2008 14th IEEE International Conference on Parallel and Distributed Systems

Response Time Constrained Top-k Query Evaluation in Sensor Networks

Weifa Liang
Dept of Computer Sci.
Australian National University
Canberra, ACT 0200, Australia

Baichen Chen
Dept of Computer Sci.
Australian National University
Canberra, ACT 0200, Australia

Jeffrey Y. Xu
Dept. of Sys. Eng. & Eng.
Chinese Uni. of Hong Kong
Shatin, NT, Hong Kong

Abstract

Existing solutions for top-k queries in wireless sensor networks mainly focused on energy efficiency and little attention has been paid to the response time to answer a top-k query as well as the relationship between the response time and the network lifetime. In this paper we address this issue explicitly by studying the top-k query problem in sensor networks with the response time constraint. We aim at finding an energy-efficient routing tree and devising an evaluation algorithm for top-k queries on the tree such that the network lifetime is significantly prolonged, provided that the query response time constraint is met too. To do so, we propose a novel joint optimization framework of finding a routing tree and devising a filter-based evaluation algorithm on the tree. We also conduct extensive experiments by simulation to evaluate the performance of the proposed algorithms. The experimental results showed that the joint optimization framework prolongs the network lifetime significantly under a given response time constraint.

1 Introduction

Query optimization in traditional databases has been extensively studied, and typical optimization metrics for queries are the query response time and space. However, in wireless sensor networks query optimization typically focused on energy efficiency in order to prolong the network lifetime. Since energy conservation has dominated most of the research in energy-constrained sensor networks, the concepts of query response time, end-to-end delay and jitters are not primary concerns in most of published works. However, the increasing interest in real-time applications such as disaster management, combat field and security surveillance, along with the introduction of imaging and video sensors has posed additional challenges. Our optimization objective is not only to prolong the network lifetime but also to meet a certain end-to-end delay constraint (e.g. the query response time).

Top-k query is one of the popular queries in sensor networks, which is to find the k nodes with the highest readings among the sensor nodes. Although several studies on top-k query evaluation and maintenance in sensor networks have been conducted recently [9, 10], none of them incorporates the query response time constraint into consideration while focusing on energy efficiency. However, in some real-time applications, the query response time is very crucial. For example, consider a sensor network used to monitor forest fires. When a forest fire happens, the forest management authority may issue a top-k query to request the vicinity images of the k sensors with the highest temperature readings in the forest. Such a query has a stringent query response time imposed, because timing is very crucial for fire fighters to distinguish the fires.

1.1 Related work

Typically, there are two types of top-k queries. One is the distributed top-k query which aims to find the k highest ranked objects, where the ranking score of an object is an aggregated value from a number of attribute values stored at distributed sources,such as TA [4], TPUT [1], etc. Another is to find the k nodes with the highest readings in a sensor network, assuming that each node has a reading. For this latter one, Wu et al [10] exploited the semantics of top-k query and proposed a novel Filter-based algorithm (FILA) for monitoring the top-k results. The energy savings delivered by their solution however is based on an assumption that each sensor is within the transmission range of the base station, each updating probe broadcast from the base station can be heard by all the sensors, and such probing energy consumption was not taken into account. In real life, this assumption may be too restrictive, the total reception energy consumption on all sensors by receiving the base station's probing message cannot be ignored, because the reception energy consumption of a sensor is usually about one third of its transmission energy consumption in most short-distance wireless communication. For example, the reception energy consumption on MICA2 mote is 14.4 mJ/sec, but its

1521-9097/08 $25.00 © 2008 IEEE
DOI 10.1109/ICPADS.2008.65

transmission energy consumption is only 36 mJ/sec [3, 9]. Therefore, if the sensing readings among the sensors are frequently updated, the probing cost will become prohibitively expensive. Silberstein et al [9] considered the top-k query problem in sensor networks by providing approximate solutions, based on top-k samples of the past readings.

A closely related problem is data gathering with end-to-end delay constraint in sensor networks and the top-k query with response time constraint is a special case of this general setting. Despite that there are several studies on data gathering that tradeoff the time delay and the network lifetime [6, 11], they are either inapplicable or have their limitations on this special case, because they assume that either message length is given in prior or the data transmission rate at each individual node is dynamically adjustable. Lindsey et al [6] proposed an optimization metric for data gathering and proposed a heuristic algorithm only for dense network, instead of finding a chain-based routing tree which is intractable in most sensor network. Yu et al [11] considered data gathering by providing an optimal schedule for packets to meet the end-to-end delay constraint. They also devised an off-line central optimal algorithm with delay and packet length given prior and on-line approximation algorithm based on dynamic programming.

1.2 Contributions

In this paper, we study the response time constrained top-k query problem in sensor networks. We first address the query response time and its effect on the network lifetime explicitly. We then propose a novel joint optimization framework that consists of finding a routing tree in the network and devising an evaluation algorithm for the top-k query evaluation on the tree in an efficient manner to meet the query response time constraint. We finally conduct extensive experiments by simulation to evaluate the performance of the proposed optimization framework for top-k queries. The experimental results show that there is a non-trivial tradeoff between the query response time and the network lifetime, and the approach based on the joint optimization framework can prolong the network lifetime significantly under the given query response time constraint.

