Data Reduction Algorithm on the Monitoring of Extreme Values in WSNs

Chun-Lung Lin
Industrial Technology Research Institute
HsinChu, Taiwan R.O.C.
Email: chunlung@itri.org.tw

Pei-Hsuan Tsai
Institute of Manufacturing Information and Systems
National Cheng Kung University
Tainan, Taiwan R.O.C.
Email: phtsai@mail.ncku.edu.tw

Hsiao-Chuan Liang and Jia-Shung Wang
Department of Computer Science
National Tsing Hua Univeristy
HsinChu, Taiwan R.O.C.
Email: jswang@cs.nthu.edu.tw

***Abstract*—Monitoring extreme values (maximum or minimum) is important to many applications in wireless sensor networks. A previous work, called Hierarchy Adaptive Threshold (HAT), proposed a tree-based structure to distribute queries efficiently and filter out the unnecessary data updates that are not extreme values. In this paper, a data reduction algorithm is presented to reduce energy consumption of the HAT due to network transmission. The proposed method utilizes historical information of extreme values and their corresponding node ID to adjust the reporting rate of sensors properly and eases the burden of the parent of extreme nodes by balancing the packets from extreme nodes to all their possible parents. We evaluate the performance of the proposed algorithm by NS-2 network simulator and real-world data traces. The results indicate that the overall network packets are reduced to 80% with 1% data error in comparison with HAT.**

***Keywords-sensor networks*; *extreme values*; *data reduction*.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been widely applied to many current and envisioned applications, including vehicle tracking and habit monitoring [1], [2], nuclear reactors, test areas, disaster management, combat field surveillance, factory temperature monitoring, border control and so on. A maximum (or minimum) query is a query, which continuously requests the sensor node with the maximum (minimum) sensor reading. That is, a maximum query continuously maintains the (node id, value) pair in the network. Monitoring maximum (or minimum) value is important for detecting abnormal or extreme behavior in many sensor network applications. For example, monitoring maximum temperature in a factory is essential to improve production yield, or finding out the node and its corresponding areas with the highest pollution index for the purpose of pollution control. A minimum query, on the other hand, can be requested to continuously monitor the sensor node with the least residual energy so that it can be instructed to adapt its sampling rate (or transmission rate) for extending network lifetime. Since the discussions of maximum and minimum queries are similar, in the following we will only focus on answering the continuous maximum monitoring problem.

The difficulty of answering a maximum query is unable to know the sensors that constitute high values in priori. Hierarchy Adaptive Thresholds (HAT) was proposed to monitor the maximum query efficiently [3]. HAT prevents the nodes which will not be maximum nodes from transmitting by giving constraints on nodes such as filter threshold. However, there is a penalty where additional queries are required to be issued when the sink cannot definitely decide the current maximum value. Moreover, the performance of HAT highly depends on the update strategy of filter thresholds at nodes. On the basis of HAT, a novel filter-based monitoring algorithm is proposed in this paper. According to the observations on real data, the locations and values of the maximum nodes are usually stable. That is,

1) Parts of nodes inside the network become the maximum node more often than the others. This suggests that some nodes are always likely to become the maximum node, and in contrast, some are unlikely to be the maximum one.
2) The maximum node often remains unchanged for some consecutive time intervals (i.e., temporal correlation between sensor readings).
3) When a new maximum node emerges, it is often nearby the previous maximum node (i.e., spatial correlation among nearby nodes).

In light of the phenomenon above, the proposed algorithm utilizes historical information of the past maximum values and the nodes that generate them. Specifically, the algorithm measures the probability of a node being the maximum node by using historical information. The quantitative relationship between reporting rate and filtering interval is also measured. According to the probability, nodes are dynamically instructed to re-adjust their filtering threshold so as to meet the required reporting rate (e.g., decrease the reporting rates of nodes with lower probability and increase the reporting rates of those with higher probability). The typical goal is to prolong the network lifetime by reducing unnecessary data updates.

The rest of this paper is organized as follows. The related work is given in Section 2. Section 3 presents the proposed algorithm. Simulation results are described in Section 4. Finally, the conclusion is drawn in Section 5.

 ISBN: 978-1-61208-207-3

II. Related Work

Monitoring aggregation functions (such as average, maximum, and minimum) in sensor networks have received extensive attention in the past few years. However, the main focus has been on how to establish the communication structure and how to apply aggregation techniques to reduce network traffic [5], [6]. Work has also been done on compressing historical sensor readings for transmission [7], [8]. These methods are applicable to archival data collection where the application wants to log historical sensor readings and analyzes them later. In contrast, we consider monitoring applications that continuously request up-to-date sensor readings. Approximate monitoring schemes have been proposed for average and sum aggregates in [9], [10]. Yoon and Shahabi [11] proposed a clustered aggregation (CAG) algorithm for approximate query processing. CAG reduces network traffic by forming clusters of nodes where the cluster heads are responsible for aggregating the readings of their children.

