

Energy-Efficient Top- k Query Processing in Wireless Sensor Networks

Baichen Chen
Australian National University
Canberra, ACT, Australia
baichen@cs.anu.edu.au

Rui Zhou
Swinburne Univ. of Tech.
Melbourne, VIC, Australia
rzhou@swin.edu.au

Weifa Liang
Australian National University
Canberra, ACT, Australia
wliang@cs.anu.edu.au

Jeffrey Xu Yu
Chinese Univ. of Hong Kong
Shatin, NT, Hong Kong
yu@se.cuhk.edu.hk

ABSTRACT

Technological advances have enabled the deployment of large-scale sensor networks for environmental monitoring and surveillance purposes. The large volume of data generated by sensors needs to be processed to respond to the users queries. However, efficient processing of queries in sensor networks poses great challenges due to the unique characteristics imposed on sensor networks including slow processing capability, limited storage, and energy-limited batteries, etc. Among various queries, top- k query is one of the fundamental operators in many applications of wireless sensor networks for phenomenon monitoring. In this paper we focus on evaluating top- k queries in an energy-efficient manner such that the network lifetime is maximized. To achieve that, we devise a scalable, filter-based localized evaluation algorithm for top- k query evaluation, which is able to filter out as many unlikely top- k results as possible within the network from transmission. We also conduct extensive experiments by simulations to evaluate the performance of the proposed algorithm on real datasets. The experimental results show that the proposed algorithm outperforms existing algorithms significantly in network lifetime prolongation.

Categories and Subject Descriptors

H.2.4 [Database Management]: Query Processing; C.2.4 [Computer-Communication Networks]: Distributed Application

General Terms

Algorithms, Experimentation, Performance

Keywords

Top- k query, Wireless sensor network, Energy optimization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...\$10.00.

1. INTRODUCTION

Technological advances in recent years have enabled the deployment of large-scale sensor networks consisting of hundreds or thousands of inexpensive sensors in an ad-hoc fashion. Such networks now are used for a variety of environmental monitoring and surveillance purposes including measurements of meteorological data (e.g. temperature, pressure, humidity), noise levels, chemicals, etc [5, 22]. During this course, a large volume of sensing data generated by sensors is needed to be processed within the network to respond to queries from users. The wireless sensor network is thus treated as a virtual database by the database community [2]. Top- k query is a fundamental operation to search for the most important objects according to object ranking obtained by various ranking techniques. Efficient processing of top- k query is crucial in many applications [7]. Unlike those studies on top- k query in traditional databases, in this paper we focus on optimizing top- k query evaluation in resource-constrained wireless sensor networks (WSNs). 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. For example, meteorologists make use of a wireless sensor network to sense meteorological attributes such as temperature, humidity and rainfall in a region of interest. A top- k query in such a WSN can easily identify the k places suffering the high temperatures for statistic and analysis purpose. Another scenario is that ornithologists who study various bird species in a given forest are interested to know where the birds are most likely to gather [24]. To do so, they place the bird feeders at various locations in the forest and install a sensor at each feeder to count the number of birds on that feeder periodically. The result of this 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 bird behaviors at a few places where the most attractive feeders are located.

Generally speaking, query processing in WSNs is essentially different from that in traditional databases due to the

unique characteristics imposed on sensors [16], which can be seen from the following aspects. Firstly, to prolong network lifetime, the energy consumption rather than the query response time or the space used is the optimization objective for query evaluation, since the battery-powered sensors will quickly become inoperative if the large quantity of sensing data is transmitted to the base station through multi-hop relays, and the lifetime of a sensor network is closely tied to the energy consumption rate of its sensors. Secondly, a wireless sensor network containing N sensors can be viewed as a distributed system of N nodes, while this special distributed system is different from the general distributed system, in which there is no any single powerful node serving as the collection center to collect data from all the sensors. Each sensor transmits its data to the base station through multi-hop relays, which consumes energy for each data transmission. This implies that it is more expensive to obtain the sensed data from the sensors far away from the base station than those near to it. Finally, as the major optimization objective for query processing in sensor networks, the network lifetime is determined not only by the total energy consumption of all sensors but also by the maximum energy consumption among the sensors. The sensors near to the base station consume more energy than the others, because they relay the data for the others and they will exhaust their batteries first. Once they run out of energy, the rest of sensors will be disconnected from the base station, no matter how much residual energy left the rest of the sensors. Consequently, the network is no longer functioning even if the total energy consumption per query is reasonably small. This implies that in terms of query optimization in WSNs, only minimizing the total energy consumption is insufficient. Minimizing the maximum energy consumption among sensors is also critical to prolong network lifetime. Hence, how to evaluate queries effectively and efficiently in WSNs poses great challenges.

The previous studies of top- k query in distributed systems mainly focused on the distributed top- k query, which is to find k objects with the highest scores, assuming that each object is distributed in multiple nodes. In each node, the object has a local score. The score of the object is combined from all of local scores by a given function [6, 14, 15]. In this paper, we deal with a different top- k query which is identical to the one proposed in [9, 24]. That is, each sensor senses one or multiple numerical values from its vicinity, and each value and its generator sensor is referred to as a *point* in the rest of this paper. A point p is represented by a tuple $\langle p.sid, p.value \rangle$, where $p.sid$ is the ID of sensor generating point p , and $p.val$ is the sensed numerical value. The top- k query in WSNs is to return the k points with the highest sensed values. Due to the difference of distributed top- k query and top- k query in WSNs, the existing algorithms for distributed top- k query are inapplicable to the top- k query evaluation in sensor networks.

1.1 Related Work

Top- k query has been extensively studied in centralized and distributed databases [6, 10, 11, 13, 14, 15]. As men-

tioned, the proposed techniques and algorithms are inapplicable to WSNs due to the unique characteristics imposed on sensors. Several studies investigated in-network processing in WSNs for query optimization, e.g., the algorithms in [1, 2, 20, 21] exploited for simple aggregation, while the others dealt with more complicated queries including order-statistics [3, 4], skyline query [23] and top- k query [8, 9, 24]. To evaluate a top- k query in WSNs, Silberstein *et al* [24] provided several approximate solutions with high probability, based on a prediction model built on the samples of previous top- k query results. They demonstrated how to improve the accuracy of the top- k results under given energy constraints, by formulating the problem as a linear programming, and developed a series of top- k query planning algorithms. Wu *et al* [8, 9] exploited the semantics of top- k query and proposed a Filter-based maintenance algorithm (FILA) to maintain the top- k points by assigning a dedicated filter for each sensor. Given the current top- k points, each sensor is assigned an interval of values serving as its filter to suppress unnecessary data from transmission. Meanwhile, the base station also maintains a copy of the filter at each sensor. However, the energy saving by their solution is based on an assumption that the probing messages broadcast by the base station can be heard by all sensors directly and the energy consumption of each sensor by receiving the probing messages is not taken into account, which is too restrictive and may not be realistic in the real world. Note that the work by Wu *et al* [8, 9] is the top- k results maintenance and they did not cope with top- k query evaluation efficiently and effectively. In addition, the proposed algorithms by both Silberstein *et al* [24] and Wu *et al* [8, 9] are centralized algorithms, which may not be suitable for the real distributed sensor networks.

1.2 Contributions

In this paper we consider top- k query evaluation in WSNs to maximize the network lifetime through devising a capable filter mechanism and striking the right balance between the total energy consumption and the maximum energy consumption among the sensors per query. The main contributions are as follows.

- We first address the challenges of query processing in WSNs, followed by introducing a realistic cost model for top- k query evaluation which takes into account not only the number of messages transmitted within the network but also the number of points contained in the messages.
- We then devise a novel, filter-based, localized algorithm for top- k query evaluation, in which a powerful filter is developed to eliminate as many unlikely top- k points as possible within the network from transmission.
- 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 top- k evaluation algorithm outperforms the existing one in terms of the total

energy consumption, the maximum energy consumption among the sensors and the network lifetime. In addition, the proposed algorithm is distributive, which is preferable to WSNs.

