TLF: Two-level Filter for Querying Extreme Values in Sensor Networks

Min Meng[1], Jie Yang[1], Yu Niu[1], Young-Koo Lee[2], Byeong-Soo Jeong[2], Sungyoung Lee[1]
Department of Computer Engineering
Kyung Hee University
Suwon, South Korea
E-mail: [1]{mengmin,yangjie,niuyu,sylee}@oslab.khu.ac.kr, [2]{jeong,yklee}@khu.ac.kr

***Abstract* — Sensor networks have been widely applied for data collection. Due to the energy limitation of the sensor nodes and the most energy consuming data transmission, we should allocate as much work as possible to the sensors, such as data compression and aggregation, to reduce data transmission and save energy. Querying extreme values is a general query type in wireless sensor networks. In this paper, we propose a novel querying method called Two-Level Filter (TLF) for querying extreme values in wireless sensor networks. We first divide the whole sensor network into domains using the Distributed Data Aggregation Model (DDAM). The sensor nodes report their data to the cluster heads using push method. The advantages of two-level filter lie in two aspects. When querying extreme values, the number of pull operations has the lower boundary. And the query results are less affected by the topology changes of the wireless sensor network. Through this method, the sensors preprocess the data to share the burden of the base station and it combines push and pull to be more energy efficient.**

***Keywords* — TLF, Two-level filter, querying extreme values, sensor networks.**

1. Introduction

Querying extreme values in wireless sensor networks is a general problem. The scenarios of continuous MAX query are popular, such as environment monitoring, maintaining maximum temperature in a factory, or getting the highest humidity in geographic areas.

Data transmission consumes most energy in wireless sensor networks. So we should reduce data transmission as much as possible. Sensor nodes have the capabilities to preprocess the data, like data aggregation and compression.

In this paper, we propose the novel querying method called Two-Level Filter (TLF) for querying extreme values in wireless sensor networks. We first divide the whole sensor network into domains using the Distributed Data Aggregation Model (DDAM). The sensor nodes in each domain form a cluster. Every cluster has a cluster head. Then we set the first level filter using FILter based monitoring Algorithm (FILA) on the sensor node level, second level filter on the cluster head. The sensor nodes report their extreme values to the cluster heads using push method. The advantages of TLF lie in two aspects. When querying extreme values, the number of pull operations has the lower boundary. In the worst case, the number of pull operations equal to the number of cluster heads. And the query results are less affected by the topology changes of the wireless sensor network.

The rest of the paper is organized as follows. We introduce the related works in section 2. We propose TLF for querying extreme values in sensor networks in section 3. Finally, in section 4, we conclude our method and describe future works.

2. Related works

We use DDAM model to divide the sensor networks into domains. This model can save more transmission energy because each sensor nodes can decide their domains by themselves and no need to communicate with other sensor nodes. Through this the model can save energy of the sensor nodes then prolong the lifetime of the whole sensor networks. M. Wu [3, 4] proposed FILA approach to set filters on each sensor nodes. We can use other methods for filter setting [2]. It first sets a rooting tree rooted at the base station and data is then aggregated and collected along the way to the base station through the routing tree. In [3], the base station computes a filter for each sensor node. In two-level filter, the parent nodes in FILA are the cluster heads and they take the job of the base station.

Some papers proposed push or pull method for data management in wireless sensor networks, such as TAG [5]. Then some papers discussed to combine push and pull and get better results [7, 8]. [9] surveyed the research issues in sensor networks. [6] proposed join method of sensor reading and predicate table. A. Silberstein [1] surveyed the push and pull approaches for querying processing in sensor networks. Push means to install conditions within the network which, when met, trigger data transmission. Pull means to have the base station request particular data. Push method is more suitable for continuous queries. Push runs as a one way trip for server to client. The server knows when the items are updated. Push met scalability issue because it requires the server to remember requests of each client. Pull method is more suitable for ad hoc queries. Pull runs as a round trip from client to server then to client. The clients risk not having the most up-to-date items. Pull doesn't require the server to store the client state.