The remainder of the paper is organized as follows. In Section 2, the sensor network model and the problem definition are introduced. In Section 3, the cost model of energy consumption and the time delay by transmitting a message is proposed, followed by the energy consumption model and the response time to answer a top-k query. In Section 4, a simple query optimization framework is proposed. In Section 5, a more efficient query optimization framework is proposed, which incorporates filtering on nodes into the design of evaluation algorithms to filter out unnecessary data. In Section 6, extensive experiments by simulation are conducted to evaluate the performance of the proposed algorithms against the other algorithms. The conclusions are given in Section 7.

2 Preliminaries

2.1 System model

We consider a sensor network consisting of n sensors randomly deployed in a region of interest, each measuring a numeric value. Assume that there is a base station with unlimited energy supply, which serves as a gateway between the sensor network and users. On the other hand, the battery-powered sensors are responsible for sensing and collecting the sensing data. They are also capable of processing sensed data and transmitting aggregated data to their neighbors. Every sensor has identical transmission range. Two sensors are *neighbors* if they are within the transmission range of each other. The energy consumptions of a sensor by transmitting and receiving a one-byte data are R mJ and r_e mJ, respectively. A sensor network can be represented by an undirected graph $G(V, E)$, where V is the set of sensor nodes and the base station, E is the set of links. We assume that a single communication channel is shared by all the sensor nodes.

2.2 Problem definition

Given a sensor network $G(V, E)$, assume that each sensor has a reading val_i, a top-k query is to find the k nodes with the highest readings in the network, $1 \leq i, k \leq n$. *The top-k query problem with response time constraint* in G is defined as follows. Given a top-k query issued at the base station and a specified query response time bound Γ, the problem is to evaluate the query in-network by providing an evaluation plan such that the network lifetime is maximized, subject to the query response time constraint Γ to be met, assuming that a routing tree rooted at the base station is used for query evaluation, where the *network lifetime* is referred to as the failure time of the first node in the network [2]. The rationale behind the network lifetime definition is the imbalance of energy consumption among the nodes. The nodes near to the base station consume much more energy than the other nodes, thus they often die first. Once they are dead, the base station will be disconnected from the rest of the nodes in the network, no matter how much residual energy left and how well connected the rest of the nodes are.

A simple algorithm `Naive-`k, without response time constraint for top-k query, is as follows: it traverses the routing tree from the bottom-up fashion within one pass. Each node simply transmits its top-k results to its parent.

Algorithm Naive-k will be used as benchmark in our experiments.

3 Energy and Query Response Time

3.1 Energy consumption and time delay of a message transfer

In sensor networks, we assume that the basic communication mechanism is messages. Two neighboring nodes can send messages to each other. A message consists of a header and a body which contains from zero to multiple sensing readings. We refer to *the energy overhead per message* as the energy overhead on both handshaking and the message header transmission. A reading of a node represented by l bytes is composed of the node's ID and the sensing value. We convert the energy consumption overhead per message into the equivalent amounts of energy on transmission of ρ readings by the formula $\mathcal{E}_H = (\rho * l) * \mathcal{E}_{byte}$, where $\mathcal{E}_{byte}$ is the energy consumption on transmitting a single byte data and $\mathcal{E}_H$ is the energy overhead per message. And we set $l_H = \rho * l$.

The time delay of transmission is dominated by the *transmission time* as there is no queuing in the network. The processing and propagation delays are negligible compared to the transmission time delay. Let t_H be the time spent for message header which consists of the time spent on handshaking and header transmission. If a message body contains k readings, its transmission time is kl/s, assuming that all sensors have an identical, fixed data transmission rate s and the transmission time of a message containing k readings is $t_H + kl/s$.

3.2 Energy consumption of a top-k query

Suppose a routing tree $\mathcal{T}$ rooted at the base station and spanning all sensor nodes will be used for query evaluation. Let $parent(v)$ denote the parent of node v and $C(v)$ the children set of v in $\mathcal{T}$. There is an algorithm $\mathcal{A}$ for top-k query on tree $\mathcal{T}$ that involves M messages and N readings contained in the M messages. The total energy consumption $\mathcal{E}_{total}(\mathcal{A}, \mathcal{T}) = (M * l_H + N_l)(R + r_e)$. Meanwhile, for a given node v with d_v children, we assume that the number of messages sent and received by v are $M_{send}(v)$ and $M_{rece}(v)$ and the number of readings in corresponding messages are $N_{send}(v)$ and $N_{rece}(v)$. The maximum energy consumption among the nodes by algorithm $\mathcal{A}$ on tree $\mathcal{T}$ is $\mathcal{E}_{max} = \max_{v \in V}((M_{send(v)} l_H + N_{send}(v)l)R + (M_{rece}(v) l_H + N_{send}(v)l) r_e)$.

To prolong the network lifetime, it is required to find a routing tree $\mathcal{T}$ and to devise an algorithm $\mathcal{A}$ on $\mathcal{T}$ such that both $\mathcal{E}_{total}(\mathcal{A}, \mathcal{T})$ and $\mathcal{E}_{max}(\mathcal{A}, \mathcal{T})$ are minimized. However, finding such a tree and an algorithm are difficult, since they are conflicting with each other. We thus focus on finding a feasible routing tree and an evaluation algorithm that tradeoff the two optimization objectives nicely.

