the given constraint of the accuracy of results.

2 Preliminaries

2.1 System Model

We consider a sensor network consisting of n stationary sensors, randomly deployed in a region of interest, each measuring a numerical value. For each sensor v, $v.id$ is the ID of sensor v. Each *point* p is represented by a tuple $< p.sid, p.val >$, where $p.sid$ is the ID of sensor generating point p, and $p.val$ is the sensing (reading) value. Assume that $p.sid$ as well as $p.val$ is represented by 4 bytes. Thus, a point p is represented by 8 bytes in total. There is a base station with unlimited energy supply, which serves as the gateway between the sensor network and users. Each sensor can communicate with the other sensors within its transmission range, where the transmission range of all sensors in the network is identical. Denote by f the probability of a link failures during the wireless radio communication, i.e., the transmission through a link is failed with probability f, $0 \leq f \leq 1$. To transmit a message containing l bytes from one sensor to its neighbors, the amount of transmission energy consumed at the sender is $\rho_t + l * R$, while the amount of reception energy consumed at the receiver is $\rho_r + l * r$, where ρ_t and ρ_r are the sum of energy overhead on handshaking and sending and receiving the header of the message, and R and r are the amounts of transmission and reception energy per byte. We assume that the computation energy consumption on sensors can be ignored, because in practice it is several orders of magnitude less than the communication energy consumption, e.g., the authors in [14,19] claimed that the transmission of a bit of data consumes as much energy as executing 1,000 CPU instructions.

2.2 Query Structure

Denote by $N(v)$ the set of *neighbors* of sensor v. Each sensor can transmit points to the base station through one or multi-hop relays. We define the *distance* of v to the base station as the *minimum number of hop relays* from sensor v to the base station. The sensors in the network can be divided into several *levels*, and the sensors at the same level are also indexed. According to their distances to the base station, $v_{i,j}$ is referred to as the sensor with jth smallest ID at the ith level and V_i is the set of sensors at the ith level of the network. The base station is in V_1 (the base station is $v_{1,1}$), and the number of hops from a sensor $v_{i,j}$ to the base station is $i-1$. Let $Up(v_{i,j})$ be the set of *upstream sensors* of $v_{i,j}$, which is the set of neighbors of v in V_{i-1}, i.e., $Up(v_{i,j}) = \{v \mid v \in V_{i-1}, v \in N(v_{i,j})\}$. Similarly, denote by $Down(v_{i,j}) = \{v \mid v \in V_{i+1}, v \in N(v_{i,j})\}$ the set of *downstream sensors* of $v_{i,j}$. A sensor v is defined as a *partner* of sensor $v_{i,j}$ from sensor u, such that v and $v_{i,j}$ have the same downstream sensor u, i.e., the set of *partner* of sensor $v_{i,j}$ from sensor u is $Par(v_{i,j})_u = \{v \mid u \in Down(v_{i,j}), u \in Down(v)\}$. In other words, a sensor in $Up(u)$ is a partner of any sensor in $Up(u)$ from sensor u. $v_{i,j}$ is also a partner of itself. Thus $Par(v_{i,j}) = \bigcup_{u \in Down(v_{i,j})} Par(v_{i,j})_u$ is the set of partners of sensor $v_{i,j}$ from all its downstream sensors. A sensor v is defined as the *ancestors* of sensor $v_{i,j}$ if the points generated at sensor $v_{i,j}$ can be sent to sensor v through one or multi-hop relays, while a sensor v is defined as the *descendants* of $v_{i,j}$ if the points from v can be sent to $v_{i,j}$ through one or multi-hop relays. The sets of ancestors and descendants of sensor $v_{i,j}$ are referred to as $Anc(v_{i,j})$ and $Des(v_{i,j})$, respectively. In addition, denote by $Down(v_{i,j})_u$ the subset of downstream sensors of sensor $v_{i,j}$, in which the sensors are the ancestors of sensor u, i.e., $Down(v_{i,j})_u = \{v \mid v \in Down(v_{i,j}), v \in Anc(u)\}$. In the previous algorithms using the multi-ring structure, the points are transmitted from a sensor to all its upstream sensors and are regarded as the duplications of the points. The points thus are transmitted to the base station through multi-paths, which significantly increases the accuracy of the query results.

2.3 Accuracy of Top-k Query Results

The accuracy of the query results in previous studies is measured by the ratio of the obtained results to the real results. However, such a measurement may not be suitable for top-k queries. The reason is as follows. Assume that the base station issues a top-100 query to a sensor network of 1,000 sensors and receives the top-100 result from its 700 sensors, and the points from the rest of the 300 sensors are lost within the transmission due to link failures. However, due to the skew data distribution, the obtained top-100 result from the 700 sensors is the actual top-100 result from the 1000 sensors. If the data distribution is changed, e.g., the actual top-100 points all come from the 300 sensors, the obtained result and the actual result are completely different. In this case, the comparison between the obtained results and the actual results does not reflect the actual number of the points lost during the data transmission. To this end, we thus propose another metric to measure the accuracy of top-k query results that will not be affected by the data distribution as follows.