Another work for approximate query processing is range caching. There is a value range installed at each node and the base station caches all the values of ranges at nodes. A node updates its new reading with the sink only if the difference between the new value and the previously reported value beyond the range. The nodes are sorted by range upper bounds, and on receiving a maximum query, the sink searches each node in order one by one. This approach is costly and inefficient because it does not take aggregation into consideration. A maximum query can be a special case of top-k query, and studies on evaluating snapshot top-k queries in distributed networks are included in [12]–[15].

In this paper, we are interested in monitoring a continuous maximum query. Although continuous monitoring could be simulated by repeatedly executing a snapshot query, many snapshot queries would be costly and inefficient if the answers remain unchanged. Recent work on monitoring continuous top-k queries in distributed networks are described in [3], [16]–[19]. In [18], a filter-based approach called FILA was proposed to exploit the semantics of top-k query. The basic idea of FILA is to install a filter at each sensor node to suppress unnecessary sensor updates. Silberstein et al. [3] explored techniques for continuously monitoring a maximum query, with the objective of minimizing network traffic. They proposed a HAT approach, which use suitable constraint settings to prevent nodes unlikely to have the maximum value from transmitting. However, HAT does not take historical information of the maximum values into account.

III. The Proposed Algorithm

As mentioned above, the purpose of the proposed algorithm is to reduce network packets when running HAT. The proposed monitoring algorithm of extreme values is on the basis of HAT algorithm. The significant improvement of the proposed method over the HAT is that it utilizes historical information of the maximum values (and the maximum nodes) to measure the probability of a node being the maximum node. The quantitative relationship between reporting rate and filtering threshold is also measured. According to the probability, nodes are dynamically instructed to re-adjust their filtering thresholds so as to meet the required reporting rate.

Consider a wireless sensor network that includes n fixed location sensors. The network is rooted at the sink with a continuous power supply, and the sensor nodes are powered by battery. It is assumed that there is a tree-based communication infrastructure installed on the network by which nodes beyond the transmission range of the sink can send their data to the sink [31]. Each sensor is assumed to periodically sample the local phenomenon, such as temperature, humidity and pollution index, at a fixed rate. Without loss of generality, the period between two successive samplings is assumed to be one time unit (round). That is, we assume the query is processed repeatedly over a series of rounds, where each node generates a value in each round. Each round is long enough for all necessary messaging to occur in order to complete the query.

In this paper, we consider a maximum query that continuously queries the sensor node with the maximum sensor reading at each round. The maximum query is to continuously maintain the (node id, value) pair for the node with the maximum value in the network. The monitoring result is logged at the sink and provided to external users. The monitoring algorithm of extreme values is to control when and how sensor readings should be sent to the sink to continuously generate maximum query results.

A. Procedure of the proposed algorithm

1) Initialization: Let C_i denote the set of the immediate children of a sensor node s_i and p_i denote the parent node of s_i in the network communication infrastructure, respectively. Denote the reading of sensor i and the maximum value at the t-th round as v_t^i and v_t^{max} respectively. Initially, each sensor node sends its reading to its parent node. When an internal node s_i receives all readings of its child nodes, it obtains the maximum value in the sub-tree rooted at s_i by sorting the sensor readings, including its own reading. That is, node s_i aggregates the messages of its child nodes and itself by only passing the packet with the highest value since it is impossible for those values to be the maximum. Then the node s_i computes a filter threshold for itself and each of its child nodes. The thresholds are sent to all of its child nodes. Similarly, we denote h_t^j as the filter threshold of node s_j at the t-th round. Note that the filter threshold of a node is by known to both itself and its parent node. This message aggregation is bottom-up, beginning at leaf nodes until the sink is reached. Accordingly, the sink obtains the initial query result, i.e., v_0^{max}, by collecting the readings from all

 ISBN: 978-1-61208-207-3

sensor nodes. Once a maximum node is obtained, the sink will also send a message to designate the node reporting the maximum value as the current maximum node, denoted by s_0^{max}.

2) Sequential Rounds: Each subsequent round proceeds in three stages: *node-initiated report*, *root-initiated query* and *reporting rate adjustment*.