3.3 The response time of a top-k query

Given a routing tree $\mathcal{T}$ rooted at the base station r, assume that a node v has d_v children $u_1, u_2, \ldots u_{d_v}$. Recall that each time only one child can communicate with its parent because of the use of a single communication channel. We further assume that an algorithm $\mathcal{A}$ for top-k query on $\mathcal{T}$ from bottom toward up fashion. An internal node v in $\mathcal{T}$ starts sending its top-k readings to its parent only if (i) it has received all necessary readings from all of its children, and (ii) its children can communicate with it as many rounds as needed before it starts to communicate with its parent. Thus, the time delay at node v is defined as the moment it gets all the necessary data from all its children and just starts its own data transmission to its parent. More precisely, *the response time* $t(r)$ to answer a top-k query at base station r can be defined recursively as follows.

Let $t(u)$ be the time delay at node u. Assume that the d_v children of node v, $u_1, u_2, \ldots, u_{d_v}$, are indexed in increasing order of their time delays, i.e., $t(u_1) \leq t(u_2) \leq \ldots \leq t(u_{d_v})$. Then, the earliest time that v received all top-k readings from its children is

$$t(v) = \begin{cases} 0, v \text{ is a leaf} \\ \max_{1 \leq i \leq d_v}\{(t(u_i) + \sum_{j=i}^{d_v} tr(u_j, M_j)) + tp(v)\}, \end{cases} \quad (1)$$

where $tr(u_j, M_j)$ is the transmission time used to transmit M_j messages from u_j to v with each message containing no more than k readings, $1 \leq j \leq d_v$, and $tp(v)$ is the processing time at v to identify the top-k readings. Obviously, $tp(v)$ is negligible compared with the transmission delay, we thus set $tp(v) = 0$. The response time to a top-k query thus is $t(r)$.

4 A Simple Joint Optimization Framework

In this section we propose a simple joint optimization framework for top-k query evaluation, which takes into account both the energy optimization and the response time constraint simultaneously. That is, we aim to find a routing tree such that the network lifetime derived by applying algorithm NAIVE-k on the tree is maximized. Meanwhile, the response time constraint is met as well.

4.1 Response time vs network lifetime

To motivate our discussion, we consider two extreme routing trees, one is the maximum degree-constrained spanning tree T_{MDST} whose maximum degree is minimized,

and the other is the Breadth-First-Search tree T_{BFS} rooted at the base station. Let Δ_{MDST} and Δ_{BFS} be the maximum node degrees of T_{MDST} and T_{BFS}, respectively. We examine the relationship between the query response time and the network lifetime when applying NAIVE-k on both trees. It can be seen that the depth of T_{MDST} is much deeper than that of T_{BFS}, while the depth of T_{BFS} is the lowest among all spanning trees rooted at the base station. This implies that the response time to answer a top-k query by applying an algorithm on T_{BFS} usually is much shorter than that by applying the same algorithm on T_{MDST}, because the former explores more parallelism of message transmissions among the nodes, while the latter is inherently sequential in the worst scenario. On the other hand, the maximum energy consumption among the nodes on T_{BFS} is significantly larger than that on T_{MDST}, because the maximum energy consumption among the nodes is fully determined by its maximum node degree. Since $\Delta_{BFS} > \Delta_{MDST}$, which implies the network lifetime delivered by T_{MDST} is substantially longer than that by tree T_{BFS}. As there is a non-trivial tradeoff between the query response time and the network lifetime, it is crucial to find an appropriate routing tree for top-k queries that not only prolong the network lifetime significantly but also meet the response time constraint. Finding a MDST is NP-complete and we use a optimal approximation algorithm instead [5], which delivers a routing tree in which the maximum node degree is $\Delta^* + 1$, where Δ^* is the maximum node degree optimally.

4.2 Algorithm for finding routing trees

We use algorithm `NAIVE-`k as the evaluation algorithm and tree T_{MDST} as the initial routing tree, which has the minimum maximum node degree among all the spanning trees. The response time to a top-k query obtained by using this tree however is prohibitively slow due to the fact that it is a skinny tree with deep depth. To meet the response time constraint, we modify the current routing tree by increasing the maximum node degree of these nodes with no less than k descendants by one and updating non-child descendants of a node into the children of the node if possible. As a result, the depth of the resulting tree is lowered and the query response time may be shortened. On the other hand, the maximum energy consumption among the nodes in the resulting tree increases and the network lifetime becomes shorter. The procedure of modifying the current routing tree continues until the response time constraint is met. The proposed strategy aims to meet the response time constraint gradually at the expense of a small fraction of network lifetime. Specifically, let UE be an upper bound on the maximum energy consumption among the nodes in the current routing tree. Let Δ be the maximum degree of nodes with no less than k descendants in the tree. It is easy to see that such a node with maximum degree Δ will consume the maximum amounts of energy among the nodes if its every child contains the top-k readings in the subtree rooted at the child. Then $UE = (l_H + kl)R + \Delta(l_H + kl)r_e = (\rho + k)lR + \Delta(\rho + k)lr_e$, where $(l_H + kl)R$ is the transmission energy consumption of the node and $\Delta(l_H + kl)r_e$ is its reception energy consumption.