Denote by P the set of points in the sensor network. If there are x points in P which are actually used to determine the results, $\frac{x}{|P|}$ is defined as the accuracy of the top-k results, where the x points consist of two types of points: the points transmitted to the base station successfully, and the points failed to be transmitted to the base station because they are discarded by the sensors receiving at least k points with higher values than theirs. For each query, assume that an expected *threshold ratio* θ is given in advance, and at least $\theta * |P|$ points can be used for determining the results. The *single point threshold* θ_p is the probability of a point that is one of the x points. Denote by Ev_x the event that x points are used to determine the query result. Therefore, the accuracy of top-k result is

$$\begin{aligned}
\theta &= Pr(Ev_{|P|}) + Pr(Ev_{|P|-1}) + \ldots + Pr(Ev_{\lceil |P|\theta \rceil}) \\
&= \theta_p^{|P|} + \theta_p^{|P|-1}(1-\theta_p)|P| + \ldots + \binom{|P|}{\lceil |P|\theta \rceil}(\theta_p^{|P|-\lceil |P|\theta \rceil})(1-\theta_p)^{\lceil |P|\theta \rceil} \\
&= \sum_{i=0}^{|P|-\lceil |P|\theta \rceil+1} \binom{|P|}{i} \theta_p^{|P|-i}(1-\theta_p)^i.
\end{aligned} \tag{1}$$

The value of θ_p in Eq. (1) can be estimated provided that θ is given, and if each point used for the query result has a probability no less than θ_p, then the accuracy of the query result will be no less than θ. Note that due to the uncertainlity of the link failures, it cannot guarantee whether the accuracy of each top-k query result meets the specified threshold.

We now give the problem statement. Given a wireless sensor network $G(V, E)$, V is the set of sensors and the base station, and E is the set of links. Assume that each sensor contains a point initially. Let P be the set of points generated by all sensors. Assume that the average link failure rate is f, which means that a sensor v in the WSN transmits data to its neighbors, the neighbors cannot receive the correct data with probability $f(v)$ on average, and the query result accuracy threshold θ is given, too. The *robust top-k query evaluation* is to return the k points with the largest values with the query result accuracy being no smaller than θ. If the link failure rate is too high to meet the accuracy threshold θ, the top-k query evaluation will return the results as accurate as possible.

3 Robust Top-*k* Query Evaluation Algorithm

In this section we propose a novel localized evaluation algorithm for robust top-k query to tradeoff the energy consumption and the query result accuracy. The proposed algorithm is as follows. Built upon the query structure, a sensor $v_{i,j}$ first collects the local information (i.e., its neighboring sensors). To respond to a top-k query, each sensor $v_{i,j}$ performs different operations according to whether it has downstream sensors. If there is no downstream sensors, $v_{i,j}$ just broadcasts its points to its upstream sensors; otherwise, it receives the points from all its downstream sensors and puts the points with the same values and the generated sensors into the same group. Sensor $v_{i,j}$ examines the groups one by one to determine whether to broadcast the points of groups to its upstream sensors. Having examined all the groups, $v_{i,j}$ broadcasts some of the received points to its upstream sensors. To determine whether to broadcast a received point, $v_{i,j}$ calculates a value of a local function. If the value of the function for the received point is less than the threshold θ_p, the point is forwarded to its upstream sensors; otherwise, the point is discarded. The calculation of the local function is based on the local information of sensor $v_{i,j}$ and its information so far. In the following, we first describe how each sensor collects the local information and then propose the algorithm `Robust Top-`k `Query Evaluation` (`RTE` in short). Note that the proposed algorithm is a localized algorithm, which is preferable by distributive sensor networks.

3.1 Local Information Collection

We show how a sensor $v_{i,j}$ collects the local information of its neighboring sensors. Firstly, each sensor $v_{i,j}$ broadcasts its ID $v_{i,j}.id$ and the number of its upstream sensors $|Up(v_{i,j})|$ to its downstream sensors, and at the same time it also received the same information from its upstream sensors. In other words, each sensor $v_{i,j}$ broadcasts $\{v_{i,j}.id, |Up(v_{i,j})|\}$ to its downstream sensors and receives $\bigcup_{v \in Up(v_{i,j})}\{v.id, |Up(v)|\}$ from its upstream sensors. Secondly, sensor $v_{i,j}$ broadcasts $\bigcup_{v \in Up(v_{i,j})}\{v.id, |Up(v)|\}$ to its upstream sensors. It also receives the information broadcast from all its downstream sensors. Having received the information from one of downstream sensor u , i.e., $\bigcup_{v \in Up(u)}\{v.id, |Up(v)|\}$, $v_{i,j}$ obtained the information of partners in $Par(v_{i,j})_u$. Therefore, sensor $v_{i,j}$ obtains the IDs and the number of upstream sensors of all its partners, i.e., $\bigcup_{v \in Par(v_{i,j})}\{v.id, |Up(v)|\}$. Finally, $v_{i,j}$ makes use of the information to calculate the local function and determines whether to forward the received points. Note that the local information collection is only executed once.