Node-initiated report: If a node s_k is the designated maximum at the last round (t-th round), i.e., $s_k = s_t^{max}$, it transmits an update containing the (node id, new value) pair to the sink only if $v_{t+1}^k \neq v_t^{max}$. (i.e., the new gathered value v_{t+1}^k differs from the maximum value in the last round.) For a node s_j, $s_j \neq s_t^{max}$, if the new reading at the $(t+1)$th round is smaller than its filter threshold h_t^j, then no update is sent to its parent node. Otherwise, an update is sent to its parent node. When an internal node s_i receives any update from its child nodes, it computes the new maximum value v_i^* in the sub-tree rooted at s_i. If $v_i^* > h_t^i$, an update with the new value v_i^* is sent to its parent; otherwise, all packets in the sub-tree rooted at s_i will be filtered out to reduce network traffic. Namely, node s_i aggregates the messages of it child nodes by only passing the packet if the highest value breaks its filter threshold. Then s_i will also update its threshold h_{t+1}^i and the filter thresholds h_{t+1}^j for its child nodes that send an update by sending a threshold update packet to them. Obviously, the threshold h_{t+1}^j is known to both sensor s_j and its parent node s_i.

Root-initiated query: If the designated maximum node s_{max} in the last round do not report, it can be known in the sink that the reading of s_{max} remains unchanged in the current round. Then the stored value of v_{max} in the last round can be used to evaluate the new maximum value. Once all of the update packets are received at the sink, it determines v_{sink}^* from the set of all returned values. It can be verified easily that if the reading of s_{max} stays the same or rises, then v_{sink}^* is the maximum value since all values higher than v_{sink}^* will definitely overtake their filter threshold. On the other hand, a node can become the new maximum node without breaking its filter threshold only if the old maximum value falls. In the case, it is possible that the new maximum value can only be discovered through root-initiated querying process. More specifically, the root sends query messages containing the temporary maximum value v_{sink}^* to those of its child nodes with filter threshold greater than v_{sink}^*. In turn, each node receiving a query packet only forwards it to its child nodes with thresholds greater than v_{sink}^*. On receiving the query packet, only those nodes with values greater than v_{sink}^* send their new readings to the sink. Similarly, all reply messages are aggregated at internal nodes along the transmission path and only the maximum of them is forwarded upward. Moreover, the filter thresholds of nodes along the transmission are also updated accordingly.

Reporting rate adjustment: Here we assume that the sink maintains a probability value γ_t^i for each node s_i at the t-th round. (We postpone the detailed discussion of γ_t^i in the next subsection.) After deciding the maximum value in the current round, the sink will re-evaluate γ_t^i and check whether $\gamma_t^i < \gamma$ or not for all nodes in the network, where γ is a given system parameter. If $\gamma_t^i < \gamma$, then s_i is classified as an unimportant node that has very low probability to become the maximum node in the next round. Therefore, the sink sends a reporting rate adjustment packet containing γ_t^i to the nodes with $\gamma_t^i < \gamma$. On receiving an reporting rate adjustment request, node s_i raises its filter threshold by

$$h_{t+1}^i = (1 - \gamma_t^i) \cdot v_t^{max} + \gamma_t^i \cdot h_t^i \quad (1)$$

Table I
GLOBAL MAXIMUM TIMES TABLE FOR 70 ROUNDS

Node	1	2	3	4	5	6	7	8
Frequency	5	0	5	0	0	0	40	20

B. Classification of Important and Unimportant Nodes

In this subsection, a strategy will presented to classify the nodes as important and unimportant nodes. In real world, most scenarios, the locations where the maximum values occurred are quite stable (i.e. the distribution of monitored value would not change rapidly). For example, the maximum temperature of a factory usually stays in the same region and the maximum humidity of a rainforest may stay in the same region as well. With this observation, we can take advantage of this phenomenon. We endow the sink node with the ability to learn the importance of all nodes. In short, we deem that a node is important if it becomes maximum node frequently. Similarly, a node is classified as an important node if it has very low chance to become maximum node. Hence, the sink node will keep a *maximum times table* used to store the times of each node being the maximum node.

According to this maximum times table, the sink node could differentiate important nodes from all nodes. However, in most cases, recent observations are much more important than older observations. Therefore, an additional table is also utilized to store recent observations in the sink node. The older observations are in *global maximum times table*, and the recent observations are stored in *temporal maximum times table*. Table I and Table II show an example of global maximum times table and temporal maximum times table. By using the maximum times tables, a node could be classified by computing the important factor as follows.

$$\gamma_i = w \cdot P(G_i) + (1 - w) \cdot P(T_i), w \in [0, 1] \quad (2)$$

γ_i represents the importance of a node (the probability of this node to become maximum node). $P(G_i)$ and $P(T_i)$ denote the probabilities of node s_i being the maximum node in Global Maximum Times Table and Temporal Maximum Times Table, respectively. And w is the weight of Global