Suppose that the current routing tree does not meet the query response time constraint, we modify the tree iteratively by increasing the node degrees and the number of descendants of some nodes to reduce the depth of the resulting tree until the resulting tree meets the response time constraint.We distinguish the nodes in the routing tree by two cases. Case (i) for a node v with less than k descendants, if a descendant of v is not its child but within its transmission range and the total energy consumption at v by adding the descendant as its child is not greater than UE, the descendant becomes a child of v. Case (ii) for a node v with no less than k descendants, if its grandparent is within its transmission range and the degree of the grandparent is strictly less than the maximum node degree Δ, node v sets itself as a child of its grandparent. After every iteration of modification, we compute the response time of applying the `Naive-`k on the modified tree. If node v is a leaf node, the time delay of the v, $t(v)$ is zero. Otherwise, $t(v) = \max_{1 \le i \le d_v}\{(t(u_i) + \sum_{j=i}^{d_v} tr(u_j, M_j))$, where u_i is the child node of v. At the end we obtain the time delay of root r which is the query response time. The modification of the tree continues until the given constraint is met.

Having the routing tree, the actual response time of applying `NAIVE-`k on the tree is no greater than the constraint and the network lifetime delivered by applying `NAIVE-`k on the routing tree is greatly prolonged in comparison with that on the T_{BFS}, which is verified by later experiments.

5 An Efficient Joint Optimal Framework

In this section we propose an energy-efficient joint optimization framework for top-k query evaluation with response time constraint, by finding a routing tree and devising a filter-based algorithm for top-k query evaluations on the tree. The filters installed at individual nodes aim to filter out unnecessary data within the network from transmission to reduce data traffic in the network, thereby reducing the query response time and prolonging the network lifetime.

5.1 Top-k query filtering algorithm

Suppose that each child u_i of node v holds the top-l_i readings in the subtree of $\mathcal{T}$ rooted at u_i, which are sorted in decreasing order and denote by $L(u_i) = \{s_{i,1}, s_{i,2}, \ldots, s_{i,l_i}\}$ the set of the top-l_i readings, $s_{i,j} \ge$

$s_{i,j'}$ if $j < j'$, $1 \leq i, j, j' \leq d_v$, $3 \leq l_i \leq k$. The top-k readings in the subtree rooted at v are in the set $S(v) = \bigcup_{u_i \in C(v)} L(u_i)$ and v's own reading, and the cardinality of $S(v)$ is computed as follows.

Assume that the number of descendants $desc(v)$ of v is given, then (i) if $desc(v) < k$, $|S(v)| = desc(v)$; (ii) otherwise, $|S(v)| = \sum_{i=1}^{d_v} \min\{k, desc(u_i) | \, u_i \, is \, a \, child \, of \, v\}$.

The proposed filtering algorithm is described as follows. Each child u_i first sends the median $s_{i,\lceil l_i/2 \rceil}$ of its top-l_i readings to its parent v, $1 \leq i \leq d_v$. After received the median sequence consisting of d_v median readings, node v then sorts the sequence in decreasing order. Let $m_{i_1}, m_{i_2}, \ldots, m_{i_{d_v}}$ be the sorted sequence, $1 \leq i_j, j \leq d_v$. A filter $x = m_{i_j}$ is chosen such that $|R(x)| = \sum_{p=1}^{j} \lceil l_{i_p}/2 \rceil \geq k$, where j is the smallest integer with $1 \leq j \leq d_v$, and such a j must exist due to $|S(v)| \geq 2k$. Third, node v broadcasts x to all its children. Each child u_i finds a smallest reading $s_{i,l_i'}$ in $L(u_i)$ such that $s_{i,l_i'} \geq x$ and transmits its top-l_i' readings except its median to v. Node v finally identifies the top-k readings in the subtree rooted at itself, using the received readings from its children and its own readings. We refer to this algorithm as `FilterA`, which has the following properties.

Lemma 1 *If there is a node v in $\mathcal{T}$ with $|S(v)| \geq 2k$, then there must be a filter $x \in S(v)$ from the median sequence of its children such that $|R(x)| \geq k$. Let $R'(x)$ consist of the elements in the lower half of $L(u_{i_t})$ of node u_{i_t} that are not in $R(x)$ with $1 \leq t < j$, assuming that $x = m_{i_j}$. Then, $|R'(x)| \leq k-1$ and $|S'(v)| \leq |S(v)|/2 + d_v + |R'(x)|$.*

Proof }□ **1** *Let $m_{i_1}, m_{i_2}, \ldots, m_{i_{d_v}}$ be the sorted median sequence and $x = m_{i_j}$. We show that such a j must exist with $1 \leq j \leq d_v$. Otherwise, $\sum_{t=1}^{d_v} \lceil l_{i_t}/2 \rceil < k$. However, given $|S(v) \geq 2k$, we have $\sum_{t=1}^{d_v} \lceil l_{i_t}/2 \rceil \geq \sum_{t=1}^{d_v} l_{i_t}/2 = |S(v)|/2 \geq k$. This results in a contradiction. Thus, $\sum_{t=1}^{d_v} \lceil l_{i_t}/2 \rceil \geq k$.*

