INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1384

Energy-efficient and localized lossy data aggregation in asynchronous sensor networks[‡]

J. Zhang [1,2,*,†], X. Shen [1], H. Zeng [1], G. Dai [1], C. Bo [1], F. Chen [1] and C. Lv [1]

[1] *College of Computer and Software, Hangzhou Dianzi University, Hangzhou 310018, China*
[2] *SITE, University of Ottawa, Ottawa, Canada K1N 6N5*

SUMMARY

In wireless sensor networks, most data aggregation scheduling methods let all nodes aggregate data in every time instance. It is not energy efficient and practical because of link unreliability and data redundancy. This paper proposes a lossy data aggregation (LDA) scheme to reduce traffic and save energy. LDA selects partial child nodes to sample data at partial time slots and allows estimated aggregation at parent nodes or a root in a network. We firstly consider that all nodes sample data synchronously and find that the error between the real value of a physical parameter and that measured by LDA is bounded respectively with and without link unreliability. Detailed analysis is given on error bound when a confidence level is previously assigned to the root by a newly designed algorithm. Thus, each parent can determine the minimum number of child nodes needed to achieve its assigned confidence level. We then analyze a probability to bound the error with a confidence level previously assigned to the root when all nodes sample data asynchronously. An algorithm then is designed to implement our data aggregation under asynchronization. Finally, we implement our experiment on the basis of real test-beds to prove that the scheme can save more energy than an existing algorithm for node selection, Distributive Online Greedy (DOG). Copyright © 2012 John Wiley & Sons, Ltd.

Received 12 December 2010; Revised 29 August 2011; Accepted 22 October 2011

KEY WORDS: lossy data aggregation; energy efficiency; data sampling; link unreliability; wireless sensor networks

1. INTRODUCTION

In wireless sensor networks (WSNs), data processing in a sensor node usually consumes much less power than data transmitting in each period [1]. Thus, it can save energy and reduce in-network interference by applying data aggregation [2] or compression before transmitting data. A number of novel data aggregation methods have been proposed recently with various optimization goals, such as reducing the energy consumption [2–4] and the delay of data aggregation [5, 6]. Most of data aggregation schemes argue that all sampled data can be successfully transmitted to a root, but it is not always feasible to query all nodes in real applications [7] as this can result in a great deal of resource consumption. Because links are not always reliable [8] and some nodes cannot work normally all through in some practical applications, some packets are inevitably and unpredictably lost by some nodes at some sampling periods or time slots with probability P_l. The final data received by parent nodes are fragmentary. In this paper, we do not analyze how to obtain these probability P_l because there are some previous papers analyzing link reliability through estimating bit error rate [9] or packet-receiving rate [10, 11]. Data aggregation methods typically let each node sample data at all slots and collect the data from all of their child nodes. Meanwhile, the sample rate is

*Correspondence to: J. Zhang, College of Computer and Software, Hangzhou Dianzi University, Hangzhou 310018, China.
†E-mail: jhzhang.zju@gmail.com
‡The conference version of this paper was previously accepted by MSN 2010.

always limited by the network capacity [12], so aggregated data are easy to be delayed before it is transmitted to the root by multi-hop fashion.

Because WSNs have exiguous recourses, another unavoidable question is whether it is worthy collecting high-precise information when relatively low-precise information is acceptable. Meanwhile, any electric products, such as sensors and sensor nodes, have 'observational error' [13] when they measure certain parameters, such as temperature and humidity. Observational error is an inherent part of measurement. When its randomness or uncertainty is modeled by probability theory, one can standardize statistical errors of some kinds of distribution, such as normal or Gaussian distribution. In fact, sensor nodes are measurement tools because each node is equipped with several sensors, which are used to measure physical parameters in its surroundings. Thus, we can use probability distributions to estimate the measurement error among sensor nodes. In this sense, it is advisable to collect data from a part of nodes or a part of data sampled in each time slot. This kind of scheme is called *lossy data aggregation* (LDA), which can save much energy when a part of nodes consume energy on sampling at part of time. We studied LDA under synchronization case in [14], which did not consider LDA under asynchronization.

In this paper, we study LDA under asynchronization based on some previous results under synchronization [14] and consider both cases of link reliability and unreliability. We firstly design LDA under a case where there is only a part of nodes sampling data in part of time and all nodes transmit data under synchronization. When each link is reliable, we positively select a part of nodes to sample data in whole work time or in a part of time. Meanwhile, all nodes in a whole network still afford of communication task. When each link is unreliable with a certain probability P_l, we analyze an error under a given confidence level when we positively adopt retransmission scheme. In this paper, the error is an estimated value to indicate the difference between the measured value of a parameter and its real one. Notice that the reliability of each link can be estimated according to its packet-receiving rate [15]. Thus, P_l can accordingly be obtained. Here, an interesting problem is how to determine the probability P_r of the successful retransmission on each link according to the value of P_l, which can be obtained previously. We establish the relation between P_l and P_r when the error under a certain confidence level is previously required. Thereafter, we study the confidence level to bound the error when all nodes sample data asynchronously. The main contributions of this paper are as follows:

1. When each node samples data and clocks of all nodes are synchronous, we analyze the error bound $\wp$ between the value D of a parameter measured by LDA and its real value D' under certain confidence level $1-\gamma$.
2. Under both of link reliability and unreliability, we give theoretical support for the confidence level allocation among *parent* nodes when the errors admitted on these parent nodes are given beforehand. An algorithm is designed to allocate the confidence level among different parent nodes.
3. We analyze the confidence level under asynchronous data sampling when a certain error is required. Furthermore, an algorithm is designed to find a uniform duration for all nodes, during which the sampled data are aggregated.
4. We establish a real test-bed of TelosB nodes to evaluate our scheme on energy efficiency and errors under different confidence level. Section 6 shows that different amount of energy can be saved when different confidence levels are required.

The rest of this paper is organized as follows. The network model is defined in Section 3, which also presents a data aggregation scheme. Our scheduling algorithms are presented and analyzed under the cases of synchronous and asynchronous sampling respectively in Sections 4 and 5. In Section 6, detailed experiments are designed to evaluate the performance of our algorithms. Section 2 outlines some related works. Section 7 concludes this paper and discusses some future works.

2. BACKGROUND

In WSNs, data aggregation has been well studied in recent years [3, 5, 16–19], which aims to reduce energy consumption and latency. However, a few of works focused on LDA.

DOI: 10.1002/dac

The first main purpose of in-network aggregation is to decrease energy consumption on communication [2]. Instead of transmitting raw data to a root, it can save much energy and decrease in-network interference to compute and transmit partially aggregated data. Hu *et al*. proposed a heuristic algorithm for constructing data aggregation trees that minimize total energy cost under a latency bound and computed the worst case delay for a sensor node to aggregate data from all its child nodes in the aggregation tree based on an analytic model for IEEE standard 802.15.4 [20]. The second main purpose is to bound the latency of data aggregation. In order to decrease the latency, Yu *et al*. developed a distributed collision-free schedule with the latency bound of $24D + 6\Delta + 16$, where D is the network diameter and Δ is the maximum node degree [6]. The tradeoff between energy consumption and time latency was also analyzed in [21]. In data aggregation, it is popular to construct a tree on the basis of a network by finding a connected dominating set (CDS) [22]. Ma *et al*. constructed an energy-based CDS and resulted in better performance [23]. Because collision and interference are unavoidable in WSNs [24], Lee *et al*. designed a collision-free scheduling when data collection was implemented [25].

Tang *et al*. differentiated the precision of data collected from different sensor nodes to balance their energy consumption [26]. Apoorva *et al*. gave a brief idea to decrease energy by letting a node transmit data in each size-optimized cluster structure [27]. But, it did not consider that the link is unreliable, and nodes may sample data asynchronously. Asynchronous data aggregation results in difficulty because the data sampled from the surrounding are spatio-temporal relation. By exploiting the relation, Golovin *et al*. presented a distributed algorithm for repeatedly selecting sensors online, only receiving feedback about the utility of the selected sensors [28]. In this way, the selected nodes only needed to sample data whereas the utility of a whole network was guaranteed. Deshpande *et al*. proposed a model-based querying approach and built a prototype BBQ to activate sensors in order to minimize prediction error [29]. These works all exploited the spatio-temporal relation of sensed data to approximate the information sensed by a part of nodes with that by all nodes [30] Shi *et al* constructed effective sensor data scheduling schemes that minimized the estimation error and satisfied the energy constraint [31]. They studied two scenarios: the sensor had sufficient computation capability and the sensor had limited computation capability.

Although there were much work focusing on data aggregation, few of them simultaneously consider the packet loss and sampling link unreliability, which are ineluctable in WSNs. Under LDA, there is very few work researching aggregation with packet loss especially under asynchronization. Meanwhile, the energy consumption on data sampling and transmitting can be reduced more when proper strategies, such as loss data aggregation, are designed, which is not considered as far as we know.