The remainder of the paper is organized as follows. Section 2 introduces the cost model and problem definitions, followed by briefly introducing an existing top- k query evaluation algorithm. Section 3 proposes an energy-efficient algorithm for top- k query evaluation and conducts theoretical analysis on the filtering capability of the devised filter. To evaluate the performance of the proposed algorithm, extensive experiments on real sensing datasets are conducted in Section 4, and the conclusions are given in Section 5.

2. PRELIMINARIES

2.1 System Model

We consider a sensor network consisting of N stationary sensors v_1, v_2, \dots, v_N , randomly deployed in a region of interest, and a base station r with unlimited energy supply located at the center of the region. For the sake of simplicity, we assume that the topology of the sensor network is a spanning tree \mathcal{T} rooted at the base station [2]. Otherwise, such a tree can be obtained by applying any spanning tree algorithm in the network like the one by TAG [1]. We further assume that the transmission ranges of all sensors are identical, and each sensor can communicate with the base station via one or multi-hop relays. In this paper, each sensor has a point p and $p.sid$ as well as $p.val$ is represented by 4 bytes. Thus, a point p is represented by 8 bytes in total. To transmit a message containing l bytes of data from a sensor to one of its neighboring sensors, the amount of transmission energy consumed at the sender is $\rho_t + R * l$ and the amount of reception energy consumption at the receiver is $\rho_r + r_e * l$, where ρ_t and ρ_r are the sum of energy overhead on handshaking between the two sensors and transmitting and receiving the message header, R and r_e are the amounts of transmission and reception energy per byte. We assume that the sensing and computation energy consumptions on sensors are not taken into account, because in practice they are several orders of magnitude less than that of wireless communication energy consumption. For example, the authors in [2, 16] claimed that the transmission of 1-bit data consumes as much energy as executing 1,000 CPU instructions. Therefore, unless otherwise specified, we only compare the communication energy consumption of different algorithms in the later performance evaluation.

2.2 Problem Definition

Given a wireless sensor network, assume that $P(v_i)$ is the set of points generated at sensor v_i , then $P = \bigcup_{i=1}^N P(v_i)$ is the point set in the whole sensor network. A top- k query is to find the k points in P with the highest values, $1 \leq k \leq N$. If there are more than k points with the highest values, k of them are selected as the result, which is determined by the rank of the other attributes of the points like the IDs

of their generator. The top- k query problem is defined as follows. Given a sequence of top- k queries with various k s issued at the base station, the problem is to propose an evaluation plan for each incoming top- k query such that the energy consumption in terms of the total energy consumption and the maximum energy consumption among the sensors for answering the query is minimized with the ultimate aim to maximize the network lifetime. The network lifetime is referred to as the time when the first sensor in the network exhausts its energy [12].

2.3 The Naive- k Algorithm

Before we proceed, we briefly review a popular algorithm Naive- k for top- k query evaluation by Silberstein et al [24], which will be served as the benchmark. Algorithm Naive- k computes the answer bottom-up in one pass over the network. In algorithm Naive- k , the leaf sensors forward their points to their parent. If a internal sensor contains k' ($< k$) points (including its own), then it forwards all the points to its parent. Otherwise, it forwards its top- k points to its parent. In the end, the base station obtains the top- k points from the collected points from its children and its own, which is the result of top- k query.

3. TOP- K QUERY ALGORITHM

In this section we propose a novel filter-based algorithm for top- k query evaluation in WSNs. For a given set S of points, assume that the points in S are sorted in decreasing order of their values. The α -quantile of S is the point whose rank is $\lceil \alpha |S| \rceil$ and the value of the α -quantile is referred to as α -quantile value, where α is constant with $0 < \alpha < 1$. For example, given a set $S = \{p_1, \dots, p_{|S|}\}$ with $p_i.val \geq p_{i+1}.val$, α -quantile of S is point $p_{\lceil \alpha |S| \rceil}$ and $p_{\lceil \alpha |S| \rceil}.val$ is the α -quantile value of S .

The basic idea of the proposed algorithm is as follows. Firstly, every sensor sorts its points in decreasing order of sensed values, and sends its α -quantile value to its parent. Secondly, the parent chooses one of the received α -quantile values as the filter and broadcasts the filter to all children. Finally the children send those points whose values are no less than the received filter to their parent. The chosen filter is called the *quantile filter*.

In the following we first provide the details of identifying the quantile filter, followed by analyzing the filtering capability of the quantile filter. We then analyze the energy saving by the quantile filter as well as the extra energy overhead on installing it. We finally present a quantile filter-based algorithm for top- k query evaluation and give the analysis of the proposed algorithm.

3.1 The Optimal Quantile Filter

Consider a sensor v in the routing tree \mathcal{T} with d_v children u_i , $1 \leq i \leq d_v$. $L(u_i) = \{p(i)_{j_1}, \dots, p(i)_{l(i)}\}$ is the set of points at u_i with $p(i)_{j_1}.val \geq p(i)_{j_2}.val$ if $j_1 < j_2$, $1 \leq j_1 \leq j_2 \leq l(i)$. $l(i)$ is the number of the points at u_i if $l(i) < k$. Otherwise, $l(i) = k$. In other words, there are at most k points in $L(u_i)$. $S(v) = \bigcup_{i=1}^{d_v} L(u_i)$ is the set of potential

top- k points from the children of v , and $|S(v)| \leq k * d_v$. The α -quantile of $L(u_i)$ is point $p(i)_{\lceil \alpha l(i) \rceil}$ and $p(i)_{\lceil \alpha l(i) \rceil}.val$ is the α -quantile value of $L(u_i)$, which is also referred to the α -quantile value of sensor u_i . The process of identifying a quantile filter that can filter out as many points in $S(v)$ as possible from transmission is described as follows.

Each child u_i sends a pair of values, $p(i)_{\lceil \alpha * l(i) \rceil}.val$, and $l(i)$ to its parent v , $1 \leq i \leq d_v$. The d_v received α -quantile values at sensor v are sorted in decreasing order. Let $q_{i_1}, q_{i_2}, \dots, q_{i_{d_v}}$ be the sorted sequence, where $q_{i_j} (= p(i_j)_{\lceil \alpha * l(i_j) \rceil}.val)$ is the value sent by child u_{i_j} . The m th largest α -quantile value q_{i_m} in the sequence is chosen as *the quantile filter* by sensor v , if an integer m is found such that $\sum_{t=1}^{m-1} \lceil \alpha * l(i_t) \rceil < k$ and $\sum_{t=1}^m \lceil \alpha * l(i_t) \rceil \geq k$, which guarantees that there are at least k points in $S(v)$ whose values are no less than the quantile filter $q_{i_m} = p(i_m)_{\lceil \alpha * l(i_m) \rceil}.val$. Sensor v then broadcasts the found quantile filter to each child u_i , and u_i sends those points in $L(u_i)$ whose values are no less than the quantile filter to parent v , $1 \leq i \leq d_v$. Denote by $Send(u_i)$ the set of points sent by u_i . Having received $Send(u_i)$ from each child u_i , sensor v obtains the top- k points by identifying the k points with the highest sensed values in $\cup_{i=1}^{d_v} Send(u_i) \cup P(v)$.

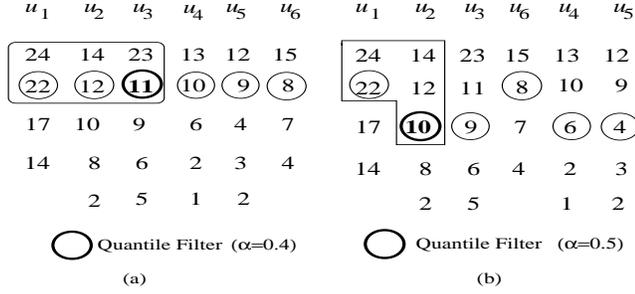


Figure 1: The examples of the quantile filter for top-5 query.