Following the definition of x, j is the minimum integer such that $\sum_{t=1}^{j} \lceil l_{i_t}/2 \rceil \geq k$, we have $\sum_{t=1}^{j-1} \lceil l_{i_t}/2 \rceil \leq k-1$. Consequently, $|R'(x)| = \sum_{t=1}^{j-1} \lfloor l_{i_t}/2 \rfloor \leq \sum_{t=1}^{j-1} \lceil l_{i_t}/2 \rceil < k \leq k-1$. Following `FilterA`, for a child u_{i_t} with $t < j$, it transmits no more than its top-l_{i_t} readings except $s_{i_t,\lceil l_{i_t}/2 \rceil}$ to v, for the child indexed with i_j, it transmits its top-$\lfloor l_{i_j}/2 \rfloor$ readings except x to v, while for a child u_{i_t} with $t > j$, it transmits no more than its top-$\lfloor l_{i_t}/2 \rfloor$ readings except $s_{i_t,\lceil l_{i_t}/2 \rceil}$ to v. Then, $|S'(v)| \leq \sum_{t=1}^{j-1} l_{i_t} + \lceil l_{i_j}/2 \rceil + \sum_{t=j+1}^{d_v} \lceil l_{i_t}/2 \rceil \leq \sum_{t=1}^{d_v} \lceil l_{i_t}/2 \rceil + |R'(x)| \leq |S(v)|/2 + d_v + k - 1$.

It is easy to verify that the top-k readings in $S(v)$ are in $S'(v) = S(v) - S_x$, since $|R(x)| \geq k$, $R(x) \subseteq S'(v)$, and every element in S_x is no greater than any element in $R(x)$. Accordingly, almost a half number of elements in $S(v)$ are filtered out, with small extra cost by transmitting the d_v median readings of the children to v and performing a broadcast from v to its children, when x is the filter at v.

5.2 Finding a routing tree with response time constraint

We now analyze the energy consumption and time delay at a node v on which with and without filtering is applied. Assume that the time delay $t(u_i)$ at u_i is given already, $1 \leq i \leq d_v$. Without loss of generality, we assume that the children of v, $u_1, u_2, \ldots, u_{d_v}$ are sorted in increasing order of their time delays, i.e., $t(u_1) \leq t(u_2) \leq \ldots \leq t(u_{d_v})$.

If no filtering is applied on node v, then the total energy consumption of v by receiving all readings in $S(v)$ from its children is $E_{nofilt}(v) = (d_v l_H + |S(v)|l)r_e$, the time delay at v associated with the reception of these readings is

$$t_{nofilt}(v) = \max_{1 \leq i \leq d_v} \{t(u_i) + \sum_{j=i}^{d_v} (t_H + \min\{k, l_j\} l/s)\}, \quad (2)$$

where child u_j contains the top-l_j readings in the subtree rooted at itself, $1 \leq j \leq d_v$.

Otherwise, the total energy consumption of v by receiving the readings in $S'(v)$ and filtering is $E_{filt}(v) = 2d_v l_H r_e + (|S(v)|/2 + d_v + |R'(x)|)lr_e + (l_H + l)R$, and the time delay at v associated with the reception of all readings in $S'(v)$ from its children and on filtering is

$$t_{filt}(v) = \max_{1 \leq i \leq d_v} \{t(u_i) + (d_v - i + 1)(t_H + l/s) + t_H + (d_v t_H + (|S(v)|/2 + d_v + k)l/s)\}, \quad (3)$$

since $|S'(v)| \leq |S(v)|/2 + d_v + |R'(x)| \leq |S(v)|/2 + d_v + k$, by Lemma 1. Denote by $\delta E(v)$ and $\delta t(v)$ the differences of energy consumption and time delay between no filtering and filtering on v, when receiving the readings from $S(v)$ and $S'(v)$ respectively, then $\delta E(v) = E_{nofilt}(v) - E_{filt}(v) = (|S(v)|/2 - d_v(\rho + 1) - |R'(x)| - (\rho + 1)R/r_e)lr_e$ and $\delta t(v) = t_{nofilt}(v) - t_{filt}(v)$. To ensure that filtering on node v results in the energy savings, $\delta E(v) \geq 0$. So that we have the following inequality which is the condition of setting the filter on node v.

$$|S(v)| \geq 2d_v(\rho + 1) + 2(k-1) + 2(\rho + 1)R/r_e \quad (4)$$

Although filtering on v guarantees the energy savings when inequality (4) is satisfied, this does not mean that it also guarantees reducing the time delay at v. Therefore, filtering on a node is performed only when it results in both energy saving and the time delay reduction at the node.

Following the above analysis, an algorithm combined with the filter for finding a routing tree `Filter Routing Tree`, meeting the response time constraint is derived

through a minor modification to the algorithm of finding the routing tree by applying Naive-k, that is, after each round of modification, we compute the query response time by using the equation (2) and equation(3) rather than equation (1). For each node v, we computes the $t_{nofilt}(v)$, $t_{filt}(v)$ and $|S(v)|$. If $t_{nofilt} \geq t_{filt}$ and $|S(v)| \geq 2d_v(\rho+1)+2(k-1)+2(\rho+1)R/r_e$, we install the filter on v and the time delay of node v, $t(v) = t_{filt}(v)$. Otherwise, $t(v) = t_{nofilt}(v)$, which means there is no filter installed at v. At the end, the query response time $t(r)$ is obtained at root r. The modification on the tree continues until $t(r) \leq \Gamma$, where Γ is the given time constraint.