3. PRELIMINARY

3.1. Network model

A network consisting of n sensor nodes is modeled by a graph $G(V, E)$. These nodes are randomly and uniformly deployed in a $\mathcal{C} \times \mathcal{C}$ area and respectively denoted as N_i, $i = 1, \cdots, n$, composing a node set $V = \{N_i | i = 1, \cdots, n\}$. Every node N_i has a transmission range r_i such that two nodes N_i and N_j can communicate with each other directly if $||N_i - N_j|| \leqslant r$, where $r = \min\{r_i, r_j\}$, and there is no other simultaneous transmission in both transmission area of N_i and N_j. The transmission range r of each node is properly set to guarantee the whole network connected [32]. We adopt an existing algorithm to construct a CDS $\mathcal{S}$, $\mathcal{S} \subset V$. Wan *et al*. presented a distributed algorithm that had an approximation factor of at most eight, $O(n)$ time complexity and $O(n \log n)$ message complexity [33]. Then, each node N_i $(N_i \notin \mathcal{S})$ can find a node N_j $(N_j \in \mathcal{S})$ and connect with it. All nodes in $N_j \in \mathcal{S}$ can connect together and send their data to a root by multi-hop. We call those nodes in $\mathcal{S}$ as *parents nodes* (or parents) and the nodes not in $\mathcal{S}$ as *child nodes* (children). All nodes transmit their data to the root directly or through relay nodes.

3.2. Data aggregation scheme

In WSNs, one of the primary tasks is that each node collects and transmits data to its destinations. Meanwhile, each node works under duty cycle mode in order to prolong the network lifetime as

DOI: 10.1002/dac

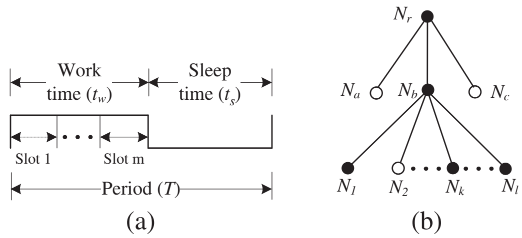


Figure 1. (a) Duty cycle. (b) A data aggregation tree DT_r with a subtree DT_3, in which white node has no child whereas black does.

shown in Figure 1(a). A node keeps itself 'active' in work time t_w and 'sleep' in sleep time t_s. This paper divides a period of work time t_w into K equal time slots. All nodes run the same duty cycle.

We denote the life time of a network as F and assume that $F = L \times T$, where L is a natural number. At pth ($p = 1, \cdots, K$) slot S_p of qth ($q = 1, \cdots, L$) period T_q, a node N_i samples a data from environment. The data are denoted as $D(p, q, i)$ so we can denote a series of data by $D(S_p, q, i)$ ($p = 1, \cdots, K$), which is sampled by a node N_i at qth period T_q. We suppose that the data series $D(p, q, i)$ ($p = 1, \cdots, K$) is relative to time t and denote the series by $\mathscr{N}(t)$. For convenience, we denote the sampled or received data as $D(p, q, i)$ and the aggregated data on node N_i as $D_f(p, q, i)$ according to some kind of aggregation function $\mathbb{f}$, that is, $D_f(S_p, T_q, i) = \mathbb{f}(D(S_p, T_q, \mathscr{S}))$, where $\mathscr{S} \subset V$ is a subset containing nodes in the aggregation tree rooted at N_i. $\mathbb{f}$ can mean the calculation max, min, average, and so on. In brief, we use D_f instead of $D_f(S_p, T_q, k)$.

This paper adopts the following *aggregation function*: $\mathbb{f}$. Each node N_i obtains k_i ($k_i \leqslant K$) data after sampling data in each period T and aggregates these k_i data into one. If $N_i \notin \mathcal{S}$, N_i sends its aggregated data to its parent. If $N_i \in \mathcal{S}$, N_i waits till it receives all data of its child nodes. Then, N_i aggregates its data and that of its child nodes into one, which is transmitted to the root by multi-hop fashion. When a child N_i transmits its own aggregated data to the root through other intermediate nodes, other nodes aggregate their data with N_i's.

Suppose that there exists a routing protocol, which constructs a *fixed routing* to the root for each node in a network. We can use the *aggregation tree* DT_r to describe the process of data aggregation and transmission as shown in Figure 1(b). The aggregation tree DT_r denotes a tree rooted at a node N_r, and the size of DT_r is denoted as $|DT_r|$, which can be worked out by the root N_r in our function $\mathbb{f}$. In a period T_q, N_r can know the number of child nodes, which gather data from the physical world.

4. SYNCHRONOUS SAMPLING

In this section, all nodes sample data synchronously. We adopt two ways to reduce the number of time slots, in which nodes sample data, and the number of nodes to transmit data. One way is called *controllable data aggregation* whereas the other is *uncontrollable data aggregation*.

In the following context, we suppose that the data $D(p, q, i)$, directly gathered from the physical world, obeys the normal distribution $N(\mu, \sigma^2)$, where u is the expectation and σ is the variance. $D(p, q, i)$ and $D(p, q, j)$ are independent from each other when $i \neq j$.

4.1. Controllable data aggregation

When links and nodes are reliable and the interference within a network can be avoided by a precisely designed schedule, we can assume that there is no link unreliability. Under this case, a root can receive the aggregated data $D(p, q, i)$ ($p = 1, \cdots, K$ and $i = 1, \cdots, n$) from all nodes at each period T_q without losing any sampled data. Thus, we can theoretically analyze the least number m of nodes needing to sample data and transmit their data to the root when we guarantee $P(|D_f - D'_f| < \wp) = 1 - \gamma$, where $\wp$ is a small positive value. So, we can design an algorithm to select only m nodes to sample data.

This paper proposes two ways to implement LDA under controllable data aggregation. The first is that a root only collects the data of m nodes ($m < n$), and these m nodes sample data in all time

DOI: 10.1002/dac

slots. The second is that the root also collects the data of m out of n nodes, but each of m nodes only samples data in part of time slots.

The first way is described as follows. When all nodes sample and aggregate data at all time slots, the root can finally obtain an aggregated data D_f^n. WSNs are large-scale networks, so we consider that n is big enough to ensure V a big sample space. Therefore, we consider D_f^n very close to the real value such that it can represent the real value. When the root randomly and uniformly selects m out of n nodes, we denote the value of the aggregated data at the root as D_f^m. Thus, we can easily obtain the following theorem according to the interval estimation [34].

Theorem 1

When there are m ($m < n$) nodes, randomly chosen, sampling and aggregating data and each of m nodes gathers data at all time slots, we have $P\{|D_f^m - D_f^n| < \wp\} = 1 - \gamma$, where $\wp = \frac{S \cdot t_{\frac{\gamma}{2}}(m-1)}{\sqrt{m}}$ [§], S is a standard deviation, $0 \leqslant \wp$ and $0 \leqslant \gamma \leqslant 1$.

Proof

Because there are totally n nodes in a network, the average value D_f^n of their sampled data $D(p,q,i)$ is $D_f^n = \frac{1}{n}\sum_{i=1}^{n} \overline{D}(S_p, T_q, i)$ at the period T_q, where $\overline{D}(S_p, T_q, i) = \frac{1}{K}\sum_{p=1}^{K} D(p, T_q, i)$. According to *the law of large numbers* [35], there is a following equation:

$$P(|D_f^n - \mu| < \epsilon) \geqslant 1 - \frac{\sigma^2}{n \cdot \epsilon^2}$$

where $\epsilon > 0$ is a small positive number. WSN is a kind of large-scale network so we can ensure that n is large enough to guarantee $D_f^n = \mu$ with high probability.

When there are only m nodes sampling, aggregating, and sending data to the root, the average value D_f^m of their sampled data $D(p,q,i)$ is $D_f^m = \frac{1}{m}\sum_{i=1}^{m} \overline{D}(S_p, T_q, i)$ in each period T_q. Notice that these m nodes are randomly chosen. The error between D_f^m and the expectation μ of the data in the real world then can be obtained by the following equation:

$$P\{|\frac{D_f^m - \mu}{\sigma}\sqrt{m}| < c\} = 1 - \gamma \tag{1}$$

where $c > 0$ and $1 - \gamma$ is a confidence level. The variance σ in physical world is usually unknown. Furthermore, the root can estimate the variance on the basis of the current data according to a following equation.

$$S = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(\overline{D}(S_p, T_q, i) - D_f^m)^2} \tag{2}$$

On the basis of Equation (2), we can obtain a following equation according to Equation (1).

$$P\{|\frac{D_f^m - \mu}{S}\sqrt{m}| < c\} = 1 - \gamma$$

As $\frac{D_f^m - \mu}{S}\sqrt{m}$ obeys $t(m-1)$ *distribution*, we have $P\{|t| < c\} = 1 - \gamma$, that is, $P\{|t| \geqslant c\} = 1 - P\{|t| < c\} = \gamma$. Therefore, we have the following equation:

$$P\{|D_f^m - \mu| < \wp\} = 1 - \gamma \tag{3}$$

where $\wp = \frac{S \cdot c}{\sqrt{m}}$ and $c = t_{\frac{\gamma}{2}}(m-1)$. This finishes the proof. □

[§] t-distribution has the probability density function: $f_\gamma(t) = \frac{\Gamma(\frac{\gamma+1}{2})}{\sqrt{\gamma\pi}\Gamma(\frac{\gamma}{2})}(1 + \frac{t^2}{\gamma})^{-(\gamma+1)/2}$, where γ is the number of degrees of freedom and Γ is the Gamma function.