3.2 Query Evaluation Algorithm

We propose the evaluation algorithm RTE for robust top-k query evaluation as follows. For each sensor $v_{i,j}$, the set of its upstream sensors is partitioned into two disjoint subsets: The first $\lceil |Up(v_{i,j})| * f \rceil$ sensors with smaller IDs are *appointed* to forward the points from $v_{i,j}$, while the other sensors will determine whether to forward the points by calculating a *local function*. Note that each sensor can easily know whether it is appointed by one of its downstream sensors because it stores the IDs of all its partners. The detailed procedure is as follows.

If sensor $v_{i,j}$ does not have any downstream sensors, it broadcasts the points to all its upstream sensors; otherwise, it checks the received points one by one. There are different groups of points at $v_{i,j}$, containing the same points from different downstream sensors. The points in the group $Gp(p)_{v_{i,j}}$ are the points whose values are $p.val$ and their generated sensors are $p.sid$. $v_{i,j}$ contains k different groups of points $Gp(p_1)_{v_{i,j}}, \ldots, (p_k)_{v_{i,j}}$, in which $p_1, \ldots, p_k$ are the k points with highest values among the points at sensor $v_{i,j}$, where for every two groups $Gp(p_x)_{v_{i,j}}$ and $Gp(p_y)_{v_{i,j}}$, $p_x.sid \neq p_y.sid$ or $p_x.val \neq p_y.val$. If sensor $v_{i,j}$ is appointed by any downstream sensors sending p_x to $v_{i,j}$ and p_x is in one of the top-k group, point p_x is marked as the point to be forwarded by $v_{i,j}$; otherwise, sensor $v_{i,j}$ calculates the function to determine whether to forward point p_x. The calculation of the local function will be introduced in the next section. If the value of the local function is not smaller than θ_p, point p_x is discarded by sensor $v_{i,j}$; otherwise, point p_x is marked to be forwarded. Having checked all the groups, sensor $v_{i,j}$ broadcasts the set of all marked points to its upstream sensors. The pseudo-code of algorithm RTE at each sensor is as follows.

Algorithm 1. `Algorithm RTE`(v)

```
begin
    collect the local information; receive θ and calculates θ_p;
    if v_{i,j} has no downstream sensors then
        broadcast the points to all the upstream sensors;
    else
        receive the points from the downstream sensors;
        group the points and obtains top-k points groups,
        G(p_1)_{v_{i,j}}, ..., G(p_k)_{v_{i,j}};
        foreach point p_x of group G(p_x)_{v_{i,j}} do
            if v_{i,j} is appointed by any downstream sensors sending p_x then
                p_x is marked to be forwarded;
            else
                calculate the local function;
                if the value of local function is smaller than θ_p then
                    p_x is marked to be forwarded;
        end
        broadcast the points that are marked to be forwarded;
end
```

3.3 Determination of Non-appointed Sensors

We then describe how a sensor $v_{i,j}$ determines whether forwarding a received point p if it is not appointed by any sender of p. The intuition of determination is that sensor $v_{i,j}$ calculates the probability of transmitting point p by its partners appointed to forward point p. If the probability is no less than the threshold, there is no need for it to forward point p; otherwise, sensor $v_{i,j}$ forwards point p. However, it is a bit difficult for sensor $v_{i,j}$ to calculate the probability of sending point p by the appointed partners, since $v_{i,j}$ only has the local rather than the global information. Therefore, an approximate value that is always smaller than that probability is designed for sensor $v_{i,j}$ as follows.

Denote by $pr(v)_p$ or $pr(V_{i-1})_p$ the probability of point p sent to sensor v or any sensor at the $i-1$th level, while $lpr(v)_p$ and $lpr(V_{i-1})_p$ are the local functions of sensor $v_{i,j}$ estimating the probability of that point p is sent to sensor v or any sensor at the $i-1$th level, respectively. Assume that point p is generated at sensor $v_{i',j'}$ and $v_{i,j}$ is the ancestor of $v_{i',j'}$. Recall that $Down(v_{i,j})_{v_{i',j'}}$ is the set of downstream sensors of $v_{i,j}$ and the ancestors of $v_{i',j'}$. In other words, point p can be transmitted from $v_{i',j'}$ to $v_{i,j}$ through sensor $u \in Down(v_{i,j})_{v_{i',j'}}$. Suppose that sensor $v_{i,j}$ receives point p successfully from m sensors $\{u_1, u_2, \dots, u_m\} \subseteq Down(v_{i,j})_{v_{i',j'}}$. Sensor v is a partner of sensor $v_{i,j}$ that is also ancestor of sensor $v_{i',j'}$. Denote by $d(v)$ the number of downstream sensors of sensor v in $\{u_1, u_2, \dots, u_m\}$. For a sensor v in $\bigcup_{t=1}^{m} Par(v_{i,j})_{u_t}$, sensor $v_{i,j}$ estimates the probability that sensor v receives point p is