In the following, we use two examples to illustrate the procedure of finding the quantile filter. Assume that sensor v has 6 children u_1, \dots, u_6 , and the values of points at each sensor are sorted as shown in Fig. 1. Assume that a top-5 query is issued at the base station and broadcast to v and its children. Fig. 1(a) and (b) show different quantile filters identified when $\alpha = 0.4$ and $\alpha = 0.5$ respectively. In Fig. 1(a), the 0.4-quantile values of all children are $q_{i_1=1} = 22$, $q_{i_2=2} = 12$, $q_{i_3=3} = 11$, $q_{i_4=4} = 10$, $q_{i_5=5} = 9$, and $q_{i_6=6} = 8$, while their 0.5-quantile values are $q_{i_1=1} = 22$, $q_{i_2=2} = 10$, $q_{i_3=3} = 9$, $q_{i_4=4} = 8$, $q_{i_5=5} = 6$, and $q_{i_6=6} = 4$, which are shown in Fig. 1(b). Because $\lceil 0.4 * l(1) \rceil + \lceil 0.4 * l(2) \rceil + \lceil 0.4 * l(3) \rceil = 6 \geq k$ and $\lceil 0.5 * l(1) \rceil + \lceil 0.5 * l(2) \rceil = 5 \geq k$, $q_{i_3=3} = 11$ and $q_{i_2=2} = 10$ are chosen as the quantile filters in Fig. 1(a) and Fig. 1(b), respectively. From the figures, we can see that the values of the points in the most left top corner regions (circled by the lines) are no less than the quantile filter. Because there are at least k points whose values are no less than the quantile filter, the points with the values

smaller than the quantile filter will be impossible to be part of the final top- k results and therefore are safely filtered out. Compared with algorithm **Naive- k** in which the number of received points by v is $|S(v)|$, the quantile filter can filter out many points in $S(v)$, and thus reduces the energy consumption by transmitting fewer points within the network.

As have been shown, the use of quantile filter can prune some points in $S(v)$ from transmission, but the number of points filtered out depends on the choice of different α s. We now aim to prune as many points from $S(v)$ as possible by identifying the optimal value of α . Consequently, we can reduce the transmission and reception energy consumptions of sensors significantly. What followed is to find such an optimal quantile filter.

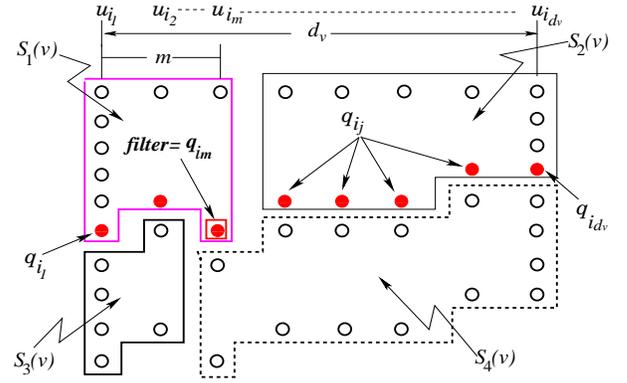


Figure 2: The partition of $S(v)$ by the quantile filter.

Recall that sensor v has d_v children u_1, \dots, u_{d_v} . Define *the shedding ratio* of the quantile filter as the number of points filtered out by the quantile filter to the number of points in $S(v)$. Assume that the points at the d_v children of sensor v are arranged into d_v columns, where $L(u_{i_j})$ occupies column i_j if its α -quantile value is ranked at the i_j position, as shown in Fig. 2. We sort the α -quantile values of the children of v in decreasing order and let $q_{i_1}, \dots, q_{i_{d_v}}$ be the sorted sequence, i.e., $q_{i_x} = p(i_x)_{\lceil \alpha l(i_x) \rceil}.val$ and $q_{i_x} \geq q_{i_{x+1}}$. For convenience, the elements in Fig. 2 represent the values of the corresponding points. Denote by $c_{x,y}$ the element located at the x th column and y th row. All the elements in the x th column are from child u_{i_x} , which are sorted in decreasing order, i.e., $c_{x,y} = p(i_x)_y.val$ and $c_{x,y} \geq c_{x,y+1}$, $1 \leq x \leq d_v$, $1 \leq y \leq l(i_x)$. Suppose that $q_{i_m} = c_{m, \lceil \alpha l(i_m) \rceil}$ is chosen as the quantile filter, where $m = \min\{w \mid \sum_{t=1}^w \lceil \alpha l(i_t) \rceil \geq k, 1 \leq w \leq d_v\}$. From Fig. 2, we can observe that the set of all elements $S(v)$ is divided into 4 subsets by element q_{i_m} and the α -quantile values, i.e., $S_1(v)$, $S_2(v)$, $S_3(v)$ and $S_4(v)$. The values of points in $S_1(v)$ are no less than q_{i_m} , while the values of points in $S_4(v)$ are smaller than q_{i_m} . Because $|S_1(v)| \geq k$, all the points in $S_4(v)$ are impossible to be top- k points. It cannot make sure that the values of the points in $S_2(v)$ and $S_3(v)$ are smaller than the quantile filter, and whether the points in $S_2(v)$ and $S_3(v)$ can be filtered out depends on the data distribution of the sensor network. In comparison with algorithm **Naive- k** , at least

$|S_4(v)|$ points are filtered out from $S(v)$ if sensor v broadcasts the quantile filter to all its children. $\frac{|S_4(v)|}{|S(v)|}$ is thus regarded as the shedding ratio of the proposed quantile filter, where $|S_4(v)| = |S(v)| - |S_1(v)| - |S_2(v)| - |S_3(v)|$. In the following we derive the optimal value of α to maximize $|S_4(v)|$. We thus have the following lemma.

LEMMA 1. $|S_4(v)| \geq (1 - \alpha)|S(v)| - d_v - \frac{1-\alpha}{\alpha}(k-1)$

PROOF. It is obvious that the sizes of $S_1(v)$ and $S_2(v)$ are $|S_1(v)| = \sum_{t=1}^m \lceil \alpha l(i_t) \rceil$ and $|S_2(v)| = \sum_{t=m+1}^{d_v} \lceil \alpha l(i_t) \rceil$, respectively. Therefore, $|S_1(v)| + |S_2(v)| = \sum_{t=1}^{d_v} \lceil \alpha l(i_t) \rceil$.

Since $\sum_{t=1}^{d_v} \lceil \alpha l(i_t) \rceil \leq \sum_{t=1}^{d_v} (\alpha l(i_t) + 1) = \alpha |S(v)| + d_v$, we have

$$|S_1(v)| + |S_2(v)| \leq \alpha |S(v)| + d_v. \quad (1)$$

Denote by $|S_1(v)'| = \sum_{t=1}^{m-1} \lceil \alpha l(i_t) \rceil$. If $|S_1(v)'| \geq k$, $q_{i_{m-1}} = p(i_{m-1})_{\lceil \alpha l(i_{m-1}) \rceil} \cdot val$ instead of $q_{i_m} = p(i_m)_{\lceil \alpha l(i_m) \rceil} \cdot val$ will be chosen as the quantile filter. Thus, $|S_1(v)'| \leq k-1$.

The size of $S_3(v)$ is estimated as follows:

$$\begin{aligned} |S_3(v)| &= \sum_{t=1}^{m-1} \lfloor (1-\alpha)l(i_t) \rfloor \leq \sum_{t=1}^{m-1} \frac{1-\alpha}{\alpha} \lceil \alpha l(i_t) \rceil \\ &= \frac{1-\alpha}{\alpha} |S_1(v)'| \leq \frac{1-\alpha}{\alpha} (k-1), \end{aligned} \quad (2)$$

while $|S_1(v)'| \leq k-1$.