DOI: 10.1002/dac

The above theorem means that the root need receive data only from m nodes if the difference $\wp$ with confidence level $1-\gamma$ is acceptable. The parameters $\wp$ and γ are previously set at the root. Thus, $n-m$ nodes can stop gathering data and save energy. Notice that these $n-m$ nodes still afford of communication task. We denote the parameters $\wp_k$ and γ_k for an arbitrary parent node N_k. When N_k selects its m_k ($m_k \leqslant m$) child nodes to sample and transmit data, $m_k = \frac{m|DT_k|}{n}$ with probability, where $|DT_k|$ denotes the cardinality of the set DT_k. We can generalize the result in Theorem 1 to the following equation.

$$P\{|D_f^{m_k} - \mu| < \wp_k\} = 1 - \gamma_k \tag{4}$$

where $\wp_k = \frac{S_{m_k} \cdot c}{\sqrt{m_k}}$, $c = t_{\frac{\gamma_k}{2}}(m_k - 1)$ and $S_{m_k} = \sqrt{\frac{1}{m_k-1}\sum_{i=1}^{m_k}(\overline{D}(S_p, T_q, i) - D_f^{m_k})^2}$.

On the basis of the first way, the second way to save energy and sampling time is that each node samples data in η out of K time slots whereas the root still randomly and uniformly chooses m nodes to sample data. Here, we use $D_f^{k\eta}$ to denote the value of the aggregated data at the node N_k when a node samples only at η time slots and there are totally m nodes to sample data. D_f^k denotes the value of aggregation data at a node N_k when N_k samples at all time slots, the error between $D_f^{k\eta}$ and D_f^k could be bounded to be at most $\widetilde{\wp}_k$ rooted at a node N_k with a given confidence level level $1-\gamma_k$, where $\widetilde{\wp}_k \geqslant 0$ and $0 \leqslant \gamma_k \leqslant 1$.

Lemma 1

When a node N_k randomly chooses m_k ($m_k \leqslant m$) nodes in its DT_k, to sample and aggregate data and each of m_k nodes samples data at η ($\eta \leqslant K$) time slots, we have $P\{|D_f^{k\eta} - D_f^k| < \tilde{\wp}_k\} = 1-\gamma_k$, where $\tilde{\wp}_k = \frac{S_k \cdot t_{\frac{\gamma_k}{2}}(m_k-1)}{\sqrt{m_k}}$, S_k is a standard deviation.

Proof

When each node N_i gathers data at η out of K time slots at period T_q, we can obtain that the average of these η sampled data is $D_f^{i\eta} = \frac{1}{\eta}\sum_{p=1}^{\eta} D(p, T_q, i)$ at period T_q. The following equation can be obtained according to Theorem 1:

$$P_i\{|D_f^{i\eta} - \overline{D}_f| < \wp_i\} = 1 - \gamma_i$$

where $\overline{D}_f = \frac{1}{K}\sum_{p=1}^{K} D(p, T_q, i)$, $\wp_i = \frac{S_i \cdot c_i}{\sqrt{\eta}}$ and $c_i = t_{\frac{\gamma_i}{2}}(\eta - 1)$ ($0 \leqslant \wp_i$ and $0 \leqslant \gamma_i \leqslant 1$). Here, S_i is described in Equation (5).

$$S_i = \sqrt{\frac{1}{\eta - 1}\sum_{p=1}^{\eta}(D(p, T_q, i) - D_f^{i\eta})^2} \tag{5}$$

When there are totally m nodes to sample data in the whole network, the expected number m_k of nodes contained in an aggregation tree DT_k is $\frac{m|DT_k|}{n}$. The average value $D_f^{m_k}$ of their sampled data $D(p, q, i)$ is $\widetilde{D}_f^{m_k} = \frac{1}{m_k}\sum_{i=1}^{m_k} \eta D_f^{i\eta}$ at T_q. Thus, the standard deviation is

$$S_k = \sqrt{\frac{1}{m_k - 1}\sum_{i=1}^{m_k}(D_f^{i\eta} - \widetilde{D}_f^{m_k})^2} \tag{6}$$

The error between $\widetilde{D}_f^{m_k}$ and μ is:

$$P\{|\widetilde{D}_f^{m_k} - \mu| < \wp_k\} = 1 - \gamma_k \tag{7}$$

where $\wp_k = \frac{S_k \cdot c_k}{\sqrt{m_k}}$ and $c_k = t_{\frac{\gamma_k}{2}}(m_k - 1)$. □

DOI: 10.1002/dac

Notice that the number p of time slots in each period T_q is usually small, so the confidence level $1-\gamma_k$ cannot be very high at certain error $\wp_k$. When we consider the value $D_f^{n\eta}$ of the aggregated data at the root and m nodes sample only at η time slots, the error between $D_f^{n\eta}$ and D_f^n could be bounded to be at most $\wp_n$ under the confidence level $1-\gamma_n$, where $\wp_n \geqslant 0$ and $0 \leqslant 1-\gamma_n \leqslant 1$.

Lemma 2
In Lemma 1, if the data are aggregated to the root, then $P\{|D_f^{n\eta} - D_f^n| < \wp_n\} = 1-\gamma_n$.

In Lemma 1, the aggregation tree DT_k extends to be a tree rooted at the root DT_s when the aggregated data are transmitted to the root finally. The average value D_f^m of their sampled data $D(p,q,i)$ is $\widetilde{D}_f^m = \frac{1}{m}\sum_{i=1}^m \eta D_f^{i\eta}$ at T_q, so the standard deviation of these data is $\tilde{S_n} = \sqrt{\frac{1}{m-1}\sum_{i=1}^m (D_f^{n\eta} - \widetilde{D}_f^m)^2}$. According to Equation (5), the error between $\widetilde{D}_f^m$ and μ is $P\{|\widetilde{D}_f^m - \mu| < \widetilde{\wp}_n\} = 1-\gamma$, where $\widetilde{\wp}_n = \frac{\widetilde{S_n}\cdot c_n}{\sqrt{m}}$ and $c_n = t_{\frac{\gamma_n}{2}}(m-1)$. At the moment, $\widetilde{D}_f^n = \widetilde{D}_f^m$. Lemma 2 means that the error between the aggregation data obtained from all time slots and that obtained from η time slots is also bounded by the error $\wp_n$ under certain confidence level $1-\gamma_n$ when m nodes sample.

4.2. *Noncontrollable data aggregation*

In practical environments, nodes and links are unreliable and interference is unavoidable. Therefore, some packets are unavoidably lost during transmission because of the unreliability of nodes and links or some fault occurring.

Unlike the case in the Section 4.1, it is controllable that some nodes need sample data without considering the link or node reliability. When the data lost does exist and all nodes sample data at all time slots, we denote the aggregated data at the root as D_f^x. Surely $D_f^x \leqslant D_f^n$.

Theorem 2
Suppose the link unreliability probability P_l happens randomly and uniformly in any time slots on each link. If each node samples data at all time slots, then a parent can receive data with packet-loss rate no bigger than $\wp_k$ under a confidence level $1-\gamma_k$ when P_l is not bigger than $\frac{|DT_k|-m_k}{u_k - v_k + \sum_{j=1}^{v_k}|N_j|}$.

Proof
Without loss of generality, suppose that a parent N_k of a tree DT_k has u_k child nodes, among which v_k child nodes have their own child nodes and others have no child nodes of their own. Thus, these child nodes may lose $(u_k - v_k)P_l$ data. If we denote the v_k child nodes as N_j, $j = 1,\cdots,v_k$, then those nodes, which have their own child nodes, may lose $P_l\sum_{j=1}^{v_k}|N_j|$ data. Thus, there are only $|DT_k| - (u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j|$ nodes, which can still send their data to N_k. According to Equation (4), $|DT_k| - (u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j|$ should not be less than m_k under the same error $\wp_k$ and confidence level $1-\gamma_k$, where m_k, $\wp_k$, and $1-\gamma_k$ are determined according to Equation (4). So, $|DT_k| - (u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j| \geqslant m_k$, that is,

$$P_l \leqslant \frac{|DT_k| - m_k}{u_k - v_k + \sum_{j=1}^{v_k}|N_j|} \tag{8}$$

□

According to Theorem 2, a parent can make decision whether it need require its child nodes to retransmit their data.

When the probability of link unreliability between any pair of nodes is P_l and we denote the aggregated data at the root as $\widetilde{D}_f^n$, the following lemma can be obtained under certain error $\wp_l$ and confidence level $1-\gamma_l$.

DOI: 10.1002/dac

Lemma 3
Suppose that the packet-loss probability P_l happens randomly and uniformly in any time slots on each link. We have $P\{|\tilde{D}_f^n - D_f^n| < \wp\} = 1-\gamma$ when $P_l \leqslant 1-\frac{m}{n}$ and all nodes transmit their data to the root.

Proof
When the aggregated data are sent to the root, $|DT_s| = n$ and $m_k = m$, the root has no child node so $u_s = v_s$. According to Theorem 2, $P_l \leqslant \frac{n-m}{\sum_{j=1}^{v_s}|N_j|} = \frac{n-m}{n} = 1-\frac{m}{n}$, where v_s is the child nodes of the root. Therefore, Theorem 1 can still be satisfied when $P_l \leqslant 1-\frac{m}{n}$. □

When permanent fault occurs on links or nodes, the topology structure should be reconstructed. The issue has been researched by many previous work. When temporary nodes or link fault occur, on the basis of the received data, a parent can make a decision whether the fault branch or child need retransmit their data.