$$lpr(v)_p = 1 - (1 - (1 - f)^{i'-i-1}(1 - f))^{d(v)}. \tag{2}$$

where f is the average of link failure rate and the value of $d(v)$ can be calculated by sensor $v_{i,j}$ as follows. Initially $d(v) = 0$. Having received point p from sensors $u_1, \dots, u_m$, sensor $v_{i,j}$ increments $d(v)$ by 1 if v is in $Up(u_t)$, $1 \le t \le m$, and i' can be obtained from $p.sid$. Thus the value of $lpr(v)_p$ can be calculated locally. Assume that there are m' partners of $v_{i,j}$, $v_1, \dots, v_{m'}$ in $\bigcup_{t=1}^{m} Par(v_{i,j})_{u_t}$ appointed to forward point p. Sensor $v_{i,j}$ calculates the value of $lpr(V_{i-1})_p$ by calculating the probability that all sensors $v_1, v_2, \dots, v_{m'}$ send point p to their upstream sensors. From Eq. (2), we have

$$\begin{aligned} lpr(V_{i-1})_p &= 1 - \prod_{t=1}^{m'} (1 - lpr(v_t)_p(1 - f^{|Up(v_t)|})) \\ &= 1 - \prod_{t=1}^{m'} (1 - (1 - (1 - (1 - f)^{i'-i})^{d(v_t)})(1 - f^{|Up(v_t)|})), \end{aligned} \tag{3}$$

The value of $|Up(v_t)|$ can be obtained by sensor $v_{i,j}$ through local information collection and all the variables in Eq. (3) can be obtained by sensor $v_{i,j}$, thus the value of $lpr(V_{i-1})_p$ can be calculated locally. For each received point p, if sensor $v_{i,j}$ is not appointed, it calculates the value of $lpr(V_{i-1})_p$. If $lpr(V_{i-1})_p > \theta_p$, sensor $v_{i,j}$ discards point p; otherwise, sensor $v_{i,j}$ sends point p to all its upstream sensors even if it is not appointed to forward point p. When $lpr(V_{i-1})_p$ is small, all the sensors need to forward the points to their upstream sensors to improve the probability no matter whether they are appointed. Consequently, the proposed algorithm is the same as algorithm SD. If the estimated probability is high, the non-appointed sensors discard the points directly. $lpr(V_{i-1})_p$ in Eq. (3) is determined by f, $|Up(v_t)|$, $d(v_t)$, and $i-i'$, while the values of f and $|Up(v_t)|$ reflect the physical condition of the sensor networks, e.g., the average link failure rate and the deployment of the sensors (the number of upstream sensors of a sensor). If f is large and $|Up(v_t)|$ is small, this means the chance to transmit point p successfully is unlikely, the value of $lpr(V_{i-1})_p$ will become small and consequently more sensors help is needed in order to forward point p to their upstream sensors. $d(v_t)$ is determined by the number of copies of point p received by sensor $v_{i,j}$ and the number of downstream sensors of sensor v_t. A smaller $d(v_t)$ implies that the number of copies of point p is small and loss of these copies will lead to the failure of the transmission of point p. When $d(v_t)$ is small, $lpr(V_{i-1})_p$ will be small as well and more sensors forward point p to increase the number of copies of p. $i-i'$ shows the number of hops from the generated sensor of point p $(v_{i',j'})$ to sensor $v_{i,j}$. The larger the $i-i'$ is, the more possible point p is in the top-k result because it has larger values than more points from the subtree rooted at sensor $v_{i,j}$. If $i-i'$ is large, $lpr(V_{i-1})_p$ will be small and the non-appointed sensors will forward point p to help increase the probability of transmitting point p successfully. In conclusion the proposed estimation of probability is a self-adapting function in the accordance with the status of the links and the routing structure of the different sensor networks, which tradeoffs the energy consumption and the accuracy of the query results well. In the following, we prove that the value of function $lpr(V_{i-1})_p$ is smaller than the probability that point p is sent to any sensor at the $i-1$th level by giving the following theorem.

Theorem 1. *Assume that point p is generated at sensor $v_{i',j'}$ and received by sensor $v_{i,j}$. Then, $Pr(V_{i-1})_p \geq\ lpr(V_{i-1})_p$.*