Table II
TEMPORAL MAXIMUM TIMES TABLE FOR RECENT 20 ROUNDS

Node	1	2	3	4	5	6	7	8
Frequency	0	0	0	0	0	0	15	5

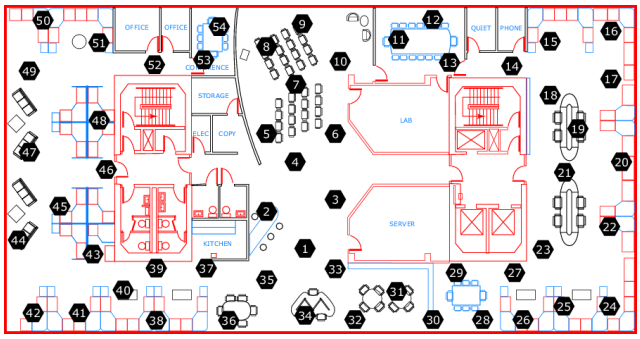

Figure 1. Sensor deployment in Intel Berkerly lab dataset.

Maximum Times Table. In practice, we would set a threshold to γ_i. A node s_i would be classified as an important node, if γ_i break this threshold. Otherwise, it would be classified as an unimportant node.

C. Adjustment Strategy of Reporting Rate

Three strategies will be adopted in the proposed method to dynamically adjust the reporting frequency of sensors according to their classification [21]–[30].

1) Round basis: The round basis strategy is to transmit the regulation packets periodically. Although this strategy could response the burst traffic in time, it also incurs a lot of control overheads. Furthermore, this strategy may transmit redundant control messages when the maximum value decreases.

2) Linear regression: As mentioned previously, the HAT would generate a lot of update messages when the maximum value suddenly increases since many nodes break their local filters. In other words, the sink node could send control messages only if the maximum value increases aggressively. Base on this observation, linear regression approach could be utilized to compute the trend of maximum values.

3) Cumulative Sum (CUSUM): CUSUM is a popular method used for change detection. Before change detection, the exponential weighted moving average could be applied to the historical maximum values to smooth short term fluctuations. This paper utilized CUSUM to detect if the smoothed values increase aggressively.

IV. PERFORMANCE EVALUATION

In this section, several simulation results will be presented to demonstrate the performance of proposed algorithm. The NS-2 network simulator [20] was used to simulate a wireless network environment. The temperature-data traces provided by the Intel Berkeley Lab Dataset [4] were used as the

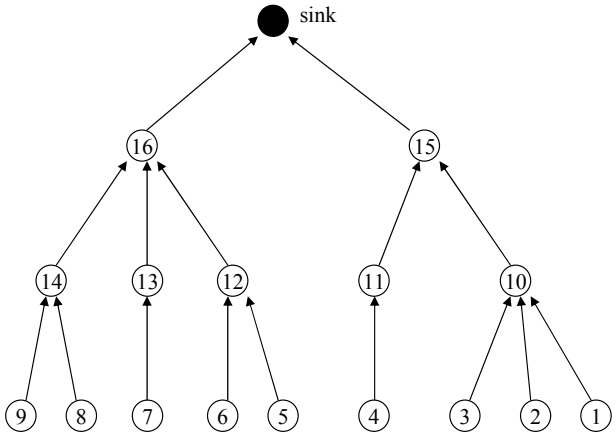

Figure 2. Network Topology.

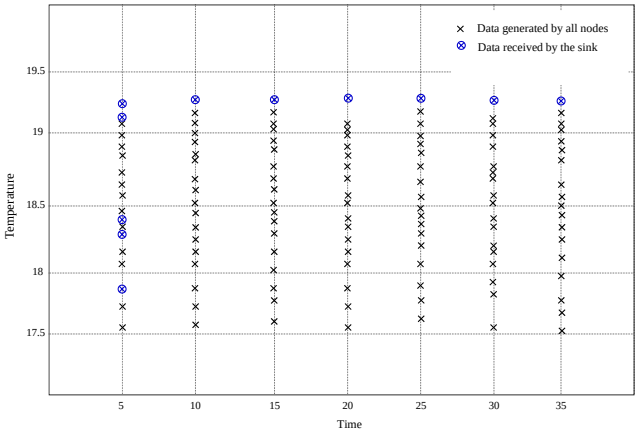

Figure 3. Performance of HAT.

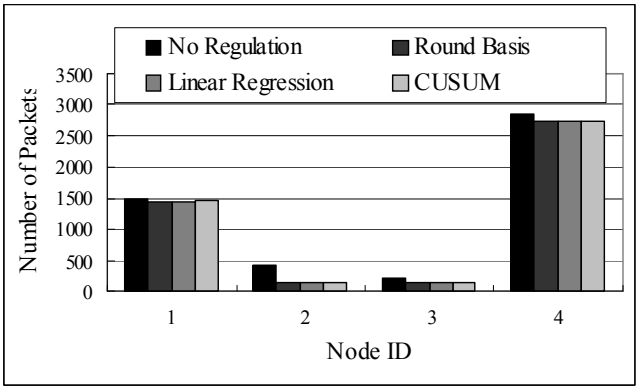

Figure 4. Number of network packets transmitted by nodes 1-4.