When the link unreliability probability of a child node is $P_l > 0$, the child will be required to retransmit its data. We suppose that data retransmission can be successful with probability P_r and define the aggregated data at the node N_k as $\overline{\overline{D}}_f^k$ after retransmission.

Lemma 4
Suppose that the link unreliability probability of a child node is $P_l > 0$ and the successful retransmission probability is P_r. When $P_l \leqslant \frac{|DT_k|-m_k}{[u_k-v_k+\sum_{j=1}^{v_k}|N_j|]P_r}$, we have $P\{|\overline{\overline{D}}_f^k - D_f^n| < \wp_r\} = 1-\gamma$, where $\wp_r$ is a given error bound.

Proof
If a parent N_k finds that the received aggregated data cannot satisfy Equation (8) when the data loss probability is P_l, N_k lets the nodes fall to transmit and retransmit their data. $[(u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j|] \times P_r$ nodes may retransmit their data successfully. Thus, there are $|DT_k| - [(u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j|] \times P_r$ nodes, which finally can transmit their data successfully. According to Equation (4), $|DT_k| - (u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j|$ should not be less than m_k under the same $\wp_k$ and $1-\gamma_k$, where m_k, $\wp_k$, and $1-\gamma_k$ are determined according to Equation (4). So, $|DT_k| - [(u_k - v_k)P_l - P_l\sum_{j=1}^{v_k}|N_j|]P_r \geqslant m_k$, that is,

$$P_l \leqslant \frac{|DT_k| - m_k}{[u_k - v_k + \sum_{j=1}^{v_k}|N_j|]P_r} \tag{9}$$

□

Algorithm 1 Non-controllable Data Aggregation

Input: n, confidence level γ, received or sampled data;
Output: Each parent decides whether its child nodes should retransmit their data or not.

```
1: Each node samples data at period T_p;
2: An arbitrary node N_u receives the data from its child nodes;
3: N_u calculates the number of received data and calculates out P_l;
4: N_u checks whether the received and sampled data can satisfy the condition in Equation (9);
5: if The condition is satisfied then
6:     N_u aggregates the data and transmits it to its parent node.
7: else
8:     N_u requires some of its child nodes, which fault to transmit their data in the previous period,
       to retransmit their data;
9:     N_u calculates P_r.
10: end if
```


DOI: 10.1002/dac

4.3. Error and confidence level allocation

One interesting task is to allocate error and confidence level within a whole network. When an error $\wp$ is acceptable under a confidence level $1-\gamma$ at the root, it is necessary to allocate the error and confidence level at some parents of different subtree. For example, if an error $\wp_b$ and a confidence level $1-\lambda_b$ at a parent N_b are known in Figure 1(b), how does N_b set the error $\wp_i$ and the confidence level $1-\lambda_i$ for N_i $(N_i \in DT_b)$ among DT_b? Here, we discuss the question under two cases: link reliability and link unreliability.

4.3.1. Allocation model. Before allocating the error under a certain confidence level within a network, we firstly introduce an allocation model.

In Figure 1(b), an aggregation tree rooted at node N_r contains three levels. Each level contains several child nodes. For convenience, an arbitrary sublayer of a tree DT_r is constructed for illustration as shown in Figure 1(b). A subtree DT_3 rooted at node N_b has l child nodes: $N_1, N_2, \cdots, N_l$, where these nodes except N_2 are also the parents of their sub-subtree. Because $N_1, \cdots, N_l$ directly transmit their data to N_b, they would not prevent other nodes from their successful transmission to their parent. Notice that we are considering a case that there is no link unreliability. Therefore, we can consider the subtree DT_b as a parallel system [36]. When we define the confidence level of each node N_i as $1-\gamma_i$ and the error $\wp_b$ under the confidence level $1-\gamma_b$ of the parent N_b is given, the probability P_b that $|D_f^{N_b} - \overline{D}_f| < \wp_b$ is $1-\gamma_b$. Therefore, the probability for the parent can be described by the probability of its child nodes.

$$P_b = 1 - \prod_{i=1}^{k}(1-P_i) = 1 - \prod_{i=1}^{k}\gamma_i \tag{10}$$

where P_i is the probability about N_i that $P_i(|D_f^{N_i} - \overline{D}_f| < \wp_i)$ is $1-\gamma_i$. Notice that some nodes, such as N_l, may not be selected to sample data and N_r only selects k $(k \leqslant l)$ data to sample data in Figure 1(b).

4.3.2. Under link reliability. When no link unreliability happens, at least m nodes should be selected to sample and transmit data in the whole network according to Theorem 1. However, how to select the m nodes depends on the error bound and confidence level given for the root.

It is easy to assign the root with an error bound and a confidence level but uneasy that the root allocates its error bound and confidence level to its child nodes and its child nodes allocate their error bound and confidence level to their child nodes till all parents are allocated their error bound and confidence level. In the following context of this paper, Algorithm 2 gives a method to allocate the confidence level between the parents and their child nodes.

We design a distributed algorithm (Algorithm 2) to allocate the confidence level $1-\gamma$ pro rata to different nodes when the error $\wp$ is given. In Algorithm 2, the given graph $G(V,\phi)$ is composed of the vertex set V and no edge set. $|V| = n$ and the node in V are randomly and uniformly deployed to satisfy the connection condition as described in [32].

Algorithm 2 confidence level Allocation

Input: A given graph $G(V,\phi)$ and error bound $\wp$ and confidence level $1-\gamma$;
Output: A connected tree with each parent N_i allocated a confidence level $1-\gamma_i$.

1: Use the algorithm in [5] to construct a CDS $\mathcal{S}$;
2: If a node N_j is not in $\mathcal{S}$, it connects to a node in $\mathcal{S}$ with the shortest Euclidian distance;
3: Set the error bound and confidence level for the root respectively with $\wp$ and $1-\gamma$;
4: Set the confidence level for each child node N_k of the parent respectively with the confidence level $1-\gamma_k$ according to Equation (16);
5: Each parent allocates its confidence level to its child nodes according to Equation (16);
6: When all parents finish the confidence level allocation, the algorithm is over.

DOI: 10.1002/dac

Theorem 3
When each link is reliable and the error bound $\wp_r$ and confidence level $1-\gamma_r$ are previously given to the root, the child nodes of the root can be allocated the confidence level γ_i, where $\gamma_i = \sqrt[k]{\gamma_r}$ and $k = |DT_s|$.

Proof
We begin our proof on the basis of the model described in Equation (10). When $k = 1$, $P_b = 1-\gamma_i = P_i$.

When $k \geqslant 2$, k, γ_i and $\wp_i$ are determined in the following text.

There are many important and practical methods to distribute the probability indexes. Here, we try to find the minimal error of $\sum_{i=1}^{k} \wp_i$ under certain confidence level P_b whereas we find a k as small as possible. A smaller k means to save more energy whereas a bigger k is needed to achieve a lower error $\wp_i$. Here, we use *Lagrange undetermined coefficients method* to find k as small as possible while guaranteeing that the error is kept at a proper value. Thus, we can construct a *Lagrange function H*:

$$H = \sum_{i=1}^{k} \wp_i + \lambda (P_b + \prod_{i=1}^{k} \gamma_i - 1) \tag{11}$$

In order to find the minimal value of $\wp_i$, we can calculate the derivative of the *Lagrange function* H and let it be zero, that is, $\frac{\partial H}{\partial \wp_i} = 0$. Because $\wp_i = \frac{S_i \cdot c_i}{\sqrt{k}}$ and $c_i = t_{\frac{\gamma_i}{2}}(k-1)$, we can obtain the following equation:

$$\wp_i = \frac{S_i \cdot t_{\frac{\gamma_i}{2}}(k-1)}{\sqrt{k}} \tag{12}$$

The probability density function of the random variable t is as follows:

$$f(t,\nu) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\nu\pi}\,\Gamma(\nu/2)} (1 + t^2/\nu)^{(\nu+1)/2}$$

where $\nu = k-1$, $t = \gamma_i/2$ and $\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} \mathrm{d}t$.