$|S_4(v)| = |S(v)| - |S_1(v)| - |S_2(v)| - |S_3(v)|$. From inequalities (1) and (2), we have

$$|S_4(v)| \geq (1 - \alpha)|S(v)| - d_v - \frac{1-\alpha}{\alpha}(k-1). \quad (3)$$

□

Therefore, the shedding ratio of the quantile filter is

$$\frac{|S_4(v)|}{|S(v)|} \geq 1 - \alpha - \frac{d_v}{|S(v)|} - \frac{(1-\alpha)(k-1)}{\alpha |S(v)|}. \quad (4)$$

From inequality (4), it is clear that the shedding ratio of the quantile filter is determined by the value of α . When $\alpha = \sqrt{\frac{k-1}{|S(v)|}}$, the value of $(1 - \alpha - \frac{d_v}{|S(v)|} - \frac{(1-\alpha)(k-1)}{\alpha |S(v)|})$ in inequality (4) is maximized, and the filter thus is referred to as the *optimal quantile filter*. The shedding ratio of the *optimal quantile filter* is

$$\eta(v) = 1 - \sqrt{\frac{k-1}{|S(v)|}} - \frac{d_v}{|S(v)|} - \left(\sqrt{\frac{|S(v)|}{k-1}} - 1 \right) \left(\frac{k-1}{|S(v)|} \right), \quad (5)$$

where $\eta(v)$ is referred to as the shedding ratio of the optimal quantile filter. The optimal quantile filter can suppress at least $|S_4(v)|$ points from transmission in case the values of the points in $S_2(v)$ and $S_3(v)$ are larger than the quantile filter. In many cases the values of the points in $S_2(v)$ or $S_3(v)$ may be smaller than the quantile filter and those points will be filtered out. Therefore, the actual shedding ratio of the quantile filter is much better than the conservative one based on the theoretical analysis.

3.2 Installation of the Quantile Filter

Although installing quantile filters at the children of v may eliminate unlikely top- k points from transmission, it does

incur extra energy overhead on the quantile filter finding as well as the filter broadcasting. We now analyze this extra energy overhead and energy saving by the filter. In the end we only install the filters at those sensors that the energy saving exceeds the overhead on the filter installation.

For a given sensor v , if no filter is installed at any of its children u_i , u_i sends its top- $l(i)$ points to v as algorithm *Naive-k* does, $1 \leq i \leq d_v$. As a result, the number of points transmitted to v is $|S(v)|$ and the total energy consumption by transmitting and receiving all $|S(v)|$ points is

$$E_{naive}(S(v)) = \rho_t d_v + |S(v)|8R + \rho_r d_v + |S(v)|8r_e. \quad (6)$$

Recall that each point is represented by 8 bytes. Within equation (6), $\rho_t d_v + |S(v)|8R$ is the total energy consumption of the d_v children transmitting the $|S(v)|$ points to sensor v , while $\rho_r d_v + |S(v)|8r_e$ is the total energy consumption of sensor v by receiving the $|S(v)|$ points.

If the quantile filters are installed at the children of v , the shedding ratio induced by them is at least $\eta(v)$, and the total energy consumption is

$$\begin{aligned} E_{filter}(S(v)) &= d_v(\rho_t + 8R + \rho_r + 8r_e) + \rho_t + 4R \\ &\quad + d_v(\rho_r + 4r_e) + d_v(\rho_t + \rho_r) \\ &\quad + (1 - \eta(v))|S(v)|8(R + r_e); \end{aligned} \quad (7)$$

where $d_v(\rho_t + 8R + \rho_r + 8r_e)$ is the total energy consumption of the children sending the α -quantile values and $l(i)$ s to sensor v and sensor v receiving the values from its children, while $\rho_t + 4R + d_v(\rho_r + 4r_e)$ is the energy consumption of broadcasting the quantile filter. $d_v(\rho_t + \rho_r) + (1 - \eta(v))|S(v)|8(R + r_e)$ is the sum of energy consumption that each child sends the points passing through the quantile filter to sensor v . Note that for any leaf sensor v , $E_{naive}(S(v)) = 0$ and $E_{filter}(S(v)) = 0$ because $|S(v)| = 0$.

In the following, we investigate the extra energy overhead on the filter finding and the energy saving of a child of sensor v . If there is no filter installed at a child u_i of v , u_i transmits its top- $l(i)$ points to sensor v and the energy consumption of u_i is

$$E_{naive}(L(u_i)) = \rho_t + l(i) * 8R; \quad (8)$$

otherwise, the energy consumption of u_i is

$$\begin{aligned} E_{filter}(L(u_i)) &= 2\rho_t + 8R + \rho_r + 4r_e \\ &\quad + (1 - \eta(v))l(i)8R. \end{aligned} \quad (9)$$

We assume that $\eta(v) * l(i)$ within equation (9) is the expected number of points filtered out by the quantile filter at sensor u_i . To guarantee that the installation of the quantile filter is beneficial, we must have $E_{filter}(S(v)) < E_{naive}(S(v))$ and $E_{filter}(L(u_i)) < E_{naive}(L(u_i))$. Meanwhile, the shedding ratio of the optimal quantile filter will be no less than 0 only when $|S(v)| \geq \frac{d_v}{1-\alpha} + \frac{k-1}{\alpha}$ from inequality (4). Combined with equations (6), (7), (8), and (9), we have

$$\begin{aligned} |S(v)| &> \theta(v) \\ &= \max \left\{ \frac{2d_v + k - 1}{1 - \alpha} + \frac{(d_v + 1)\rho_t + 2d_v\rho_r + 4(R + d_v r_e)}{8(R + r_e)(1 - \alpha)}, \right. \\ &\quad \left. \frac{d_v}{1 - \alpha} + \frac{k - 1}{\alpha} \right\}, \end{aligned} \quad (10)$$

where $\alpha = \sqrt{\frac{k-1}{|S(v)|}}$, $0 < \alpha < 1$, and

$$l(i) > \frac{\rho_t + 8R + \rho_r + 4r_e}{\eta(v)8R}. \quad (11)$$

The values of $\theta(v)$ in inequality (10) and the shedding ratio $\eta(v)$ in inequality (11) are determined by $|S(v)|$. Note that $|S(v)| = \sum_{i=1}^{d_v} l(i)$, where $l(i) = \min\{k, |L(u_i)|\}$. In practice, the value of $l(i)$ is not known by v beforehand. A naive method for this is to employ an extra phase, in which each child u_i sends its $l(i)$ to sensor v and receives the optimal α from sensor v , this method however consumes additional energy. To this end, an approximation $|S(v)'|$ of $|S(v)|$ is proposed to find the optimal $\alpha' = \sqrt{\frac{k-1}{|S(v)'|}}$ and determine whether the quantile filters should be installed at the children of sensor v . $|S(v)'|$ is the number of points received by sensor v if algorithm Naive- k is applied for top- k query evaluation. Denote by $desc(v)$ the set of descendants of sensor v in the routing tree. As mentioned, each sensor contains one point, and thus $|desc(v)|$ is the number of points contained by the descendants of v . For each sensor v , $|S(v)'| = \sum_{i=1}^{d_v} \min\{k, |desc(u_i)|\}$, where u_i is a child of v . In other words, if a child contains more than k descendants, it transmits its top- k points to v ; otherwise, the child transmits all points from its descendants to v . Obviously, $|S(v)'|$ for each sensor v can be obtained when the routing tree is built. In the following, we prove that the $|S(v)| = |S(v)'|$ by the following theorem.

