Adaptive Data Storage for Querying in Wireless Sensor Networks

Zhidan Liu Wei Xing* Dongming Lu Yongchao Wang
College of Computer Science and Technology, Zhejiang University, Hangzhou, China
E-mails: {danielliu, wxing, ldm, ychwang}@zju.edu.cn

Abstract—**In this paper, an adaptive storage method which can dynamically adopt suitable storage mode based on the computation of potential costs according to statistics of event data rate and query rate is proposed. We envision queries for event data following a biased distribution, i.e. *Zipf-like* distribution, just like the situation in real file sharing systems. In our proposed method, system can switch the storage mode between external storage and *data-centric* storage adaptively for different types of event data according to their popularity. Simulation results demonstrate that our proposed method is a feasible and energy-efficient data storage method.**

I. Introduction

Recent technology trend in new generation NAND flash memories makes sensor nodes equipped with several gigabytes of low-power flash storage feasible, meanwhile depresses energy cost of storage to a comparable degree as computation which is two orders of magnitude cheaper than communication [1]. These advances indicate that data storage can be an appropriate method to archival sensed data and process queries. Basically, all data storage methods can be classified into following three canonical data storage methods [2]: (1) *External Storage* (*ES*), in which mode sensor nodes send event data to specific external storages, referred as *Sink* nodes and future queries can be performed there too; (2) *Local Storage* (*LS*), in which mode sensor nodes store event data at internal storages upon events are detected, and queries are broadcasted into the whole WSN when data request is issued; (3) *Data-Centric Storage* (*DCS*), where event data are named and storage and queries are both performed according to this name. These methods have very different energy cost structures and each one suits for different application conditions according to the frequencies of event and query [3] [4] [5].

Research reveals that the query distribution in a P2P file sharing system follows a very biased distribution [6], e.g., *Zipf-like* distribution, and we argue that this kind of skewed query distribution could hold true for querying event data in WSNs, as it's reasonable to assume that some types of data are more popular than others and then are queried more times. In this paper, we propose an adaptive data storage method which dynamically adjusts storage mode between ES and DCS for each type of event data according to the potential costs of both storage modes when skewed query distribution exists. We expect that event data of more popular types can be stored in the region near the query entrance which is efficient in executing queries while data of less popular types can be stored by name in network, which can route queries directly without being flooded into the whole network. Our aim is to reduce the communication traffic and save sensor nodes' energy to the best efforts.

The remainder of this paper is organized as follows: section II briefly describes the related works on data storage methods in WSNs. In section III, we present the adopted network model and some assumptions. The proposed adaptive data storage method is elaborated in section IV. Performance evaluation of our adaptive data storage method is presented in section V. We conclude the whole paper in section VI.

II. Related Works

A representative example of *data-centric* routing is Directed Diffusion [7], which routes event data based on the name rather on the identity of destination node. As event data are stored in the detecting node, Directed Diffusion can be considered as an energy-efficient data dissemination method compared to flooding in LS. *Data-centric* storage is a similar concept as *data-centric* routing, and Geographic Hash Table (GHT) [3] is a classical implementation of DCS. In GHT, event data are associated with keys, and GHT hashes keys into geographic coordinates. Based on geographic routing protocol such as GPSR [8], GHT uses ***Put(k, v)*** and ***Get(k)*** operations to store and query event data associated with *k* at the sensor node which is geographically closest to the hashed coordinates.

Combination of the aforementioned three canonical data storage methods yields hybrid storage methods which can save energy further because they can adaptively adjust to the application scenarios. Gwo-Jong Yu [4] proposed an adaptive storage policy which switches between LS and DCS and Scoop [5] can switch between LS and ES adaptively. Both of these methods rely on some switching strategies built on analyzing of the collected information about event generating frequency and querying frequency. However, switching rules in adaptive storage policy only consider the numbers of events and queries in a period time and the adopted storage mode is applied to the whole network. In Scoop, *Sink* node needs to collect statistics from all sensor nodes and create storage assignments for each node periodically. As all statistics are shipped to *Sink* node and storage assignments are created and disseminated only by *Sink*, we can imagine Scoop will encounter the same bottleneck as ES method, i.e., nodes close to *Sink* will exhaust energy more quickly than others.

