Monitoring Top-k Query in Wireless Sensor Networks

Minji Wu Jianliang Xu*
Hong Kong Baptist Univ.
Kowloon Tong, Hong Kong
xujl@comp.hkbu.edu.hk

Xueyan Tang
Nanyang Technological Univ.
Singapore
asxytang@ntu.edu.sg

Wang-Chien Lee†
Penn State Univ.
University Park, PA
wlee@cse.psu.edu

Abstract

Top-k monitoring is important to many wireless sensor applications. This paper exploits the semantics of top-k query and proposes a novel energy-efficient monitoring approach, called FILA*. The basic idea is to install a filter at each sensor node to suppress unnecessary sensor updates. The correctness of the top-k result is ensured if all sensor nodes perform updates according to their filters. We show via simulation that FILA outperforms the existing TAG-based approach by an order of magnitude.*

1 Introduction

Owing to the rapid advances in sensing and wireless communication technologies, wireless sensor networks have been available for use in a wide range of *in-situ* sensing applications, such as habitat monitoring, wild-fire prevention, and environmental monitoring [4]. A wireless sensor network typically consists of a base station and a group of sensor nodes (see Figure 1). The base station serves as a gateway for the sensor network to exchange data with external users. The sensor nodes, on the other hand, are responsible for sensing and collecting data from their local environments. They are also capable of processing sensed data and communicating with their neighbors and the base station.

Monitoring of aggregate functions is important to many sensor applications and has drawn a lot of research attention [3, 4]. Among those aggregates, a top-k query continuously retrieves the set of k sensor nodes with the highest (or lowest) readings [1, 2]. However, how to energy-efficiently answer top-k queries is a great challenge to wireless sensor networks. The sensor nodes usually operate in an unattended manner and are battery powered; replacing the batteries is not only costly but also impossible in many situations (e.g., in a hard-to-reach area). If a certain portion of the nodes run out of their power and lose their coverage, the whole network will be down. Thus, in addition to reducing network traffic, a distinguished requirement for wireless sensor networks is to balance the energy consumption at the sensor nodes to prolong network lifetime.

*Minji Wu and Jianliang Xu were supported in part by Research Grants Council of Hong Kong under Project No. HKBU 2115/05E.

†Wang-Chien Lee was supported in part by National Science Foundation grant IIS-0328881.

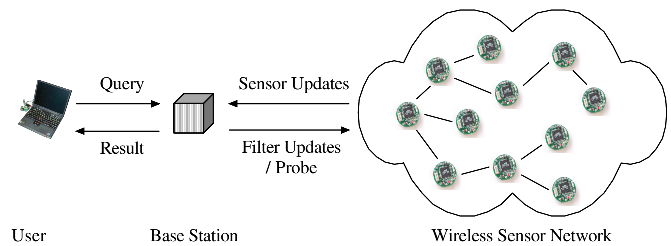
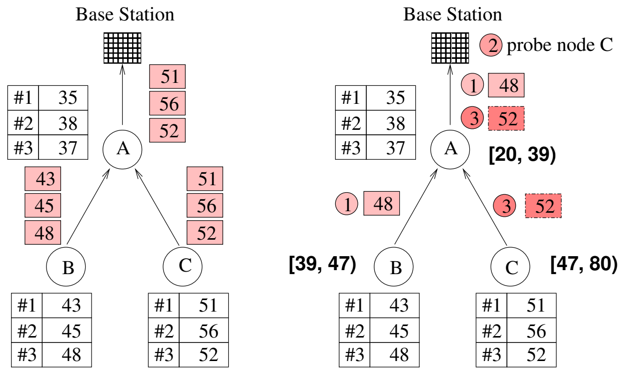


Figure 1. The System Architecture

A basic implementation of monitoring top-k query would be to use a centralized approach where all sensor readings are collected by the base station, which then computes the top-k result set. In order to reduce network traffic for data collection, an *in-network data aggregation* technique, known as *TAG*, has been proposed [3]. Specifically, a routing tree rooted at the base station is first established and the data is then aggregated and collected along the way to the base station through the routing tree. Consider a simple example shown in Figure 2a, where sensor nodes A, B, and C form a routing tree. The readings of these sensor nodes at three successive sampling instances are shown in the tables of Figure 2a. Suppose we are monitoring a top-1 query. Employing TAG, at each sampling instance, nodes B and C send their current readings to the parent (i.e., node A), which aggregates the data received with its own reading and sends the highest (i.e., the readings from node C in this example) to the base station. The top-1 result is always node C, but nine update messages (three at each sampling instance) are used. As such, this approach incurs unnecessary updates in the network and, hence, is not energy efficient.