THEOREM 1. *Given a sensor network $G(V, E)$, $|S(v)| = |S(v)'|$ for each sensor $v \in V$ if the proposed optimal quantile filter strategy is applied in the network.*

PROOF. If v is a leaf sensor, $|S(v)| = |S(v)'| = 0$; otherwise, assume that u_1, u_2, \dots, u_{d_v} are the children of v . We partition the children of v into three subsets, referred to as U_1, U_2 and U_3 , respectively. $U_1 \cup U_2$ is the set of the children of v whose descendants are not installed the filters, where $|desc(u_i)| < k$ if $u_i \in U_1$, and $|desc(u_i)| \geq k$ if $u_i \in U_2$. For each sensor $u_i \in U_1 \cup U_2$, $|S(u_i)| = |S(u_i)'|$ because there is no filter installed at any descendant of u_i and thus algorithm Naive- k is applied on the subtree rooted at u_i . If $u_i \in U_1$, $|S(u_i)| = |S(u_i)'| = desc(u_i)$; otherwise, u_i receives at least k points, and $|L(u_i)| = k$. U_3 is the set of the sensors that at least one descendant is installed a filter. Assume that $u' \in desc(u_i)$ is installed a quantile filter where $u_i \in U_3$. It is clear that $|desc(u_i)| > |desc(u')| > |S(u')| > \theta(u') > k$. Thus, u_i receives at least k points from its descendants, and consequently $|L(u_i)| = k$. For each sensor $v \in V$ we have $|S(v)'| = \sum_{u \in U_1} desc(u) + |U_2| * k + |U_3| * k$, and $|S(v)| = \sum_{u \in U_1} desc(u) + \sum_{u \in U_2} |L(u)| + \sum_{u \in U_3} |L(u)| = \sum_{u \in U_1} desc(u) + |U_2| * k + |U_3| * k = |S(v)'|$. \square

3.3 Algorithm

Having described the quantile filter, we now present the Quantile Filter-based algorithm (algorithm QF for short) as follows.

Suppose that sensor v has d_v children u_1, u_2, \dots, u_{d_v} . Each child u_i sends the number of descendants $|desc(u_i)|$ to

parent v and v broadcasts d_v and the set $\{|desc(u_i)|, |desc(u_2)|, \dots, |desc(u_{d_v})|\}$ to its children once. Having received a top- k query, each child u_i can obtain $|S(v)'| (= |S(v)|)$ and $\alpha = \frac{k-1}{|S(v)'|}$. For a child u_i , if either $0 < \alpha < 1$ or inequality (10) or (11) is not satisfied, u_i sends its top- $l(i)$ points to sensor v directly, and no filter is installed at u_i ; otherwise, a quantile filter will be installed at sensor u_i . u_i first sends the α -quantile value of $L(u_i)$ and $l(i)$ to parent sensor v . Sensor v then sorts the received d_v values and broadcasts the m th largest value as the quantile filter to its children, where $m = \min\{w \mid \sum_{t=1}^w \lceil \alpha l(i_t) \rceil \geq k\}, 1 \leq w \leq d_v$. Only child u_i with $l(i) > \frac{\rho_t + 8R + \rho_r + 4r_e}{\eta(v)8R}$ needs to receive the quantile filter and then sends the points whose values are no less than the filter to parent v . The other children of sensor v have sent their top- $l(i)$ points to v already. The pseudo-code of algorithm QF at each sensor is shown below. Note that each non-leaf sensor serves as both a parent and a child, while a leaf sensor only serves as a child.

Algorithm 1 Child_Sensor ($u_i, k, v, L(u_i)$)

begin

compute $|S(v)'|$ and $\alpha = \sqrt{\frac{k-1}{|S(v)'|}}$;
 $\theta(v) \leftarrow \frac{2d_v + k - 1}{1 - \alpha} + \frac{(d_v + 1)\rho_t + 2d_v\rho_r + 4(R + d_v r_e)}{8(R + r_e)(1 - \alpha)}$;
if $|S(v)'| < \max\{\theta(v), \frac{d_v}{1 - \alpha} + \frac{k-1}{\alpha}\}$
or $l(i) \leq \frac{\rho_t + 8R + \rho_r + 4r_e}{\eta(v)8R}$ **or** $\alpha \geq 1$ **or** $\alpha \leq 0$ **then**
transmits top- $l(i)$ points to v ;

else

sends a pair of values $(p(i)_{\lceil \alpha l(i) \rceil}, val, l(i))$ to v , where $p(i)_{\lceil \alpha l(i) \rceil} \in L(u_i)$ and $l(i) = |L(u_i)|$;
if receive the quantile filter from v **then**
transmits the points whose values are no less than the quantile filter to v ;

end

Algorithm 2 Parent_Sensor ($v, k, P(v)$)

begin

compute $|S(v)'|$ and $\alpha = \sqrt{\frac{k-1}{|S(v)'|}}$;
 $\theta(v) \leftarrow \frac{2d_v + k - 1}{1 - \alpha} + \frac{(d_v + 1)\rho_t + 2d_v\rho_r + 4(R + d_v r_e)}{8(R + r_e)(1 - \alpha)}$;
receive the values from its children;
if $|S(v)'| > \max\{\theta(v), \frac{d_v}{1 - \alpha} + \frac{k-1}{\alpha}\}$ **and** $0 < \alpha < 1$ **then**
the quantile filter q_{i_m} is chosen,
 $m \leftarrow \min\{t \mid \sum_{i=1}^t \lceil \alpha l(i_t) \rceil \geq k\}$;
broadcast the quantile filter to children;
if receive all the points from children **then**
compute top- k points from the received points and v 's own point;

end

An illustrative example for the execution of algorithm QF is given in Fig. (3). Sensor v has 6 children, u_1, \dots, u_6 , and each sensor contains several points as shown in Fig.3(a). A top-10 query is issued at the base station and broadcast to sensor v and its children. $k = 10$ and $|S(v)| = 40$, thus $\alpha = \sqrt{\frac{k-1}{|S(v)|}} = \sqrt{\frac{9}{40}} = 0.47$. Suppose that $\frac{\rho_t + \rho_r + 8R + 4r_e}{6R} = 3$, and consequently sensor u_4 with $l(4) = 3$ will send all of its

points to sensor v . The other sensors send their α -quantile values to sensor v with $\alpha = 0.47$. The circled points in Fig.3(b) are sent to sensor v in the first phase. The six received quantile values by sensor v are sorted in decreasing order and the sorted sequence is $q_{i_1=4}(= 25)$, $q_{i_2=3}(= 21)$, $q_{i_3=6}(= 19)$, $q_{i_4=2}(= 16)$, $q_{i_5=5}(= 11)$, $q_{i_6=1}(= 6)$. The quantile filter $q_{i_3=6} = 19$ is chosen, because $\lceil 0.47 * l(4) \rceil + \lceil 0.47 * l(3) \rceil + \lceil 0.47 * l(6) \rceil = 10 \geq k$. Sensor v broadcasts q_{i_3} to its children and all children except u_4 send the points with values no less than q_{i_3} to v . In Fig.3(b), all the points above the broken line are sent to v in the second phase.

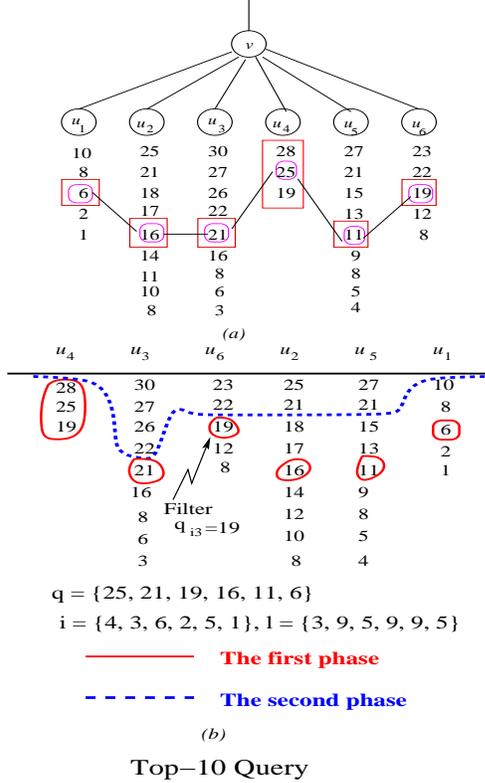


Figure 3: An example of algorithm Quantile-Filter.