* Corresponding author.

978-1-61284-683-5/12/$31.00 ©2012 IEEE

III. Network Model and Assumptions

In this paper, we consider a static $2-dimension$ wireless sensor network with a single *Sink* node as the external storage point and the only entrance for data queries and exit for data replies. We assume that *Sink* node is equipped with more storage capacity and computation capability. *Sink* node locates at coordinates $(0,0)$ and other sensor nodes in the network are distributed uniformly in a determinate space. We assume that all of the sensor nodes have the same transmission range and equal initial energy. In our network model, each sensor node is embedded with two storage modes, ES mode and DCS mode, and can switch between them according to our adaptive storage algorithm. We assume that all nodes know their geographic location which can be realized through the equipped GPS or other localization techniques. We adopt the modified GPSR described in GHT as our underlying routing protocol.

We assume that there are several event types needed to be sensed. Each sensor node is assigned with a specific event type and generates event data of the assigned event type randomly following *Poisson* distribution. We assume queries for event types follow *Zipf-like* distribution with parameter α.

We model the event data queries as $Query(k, \Delta t)$, in which k means clients' interested event type and Δt indicates the freshness of event data that meet the requirement of queries. System will reply this kind of queries with the latest event data and summary information of event data during the time range. Summary information could be average value or total number of these data gathered during the time span.

IV. Adaptive Data Storage Design

A. Overview of Adaptive Data Storage Method

We propose adaptive data storage method (denoted as ADS), in which sensor nodes can adjust storage mode switching between DCS mode and ES mode according to the event rate and query rate of a specific event type. In our ADS method, nodes are classified into three roles:

- *MEMBER*: common sensor node which generates event data randomly and can forward messages.
- *DCN*: node becomes *DCN* role when it becomes storage node, and it will return to *MEMBER* role when no more event data are sent to this node during a period of time.
- *SINK*: the *Sink* node and will never change its role.

We divide the time into intervals, and during each interval, *DCN* and *SINK* nodes collect statistical information of event rate and query rate for each event type. At the end of each interval, potential costs of ES mode and DCS mode are estimated by *DCN* and *SINK* nodes, and storage mode switching decisions are made based on our adaptive data storage algorithm which will be described in next subsection. If switching decision is made, *DCN* or *SINK* nodes will send switching messages to related sensor nodes. For example, sensor node may switch storage mode from DCS mode to ES mode if its event data are queried frequently in the last several intervals.

B. Adaptive Data Storage Algorithm

We assume there are n nodes to gather information of event type k, and distance between the i-th sensor node and *Sink* node is denoted as $h(ss)_i$ hops which can be obtained from Hello message broadcasted by *Sink* node. During T, number of events generated by the i-th node is denoted as E_i. Then we can compute potential cost of ES mode at *DCN* node or *Sink* node for event type k as follows:

$$pc(ES)_k = \sum_{i=1}^{n} E_i \times h(ss)_i \tag{1}$$

As all event data are sent to the *Sink* node and no energy consumption for queries in network, the potential cost $pc(ES)_k$ is only decided by the event rate. While potential cost of DCS mode is decided by both event rate and query rate, and the computation scheme for event type k is as follows:

$$pc(DCS)_k = \sum_{i=1}^{n} E_i \times h(sd)_i + 2 \times Q_k \times h(ds) \tag{2}$$