In this paper, we exploit the semantics of top-k query



(a) TAG (b) FILA

Figure 2. An Example of Top-k Monitoring

and propose a novel filter based monitoring approach called *FILA*. The basic idea is to install a filter at each sensor node to suppress unnecessary sensor updates. The base station also keeps a copy of the filter setting to maintain a *view* of each node's reading. A sensor node reports the reading update to the base station only when it passes the filter. The correctness of the top-k result is ensured if all sensor nodes perform updates according to their filters. Figure 2b shows an example, where the base station has collected the initial sensor readings and installed three filters [20, 39), [39, 47), and [47, 80) at sensor nodes A, B, and C, respectively. At sampling instances 1 and 2, no updates are reported since all updates are filtered out by the nodes' respective filters. At instance 3, the updated reading of node B (i.e., 48) passes its filter [39, 47). Hence, node B sends the reading 48 to the base station via node A (step ①). Since 48 lies in the filtering window of node C (i.e., [47, 80)), the top-1 result becomes undecided as either node B or C can have the highest reading. In this case, we probe node C for its current reading to resolve the ambiguity (steps ② and ③). Thus, a total of four update messages and one probe message are incurred in this approach.[1] Compared with the aforementioned TAG-based aggregation approach, five update messages are saved at the cost of one probe message. Obviously, this approach achieves a better performance than the TAG approach.

Yet, in order to make FILA to work efficiently, two fundamental issues arising at the base station server have to be addressed:

- How to set the filter for each sensor node in a coordinated manner such that the top-k result set is correctly returned if all nodes perform updates according to their filters? The filter setting is critical to the performance of FILA. In the above example, if nodes B and C have the filters set to [39, 50) and [50, 80), respectively, no updates need to be reported for all three samplings.
- Upon receiving an update from a sensor node, how to reevaluate the top-k result and how to update the affected filters?

[1]For simplicity, the overhead for initial data collection and filter setting is not shown here, but counted in our experiments.

We answer in this paper the above two questions with the objective of reducing network traffic and prolonging network lifetime.

2 System Model and Problem Definition

We consider a wireless sensor network as depicted in Figure 1. It is assumed that the base station has continuous power supply and its radio strength is strong enough to cover all sensor nodes. In other words, a probe message broadcast by the base station can reach all sensor nodes in a single hop. In contrast, the sensor nodes are powered by battery. Their radio coverage is constrained to a local area. When the base station is beyond a sensor node's radio coverage, an underlying routing infrastructure (e.g., a TAG tree [3]) is used to route data to the base station.

Each sensor node i measures the local physical phenomenon v_i (e.g., pollution index, temperature, or residual energy, etc.) at a fixed sampling rate. Without loss of generality, we consider a top-k monitoring query that continuously retrieves the (ordered) set of sensor nodes $\mathcal{R}$ with the highest readings, i.e.,

$$\mathcal{R} = < n_1, n_2, \cdots, n_k >,$$

where $\forall i > j, v_{n_i} \leq v_{n_j}$ and $\forall l \neq n_i (i = 1, 2, \cdots, k), v_l \leq v_{n_k}$. The monitoring result is maintained by the base station and updated to the user. To produce continuous query results, the proposed monitoring approach controls when and how to collect sensor reading updates to the base station.

3 FILA Overview

Initially, the base station collects the readings from all sensors. It then sorts the sensor readings and obtains the initial top-k result set. Next, the base station computes a filter (represented by a window of $[l_i, u_i)$) for each sensor node i and sends it to the node for installation. At the next sensor sampling instance, if the new reading of sensor node i is within $[l_i, u_i)$, no update to the base station is needed. Otherwise, if the new reading goes beyond the filtering window and passes the filter, meaning the top-k order might be violated, an update is sent to the base station. The base station will then reevaluate the top-k result and adjust the filter setting(s) for some sensor node(s) if necessary. The query reevaluation algorithm is discussed in detail in [6].