On the basis of Equation (11), the partial derivative of H with respect to the variable $\wp_i$ is given as follows:

$$\frac{\partial H}{\partial \wp_i} = 1 + \lambda \prod_{j=1, j\neq i}^{k} \gamma_j \frac{\partial \gamma_i}{\partial \wp_i} \tag{13}$$

According to Equation (12), we can obtain the partial derivative of γ_i with respect to the variable $\wp_i$.

$$\begin{aligned}
\frac{\partial \wp_i}{\partial \wp_i} &= \frac{S_i}{\sqrt{k}} \frac{\partial t_{\frac{\gamma_i}{2}}(k-1)}{\partial \wp_i} \\
\Rightarrow 1 &= \frac{S_i}{\sqrt{k}} \frac{\partial f(\gamma_i/2, k-1)}{\partial \wp_i} \\
\Rightarrow \frac{C}{S_i} &= (1 + \frac{\gamma_i^2}{4(k-1)})^{k/2-1} \frac{\gamma_i}{2(k-1)} \frac{\partial \gamma_i}{\partial \wp_i}
\end{aligned} \tag{14}$$

DOI: 10.1002/dac

where $C = \frac{2\sqrt{(k-1)\pi}\Gamma((k-1)/2)}{\sqrt{k}\Gamma((k)/2)}$. Let $\frac{\partial H}{\partial \wp_i} = 0$ in Equation (13) and on the basis of Equation (14), we can obtain that

$$\begin{aligned} 0 &= 1 + \lambda \frac{1-P_b}{\gamma_i}\frac{\partial \gamma_i}{\partial \wp_i} \\ \Rightarrow \lambda &= -\frac{\gamma_i}{(1-P_b)\frac{\partial \gamma_i}{\partial \wp_i}} \\ &= -(1+\frac{\gamma_i^2}{4(k-1)})^{k/2-1}\frac{\gamma_i^2 S_i}{2C(k-1)(1-P_b)} \end{aligned}$$

Notice that the above equation is tenable for each γ_i $(i = 1, \cdots, k)$, so we can obtain the following equation when $i \neq j$.

$$(4(k-1)+\gamma_i^2)^{k/2-1}\gamma_i^2 S_i = (4(k-1)+\gamma_j^2)^{k/2-1}\gamma_j^2 S_j \tag{15}$$

According to Equations (10) and (15), γ_i $(i = 1, \cdots, k)$ can be solved. From Equation (15), we can obtain that

$$\begin{aligned} &(k/2-1)\ln(4(k-1)+\gamma_i^2) + 2\ln\gamma_i + \ln S_i \\ &= (k/2-1)\ln(4(k-1)+\gamma_j^2) + 2\ln\gamma_j + \ln S_j \qquad i \neq j \end{aligned}$$

Because the value of k is the same for all nodes in the same aggregation tree DT_r, the $\gamma_i = \gamma_j$ when $i \neq j$. According to Equation (10), γ_i can be obtained in the following equation.

$$\gamma_i = \sqrt[k]{1-P_b} = \sqrt[k]{\gamma_r} \tag{16}$$

□

Now, we can design an algorithm to positively control whether a node samples data or not as shown in Algorithm 3. On the basis of the error bound $\wp_r$ at the root and the confidence level allocation obtained on Algorithm 2, Algorithm 3 makes each parent know the number of its child nodes to sample.

Algorithm 3 Control Data Aggregation

Input: The error bound $\wp_r$ at the root and the confidence level allocation obtained in Algorithm 2;
Output: The number of child nodes required by a parent to sample data.

1: The root calculates the number of nodes required to sample according to the given error bound $\wp_r$ and the confidence level $1-\gamma$ obtained in Theorem 1;
2: Based on the confidence level allocated in Algorithm 2 and its error bound, each parent N_i calculates the needed number of sampling nodes according to Equation (4);
3: After all parents finish calculating the number of data required to sample, the algorithm is over.

4.3.3. Under link unreliability. When the link is unreliable, we suppose a data would be lost with the probability P_l.

Theorem 4
When the link is unreliable with the probability P_l, and the error bound $\wp_r$ and confidence level $1-\gamma_r$ are previously given to the root, the child nodes of the root can be allocated a confidence level γ_i', where $\gamma_i' = 1 - \frac{1-\sqrt[k]{\gamma_r}}{P_l}$ and $k = |DT_s|$.

DOI: 10.1002/dac

Proof
When the data loss probability is P_l, according to Equation (10), we have the following equation.

$$P_b = 1 - \prod_{i=1}^{k}(1 - P_l P_i) \tag{17}$$

If we define $P_i' = P_l P_i$ and $\gamma_i' = 1 - P_i'$, then we can obtain the following equation from Equation (17).

$$P_b = 1 - \prod_{i=1}^{k} \gamma_i' \tag{18}$$

We also use the Lagrange undetermined coefficient method to allocate the confidence level for each parent as Equation (11) and can obtain the results similar to that in Equation (16). The relation between γ_i and γ_i' is

$$\gamma_i' = 1 - P_i' = 1 - P_l P_i = 1 - P_l(1 - \gamma_i) \tag{19}$$

Therefore, the confidence level allocation function is

$$\gamma_i' = 1 - \frac{1 - \gamma_i}{P_l} = 1 - \frac{1 - \sqrt[k]{\gamma_r}}{P_l}$$

□

4.4. *Energy saving*

Here, we consider the energy saving under both cases of controllable and uncontrollable data aggregation besides the inherent energy saving of data aggregation.

When we consider that there is no link unreliability as Section 4.1, the energy saving is mainly due to the data transmission reduction, that is, some nodes need not sample and transmit the data under some time slots. Although some packets are inevitably retransmitted because of collision in wireless channel, it has less retransmission to adopt LDA schedule as there are less data to transmit.

We define the energy cost to sample data in a slot as E_s and the energy cost to transmit a packet in one hop as E_p.

Lemma 5
When each link is reliable and only m nodes gather data at η time slots out of K, the saved energy E_a is $(n-m)((1-\eta)E_s + E_p)$.

Proof
There are $n - m$ nodes, which can save energy as they need not sample and transmit data. The total saved energy E_t mainly contains two parts: sampling energy and transmitting energy. Notice that each node N_i samples data and transmits its data to its parent N_j. N_j aggregates the N_i's data and that of itself into one packet. The energy to transmit a packet in one hop is saved. $E_a = (n-m) \times (1-\eta) \times E_s + (n-m) \times E_p = (n-m)((1-\eta)E_s + E_p)$. □

Because some data are unavoidably lost, the error under the same confidence level is increased. When we positively argue to retransmit the lost data, suppose that each transmission is unreliable with probability P_l and only m nodes gather data. The consumed energy to retransmit is $m \times P_l \times E_p$ because the number of lost packets is expectably $m \times P_l$. Under the case, the saved energy is $\max\{0,\ (n-m)((1-\eta)E_s + E_p) - m \times P_l \times E_p\}$. If the retransmission scheme is given, the energy cost increases as $P_r * m * E_p$. At this moment, the saved energy is $\max\{0,\ (n-m)((1-\eta)E_s + E_p) - m \times E_p \times (P_l - P_r)\}$.

DOI: 10.1002/dac

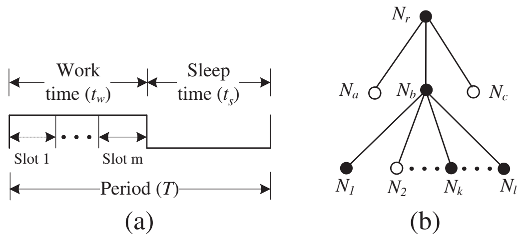


Figure 2. (a) The beginning time is different when all nodes sample asynchronously. (b) Some node pairs have similar sampling data piece in two 'asynchronous' duration.

5. ASYNCHRONOUS SAMPLING

Because the clock in a sensor node is easy to skew in WSNs or other reasons, each node samples data asynchronously. Suppose that the beginning of all nodes' sampling time is a Poisson process $P_k(t)$, as shown in Figure 2(a). For example, the sample time t_{10}, t_{23}, t_{35}, and t_{n8} at the nodes N_1, N_2, N_3, and N_n correspond to the same time, but the nodes detect the sample times as different time.

If the data sampled by each node is relative on the geometric position and suppose that these data obey a certain kind of probabilistic distribution, the distribution can be estimated and the *homogeneous region* (HR) in which a group of sensors have similar underlying distribution also can be detected [37]. The data sampled by any two different nodes, such as N_i and N_j in Figure 2(b), contain time offset, so they cannot be aggregated directly. When the two data obey the same distribution, a part of both of data can be aggregated if the data are properly intercepted.

In this paper, we focus on the time-relative data distribution and find the beginning time, at which two different nodes at the same HR have the same sampled data function. In Figure 2(b), the distribution function of the sampled data is defined as $F_i(t_i)$ for node N_i. We use two methods to calculate the time t_i $(i = 1, \cdots, n)$, which correspond to the same time.

The first method is that each node estimates the distribution function of the sampled data.

The second method is that each node calculates the slope of the sampled data in each sample slot.

5.1. Without link unreliability

In this case, we do not consider the link unreliability. When a node aggregates the data from other nodes, the data should be sampled at the same time slots in a period. Otherwise, the aggregated data are invaluable.

Lemma 6
When the beginning time of each node is a Poisson process, the case in Theorem 1 cannot be guaranteed.

We set a period as Δt. In the period, the probability that there are more than one node beginning to sample is zero as Δt goes to zero, that is, $\Delta t \to 0$. Therefore, there is no probability that there are sufficient number of nodes, which can sample data simultaneously in the same work time t_w. Therefore, the case in Theorem 1 cannot be guaranteed. But, the case that m nodes synchronously sample data at η common time slots does exist.

Theorem 5
When the sampling begin time of each node is a Poisson process with a rate parameter λ, the confidence level is at most $\frac{(1-\gamma)m^m}{m!}e^{-m}$ in order to guarantee the error $\widetilde{\wp}$.