Table III
THE OCCURRENCE HISTOGRAM OF MAXIMUM VALUES

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Frequency	185	0	0	433	0	0	20	0	0	0	0	0	0	19	37	0

Table IV
ERROR RATE COMPARISON

Method	Round basis	Linear regression	CUSUM
Maximum error	0.102	0.102	0.06
Average error	0.04	0.05	0.04
Error rate	1.34 %	0.89 %	0.45 %

test data on which to run the proposed algorithm. The data traces consist of temperature readings, which were regularly collected from 54 nodes spread around the lab, as depicted in Fig. 1. However, there are some missing values for a few of nodes in the original data traces. Therefore, the missing values were replaced with the average value from the previous and subsequent readings. The simulated network topology was depicted in Fig. 2, which consists of the sink node and 16 sensor nodes. The 802.11 and UDP network protocols were used as the simulated MAC protocol and communication protocol, respectively. The sensor nodes were assumed to have a maximum transmission range of 25m.

In the simulations, every four seconds are called a round, and the maximum query is executed round-by-round, i.e., the maximum query is executed every four seconds. The sampling rate and reporting rate are set to two packets per second. Two data segments of Intel Berkeley Lab Data [4] were selected randomly as the input data in each simulation, and the simulator was executed 20 times to obtain the average results. Each data trace contains nearly 5500 temperature values (i.e., simulation time is about 2700 seconds). The length of Temporal Times Table is set to 10 to store ten historical values. The Global Times Table was set to contain all historical data except the data recorded in Temporal Times Table. The weight of the past data of the weighted exponential moving average is set to 0.2. If the importance of a node i is smaller than 0.2, the reporting rate of the node will be set to 2/5 packets per second as compared to the original reporting rate. If the importance of a node i is between 0.2 and 0.35, the reporting rate of the node will be set to 2/3 packets per second. Otherwise, the reporting rate keeps unchanged, i.e., two packets per second. The regulation packets will be sent once the classification of nodes has been changed since last transmission. The first 50 rounds are training stage that the initial regulation packets will be sent after the training phase. When the maximum node consistently replies their readings to the sink node, it will result in heavy loads on its parent node. We adopt a simple balance traffic mechanism which can share these traffics among all possible parent nodes of the maximum node.

Firstly, we demonstrate the basic function of HAT in Fig. 3. Hence, we set slow sampling rate and regulation rate (0.33 packet/second) with an attempt to illustrate the results clearly. In Fig. 3, we observed that at the first time

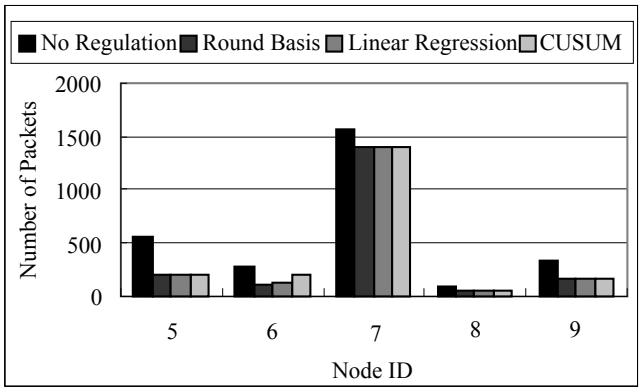

Figure 5. Number of network packets transmitted by nodes 5-9.

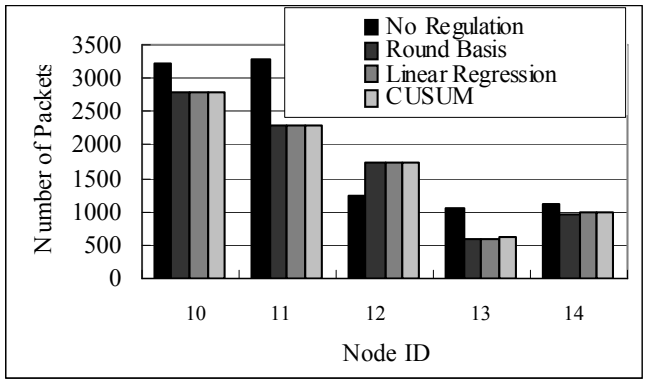

Figure 6. Number of network packets transmitted by nodes 10-14.