In TLF, the first level filter on sensor nodes uses the push method, when data passes the filter triggers data transmission to the cluster heads. Second level filter can be event driven. When there are queries for extreme values, triggers data transmission. Push is more suitable for continuous queries and pull is for ad hoc queries.

3. TLF for querying extreme values in sensor networks

3.1 DDAM to divide the WSN into domains

We use DDAM model to divide the wireless sensor network into domains. Any sensor node p, assume its relative coordination to the sink node R is (x,y). All sensor nodes including sink node R have the same communication radius r, then we call p belongs to domain (m,n) if and only if

$$m = [x \Big| \frac{r}{\sqrt{2}}],\ n = [y \Big| \frac{r}{\sqrt{2}}] \quad (1)$$

As shown in Figure 1.

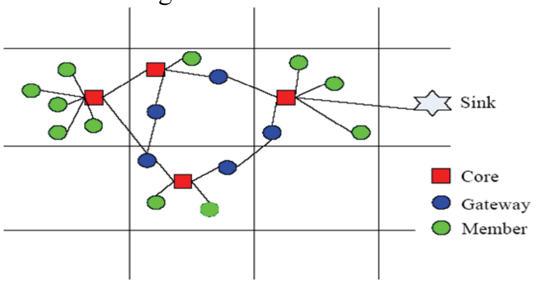

Figure 1. Domain division using DDAM model

Proof: Every domain is a square with length and width equal to $\frac{r}{\sqrt{2}}$, shown as below. The longest distance between two nodes in the domain is $\sqrt{\left(\frac{r}{\sqrt{2}}\right)^2 + \left(\frac{r}{\sqrt{2}}\right)^2} = r$. So every two sensor nodes are within the communication radius of each other. All nodes in the same domain are neighbors. As shown in Figure 2.

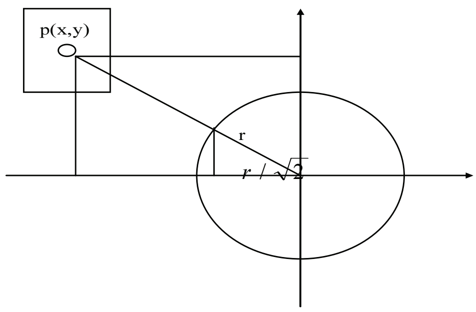

Figure 2. Proof of DDAM model

3.2 TLF for querying extreme values

The basic idea of FILA is to install a filter at each sensor node to suppress unnecessary sensor updates. The cluster head of each domain 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 cluster head only when it passes the filter. For querying extreme values, we only need to set two filters. One filter for extreme values, the non extreme values share another same filter.

There are two critical issues under the FILA approach, filter setting and query reevaluation. We use the algorithms proposed in [4]. There are two filter setting algorithms, uniform and skewed. As to the uniform filter setting algorithm, we make the sensor nodes in decreasing order of their readings, i.e., $v_1 > v_2 > \ldots > v_n$, where n is the number of sensor nodes under monitoring. A filter setting scheme, represented as $\{(l_i, u_i) | i = 1, \cdots, k+1\}$, must satisfy the following conditions:

$$\begin{cases} u_1 > v_1; \\ v_{i+1} < u_{i+1} \le l_i \le v_i, (1 \le i \le k); \ (2) \\ l_{k+1} \le v_N. \end{cases}$$

The simple way to set the filter bound is at the midpoint of two sensor readings. $u_{i+1} = l_i = \frac{v_i + v_{i+1}}{2}$ (3)