In equation (2), $h(sd)_i$ denotes the distance between the i-th node and its storage node, which is decided by the global hash function with event type k. Q_k is the number of queries for event type k , and $h(ds)$ denotes the distance between storage node and *Sink* node. In *DCN* node, this distance is the same as $h(ss)$; while in *SINK* node, we assume the coordinates of *Sink* node is (x_s, y_s), and coordinates of storage node for event type k is (x_k, y_k) which can be obtained by the hash function with parameter k, again we denote the transmission radius of radio as R, then we can estimate the approximate distance as:

$$h(ds) = \frac{\sqrt{(x_k - x_s)^2 + (y_k - y_s)^2}}{R} \tag{3}$$

Until now, we set up the estimated potential costs of ES mode and DCS mode. We can define the cost ratio of both storage modes for this interval time T as:

$$R(k) = \frac{pc(DCS)_k}{pc(ES)_k} \tag{4}$$

As temporal correlation for event distribution and query distribution, recent event rates and query rates are likely to be a good predictor for the future event rate and query rate. Relying on this insight, let $R(k)^t$ to be the cost ratio of event type k in the t-th interval time, and we can define the following function as ADS's prediction function for the next interval time:

$$f(k)^{t+1} = \sum_{i=1}^{t} w_i \times R(k)^i \ , \ \sum_{i=1}^{t} w_i = 1 \tag{5}$$

In equation (5), we assign different weights for cost ratios of different intervals, and the interval closer to current interval is assigned with a greater weight as closer interval can reflect the trend of event generating and querying. To make things easy, we can simplify equation (5) and only consider the last two intervals. We assign the latest interval with weight of 0.8 and the other one with weight of 0.2 following *Pareto Principle*. The simplified form of $f(k)^{t+1}$ could be:

$$f(k)^{t+1} = 0.8 \times R(k)^t + 0.2 \times R(k)^{t-1} \tag{6}$$

If the switching threshold is denoted as Td, then switching rule can be formulized as follows:

$$f(k)^{t+1} < Td - \Delta \tag{7}$$

$$f(k)^{t+1} > Td + \Delta \tag{8}$$

Δ is a tuning factor to avoid *Ping-Pong effect* of storage mode switching between ES mode and DCS mode too frequently. If equation (7) is satisfied, sensor nodes for gathering data of event type k will switch storage mode from ES to DCS. Actually, this kind of switching decisions will happen at *SINK* node. Storage mode switching messages are sent by *Sink* node to the relevant nodes and these nodes will adopt DCS storage mode in the next interval. If equation (8) is satisfied, *DCN* node will send switching messages to those nodes which are responsible to collect data of event type k, and relevant nodes will adopt ES storage mode in the next interval.

V. Simulation

A. Simulation Setup

We implement GHT [3] without structured replication as the DCS storage method in our comparative simulations. To avoid hashing keys to coordinates outside the network or near the boundary, we constrain the hashing space to the center of the network, e.g., the hashed X coordinate could be constrained in range of $[\mu_x R, length - \mu_x R]$ and the hashed Y coordinate could be constrained in range of $[\mu_y R, width - \mu_y R]$, where μ_x and μ_y are selective controllable factors, $length$ and $width$ are the network size.

We carry out our simulations with *ns-2* [9] which includes detailed model of wireless network's MAC and physical layer. In our simulation network model, there are 100 sensor nodes distributed uniformly in a determinate space. At the beginning of simulation, each sensor node is attached with a random event type and the sensor nodes will generate event data randomly following *Poisson* distribution with parameter λ. We ensure that each event type will be allocated with roughly the same number of nodes. Our simulation will last $555s$, and the first $50s$ is reserved for sensor nodes to exchange neighbor information, the last $5s$ is reserved to make sure all replies can be received by *Sink* node. Queries are started at time $50s$, and queries for event types are randomly generated following *Zipf-like* distribution with parameter α. Besides, we adopt the same energy model as [4]. Table I presents the detailed parameters of our *ns-2* simulations.