unit there are only five packets received by the sink node instead of 16 packets. As mentioned previously, HAT is a double sided filter which can remove the values having no chance to compete for maximum. Thus, there are only five packets can break the thresholds along the path to the root. After the decision of maximum, the sink node sends a maximum designated packet to the node having maximum value. After receiving the maximum designated packet, the maximum node reports its readings if it has changed since last transmission. Thus, we can notice that the readings of maximum node are transmitted to the sink node continuously. Besides, after the first time unit, there are no other packets received by the sink node except the

 ISBN: 978-1-61208-207-3

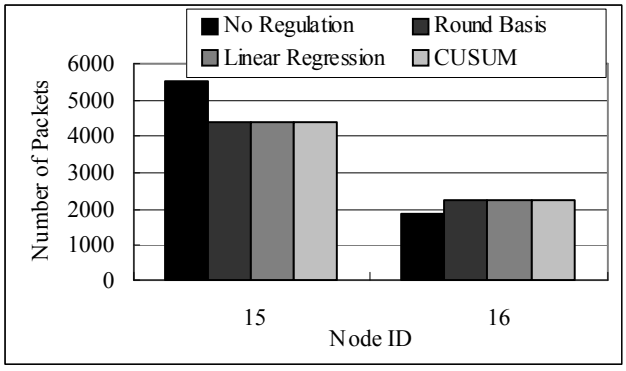

Figure 7. Number of network packets transmitted by node 15 and node 16.

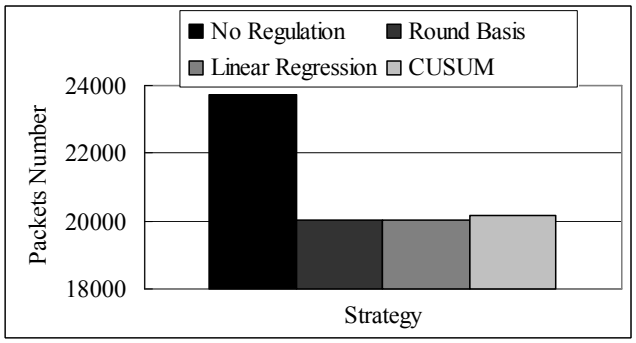

Figure 8. Total number of network packets.

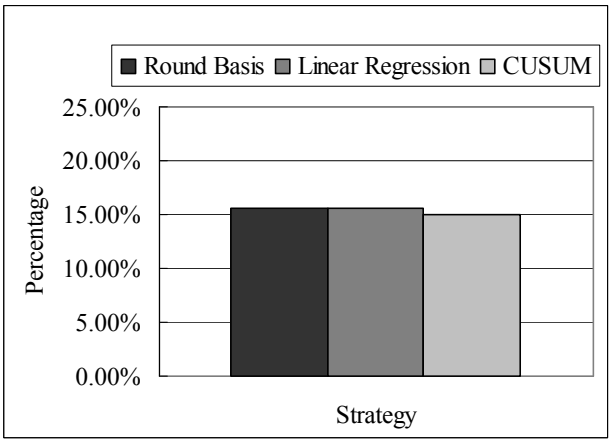

Figure 9. Network packet reduction.

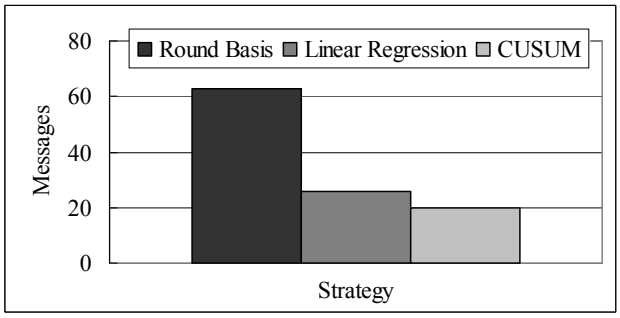

Figure 10. Control overhead.

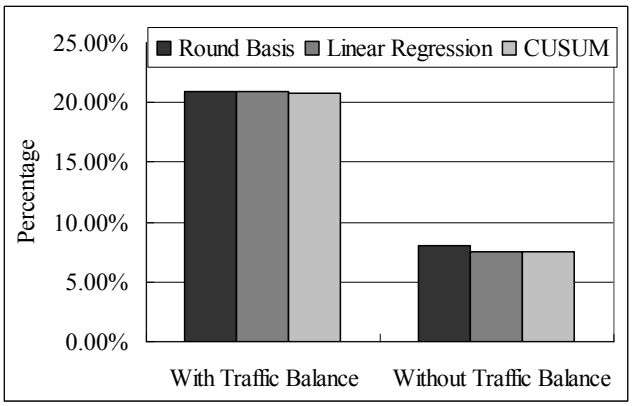

Figure 11. Packet reduction on node 15 and node 16.