3.4 Analysis

In this section we analyze the filtering capability of algorithm QF by comparing the number of transmitted points of algorithms QF and Naive- k .

Recall that u_1, \dots, u_{d_v} are the children of sensor v . If the quantile filters are installed at the children of v , the points in $S_4(v)$ are filtered out, and the number of points sent to v is $(1 - \eta(v))|S(v)| + d_v$, where d_v is the number of pairs of values $\{p(i)_{al(i)}, l(i)\}$ sent by the children of v for quantile filter finding, where $1 \leq i \leq d_v$; otherwise, $|S(v)|$ points are sent to v . We thus partition the set of sensors V into two subsets V_{naive} and V_{filter} , where V_{naive} is the set of sensors whose children are not installed quantile filters, and V_{filter} is the set of sensors whose children are installed quantile filters. Let ζ be the ratio of the number of points

transmitted within the network by algorithm QF to that by algorithm Naive- k , which is

$$\zeta = \frac{\sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{filter}} ((1 - \eta(v))|S(v)| + d_v)}{\sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{filter}} |S(v)|}. \quad (12)$$

As different topologies of routing trees have different sets V_{naive} and V_{filter} , the analysis of the bound of ζ is generally difficult. In the following we thus analyze the bound of ζ on a special routing tree, a complete d -ary tree. Assume that the complete d -ary tree has h layers and includes $\frac{d^h - d}{d - 1}$ sensors. The base station is the root of the routing tree, which is located in the first layer, and the sensors located at the smaller layers are closer to the base station. Each sensor at the i th layer has $\frac{d^{h-i+1} - d}{d - 1}$ descendants. Assume that h_1 is the maximum layer at which the sensors have more than k descendants, i.e., $\frac{d^{h-h_1+1} - d}{d - 1} > k$ and $\frac{d^{h-h_1} - d}{d - 1} < k$. Thus $h_1 = \lfloor h - \log_d(k(d - 1) + d) \rfloor + 1$. In the realistic sensor network, the average number of neighbors of a sensor is a constant. Thus the routing tree can be treated as a complete d -ary tree on average. We have the following theorem.

THEOREM 2. *Given a sensor network $G(V, E)$, assume that the routing tree rooted at the base station is a d -ary tree, which includes all sensors. Algorithms QF and Naive- k are applied for top- k query evaluation. $\zeta(d, h, k)$ is defined as the ratio of the number of points transmitted in this network by algorithm QF to that by algorithm Naive- k . The bound of $\zeta(d, h, k)$ is*

$$\begin{cases} \zeta(d, h, k) = 1, & \text{if } \theta(v) \geq d * k \text{ or } \frac{\rho_t + 8R + \rho_r + 4r_e}{\eta 8R} \geq k. \\ \zeta(d, h, k) \leq \frac{(d^h - d^{h_1})(h - h_1) + (k(1 - \eta) + 1)(d^{h_1} - d)}{(d^h - d^{h_1})(h - h_1) + k(d^{h_1} - d)}, & \text{otherwise.} \end{cases}$$

where $\theta(v)$ is the condition of installation of the quantile filter from inequality (10), h is the number of layers of the routing tree, $h_1 = \lfloor h - \log_d(k(d - 1) + d) \rfloor + 1$, and $\eta = \frac{d+2}{d} - (\frac{\sqrt{k}}{\sqrt{d(k-1)}} + \frac{\sqrt{k-1}}{\sqrt{dk}} + \frac{\sqrt{k-1}}{d\sqrt{dk}})$.

PROOF. The definition of $\zeta(d, h, k)$ follows the previous definition of ζ and it is used to represent a special ζ on the d -ary routing tree of h layers. If algorithm Naive- k is applied for top- k query evaluation, $|S(v)| = d * \frac{d^{h-i} - 1}{d - 1}$ for a sensor v at the i th layer where $i \geq h_1$, because $\frac{d^{h-i} - d}{d - 1} < k$, while $|S(v)| = d * k$ for a sensor v at the layer smaller than h_1 . And there are d^{i-1} sensors at the i th layer of a d -ary tree. Therefore, the number of transmitted points by algorithm Naive- k is

$$\begin{aligned} \sum_{v \in V} |S(v)| &= \sum_{i=h_1}^{h-1} (d^{i-1} * \frac{d^{h-i+1} - d}{d - 1}) + \sum_{i=1}^{h_1-1} (d^i * k) \\ &= \sum_{i=h_1}^{h-1} \frac{d^h - d^i}{d - 1} + \frac{(d^{h_1} - d)k}{d - 1}. \end{aligned} \quad (13)$$

The threshold $\theta(v)$ is used to determine whether the children of sensor v are installed quantile filters, which can be further divided into two cases as follows.

Case 1: If $\theta(v) \geq d * k$ or $\frac{\rho_t + 8R + \rho_r + 4r_e}{\eta 8R} \geq k$, no sensor is installed the filter because each sensor v in a d -ary tree has at most k points in $L(v)$ and $d * k$ points in $S(v)$. Thus, $\zeta(d, h, k) = 1$.

Case 2: If $\theta(v) < d * k$, quantile filters are installed at the children of each sensor v located at the layer smaller than h_1 . For such a sensor, $|S(v)| = dk$, and consequently $\eta(v) = \frac{d+2}{d} - (\frac{\sqrt{k}}{\sqrt{d(k-1)}} + \frac{\sqrt{k-1}}{\sqrt{dk}} + \frac{\sqrt{k-1}}{d\sqrt{dk}})$. The shedding ratio of all sensors with the filters are the same, and η is used to represent the value of this shedding ratio. Thus, the number of transmitted points by algorithm QF is

$$\begin{aligned} & \sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{filter}} ((1 - \eta(v)) |S(v)| + d) \\ = & \sum_{i=h_1}^{h-1} \frac{d^h - d^i}{d-1} + \sum_{i=1}^{h_1-1} (d^{i-1} ((1 - \eta) dk + d)) \\ = & \sum_{i=h_1}^{h-1} \frac{d^h - d^i}{d-1} + \frac{((1-\eta)k+1)(d^{h_1}-d)}{d-1}. \end{aligned} \quad (14)$$

Denote by

$$\begin{aligned} w_1 &= \frac{\sum_{v \in V_{filter}} ((1-\eta(v)) |S(v)| + d)}{\sum_{v \in V_{filter}} (|S(v)|)} = \frac{((1-\eta)k+1)(d^{h_1}-d)}{(d^{h_1}-d)k} \\ &= 1 - \eta + \frac{1}{k}, \end{aligned}$$

and

$$\begin{aligned} w_2 &= \frac{\sum_{v \in V_{filter}} |S(v)|}{\sum_{v \in V_{naive}} |S(v)|} = \frac{(d^{h_1}-d)k}{\sum_{i=h_1}^{h-1} (d^h - d^i)} \\ &\geq \frac{(d^{h_1}-d)k}{\sum_{i=h_1}^{h-1} (d^h - d^{h_1})} = \frac{(d^{h_1}-d)k}{(d^h - d^{h_1})(h-h_1)}, \end{aligned} \quad (15)$$