B. Performance Evaluation

As what we expected, our adaptive storage method works well and the comparative simulation results demonstrate that ADS is a competitive and energy-efficient data storage method when compared with sole ES method and DCS method. Besides, ADS offers nearly perfect availability of queried event data. Fig. 1 shows that our method outperforms the other two methods on both average energy consumption and maximum energy consumption. From Fig. 1, we observe the hot-spot problem in ES method as the maximum energy consumption exceeds the average energy consumption of ES method quite a lot. Table II lists simulation results of other evaluation metrics, and we can find that our method owns a high success reply rate

TABLE I
Simulation Parameters

Parameters	***Value***
Routing Protocol	$GPSR$
Network Size	$200m \times 200m$
Number of Nodes	100
Radio Transmission Range(R)	$40m$
λ of Event *Poisson* Distribution	1.0
Event Types	20
Simulation Time	$555s$
Interval T	$10s$
Total Queries	2000
Queries Generation Rate	$4qps$
α of Query *Zipf-like* Distributions	$0, 0.5, 1, 1.5$ $2, 2.5, 3, 3.5, 4$
Tuning Factor Δ	0.1
Initial Energy	$5.0J$
Tx Power Consumption	$0.85mW$
Rx Power Consumption	$0.10mW$

TABLE II
Comparison of other metrics for various storage methods. Results are the means of 10 simulations on *Zipf-like* query distributions with parameter α.

Storage Method	***Success Reply Rate***	***Sink Reply Rate***	***Total Msgs***
ES	100%	100%	182547
DCS-GHT	99.33%	0	114206
ADS	99.66%	45.77%	107022

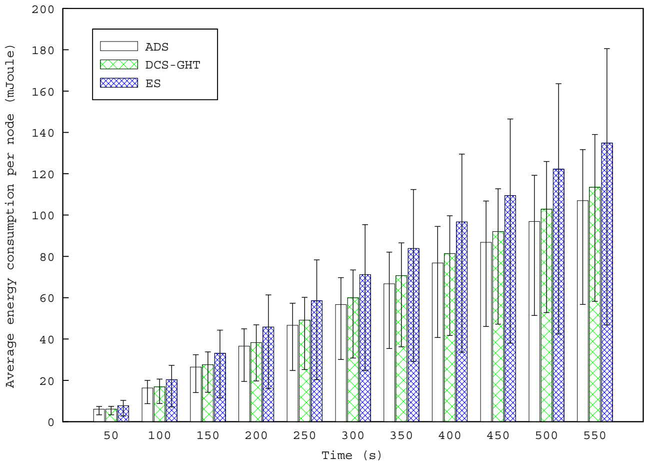

Fig. 1. Comparison of average energy consumption for our proposed ADS method, ES method and DCS-GHT method. Results are the means of 10 simulations on *Zipf-like* query distributions with parameter $\alpha = 1.5$.

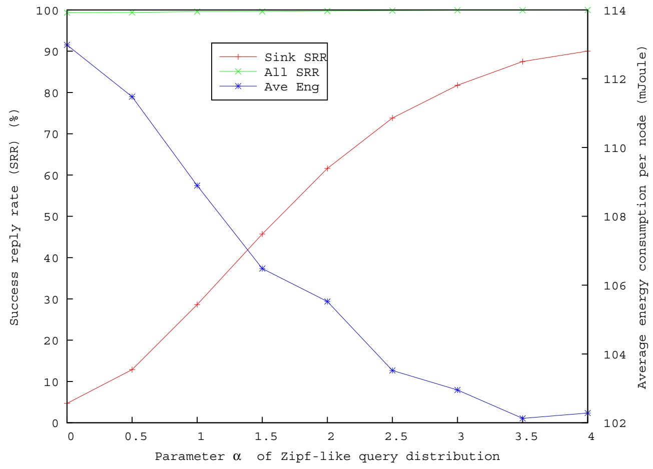

Fig. 2. *Sink* reply rate and average energy consumption per node for *Zipf-like* query distributions with different α. Results are the means of 3 simulations for each value of α.