Proof
During a period Δt, the probability that there are m nodes beginning to sample is that

$$P_m(\Delta t) = \frac{(\lambda \Delta t)^m}{m!} e^{-\lambda \Delta t} \tag{20}$$

DOI: 10.1002/dac

The expectation is $E(\Delta t) = \lambda \Delta t$ and the variance is $D_m(\Delta t) = \lambda \Delta t$, because there are K time slots in a work time t_w and a slot lasts $\frac{t_w}{K}$. When each of m node should have η common slots to sample data, the period Δt in which m nodes begin to sample is at most $2t_w - \frac{\eta t_w}{K}$, that is, $\Delta t \leqslant 2t_w - \frac{\eta t_w}{K}$, which insures that two nodes must have η common time slots.

By $\frac{\partial P_m(\Delta t)}{\partial \Delta t} = 0$, it is easy to find that $\Delta t = \frac{m}{\lambda}$. When $0 \leqslant \Delta t \leqslant \frac{m}{\lambda}$, $P_m(\Delta t)$ is an increasing function about Δt. Otherwise, it is a decreasing one. Thus, $P_m(\Delta t) \leqslant P_m(\frac{m}{\lambda})$.

Define $event_1$ in that there are m nodes that are beginning to sample in period Δt, where $\Delta t \leqslant 2t_w - \frac{\eta t_w}{K}$.

Define $event_2$ in that the error is less than $\wp_i$.

Define $event_3$ in that the error is less than $\wp_i$ when $event_1$ happens.

Then, we have

$$P\{event_3\} = P\{event_1 \cdot event_2\} = P\{event_1\} \cdot P\{event_2\} \tag{21}$$

According to Theorem 1,

$$\begin{aligned} P\{event_3\} &= (1-\gamma) P_m(\Delta t) \\ &\leqslant (1-\gamma) P_m(\frac{m}{\lambda}) = \frac{(1-\gamma) m^m}{m!} e^{-m} \end{aligned} \tag{22}$$

□

5.2. *With link unreliability*

It is a challenge problem to consider the link unreliability whereas all nodes sample data asynchronously. We also suppose that each link has unreliable probability P_l. Under this case, there are less number of nodes able to transmit data successfully.

Lemma 7
When the link unreliability probability of each node is P_l, there are still $|DT_k| - (u_k - v_k)P_l - P_l \sum_{j=1}^{v_k} |N_j|$ nodes, which can still sample and transmit data normally.

Proof
According to Equation (20), the probability that m out of n nodes begins to sample in a period Δt is $P_m(\Delta t)$. For a root node N_i, it has $|DT_i|$ neighbors. In the period Δt, there are $\frac{m|DT_i|}{n}$ nodes beginning to work with probability $P_m(\Delta t)$. Thus, there are expectably $\frac{m|DT_i|}{n} P_m(\Delta t)$ nodes in the period Δt.

According to Theorem 2, there are only $|DT_k| - (u_k - v_k)P_l - P_l \sum_{j=1}^{v_k} |N_j|$ nodes, which can still send their data to N_k. □

Because all nodes sample data asynchronously, their beginning times are different as shown in Figure 2(a). But the data sampled by a pair of nodes are the same or similar in the same duration, as shown in Figure 2(b),in spite that the two durations seem to be 'asynchronous'. Suppose that two nodes N_i and N_j sample data in each time slot, and they respectively obtain a series of sampled data in a certain period of their own. If the data sampled by the two nodes are spatial correlation and a part of one's sampling time overlaps that of others, for example, the data sampled by N_i during $[t_1, t_3]$ overlap that by N_j during $[t_1, t_3]$. Notice that $[t_1, t_3]$ and $[t_2, t_4]$ can be considered to be the same time duration. Then, the time t_1 and t_2 point to the same moment. On the basis of the idea above and the KMP algorithm [38], we design Algorithm 4 to find the same moments of different nodes' clocks, which point to the same true time moment. In this way, we can know the relative time difference among nodes, so we can aggregate the data sampled asynchronously. In Algorithm 4, we set a time threshold t_t in order to make the algorithm feasible.

After Algorithm 4 is implemented, a node can know the clock difference between its own and others before it aggregates its data with other node. Hence, the node can aggregate data with others in an exact time, and the data aggregation schemes adopted in the afterward periods could be same with the case in Section 4.

DOI: 10.1002/dac

Algorithm 4 Matching sampling time

Input: $D(p, T_q, i)$ and $D(p, T_q, j)$, $i \neq j$;
Output: A time quantum τ, which satisfies $D([p, p + t_t], T_q, i) = D([p + \tau, p + t_t + \tau], T_q, j)$, where $0 \leq \tau < K - t_t$.

Each parent node u counts the number of child nodes;
for $p = 1, \cdots, K - \tau$ **do**
  for $\tau = t_t, \cdots, K$ **do**
    Let w=KMP($D([p, p + t_t], T_q, i)$, $D([p + \tau, p + t_t + \tau], T_q, j)$);
    if w is true **then**
      Output τ.
    end if
  end for
end for
Output τ is not found.

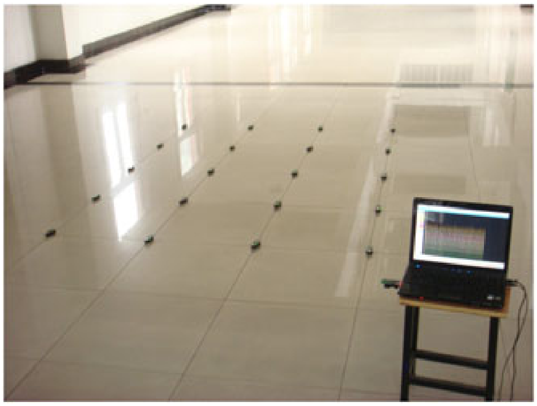


Figure 3. The deployment scenario.

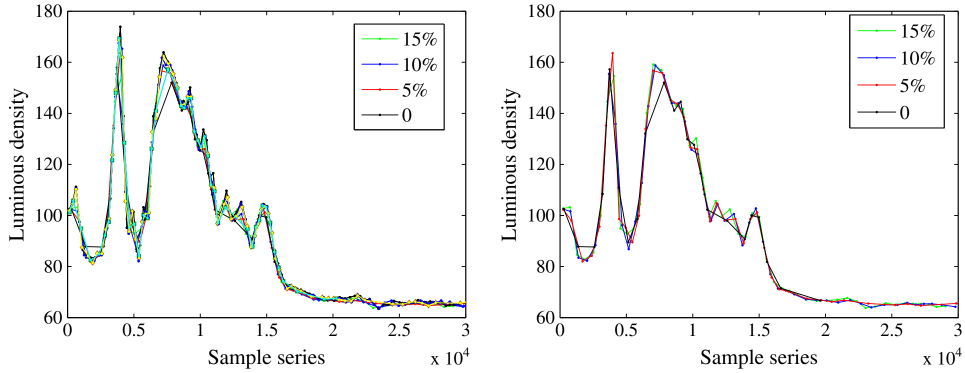
6. EXPERIMENT ANALYSIS

In this section, we designed a real test-bed and detailed experiment to evaluate our LDA schemes. We design three groups of experiments. The first group is that the nodes sample data in synchronous way. The second one is that the nodes sample data in a asynchronous way. The third one is that there exists link unreliability. In each group, there contains two cases. Under one, all nodes sample data whereas only a part of nodes sample data under the other.

6.1. *Test-bed setting*

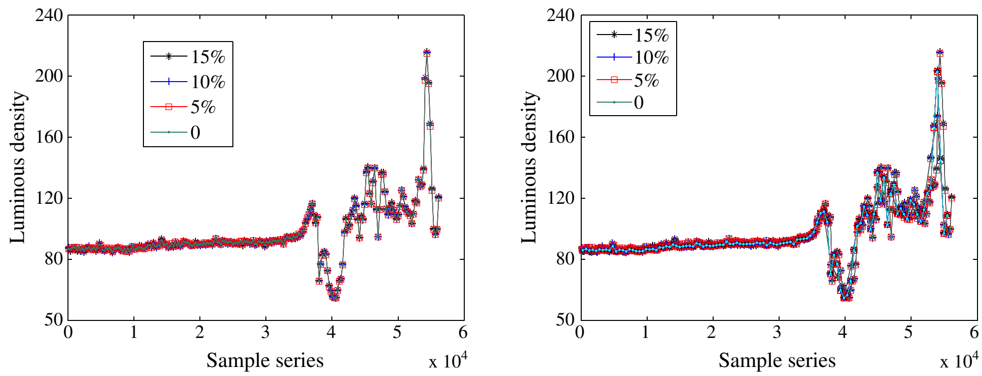
The test-bed is composed of 20 TelosB sensor nodes whereas an additional sensor node acts as the root. The test-bed is deployed in an indoor scenario as shown in Figure 3. All nodes are deployed into four rows and five columns. In the same rows and columns, each pair of neighboring nodes are 60 cm away from each other. Each TelosB node runs in a TinyOS 1.1 [39]. In the experiment of this paper, we let each sensor node sample the light intensity and pack the sampled data into data packets. At first, we run the breadth-first search algorithm to construct a rooted spanning tree. Then, we use the algorithm of Wan *et al*. [33] to construct a dominating tree. In the experiments, the sample period of all nodes is set as 5 s, and each light sensor periodically samples the luminous intensity. In order to obtain synchronous time, we use a flooding method to synchronize the network time [40]. Then, we can controllably set the clocks of all nodes to be asynchronous time by allocating each node with beginning time obeying the Gaussian random distribution.