Figure 12. Packet reduction on nodes 10-14.

packets generated by the maximum node. The reason of this phenomenon is that the temperatures sensed by these nodes are decreasing. Hence, they cannot break their thresholds resulting in no other packets arrived at the sink node.

In the following, the proposed method with these three adjustment strategies of reporting rate is evaluated. In Fig. 4, it can be seen that the reduction on network traffic is nearly equal using the three adjustment strategies. This is because that these nodes become maximum node frequently, as shown in Table III. Hence, their reporting rates will be kept as normal reporting rate in order to reduce error rate. In Fig. 5, the three strategies obviously reduce the network

 ISBN: 978-1-61208-207-3

traffic as compared with no regulation. This is because the sensed data of the nodes have little chance to become the maximum one. Accordingly, the reporting rate of the nodes will be reduced with an attempt to save energy.

In Fig. 6, the three adjustment strategies incur additional network traffic at node 12 as compared with no adjustment. The reason why our methods generated more packets at node 12 is that we adopt a traffic balance method which can share the traffic loads generated by the maximum node to all of its possible parents. For instance, when node 4 becomes the maximum node, it will transmit its data values continuously to all of its possible parents (i.e., nodes 10, 11, 12) instead of putting all loads on its direct parent (node 11). Thus, a lot of network packets could be avoided on the direct parent of the maximum node. The same phenomenon can be observed in 7. Fig. 8 shows the total number of reduced network packets. Fig. 9 suggests that the proposed strategies could obtain nearly 15% packet reduction. Among the three adjustment strategies, Round Basis has the highest percent of reduction. However, it also incurs the most overhead as well in Fig. 10.

The control overhead of Linear Regression method and CUSUM detection method is 41% and 36% as compared to Round Basis. After the illustration of packet reduction, we illustrate error rate, maximum error and average error due to packets reduction in Table IV. From Table IV, we can notice that the highest error rate is Round Basis. This is because that Round Basis reduces the most packets among the three regulation policies. Thus, it incurs the highest error rate. In addition to the error rate comparison, we further compare the method with traffic balance and without balance in Fig. 11 and Fig. 12.

In Fig. 11, node 15 and node 16 are two nodes one-hop away from the sink. In Fig. 12, nodes 10-14 are the nodes two-hop away from the sink. As can be seen, the reporting rate regulation combined with traffic balance scheme could achieve better balance than only reporting rate regulation methods. Furthermore, if we purely implement reporting rate regulation but not combined with traffic balance scheme, the extended life time of this network is limited. This is because that the nodes regulated to slow reporting rate are always the nodes with low probability to become maximum node. Hence, if we do not combine our method with a traffic balance scheme, the improvement of life time is limited.

V. CONCLUSIONS

The performance of the proposed algorithm was evaluated using NS-2 network simulator [20] and real-world data traces. The results indicate that the overall network packets are reduced to 80% with 1% data error in comparison with HAT. Besides, this paper also further keeps the error caused by reduced packets transmission below 1%.

According to our observation, a node will likely become the maximum node in the subsequent rounds once it is designed as the maximum node in the current round. This suggests that the maximum node would put heavy loading on its immediate parent due to continuous monitoring (reporting) of the maximum node. As a result, some nodes in the network would exhaust their power more quickly than the others. In this case, the balancing of communication tree should be taken into consideration in the proposed method in the future.

ACKNOWLEDGMENT

The research reported in this article was partially supported by NSC under Grant No. 100-2218-E-006-035 and by the Ministry of Economic Affairs in Taiwan under Grant No. B301AR2E10.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE communications magazine, vol. 40, pp. 102114, Aug 2002.

[2] C. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," in Proc. of IEEE, vol. 91, no. 8, pp. 12471256, Aug 2003.

[3] A. Silberstein, K. Munagala, and J. Yang, "Energy-Efficient Monitoring of Extreme Values in Sensor Networks," in Proc. ACM SIGMOD Intl Conf. on Management of Data, pp. 169-180, 2006.

[4] Intel Berkeley Research Lab. http://berkeley.intel-research.net/labdata/

[5] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A tiny aggregation service for ad-hoc sensor networks," in Proc. USENIX OSDI02, pp. 131146, Dec. 2002.

[6] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis, "TiNA: A scheme for temporal coherency-aware in-network aggregation," in Proc. ACM MobiDE03, Sep. 2003, pp. 6976.

[7] A. Deligiannakis, Y. Kotidis, N. Roussopoulos, "Compressing Historical Information in Sensor Networks," SIGMOD04, pp. 527-538, 2004.

[8] A. Deligiannakis, Y. Kotidis, N. Roussopoulos, "Dissemination of compressed historical information in sensor networks," VLDB J. vol. 16, no. 4, pp. 439-461, 2007.