5.3 Query evaluation algorithm with response time constraint

Having the routing tree $\mathcal{T}$, algorithm FilterA for the top-k query problem is proposed, which consists of three subroutines Initialization, Child_Node, and Parent_Node as follows.

Algorithm 1 $Initialization(G, r, k)$

1: Construct $\mathcal{T}$ in G by call Find_Routing_Tree_Filter ($\mathcal{T}$, k, r, Γ);
2: Compute the numbers of descendants $desc(v)$ and children d_v of each node v;
3: Each child u of node v sends its $desc(u)$ value to v;
4: each parent node v broadcasts $desc(u)$ of each child u, d_v,and $filter(v)$ to all its children;

Note that the initialization is only done once. Upon its execution, each node u_i knows the number of siblings d_v it has, the number of descendants $desc(u_j)$ of each sibling u_j with $i \neq j$, and whether a filter $filter(parent(u_i))$ is installed at its parent v, $1 \leq i, j \leq d_v$. Each node in $\mathcal{T}$ acts properly by performing the following pseudo-code, according to the role (parent or child node) it plays.

Algorithm 2 $Child\_Node(\mathcal{T}, u_i, k)$

1: **if** $filter(parent(u_i)) = 0$ **then**
2: send its top-l_i readings to node v;
3: **else**
4: send the median $s_{i,\lceil l_i/2 \rceil}$ of its top-l_i readings to v and wait for a filter broadcast x from it.
5: identify the smallest element $s_{i,l_i'}$ in $L(u_i)$ such that $s_{i,l_i'} \geq x$ and transmit its top-l_i' elements in $L(u_i)$ to v;

6 Performance Study

In this section we evaluate the performance of the proposed algorithms in sensor networks, in terms of the total energy consumption, the maximum energy consumption among the nodes, the query response time, and the network lifetime through experimental simulation.

Algorithm 3 $Parent\_Node(\mathcal{T}, v, k)$

1: **if** $filter(v) = 0$ **then**
2: /* no filtering on its parent v */
3: wait for receiving all readings in $S(v)$;
4: **else**
5: wait for receiving the medians from its children and sort the medians in decreasing order;
/* Let $m_{i_1}, m_{i_2}, \ldots, m_{i_{d_v}}$ be the sorted sequence. */
6: filter $x = m_{i_j}$ such that $|R(x)| = \sum_{p=1}^{j} \lceil l_{i_p}/2 \rceil \geq k$ and j is the smallest integer with $1 \leq j \leq d_v$.
7: broadcast x to and receive data from its children;
8: identify the top-k readings in the subtree rooted at v,using the received readings and its own reading;

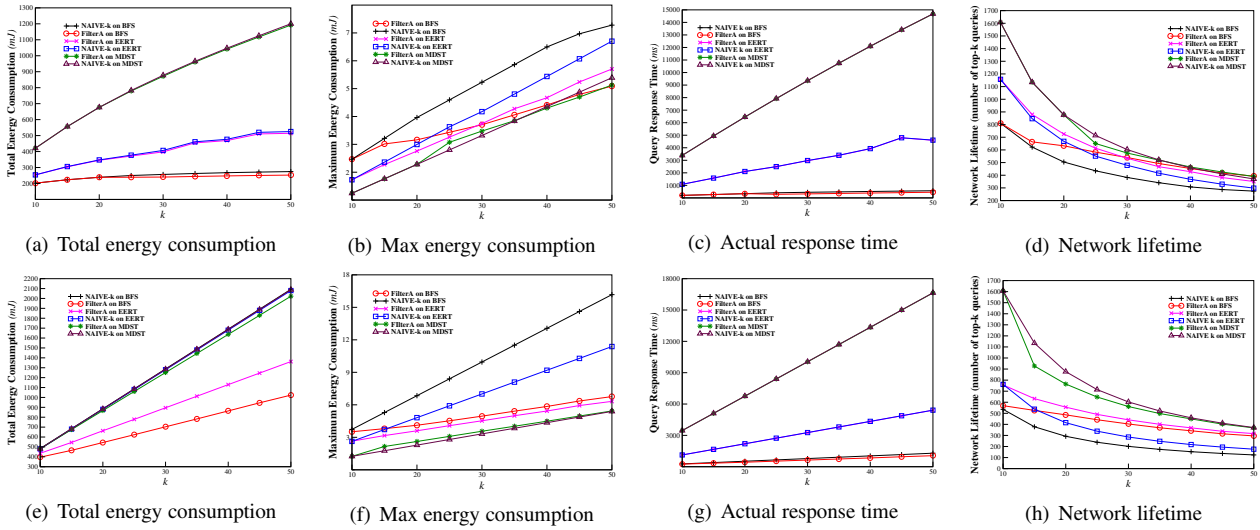
6.1 Simulation environment

We assume that the sensor network is used to monitor a $100m \times 100m$ region of interest and 500 sensor nodes are randomly deployed by the $NS-2$ simulator [8] within the region. The base station is located at the square center. Two nodes can communicate to each other if they are within the transmission range of each other. We further assume that all sensor nodes have the same transmission range (10 meters). We take into account the transmission and reception energy consumptions of sensor nodes. In our experiments, we set $R = 0.0144\ mJ$ and $r_e = 0.00576\ mJ$ as the transmission and reception energies per byte, respectively. The time delay for transmitting one byte is $0.4\ ms$ and $t_H = 0.8\ ms$. The initial energy capacity at each sensor node is set to be $2,000\ mJ$. We assume that each node contains one or k readings initially, $10 \leq k \leq 50$. The sensing readings in our experiments are simulated by using the real temperature trace of Live from Earth to Mars(LEM) project at the University of Washington [7]. Each readings is represented by $l = 4$ bytes and $l_H = 8$ bytes.