DOI: 10.1002/dac



(a) 50 period intervals (b) 100 period intervals

Figure 4. Synchronous sample with different period intervals.



(a) 50period intervals (b)100 period intervals

Figure 5. Asynchronous sample with different period intervals.

Table I. Average error.

Confidence level	15%	10%	5%	0%
Synchronous sample	0.25961	−0.2268	−1.11392	−1
Asynchronous sample	0.05	−0.20909	−0.15	−4.074

6.2. Result analysis

We give the first group of experiments in which all nodes sample data synchronously. The group of experiment is implemented from 13:30:39 to 22:04:04 on 29 July. The results are shown in Figure 4. Each subfigure presents four experiment, in which there are respectively 15%, 10%, 5%, and 0% nodes not to sample data. The sample time cannot be set arbitrarily on a real sensor node, such as TelosB. So, we let sensor nodes sample data with high rate to make up the too short work time t_w. Figure 4 presents two kinds of sample rate although the periods are same. In (a), the interval is 50 periods between each two adjacent samples whereas the interval is 100 in (b).

Under the asynchronous sampling, the beginning time of all nodes are random. We suppose that their beginning time obeys the Gaussian random distribution and set the average of the Gaussian distribution as $\mu = 7500$ ms and the deviation as $\sigma = 1000$ ms. The experimental results are shown in Figure 5, in which the interval setting is same with that in Figure 4. In both cases of synchronous and asynchronous samples, the errors among the sample values are very small when different percent sensor nodes stop sampling data. Table I gives the average error in whole work lifetime under both cases. We can easily find that the errors of luminous density are quite small.

DOI: 10.1002/dac

Table II. Energy saving (mW).

	Period interval	15%	10%	5%	0%
Synchronous sample	50	83.037	89.996	92.963	98.897
	100	38.013	40.748	42.056	44.504
Asynchronous sample	50	112.296	117.813	121.499	126.588
	100	63.012	65.023	67.846	69.623

In Table II, we illustrate the energy consumption in our experiment. The energy consumption contains two parts: communication and sampling. The transmission power of each node is −18 dBm. From Table II, we can observe that energy consumption is higher when a network works under higher confidence level. Thus, the LDA scheme in this paper can save energy. For example, 9% energy is saved under synchronous sample when the confidence level is 10%, and period interval is 50.6.61% energy is saved under asynchronous sample when the confidence level is 10% and period interval is 100.

6.3. *In comparison with an existing method*

We also set up an outdoor experiment on the same test-bed as shown in Figure 3 and compare our scheme with the existing method DOG [28]. In this experiment, we measure luminous density from 6:00 PM to 5:00 AM. Sampling rate is set to be once per minute. The transmission power of each node is −5 dBm because a network is deployed regularly and each pair of neighboring nodes is no more than 1 m apart. When no strategy is used, all nodes participate in the measurement of luminous density in every minute. The experimental result is illustrated in Figure 6.

When comparing our scheme with DOG [28], we set confidence level as 10% in LDA and $k = 5$ in DOG, where k is the number of nodes selected each time. DOG is a distributive algorithm for node selection, in which k nodes are selected each time to measure some natural parameters. The purpose in selecting k nodes is to minimize the *expected mean squared prediction error* (EMSPE), which can converge to a stable small value after taking a large number of iterations to find a good set of nodes. Two models, broadcast model and p2p, are considered in [28]. Under both models, DOG has to cost much time on node selection and consume much energy on communication.

Figure 7 shows the percentage of nodes to participate in measurement. Although DOG needs only five nodes to measure the natural parameter, luminous density, much more than this number of nodes are involved because DOG has to select five from the actually involved nodes to minimize EMSPE. The number of those involved nodes is quite affected by the variation of value of the parameter the network is measuring. For example, the value of luminous density varies greatly from

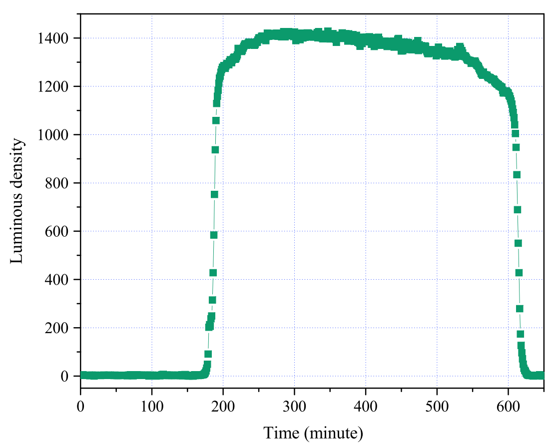


Figure 6. Luminous density measured with no strategy.

DOI: 10.1002/dac

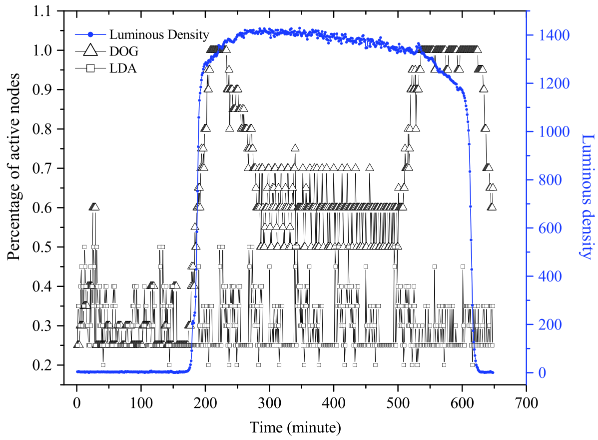


Figure 7. Percentage of active nodes.

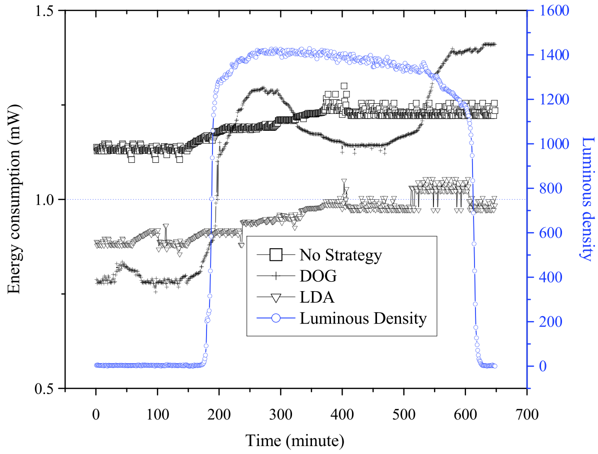


Figure 8. Energy consumption per node in each minute.

time 170 to 250 and from 500 to 605 in Figure 7. Thus, the percentages of nodes to measure luminous density increase dramatically, even up to 100%, in these two periods by DOG. Comparatively, the percentage keeps between 25% and 40% at most of time by our scheme LDA. As a result, DOG consumes much more energy than that by LDA when the value of luminous density varies greatly during those two periods as shown in Figure 8. It deserves to be specially noted that DOG consumes more energy than that by no strategy from time 220 to 320 and from 550 to 650 in Figure 8. That is because DOG spends much time and energy on node selection by broadcast or p2p, as we address that communication is much more energy consuming than that caused by processing data.

However, DOG has its advantage on minimizing EMSPE. In the whole period from time 0 to 650, the maximal error, compared with that by no strategy, obtained by DOG is not more than 5 and that by LDA is not more than 10. Therefore, LDA consumes less energy, involves less number of nodes than DOG, and achieves a little higher error bound. Actually, DOG is power and time consuming whereas LDA can save energy.

7. CONCLUSION

This paper firstly studied the error between the value obtained from the sampled data of all or a part of nodes, and its real one is bounded when assuming that there is no link unreliability. The relation between the error bound and the number of sampling nodes or time slots was given out when the confidence level was previously given. In order to minimize the number of sampling child nodes

DOI: 10.1002/dac

and time slots, we designed algorithms to assign confidence levels to child nodes of the root on the basis of their corresponding confidence levels. We studied the case when link unreliability exists and computed the probability bound when the confidence level and error bound are given. By our LDA scheme, different amounts of energy could be saved whereas different confidence levels were required. We also designed an algorithm to implement LDA under asynchronization. Our scheme was implemented on real test-beds of TelosB nodes, and its results were compared with an existing algorithm DOG. The experimental results illustrated that our scheme can save energy extremely.

There are some works waiting to be solved in the future. When the distribution of the sampled data does not obey Gaussian distribution, we will estimate the probability density function in a complex and time-variation environment. As the paper has analyzed the error bound under the certain confidence level when the link unreliability probability $P_l > 0$, we will design the algorithm that allocates the error bound and confidence level with the existence of the link unreliability probability.

We will analyze the error and confidence level allocation when the beginning time of each node to sample data is asynchronous, as the clock offset in WSNs is unavoidable and relatively large [41] and it costs much extra energy to synchronize time in the network.