[9] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," in Proc. IEEE ICDE04, pp. 449460, Mar. 2004.

[10] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in Proc. ACM SenSys04, pp. 250262, Nov. 2004.

[11] S. Yoon, C. Shahabi, "The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks," ACM Trans. On Sensor Networks, vol. 3, no. 1, article 3, Mar. 2007.

 ISBN: 978-1-61208-207-3

[12] P. Cao and Z. Wang, ”Efficient Top-k Query Calculation in Distributed Networks,” in Proc. ACM SIGACT-SIGOPS Symp. Principles on Distributed Computing (PODC 04), July 2004.

[13] R. Cheng, B. Kao, S. Prabhakar, A. Kwan, and Y.-C. Tu, ”Adaptive Stream Filters for Entity-Based Queries with Non-Value Tolerance,” in Proc. 31st Intl Conf. Very Large Data Bases (VLDB 05), 2005.

[14] R. Fagin, A. Lotem, and M. Naor, ”Optimal Aggregation Algorithms for Middleware,” in Proc. ACM Symp. Principles of Database Systems (PODS 01), Aug. 2001.

[15] U. Gu ntzer, W.-T. Balke, and W. Kie ling, ”Optimizing Multi-Feature Queries for Image Databases,” in Proc. Intl Conf. Very Large Data Bases (VLDB 00), 2000.

[16] B. Babcock and C. Olston, ”Distributed Top-k Monitoring,” in Proc. ACM SIGMOD Intl Conf. Management of Data (SIGMOD 03), pp. 28-39, June 2003.

[17] C. Olston, J. Jiang, and J. Widom, ”Adaptive Filters for Continuous Queries over Distributed Data Streams,” in Proc. ACM SIGMOD Intl Conf. Management of Data (SIGMOD 03), pp. 563- 574, June 2003.

[18] M. Wu, J. Xu, X. Tang, and W.-C. Lee, ”Top-k Monitoring in Wireless Sensor Networks,” IEEE Trans. on Knowledge and Data Engineering, vol. 19, no. 7, July 2007.

[19] A. Silberstein, R. Braynard, C. Ellis, K. Munagala, and J. Yang, ”A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks,” in Proc. Intl Conf. Data Eng. (ICDE 06), Apr. 2006.

[20] The Network Simulator NS-2. http://www.isi.edu/nsnam/ns.

[21] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, Forecasting: Methods and Applications -3rd edition, John Wiley and Sons, Inc., 1998

[22] C. C. Zou, W. Gong, D. Towsley, and L. Gao, ”The Monitoring and Early Detection Internet Worms,” in IEEE/ACM Trans. on Networking, Oct. 2005.

[23] H. Wang, D. Zhang, and K.G. Shin, ”Change-Point Monitoring for Detection of DOS Attacks,” in IEEE Trans. on Dependable and Secure Computing, vol. 1, no. 4, 2004.

[24] Z. Chen, L. Gao, and K. Kwiat, ”Modeling the Spread of Active Worm,” in Proc. IEEE Intl Conf. on Computer Communications (INFOCOM), 2003.

[25] J. O. Kephart, and S. R. White, Measuring and Modeling Computer Virus prevalence, in Proc. IEEE Symp. on Security and Privacy, pp. 2-15, 1993.

[26] T. Bu, A. Chen, S. V. Wiel, and T. Woo, ”Design and Evaluation of a Fast and Robust Worm Detection Algorithm,” in Proc. IEEE Intl Conf. on Computer Communications (INFOCOM), 2006.

[27] C.T. Ee and R. Bajcsy, ”Congestion Control and Fairness for May-to-One Routing in Sensor Networks,” in Proc. Intl Conf. on Embedded Networked Sensor Systems (ACM SenSys), Nov. 2004.

[28] C. Wang, B. Li, K, Sohraby, M. Daneshmand, and Y. Hu, ”Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization,” IEEE journal on Selected Areas in Communications, vol. 25, no. 4, May 2007.

[29] B. Hull, K. Jamieson, and H. Balakrishnan, ”Mitigating Congestion in Wireless Sensor Networks,” in Proc. Intl Conf. on Embedded Networked Sensor Systems (ACM SenSys), 2004.

[30] K. Karenos, V. Kalogeraki, and S. V. Krishnamurthy, ”Cluster-based Congestion Control for Supporting Multiple Classes of Traffic in Sensor Networks,” in Proc. 2nd IEEE workshop on Embedded Networked Sensors, 2005.

[31] J. Gao, and L. Zhang, ”Load Balanced Shortest Path Routing in Wireless Networks,” in Proc. IEEE Intl Conf. on Computer Communications (INFOCOM), 2004.

 ISBN: 978-1-61208-207-3