It is obvious that $0 < w_1 \leq 1$ and $w_2 > 0$. Therefore,

$$\begin{aligned} \zeta(d, h, k) &= \frac{\sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{filter}} ((1-\eta(v)) |S(v)| + d)}{\sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{filter}} |S(v)|} \\ &= \frac{\sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{naive}} |S(v)| w_1 w_2}{\sum_{v \in V_{naive}} |S(v)| + \sum_{v \in V_{naive}} |S(v)| w_2} \\ &= \frac{1+w_1 w_2}{1+w_2} \\ &\leq \frac{(d^h - d^{h_1})(h-h_1) + (k(1-\eta)+1)(d^{h_1}-d)}{(d^h - d^{h_1})(h-h_1) + k(d^{h_1}-d)}, \end{aligned} \quad (16)$$

because the value of $\frac{1+w_1 w_2}{1+w_2}$ decreases, with the increase of w_2 , where $h_1 = \lfloor h - \log_d(k(d-1) + d) \rfloor + 1$, and $\eta = \frac{d+2}{d} - (\frac{\sqrt{k}}{\sqrt{d(k-1)}} + \frac{\sqrt{k-1}}{\sqrt{dk}} + \frac{\sqrt{k-1}}{d\sqrt{dk}})$. \square

4. PERFORMANCE STUDY

In this section we evaluate the performance of the proposed algorithm QF in terms of the total energy consumption, the maximum energy consumption among sensors, the network lifetime, and the average energy consumption of sensors at different layers of the routing tree. The network lifetime is defined as the number of top- k queries answered before the first sensor exhausts its energy. In other words, the network having a longer lifetime can answer more top- k queries. In the following, algorithm QF with the optimal value of α and with fixed $\alpha = 0.3$ and $\alpha = 0.5$ are referred to as algorithm **Optimal-QF**, algorithm **0.3-QF** and algorithm **0.5-QF**, respectively. The performance of algorithm **Naive- k** in [24] will be used as benchmark for comparison purpose.

4.1 Experiment Setting

We assume that the sensor network is deployed to monitor a $100m \times 100m$ region of interest, in which N sensors are randomly deployed by the **NS-2** simulator [19] and the base station is located at the square center. Assume that all the

sensors have the same transmission range (5 meters in this paper). The initial energy at each sensor is 10^5 mJ. The topology of the sensor network is the *TAG* routing tree [1] (a *Breadth-First-Search-tree*). As mentioned, the energy consumption on wireless communication dominates all the other energy consumptions at a sensor, and we thus only take into account the radio energy consumption. In our experiments we adopt the transmission and reception energy consumption parameters of a real sensor MICA2 mote [17], where the energy consumptions on transmitting and receiving a header and handshaking are $\rho_t = 0.4608$ mJ and $\rho_r = 0.1152$ mJ, and the energy consumptions of transmitting and receiving one byte are $R = 0.0144$ mJ and $r_e = 0.00576$ mJ, respectively [17]. The sensed points are drawn from a real dataset of temperature traces, collected by the Intel Berkeley Research Lab from February 28 to April 5, 2004 [18].

4.2 Impact of k on Performance of Algorithms

We first evaluate the performance of various algorithms for top- k query evaluation with the range of k being from 80 to 150. In our experiments, two instances of sensor networks consisting of 1,500 and 2,500 sensors are considered.

Fig. 4 shows the curves of various performance metrics. It can be seen that the total energy consumption and the maximum energy consumption among the sensors by algorithms **Optimal-QF**, **0.3-QF** and **0.5-QF** are substantially less than those by algorithm **Naive- k** . Algorithm **Optimal-QF** outperforms algorithms **0.3-QF** and **0.5-QF** in terms of various performance metrics, which implies that the optimal quantile filter can filter out more points from transmission. The network lifetime by algorithm **Optimal-QF** is 1.25 times, 1.15 times or 1.4 times longer than that by algorithm **0.3-QF**, **0.5-QF** or **Naive- k** on average for both network instances.

Fig. 5 plots the curves of average energy consumption of sensors at different layers of the routing tree by various algorithms in the network instances consisting 1,500 and 2,500 sensors when $k = 80$, $k = 100$, and $k = 120$, respectively. A sensor at the i th layer is i -hop away from the base station and the sensors near to the base station play more important roles in network lifetime prolongation. As shown in Fig. 5, the average energy consumption of sensors at the first 10 layers by algorithms **Optimal-QF**, **0.3-QF** and **0.5-QF** are less than that by algorithm **Naive- k** , while the average energy consumption of sensors in the first 3 layers by algorithm **Optimal-QF** is less than those by algorithms **0.3-QF** and **0.5-QF**. The curves imply that algorithm **Optimal-QF** efficiently balances the load among the sensors in the network, thereby reduces the energy consumption of the sensors near to the base station.

5. CONCLUSIONS

In this paper we have tackled the problem of processing top- k query in wireless sensor networks with objective to maximize the network lifetime. We proposed a filter-based localized algorithm for top- k query evaluation through the design of a novel filter which is capable to filter out as many

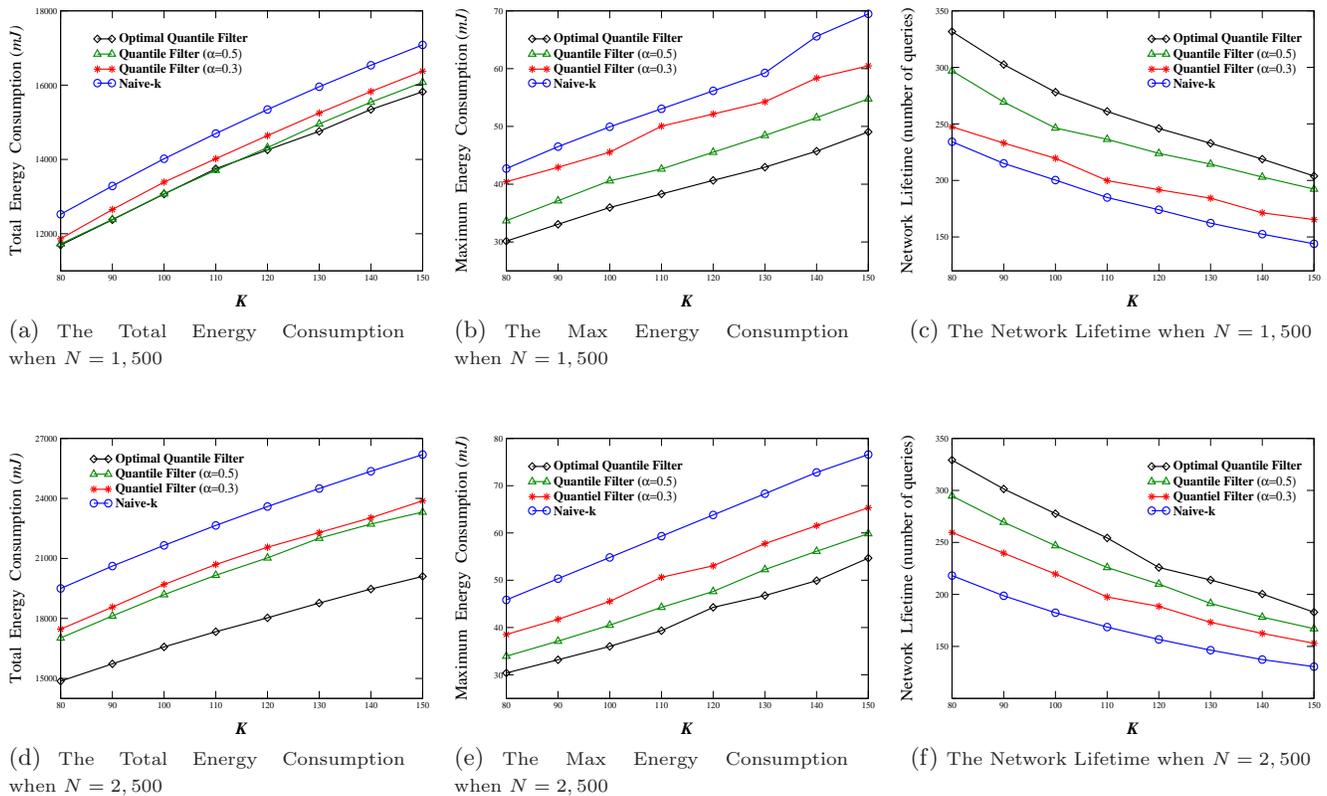
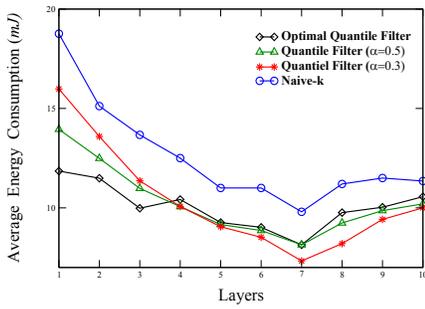


Figure 4: Performance of various algorithms for top- k query evaluation with $80 \leq k \leq 150$.