6.2 Impact of different size k

We first evaluate the impact of different algorithms for top-k queries on network lifetime by varying the size of k. We evaluate the performance of algorithms NAIVE-k and FilterA on three different routing trees, T_{BFS}, T_{MDST}, and the energy-efficient tree (EERT for short) obtained by applying Filter Routing Tree. The performance of various algorithms is depicted by Fig. 1 (a)-(h).

Consider the case where each node contains just one reading initially. Fig. 1(a) implies that the total energy consumption is proportional to the size of k. With the increase of k, the total energy consumption by using MDST is significantly higher than that by using EERT and BFS, and the total energy consumption by BFS is the minimum one among the trees, regardless of which algorithm being used. It can be seen from Fig. 1(b) that the maximum energy consumption by applying algorithm FilterA on BFS is less than that by applying the same algorithm on EERT when $k \geq 30$, which is explained as follows.



(a) Total energy consumption (b) Max energy consumption (c) Actual response time (d) Network lifetime

(e) Total energy consumption (f) Max energy consumption (g) Actual response time (h) Network lifetime

Figure 1. The performance of different routing algorithms and various routing trees for top-k query evaluation in terms of the total energy consumption, the maximum energy consumption among the nodes, the query response time, and the network lifetime, assuming that the network contains 500 sensor nodes and each node contain either one reading (figures (a)-(d)) or k readings (figures (e)-(h)) with $10 \leq k \leq 50$.

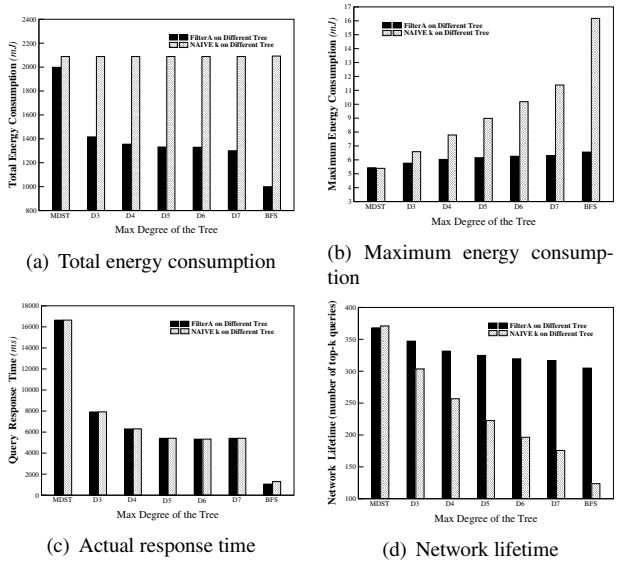
Let Δ_{EERT} be the maximum degree of nodes with no less than k descendants in EERT and Δ_{BFS} the maximum degree of BFS. When k is small in comparison with the network size n, for the top-k query using the proposed filtering algorithm on either of the two trees, the maximum energy consumption among the nodes is the maximum degree nodes in the corresponding tree that each child has at least k descendants. When k becomes larger without changing the size of n, the average number k' of descendants of each child of the maximum degree node will be less than k, i.e., $k' < k$. Meanwhile, it is known that the reception energy consumption at a node is proportional to the number of children and the average number of top-k readings of the children, assuming that the number of descendants of some children is no more than k. Thus, the maximum energy consumption among the nodes in BFS or EERT for top-k queries is proportional to $\Delta_{BFS}k_{BFS}$ or $\Delta_{EERT}k_{EERT}$, where k_{BFS} and k_{EERT} are the average number of top-k readings of children of a maximum degree node in BFS and EERT. Clearly, $k_{EERT} > k_{BFS}$ and $1 \leq k_{EERT}, k_{BFS} < k$, because $\Delta_{EERT} < \Delta_{BFS}$ and the network size is given. The maximum energy consumption derived by tree EERT is larger than that by BFS if $\Delta_{EERT}k_{EERT} > \Delta_{BFS}k_{BFS}$. In other words, the claim holds if $\frac{\Delta_{EERT}}{\Delta_{BFS}} > \frac{k_{BFS}}{k_{EERT}}$. By this inequality, once $\frac{\Delta_{EERT}}{\Delta_{BFS}}$ is fixed, it is easy to see the increasing rate of k_{EERT} is faster than that of k_{BFS} with the growth of k. In other words, let k becomes $k + \delta k$, correspondingly, k_{BFS} and k_{EERT} become $k_{BFS} + \delta k_{BFS}$ and $k_{EERT} + \delta k_{EERT}$, and $\delta k_{EERT} > \delta k_{BFS}$. Thus, $\frac{k_{BFS}}{k_{EERT}} > \frac{k_{BFS}+\delta k_{BFS}}{k_{EERT}+\delta k_{EERT}}$, and the inequality continues to be held, with the growth of k. Note that in our experiment case, $\Delta_{BFS} = 11$ while $\Delta_{EERT} = 5$, and $n = 500$ nodes are deployed. The maximum degree node in tree BFS contains no more than $5 * k = 5 * 30 = 150$ descendants when $k \geq 30$. In practice, the degree of each node in a large size network is a fixed small constant. When k is relatively small compared with the network size n, for a maximum degree node in a routing tree, the condition that each of its children contains at least k descendants is easily met. As a result, in terms of the maximum energy consumption among the nodes by employing the filtering algorithm on either of them for top-k query, EERT outperforms BFS, this can be seen from Fig. 1(f), where each node contains k readings initially.