Proof. Recall that $Par(v_{i,j})_u$ is the set of partners of sensor $v_{i,j}$ from u, in which the sensors and $v_{i,j}$ have the same downstream sensor u. If sensor $v_{i,j}$ receives point p from some of its downstream sensors in $Down(v_{i,j})_{v_{i',j'}}$, it is obvious that the sensors in $\bigcup_{u\in Down(v_{i,j})_{v_{i',j'}}} Par(v_{i,j})_u$ are the ancestors of sensor $v_{i',j'}$ and likely to receive point p. For a partner $v \in \bigcup_{u\in Down(v_{i,j})_{v_{i',j'}}} Par(v_{i,j})_u$, the probability of that it received p is

$$pr(v)_p = 1 - \prod_{u\in Down(v)_{v_{i',j'}}} (1-pr(u)_p)(1-f)), \tag{4}$$

where v is at the ith level and each u in $Down(v)_{v_{i',j'}}$ is the downstream sensor sending point p to v. There may be a sensor $u \in Down(v)_{v_{i',j'}}$ that is not a downstream sensor of sensor $v_{i,j}$. Thus, set $Down(v)_{v_{i',j'}}$ and the value of $pr(u)_p$ in Eq. (4) is not known by sensor $v_{i,j}$, because every sensor only has the local information and consequently the value of $pr(V_{i-1})_p$ is not known either.

The probability that point p is sent to the upstream sensors of v is $pr(v)_p * (1 - f^{|Up(v)|})$. Thus, the probability of sending point p to any sensor in V_{i-1} is

$$pr(V_{i-1})_p = 1 - \prod_{v_{i,k} \in Anc(v_{i',j'})} (1 - pr(v_{i,k})_p(1 - f^{|Up(v_{i,k})|})), \tag{5}$$

where $v_{i,k}$ at the ith level is an ancestor of sensor $v_{i',j'}$.

Within Eq. (4), $pr(u)_p$ is the probability that p is sent from sensor $v_{i',j'}$ to a downstream sensor u of v at the $(i+1)$th level. The value of $pr(u)_p$ is minimum when point p is sent to sensor u through single path with $i' - i - 1$ hops, that is, $pr(u)_p \geq (1-f)^{i'-i-1}$. And $d(v)$ is the number of sensors in $Down(v)_{v_{i',j'}} \cap \{u_1, u_2, \ldots, u_m\}$, where $\{u_1, \ldots, u_m\}$ are the downstream sensors sending point p to $v_{i,j}$ successfully. Obviously $d(v) \leq Down(v)_{v_{i',j'}}$. Because $(1-f)^{i'-i-1} \leq pr(u)_p$ and $d(v) \leq |Down(v)_{v_{i',j'}}|$, from Eqs. (4) and (2), $lpr(v)_p \leq pr(v)_p$. $lpr(v)_p \leq pr(v)_p$ and $\{v_1, \ldots, v_{m'}\} \subseteq Anc(v_{i',j'})$, we have

$$\begin{aligned} lpr(V_{i-1})_p &= 1 - \prod_{t=1}^{m'} (1 - lpr(v_t)_p(1 - f^{|Up(v_t)|})) \\ &\leq 1 - \prod_{v_{i,k} \in Anc(v_{i',j'})} (1 - pr(v_{i,k})_p(1 - f^{|Up(v_{i,k})|})) \\ &= pr(V_{i-1})_p. \end{aligned} \tag{6}$$

Thus, if $lpr(V_{i-1})_p \geq \theta_p$, then $pr(V_{i-1})_p \geq \theta_p$.

Therefore, if a point p generated at $v_{i',j'}$ is forwarded to sensors at the ith level and there is a non-appointed sensor $v_{i,j}$ discarding point p because $lpr(V_{i-1})_p > \theta_p$, the probability that point p is sent to any sensor in V_{i-1} is also larger than θ_p. If a sensor $v_{i-1,k}$ receives point p and other k points with larger values than $p.val$, p is impossible to be part of top-k result and it is transmitted successfully; otherwise, if there is a sensor $v_{i-1,k}$ receiving point p but discarding it because $lpr(V_{i-1})_p \geq \theta_p$, the probability of sending point p to the sensors in V_{i-2} is not smaller than θ_p. Eventually point p is sent to the base station with probability not smaller than θ_p if there is at least a sensor at the second level discarding point p because the locally estimated probability is not smaller than θ_p.

3.4 The Extension of the Algorithm

It is well known that link failure rates in wireless sensor networks are not identical. Denote by $f_{u,v}$ the failure rate of a link from sensor u to sensor v. Our proposed algorithm can be extended for this generalized scenario as well. First, in local information collection phase, each sensor broadcasts the link failure rates of the links between itself and its upstream and downstream sensors in addition to the number links between them. Second, each sensor calculates the probabilities of receiving a point and forwarding the point along with the link failure probability. Finally, the local function is modified to suit for WSNs with different link failure rates. Denote by $pr(p)_v$ the probability of sensor v receiving point

p. If v is the generator of p, $pr(p)_v = 1$ and v broadcasts point p with $pr(p)_v$; otherwise, $pr(p)_v$ is calculated as follows. Assume that sensor v received point p with probabilities $pr(p)_{u_1}, \ldots, pr(p)_{u_m}$ from sensors $u_1, \ldots, u_m$. The probability that v receives p is $pr(p)_v = 1 - \prod_{i=1}^{m}(1 - pr(p)_{u_i} * f_{u_i,v})$. Therefore, sensor v estimates the probability that sensor v' in $\bigcup_{t=1}^{m} Par(v_{i,j})_{u_t}$ if it receives point p from sensor $u_1, u_2, \ldots, u_m$, which is

$$lpr(v')_p = 1 - \prod_{v' \in Up(u_t)} (1 - pr(p)_{u_t}(1 - f_{u_t,v'})), \tag{7}$$

where $1 \leq t \leq m$. $pr(p)_{u_t}$ is received by sensor v and $f_{u_t,v'}$ can be obtained through local information collection. Therefore the local function is modified as

$$lpr(V_{i-1})_p = 1 - \textstyle\prod_{t=1}^{m'}(1 - lpr(v_t)_p(1 - \prod_{x=1}^{|Up(v_t)|} f_{v_t,n_x})), \tag{8}$$