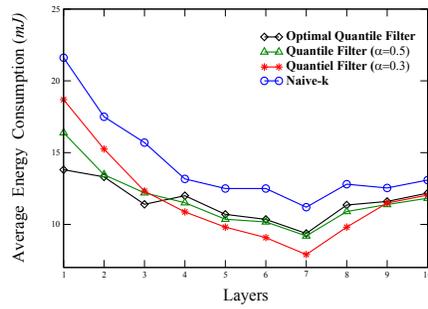
unlikely top- k points as possible within the network from transmission, and we analyzed the filtering capability of the proposed algorithm. We conducted extensive experiments by simulations on real sensing datasets to evaluate the performance of the proposed algorithm. The experimental results show that the proposed algorithm is efficient and can prolong the network lifetime significantly.

6. REFERENCES

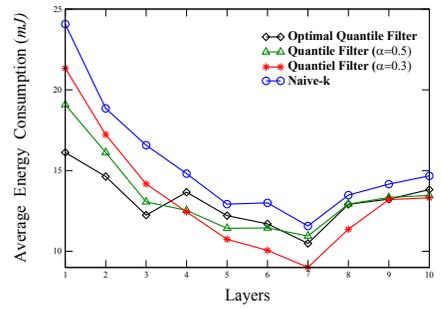
- [1] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong. TAG: a tiny aggregation service for ad hoc sensor networks. *ACM SIGOPS Operating Systems Review*, Vol.36, pp.131–146, 2002.
- [2] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong. The design of an acquisitional query processor for sensor networks. *Proc. ACM SIGMOD*, pp.491–502, 2003.
- [3] M. B. Greenwald, S. Khanna. Power-Conserving Computation of Order-Statistics over Sensor Networks. *Proc of ACM PODS*, ACM, pp.275–285, 2004.
- [4] N. Shrivastava, C. Buragohain, D. Agrawal, S. Suri. Medians and beyond: new aggregation techniques for sensor networks. *Proc of ACM SenSys*, ACM, pp.239–249, 2004.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, pp.102–114, 2002.
- [6] B. Babcock and C. Olston. Distributed top- k monitoring. *Proc. of ACM SIGMOD*, ACM, pp.28–39, 2003.
- [7] I. F. Ilyas, G. Beskales, M. A. Soliman. A survey of top- k query processing techniques in relational database systems. *ACM Computing Survey*, Vol.40, issue 4, 2008.
- [8] M. Wu, J. Xu, X. Tang, W-C. Lee. Monitoring top- k query in wireless sensor networks. *Proc. of ICDE*, IEEE, 2006.
- [9] M. Wu, J. Xu, X. Tang, W-C. Lee. Top- k monitoring in wireless sensor networks. *IEEE Trans. Knowledge and Data Engineering*, Vol.19, No.7, July, pp.962–976, 2007.
- [10] R. Fagin, A. Lotem, M. Naor. Optimal aggregation algorithms for middleware. *JCSS*, Vol.1, pp.614–656, 2001.
- [11] A. Marian, L. Gravano, N. Bruno. Evaluating top- k queries over Web-accessible databases. *ACM Trans. Database Systems*, Vol.29, issue.2, pp.319–362, 2004.
- [12] J-H Chang and L. Tassiulas. Energy conserving routing in wireless ad hoc networks. *Proc. INFOCOM'00*, IEEE, 2000.
- [13] G. Das, D. Gunopulos, N. Koudas, N. Sarkas. Ad-hoc



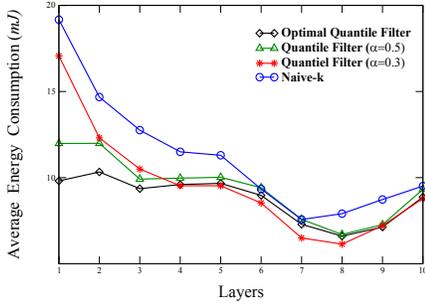
(a) The Average Energy Consumption when $N = 1,500$, $k = 80$



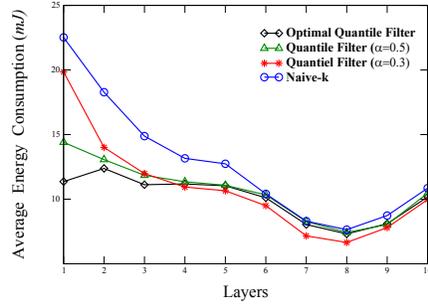
(b) The Average Energy Consumption when $N = 1,500$, $k = 100$



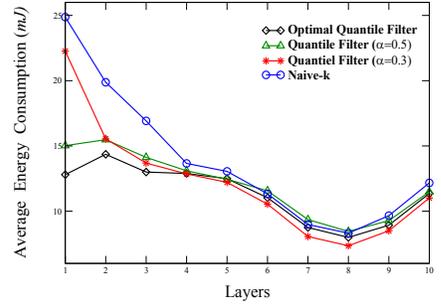
(c) The Average Energy Consumption when $N = 1,500$, $k = 120$



(d) The Average Energy Consumption when $N = 2,500$, $k = 80$



(e) The Average Energy Consumption when $N = 2,500$, $k = 100$



(f) The Average Energy Consumption when $N = 2,500$, $k = 120$

Figure 5: The average energy consumption of sensors in different layers.

- top- k query answering for data streams. *Proc of VLDB*, 2007.
- [14] D. Zeinalipour-Yaztia, Z. Vagenab, V. Kalogerakic, D. Gunopulos, V. J. Tsotrase, M. Vlachosf, N. Koudas, D. Srivastava. Finding the K highest-ranked answers in a distributed network. *The International Journal of Computer and Telecommunications Networking*, volume 53, Issue 9, pp.1431–1449, 2009.
- [15] D. Zeinalipour-Yaztia, Z. Vagenab, V. Kalogerakic, D. Gunopulos, V. J. Tsotrase, M. Vlachosf, N. Koudas, D. Srivastava. The threshold join algorithm for top- k queries in distributed sensor networks. *Proc of DMSN*, 2005.
- [16] G. J. Pottie, W. J. Kaiser. Wireless Integrated Network Sensors. *Communication of ACM*, Vol.43 No.5, pp.51–58, 2000.
- [17] Crossbow Inc. *MPR-Mote Processor Radio Board Users Manual*.
- [18] <http://db.csail.mit.edu/labdata/labdata.html>
- [19] Network Simulator-ns2. <http://www.isi.edu/nsnam/ns>, 2006.
- [20] Y. Yao, J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, Vol. 31, pp.9–18, 2002.
- [21] Y. Yao, J. Gehrke. Query processing for sensor networks. *Proc of CIDR*, Jan., 2003.
- [22] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson. Wireless sensor networks for habitat monitoring. *Proc. of Int'l Workshop on Wireless Sensor Networks and Applications*, ACM, 2002.
- [23] W. Liang, B. Chen, J. X. Yu. Energy-efficient skyline query processing and maintenance in sensor networks. *Proc of ACM CIKM*, pp.1471–1472, 2008.
- [24] A. Silberstein, R. Braynard, C. Ellis, K. Munagala, J. Yang. A sampling-based approach to optimizing top- k queries in sensor networks. *Proc. of ICDE*, IEEE, pp.68–79, 2006.