As can be seen, the purpose of using filters is to filter out some local sensor updates and hence suppressing the traffic in the network. The correctness of the top-k result must be guaranteed provided that all sensor nodes perform updates according to their filters. Thus, the filter settings have to be carefully planned in a coordinated manner. Denote the current reading of node i by v_i. Without loss of generality, we number the sensor nodes in decreasing order of their sensor readings, i.e., $v_1 > v_2 > \cdots > v_N$, where N is the number of sensor nodes under monitoring. Intuitively, to maintain the monitoring correctness, the filters assigned to the nodes in the top-k result set should cover their current readings but not overlap with each other. On the other hand, the nodes in the non-top-k set could share the same filter setting. Thus, we consider the filter settings only for the top-k+1 nodes. A *feasible* filter setting scheme, represented as $\{[l_i, u_i) \mid i = 1, \cdots, k+1\}$, must satisfy the following conditions:

$$\begin{cases} u_1 > v_1; \\ v_{i+1} < u_{i+1} \le l_i \le v_i, \qquad (1 \le i \le k); \\ l_{k+1} \le v_N. \end{cases} \tag{1}$$

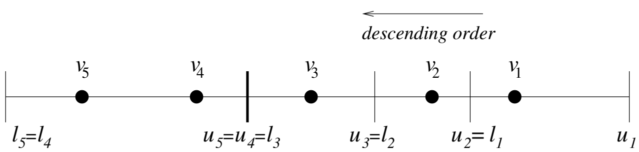


Figure 3. Filter Settings for Top-3 Monitoring

Figure 3 shows a feasible filter setting for top-3 monitoring, where nodes 4 and 5 share a filter setting and u_{i+1} is set equal to l_i for $1 \le i \le 3$ in order to maximize the filtering capability. Intuitively, *a filter setting is a (constrained) partitioning of the data space*. A straightforward way is to set the filter bound at the midpoint of two sensor readings, i.e.:

$$u_{i+1} = l_i = \frac{v_i + v_{i+1}}{2}, \quad (1 \le i \le k). \tag{2}$$

We call it *uniform* filter setting. It is favorable in the case where the sensor readings from all sensor nodes follow a similar changing pattern. A more sophisticated scheme for filter setting is described in [6].

We have developed a simulator based on ns-2 and NRL's sensor network extension to evaluate the proposed FILA approach. Figure 4 shows the results against TAG [3] for a wide range of real traces [5]. We can see that FILA improves network lifetime over TAG by an order of magnitude while achieving a much lower average energy consumption.

4 Conclusions

This paper proposed a novel energy-efficient approach called FILA for top-k monitoring in wireless sensor networks. As for future work, we plan to extend the proposed monitoring approach to other aggregate functions such as kNN, average, and sum. We are going to build a prototype based on Motes and measure the performance in real environments. We are also interested in monitoring spatial queries in object-tracking sensor networks.

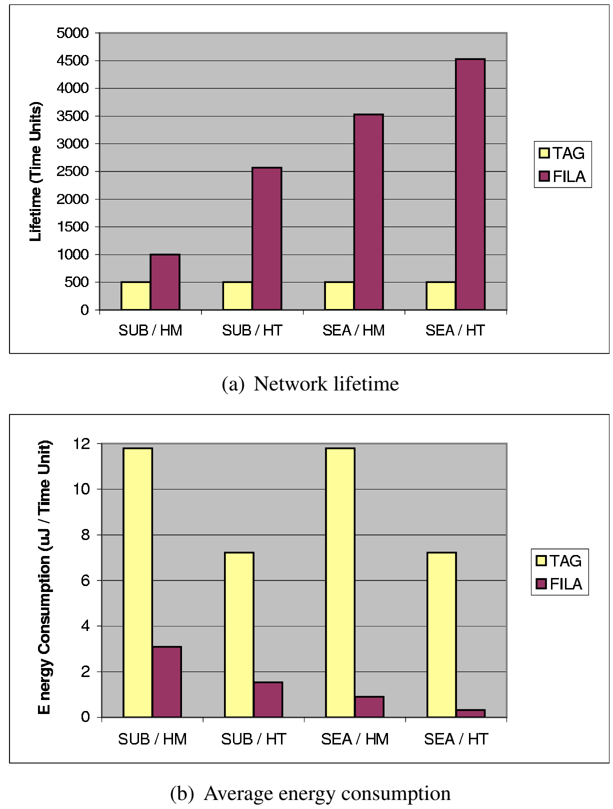


Figure 4. Performance Comparison with TAG

References

[1] B. Babcock and C. Olston. Distributed top-k monitoring. In *Proc. ACM SIGMOD*, pages 28–39, June 2003.

[2] P. Cao and Z. Wang. Efficient top-k query calculation in distributed networks. In *Proc. PODC*, July 2004.

[3] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *Proc. USENIX OSDI*, pages 131–146, December 2002.

[4] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, June 2004.

[5] Tropical Atmosphere Ocean (TAO) Project. http://www.pmel.noaa.gov/tao/data_deliv.

[6] M. Wu, J. Xu, X. Tang, and W.-C. Lee. Top-k monitoring in wireless sensor networks. Technical Report, Hong Kong Baptist University, June 2005.