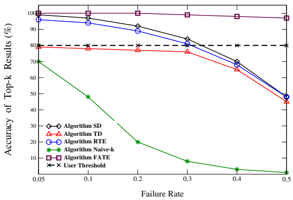
where $n_1, n_2, \ldots, n_x$ are the upstream sensors of v_t. All the variables in the modified function can be obtained by sensor v, and this indicates that the modified function can also be calculated by sensor v locally.

4 Performance Study

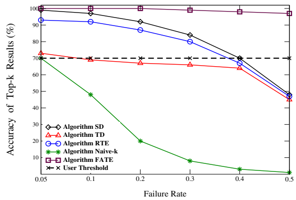
In this section we evaluate the performance of the proposed algorithm in terms of the total energy consumption, the maximum energy consumption among sensors, and the accuracy of the top-k query results. We assume that the sensor network is used to monitor a $100m \times 100m$ region of interest. Within the region, 1000 sensors are randomly deployed by the `NS-2` simulator [30] and the base station is located at the square center. There is a communication channel between two sensors if they are within the transmission range (5 meters in this paper) of each other. Each point is represented by 8 bytes. It is supposed that the energy overhead on transmitting and receiving a header and the handshaking are $\rho_t = 0.4608\ mJ$ and $\rho_r = 0.1152\ mJ$. The energy consumption of transmitting and receiving one byte are $R = 0.0144\ mJ$ and $r_e = 0.00576\ mJ$, respectively, following the parameters given in a commercial sensor MICA2 [28]. In our experiments, we use the real sensing dataset [29]. The performance of algorithm `SD` in [18], algorithm `TD` in [17], algorithm `FATE` based on the re-transmission mechanism in [12] and a well-known algorithm `Naive-`k in [22] on the tree structure is used as the benchmark for comparison purpose.

4.1 Performance Comparison with Equal Link Failure Rates

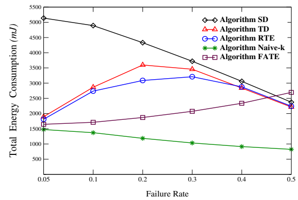
We first study the performance of various algorithms with different thresholds θ and equal link failure rates f, where θ is 0.7 or 0.8, while f is ranged from 0.05 to 0.5. Assume that a top-30 query is broadcast to all sensors. The results of the experiments are the average of running the top-30 query by different algorithms 1,000 times, and in each time the status of each link in the network is randomly determined, according to the given link failure rate.



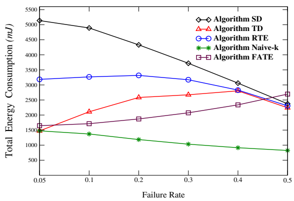
(a) Accuracy of Top-k Results when θ=0.8



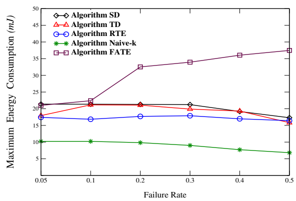
(b) Accuracy of Top-k Results when θ=0.7