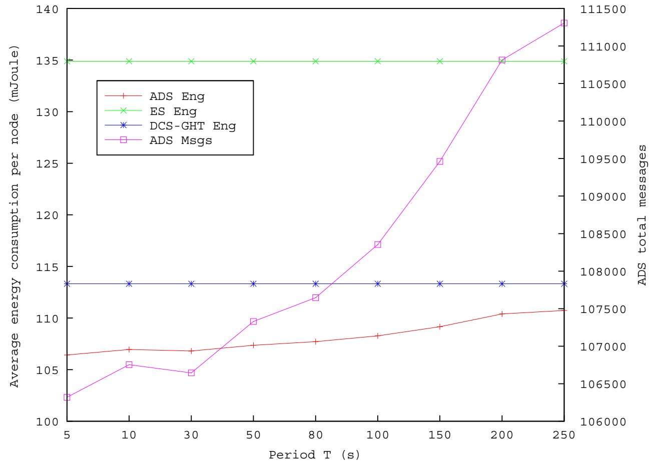

Fig. 3. Relation between period T and average energy consumption per node and total messages of ADS. Results are the means of 3 simulations for each value of T.

nearly 100% but with less communication messages which are in accord with average energy consumptions.

We design the adaptive data storage method based on the insight that in real-world systems, user queries may be skew to some popular event data. In ADS method, popular event data can be stored to *Sink* node which is the query source and if these data are not popular in future, they can be shifted to storage nodes in network according to DCS-GHT storage method. Fig. 2 demonstrates that our method can adaptively adopt suitable storage mode for event data according to their popularity degree. With the same event generation model, our method can preserve more energy if queries are concentrated on minority event types. As the value of α increases, the popular event data can be queried at the *Sink* node and less query messages will be sent into the network. In Fig. 2, as *Sink* reply rate increases along with increase of α, the average energy consumption decreases. Furthermore, we can find that our proposed method performs stably with high SRR approaching to 100% on various *Zipf-like* query distributions.

Theoretically, there are two important tunable parameters in our adaptive storage algorithm: Δ and T. Δ decides in what condition the storage mode changes and T affects how often storage mode switching checking will happen. However, we observe that there are no obvious difference on energy consumption among various values of Δ through our simulations. Fig. 3 presents the relation between T and energy consumption. Along with increase of period T, the average energy consumption increases at the same time, but the energy consumption is still less than DCS-GHT method ($113.3mJoule$) and much less than ES method ($134.9mJoule$). Actually, a smaller T will make system adjust storage mode in time according to the rates of event and query, though more controlling messages will be generated. On the other hand, a lager value of T will generate less controlling messages, but it makes the system less sensitive to the storage requirements of network. We argue that a suitable value of period T depends on the event rate and query rate of reality applications.

VI. Conclusions

In this paper, we propose an adaptive data storage method which switches storage mode between ES and DCS based on the computation of potential costs according to the statistics of event data rate and query rate. Simulations conducted with *ns-2* demonstrate that our ADS method outperforms the pure ES method and DCS method and can save more energy. The objective of data storage is to facilitate querying, and our proposed method can store different types of event data at the right places, which are more convenient for processing queries and energy-preserving, according to their popularity.

References

[1] Y. Diao, D. Ganesan, G. Mathur, and P. Shenoy. Rethinking data management for storage-centric sensor networks. In *Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR)*. Citeseer, 2007.

[2] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensornets. *ACM SIGCOMM Computer Communication Review*, 33(1):137–142, 2003.

[3] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A geographic hash table for data-centric storage. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 78–87. ACM, 2002.

[4] G.J. Yu. Adaptive storage policy switching for wireless sensor networks. *Wireless Personal Communications*, 48(3):327–346, 2009.

[5] T.M. Gil and S. Madden. Scoop: A hybrid, adaptive storage policy for sensor networks. Technical report, Citeseer, 2006.

[6] A. Klemm, C. Lindemann, M.K. Vernon, and O.P. Waldhorst. Characterizing the query behavior in peer-to-peer file sharing systems. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 55–67. ACM, 2004.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67. ACM, 2000.

[8] B. Karp and H.T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.

[9] S. Mccanne and S. Floyd. *ns* network simulator. http://isi.edu/nsnam/ns/.