But the uniform setting doesn't consider the changing patterns of sensor readings so that it may result in unbalanced energy consumption. So s skewed filter setting algorithm is proposed. If the average time for reading of node i to go beyond δ is $f_i(\delta)$. Then the node update rate should be $\frac{1}{f_i(\delta)}$. In order to balance the energy consumption of nodes $i+1$ and i, we should make their update rates equal:

$$\frac{1}{f_{i+1}(u_{i+1} - v_{i+1})} = \frac{1}{f_i(v_i - l_i)} \quad (4)$$

Assume the readings change follow a random walk model, the value changes by an amount of d. Finally we get

$$u_{i+1} = l_i = v_{i+1} + \frac{d_{i+1}}{d_i + d_{i+1}}(v_i - v_{i+1}), (1 \le i \le k), (5)$$

The filter setting for the upper bound of the top-1's filter u_1 should be set to $u_1 = 2v_1$, twice of its current reading v_1, the lower bound of non-top-k node's filter $l_{k+1} = \frac{v_N}{2}$, half of the lowest sensor reading.

There are three scenarios that may trigger query reevaluation. The update is originated from a top-k node and jumps over the critical bound; the update is originated from a non-top-k node and jumps over a critical bound; the update is from a top-k node but does not jump over the critical bound.

We use filter instead of one number, such as the kth largest number for each sensor node because when a non-top-k data is beyond the critical bound, we can easily know which sensor node should be eliminated from the top-k nodes. And the queries from the clients need to know the order of the data in the top-k nodes. There are two filter updating approaches, eager filter updating and lazy filter updating. The lazy approach can help save filter updating messages.

As to the second level filter on the cluster head level, when there is a query for extreme value, we can broadcast the query only to the cluster head level, and no need to go deep to every sensor nodes. So this TLF method is more energy efficient and prolongs the lifetime of the sensor network. And the topology of wireless sensor networks are changing frequently, using TLF, if sensor nodes only move within the domain, the extreme values of the domain stored on the cluster head will not be affected.

4. Conclusions and Future Works

This paper proposes a novel energy-efficient approach TLF for querying extreme values in wireless sensor networks. It first divides the sensor networks into domains, sensor nodes in every domain form a cluster. Within the domain we use push method to store the extreme values on the cluster heads. When the queries come, we use pull method to report to the clients. As for future works, we plan to simulate the proposed method, and consider other methods for second level filter on the cluster head level.

REFERENCES

[1] A. Silberstein. Push and Pull in Sensor Network Query Processing. In Proc. of the 2006 Southeast Workshop on Data and Information Management (SWDIM '06), Raleigh, North Carolina, USA, March 2006.
[2] A. Silberstein, K. Munagala, and J. Yang. Energy-Efficient Monitoring of Extreme Values in Sensor Networks. In Proc. of the 2006 ACM SIGMOD Intl. Conf. on Management of Data, Chicago, Illinois, USA, June 2006.
[3] M. Wu, J. Xu, X. Tang, and W. Lee. Monitoring Top-k Query in Wireless Sensor Networks. In Proc. of the 2006 Intl. Conf. on Data Engineering, Atlanta, Georgia, USA, Apr. 2006.
[4] M. Wu, J. Xu, X. Tang, and W.-C. Lee. Monitoring Top-k monitoring in wireless sensor networks. Technical Report, Hong Kong Baptist University, June 2005.
[5] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks. In OSDI, 2002.
[6] D. Abadi, et al. REED: Robust, Efficient Filtering and Event Detection in Sensor Networks. In Proc. of the 31st VLDB conference, Trondheim, Norway, 2005.
[7] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajaraman. “Hybrid Push-Pull Query Processing for Sensor Networks”, In Proceedings of the Workshop on Sensor Networks as part of the GI-Conference Informatik 2004. Berlin, Germany, September 2004.
[8] Shyam Kapadia and Bhaskar Krishnamachari1, Comparative Analysis of Push-Pull Query Strategies for Wireless Sensor Networks, DCOSS 2006, LNCS 4026, pp. 185–201, 2006.
[9] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communication Magazine, Vol.40, No.8, 2002, pp.102-114.