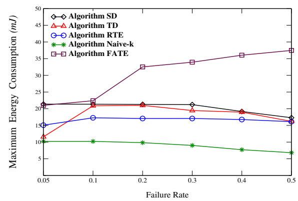
(c) Total Energy Consumption when θ=0.8



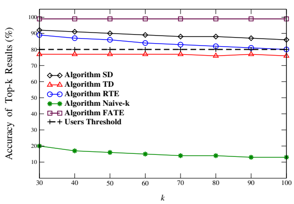
(d) Total Energy Consumption when θ=0.7



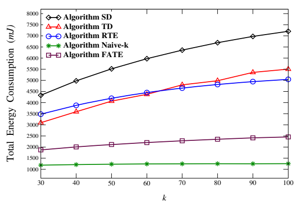
(e) Max Energy Consumption when θ=0.8



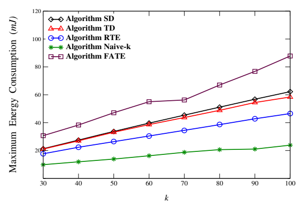
(f) Max Energy Consumption when θ=0.7



(g) Accuracy of Top-k Results when θ=0.8, f = 0.2



(h) Total Energy Consumption when θ=0.8, f=0.2



(i) Max Energy Consumption when θ=0.8, f=0.2

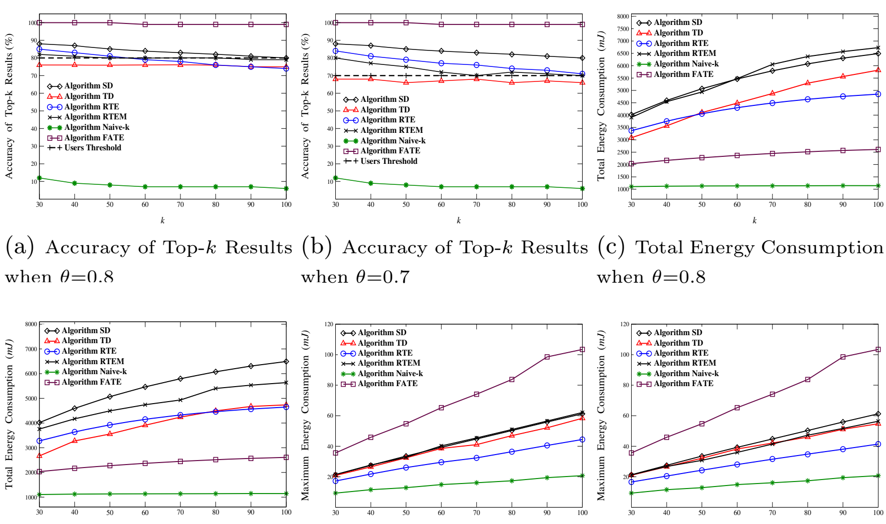
Fig. 2. The performance by various algorithms with identical link failure rates

From Fig. 2, we can observe that the query results delivered by algorithms FATE and SD has a higher accuracy than that of the other algorithms, but algorithm FATE has the largest maximum energy consumption and SD has the largest total energy consumption. The accuracy of query result by algorithm Naive-k drops sharply when the failure rate increases, which implies that it is not robust under the unstable communication environment. Compared to another adaptive algorithm TD, the results delivered by algorithm RTE is more accurate but with less energy consumption in overall. Figs. 2(g), 2(h), and 2(i) indicate the performance of various algorithms with different ks when $\theta = 0.8$ and $f = 0.2$. It can be seen that the accuracy of query results by algorithm RTE is above the threshold and the total and the maximum energy consumptions by it is smaller than these by the other mentioned algorithms. This implies algorithm RTE makes a better trade off between the accuracy of the query result and the energy consumption.

4.2 Performance Comparison with Different Failure Rates

We then evaluate the performance of various algorithms with different link failure rates. We assume that the link failure rate of each link is randomly generated and with within the range from 0 to 0.5. The modified version of algorithm `RTE` for this general case is referred to as `RTEM`.

Fig. 3 illustrates the performance curves of different algorithms, in which we can see that with various values of θ, the query results delivered by algorithm `FATE` are the most accurate, while the accuracy of the results by algorithm `Naive-`k is the worst. As shown in Figs. 3(a) and 3(b), the accuracy of algorithm `RTEM` is above the broken line representing θ and they are closer to the curves of the accuracy of algorithm `SD` with the increase of k. Although there is no guarantees that the accuracy of query results by algorithms `RTE` and `TD` meets the given threshold, the query result accuracy by both algorithms is close to the threshold, due to the adaptive mechanisms embedding in both the algorithms. From Figs. 3(c) and 3(d), with the decrease of the threshold, the energy consumption by algorithm `RTEM` is close or smaller than that by algorithm `SD`, since the dynamic decision plays a crucial role in making the trade-off between the energy consumption and the accuracy of query results. And it is observed from Figs. 3(e) and 3(f) that algorithm `RTE` has the better performance in maximum energy consumption than any other algorithms except algorithm `Naive-`k.



(a) Accuracy of Top-k Results when θ=0.8 (b) Accuracy of Top-k Results when θ=0.7 (c) Total Energy Consumption when θ=0.8

(d) Total Energy Consumption when θ=0.7 (e) Max Energy Consumption when θ=0.8 (f) Max Energy Consumption when θ=0.7

Fig. 3. The performance by various algorithms with random link failure rates when θ ranges from 0.7 to 0.9