With maintaining the same response time, Fig. 1(b),(c) and Fig. 1(f),(g) implies that algorithm `FilterA` significantly outperforms algorithm `NAIVE-`k in terms of the maximum energy consumption and network lifetime, no matter which routing tree is used. Meanwhile, the difference of network lifetime between the two algorithms becomes insignificant when applying them on T_{MDST} because in our experiments, the maximum degree of the T_{MDST} is only 2, for which algorithm `FilterA` does not have much effect on filtering. Fig. 1(c) and (g) show that the actual query response time grows with the increase of the value of k. Although applying the algorithms on T_{MDST} leads to a reasonable longer network lifetime (Fig. 1(d) and (h)), the corresponding query response time is the longest as well. On the other hand, although applying `NAIVE-`k on T_{BFS} brings the shortest response time, it will also re-

sult in the shortest network lifetime. Accordingly, applying algorithm FilterA on $EERT$ is nice choice to obtain a reasonably longer network lifetime and yet meet a certain response time constraint simultaneously.

6.3 Impact of the maximum node degree of different routing trees

We then investigate the impact of the maximum node degree of different routing trees on the network lifetime while meeting the given query response time constraint when $k = 50$. We assume that each node contains k readings initially. The performance curves are plotted in Fig. 2(a)-(h), where "$D\ \Delta$" at the x-coordinate represents a routing tree with maximum node degree of Δ, $3 \leq \Delta \leq 7$. From



(a) Total energy consumption

(b) Maximum energy consumption

(c) Actual response time

(d) Network lifetime

Figure 2. The performance of different routing trees for top-k query evaluation in terms of the total energy consumption, the maximum energy consumption among the nodes, the query response time, and the network lifetime, assuming that the network contains 500 nodes and each node contains $k = 50$ readings.

Fig. 2(a) and (b), there is no difference among different routing trees by applying NAIVE-k in terms of total energy consumption and the maximum energy consumption derived by applying NAIVE-$k$is proportional to the maximum node degree of the routing tree. In contrast, there are no significant differences among different routing trees on which algorithm FilterA is applied. Furthermore, FilterA outperforms NAIVE-k substantially, in terms of the maximum energy consumption and network lifetime (Fig. 2(b) and (d)). Fig. 2(c) shows that the query response time by applying NAIVE-k and FilterA are inversely proportional to the maximum degree of a routing tree. Given the same routing tree and time constraint, algorithm FilterA has much better performance than algorithm NAIVE-k.

7 Conclusions

We have studied the top-k query problem in sensor networks with response time constraint. We first formulated the problem by giving a cost model of energy consumption to answer a top-k query. We then proposed a novel joint optimization framework consisting of finding a routing tree in the network, devising a filter-based evaluation algorithm on the tree for top-k query evaluation. We finally conducted extensive experiments by simulation on real datasets to evaluate the performance of the proposed algorithms. The experimental results showed that there is a non-trivial tradeoff between the query response time and the network lifetime, and for a given query response time constraint, the proposed approach can prolong the network lifetime significantly.

References

[1] P. Cao and Z. Wang. Efficient top-k query calculation in distributed networks. *Proc. of ACM PODC*, ACM, 2004.

[2] J-H Chang and L. Tassiulas. Energy conserving routing in wireless ad hoc networks. *Proc. INFOCOM'00*, IEEE, 2000.

[3] Crossbow Inc. *"MPR-Mote Processor Radio Board Users Manual"*.

[4] R. Fagin, A. Lotem, and N. Naor. Optimal aggregation algorithms for middleware. *Proc. of ACM PODS*, 2001.

[5] M. Fürer and B. Raghavachar. Approximating the minimum degree spanning tree to within one from the optimal degree. *Proc. 3rd Symp. on Discrete Algorithms*, ACM-SIAM, 1992.

[6] S. Lindsey, C. S. Raghavendra, and K. M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *IEEE TPDS*, Vol.13, pp.924–935, 2002.

[7] Live from Earth and Mars (LEM) Project. http://www-k12.atoms.washington.edu/k12/grayskies, 2006.

[8] The Network Simulator-ns2. http://www.isi.edu/nsnam/ns, 2006.

[9] A. Silberstein R. Braynard, C. Ellis, K. Munagala and J. Yang. A sampling-based approach to optimizing top-k queries in sensor networks. *Proc. of ICDE.*, IEEE, 2006.

[10] M. Wu, J. Xu, X. Tang, and W-C. Lee. Monitoring top-k query in wireless sensor networks. *Proc. of 22nd Int'l Conf. on Data Engineering*, IEEE, 2006.

[11] Y. Yu, B. Krishnamachari, and V. K. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. *Proc of INFOCOM*, IEEE, 2004.