5 Conclusion

In this paper we have tackled the problem of robust top-k query processing in wireless sensor networks. The objective is to minimize the energy consumption with the accuracy of query results constraint. We proposed a localized algorithm to strive the finest tradeoff between the energy consumption and the accuracy of the results. We conducted extensive experiments by simulations to evaluate the performance of the proposed algorithm. The experimental results show that there is a non-trivial tradeoff between the energy consumption and the accuracy of query results. The proposed algorithm is more energy-efficient than that of the existing algorithms while meeting the specified accuracy requirement on query results.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine, 102–114 (2002)
2. Considine, J., Hadjieleftheriou, M., Li, F., Byers, J., Kollios, G.: Robust approximate aggregation in sensor data management systems. ACM TODS 34, 1–33 (2009)
3. Considine, J., Li, F., Kollios, G., Byers, J.: Approximate aggregation techniques for sensor databases. In: Proc. of ICDE, pp. 449–460. IEEE, Los Alamitos (2004)
4. Chen, B., Liang, W., Yu, J.X.: Progressive skyline query evaluation and maintenance in wireless sensor networks. In: Proc. of CIKM, pp. 1445–1448. ACM, New York (2009)
5. Chen, B., Liang, W., Yu, J.X.: Online time interval top-k queries in wireless sensor networks. In: Proc. of MDM, pp. 177–182. IEEE, Los Alamitos (2010)
6. Chen, B., Liang, W., Zhou, R., Yu, J.X.: Energy-efficient top-k query processing in wireless sensor networks. In: Proc. of CIKM, pp. 329–338. ACM, New York (2010)
7. Gobriel, S., Khattab, S., Mosse, D., Brustoloni, J., Melhem, R.: RideSharing: fault tolerant aggregation in sensor networks using corrective actions. In: IEEE SECON, pp. 595–604 (2006)
8. Ilyas, I.F., Beskales, G., Soliman, M.A.: A survey of top-k query processing techniques in relational database systems. ACM Computing Survey 40(4) (2008)
9. Jurca, O., Michel, S., Herrmann, A., Aberer, K.: Continuous Query Evaluation over Distributed Sensor Networks. In: Proc. of ICDE, pp. 912–923. IEEE, Los Alamitos (2010)
10. Jiang, H., Cheng, J., Wang, D., Wang, C., Tan, G.: Continuous multi-dimensional top-k query processing in sensor networks. In: Proc. of INFOCOM. IEEE, Los Alamitos (2011)
11. Liang, W., Chen, B., Yu, J.X.: Top-k query evaluation in sensor networks under query response time constraint. Information Sciences 181, 869–882 (2011)
12. Lazaridis, I., Han, Q., Mehrotra, S., Venkatasubramanian, N.: Fault tolerant evaluation of continuous selection queries over sensor data. International Journal of Distributed Sensor Networks 5, 338–360 (2009)
13. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TAG: a tiny aggregation service for ad hoc sensor networks. ACM SIGOPS Operating Systems Review 36, 131–146 (2002)

14. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: The design of an acquisitional query processor for sensor networks. In: Proc. of SIGMOD, pp. 491–502. ACM, New York (2003)
15. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: TinyDB: An acquisitional query processing system for sensor networks. ACM Trans. Database Systems 30, 122–173 (2005)
16. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proc. of International Workshop on Wireless Sensor Networks and Applications, pp. 88–97. ACM, New York (2002)
17. Manjhi, A., Nath, S., Gibbonsgh, P.B.: Tributaries and deltas: efficient and robust aggregation in sensor network streams. In: Proc. of SIGMOD, pp. 287–298. ACM, New York (2005)
18. Nath, S., Gibbons, P.B., Seshan, S., Anderson, Z.R.: Synopsis diffusion for robust aggregation in sensor networks. In: Proc. of SenSys, pp. 250–262. ACM, New York (2004)
19. Pottie, G.J., Kaiser, W.J.: Wireless integrated network sensors. Communication of the ACM 43(5), 51–58 (2000)
20. Stern, M., Buchmann, E., Bohm, K.: Towards efficient processing of general-purpose joins in sensor networks. In: Proc. of ICDE, pp. 126–138. IEEE, Los Alamitos (2009)
21. Stern, M., Bohm, K., Buchmann, E.: Processing continuous join queries in sensor networks: a filtering approach. In: Proc. of SIGMOD, pp. 267–278. ACM, New York (2010)
22. Silberstein, A., Braynard, R., Ellis, C., Munagala, K., Yang, J.: A sampling-based approach to optimizing top-k queries in sensor networks. In: Proc. of ICDE, pp. 68–79. IEEE, Los Alamitos (2006)
23. Wu, M., Xu, J., Tang, X., Lee, W.-C.: Top-k monitoring in wireless sensor networks. IEEE Trans. Knowledge and Data Engineering 19(7), 962–976 (2007)
24. Yao, Y., Gehrke, J.: The cougar approach to in-network query processing in sensor networks. ACM SIGMOD Record 31, 9–18 (2002)
25. Yao, Y., Gehrke, J.: Query processing for sensor networks. In: Proc. of CIDR (January 2003)
26. Ye, F., Zhong, G., Lu, S., Zhang, L.: GRAdient broadcast: a robust data delivery protocol for large scale sensor networks. Wireless Networks 11, 285–298 (2005)
27. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proc. of SenSys, pp. 1–13. ACM, New York (2003)
28. Crossbow Inc.: MPR-Mote Processor Radio Board Users Manual
29. `http://db.csail.mit.edu/labdata/labdata.html` (2004)
30. The Network Simulator-ns2 (2006), `http://www.isi.edu/nsnam/ns`