ACKNOWLEDGEMENTS

This paper is supported by the National Natural Science Foundation of China under Grants No.61003298, and No.60803126 and the sub-foundation of Zhejiang Provincial Key Innovation Team on Sensor Networks under Grant No.2011R09046-07 and the Natural Science Foundation of Zhejiang Province under Grant No.Z1080979.

REFERENCES

1. Estrin D. Wireless sensor network, part IV:sensor network protocols. In *The ACM/IEEE 18th Annual International Conference on Mobile Computing and Networking (MOBICOM*, Westin Peachtree Plaza, Atlanta, Georgia, USA, Sept. 23–28 2002; 23–28.
2. Krishnamachari L, Estrin D, Wicker S. The impact of data aggregation in wireless sensor networks. In *IEEE The 22nd International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, 2-5 July 2002; 575–578.
3. Wu Y, Li XY, Liu YH, Lou W. Energy-efficient wake-up scheduling for data collection and aggregation. *IEEE Transactions on Parallel and Distributed Systems* 2010; **21**(2):275–287.
4. Eskandari Z, Yaghmaee MH, Mohajerzadeh AH. Energy efficient spanning tree for data aggregation in wireless sensor networks. In *Proceedings of 17th International Conference on Computer Communications and Networks (ICCCN)*, St.Thomas U.S. Virgin Islands, August 3–7 2008; 1–5. IEEE.
5. Wan PJ, Huang CH, Yao F. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *26th Annual IEEE Conference on Computer Communications (INFOCOM)*, Anchorage, Alaska, USA, May 6–12 2007; 366–372.
6. Yu B, Li J, Li Y. Distributed data aggregation scheduling in wireless sensor networks. In *28th Annual IEEE Conference on Computer Communications (INFOCOM)*, Rio de janeiro, Brazil, 19th–25th April 2009; 2159–2167.
7. Gao J, Guibas L, Milosavljevic N, Hershberger J. Sparse data aggregation in sensor networks. *In the 6th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Cambridge, Massachusettes,U.S.A, April 25–27 2007; 430–439.
8. Li XY, Wang Y, Chen H, Chu X, Wu Y, Qi Y. Reliable and energy-efficient routing for static wireless ad hoc networks with unreliable links. *IEEE Transactions on Parallel and Distributed Systems* 2008; **20**(10):1408–1421.
9. Vutukuru M, Balakrishnan H, Jamieson K. Cross-layer wireless bit rate adaptation. *ACM SIGCOMM Computer Communication Review* 2009; **39**(4):3–14.
10. Guo S, Gu Y, Jiang B, He T. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In *The ACM/IEEE 15th Annual International Conference on Mobile Computing and Networking (MOBICOM)* 2009; 133–144. ACM.
11. Chen J, Xu W, He S, Sun Y, Thulasiraman P, Shen X. Utility-based asynchronous flow control algorithm for wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 2010; **28**(7):1116–1126.
12. Li X, Tang S, Frieder O. Multicast capacity for large scale wireless ad hoc networks. In *The ACM Thirteenth Annual International Conference on Mobile Computing and Networking (MOBICOM)*, Montréal, Québec, Canada, September 9–14 2007; 266–277.
13. Dodge Y. *The Oxford Dictionary of Statistical Terms*. Oxford University Press: USA, 2006.
14. Zhang J, Shen X, Dai G, et al. Energy-efficient lossy data aggregation in wireless sensor networks. *Ad hoc and Sensor Wireless Networks* 2011; **11**(1–2):111–135.
15. Son D, Krishnamachari B, Heidemann J. Experimental study of concurrent transmission in wireless sensor networks. In *Proceeding of ACMthe 4th international conference on Embedded networked sensor systems (Sensys)*, Boulder, Colorado, USA, November 1–3 2006; 237–250.

DOI: 10.1002/dac

16. Xu XH, Wang SG, Mao XF, Tang SJ, Li XY. An improved approximation algorithm for data aggregation in multi-hop wireless sensor networks. In *Proceedings of the 2nd ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing*, New Orleans, USA, May 18, 2009; 47–56.
17. Wang L, Wan Z-Y, Wan P-J, Huang C-H, Jia X. Minimum-latency aggregation scheduling in multihop wireless networks. *In Proceeding of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, New Orleans, Louisana, USA, May 18-21 2009; 185–194.
18. Wei G, Ling Y, Guo B, Xiao B, Vasilakos AV. Prediction-based data aggregation in wireless sensor networks: combining grey model and Kalman filter. *Computer Communications* 2011; **34**(6):793–802.
19. Zhang J, Shen X, Tang S, Dai G. Energy efficient joint data aggregation and link scheduling in solar sensor networks. *ElSEVIER Computer Communications* 2011; **34**:2217–2226.
20. Hu Y, Yu N, Jia X. Energy efficient real-time data aggregation in wireless sensor networks. In *the ACM international conference on Wireless communications and mobile computing (IWCMC)*, Vancouver, Canada, July 3–6 2006; 803–808.
21. Yu Y, Krishnamachari B, Prasanna VK. Energy-latency tradeoffs for data gathering in wireless sensor networks. *In 23th Annual IEEE Conference on Computer Communications (INFOCOM), volume 1*, Hong Kong, China, March 7–11 2004; 244–255.
22. Li M, Wan PJ, Yao F. Tighter approximation bounds for minimum CDS in wireless ad hoc networks. *Algorithms and Computation* 2009:699–709.
23. Ma Y, Guo Y, Ghanem M. RECA: referenced energy-based CDS algorithm in wireless sensor networks. *WILEY International Journal of Communication Systems* 2010; **23**(1):125–138.
24. Hartl G, Li B. Loss inference in wireless sensor networks based on data aggregation. In *Proceedings of the ACM 3rd international symposium on Information processing in sensor networks (IPSN)*, Berkeley, California, USA, April 26–27 2004; 396–404.
25. Lee H, Keshavarzian A. Towards energy-optimal and reliable data collection via collision-free scheduling in wireless sensor networks. In *Proceeding of IEEE The 27th Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, April 13-18 2008; pages 2029–2037.
26. Tang X, Xu J. Extending network lifetime for precision-constrained data aggregation in wireless sensor networks. In *the IEEE 25th International Conference on Computer Communications (INFOCOM)*, Barcelona, Catalunya, Spain, April 23–29 2006; pages 1–12.
27. Jindal A, Psounis K. A clustering method that uses lossy aggregation of data. In *Proceedings of the ACM 2nd international conference on Embedded networked sensor systems (SenSys)*, Baltimore, MD, USA, November 3–5 2004; 269–270.
28. Golovin D, Faulkner M, Krause A. Online distributed sensor selection. In *the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Stockholm, Swenden, April 12–16 2010; 220–231.
29. Deshpande A, Guestrin C, Madden SR, Hellerstein JM, Hong W. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume (VLDB)*, Toronto, Ontario, Canada, 2004; 588–599. VLDB Endowment.
30. Banerjee T, Chowdhury KR, Agrawal DP. Using polynomial regression for data representation in wireless sensor networks. *WILEY International Journal of Communication Systems* 2007; **20**(7):829–856.
31. Shi L, Cheng P, Chen J. Sensor data scheduling for optimal state estimation with communication energy constraint. Elsevier. *Automatica* 2011; **47**(8).
32. Penrose MD. The longest edge of the random minimal spanning tree. *The Annals of Applied Probability* may 1997; **7**(2):340–361.
33. Wan P-J, Alzoubi KM, Frieder O. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.* 2004; **9**(2):141–149.
34. Kendall MG, Stuart A. *Inference and relationship*, (3rd edn), The Advanced Theory of Statistics, Vol. 2. Griffin: London, 1973.
35. Klenke A. *Probability Theory: A Comprehensive Course*. Springer Verlag, 2008.
36. Ebeling C. *An Introduction to Reliability and Maintainability Engineering*. The McGraw-Hill Companies, 1997.
37. Subramaniam S, Kalogeraki V, Palpanas T. Distributed real-time detection and tracking of homogeneous regions in sensor networks. In *the IEEE 12th Real-Time and Embedded Technology and Applications Symposium (RTSS),* San Jose, California, USA, April 4–7 2006; 401–411.
38. Knuth D, Morris JH, Pratt Jr V. Fast pattern matching in strings. *SIAM Journal on Computing* 1977; **6**(2):323C350.
39. Levis P, Madden S, Polastre J, Szewczyk R, Whitehouse K, Woo A, Gay D, Hill J, Welsh M, Brewer E, *et al.* Tinyos: An operating system for sensor networks. *Springer Ambient Intelligence* 2005; **35**.
40. Maróti M, Kusy B, Simon G, Lédeczi Á. The flooding time synchronization protocol. In *ACM the 2nd International Conference on Embedded Networked Sensor Systems (SenSys*, New York, NY, USA, 2004; 39–49.
41. Hamilton BR, Ma X, Zhao Q, Xu J. ACES: adaptive clock estimation and synchronization using Kalman filtering. In *The ACM/IEEE 24th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, 2008; 152–162.

DOI: 10.1002/dac

AUTHORS' BIOGRAPHIES

Jianhui Zhang received his PhD degree and BS degree from Zhejiang University and Northwestern Polytechnical University, China in 2008 and 2000, respectively. He is an assistant professor of computer science at the Hangzhou Dianzi University now. His research interests include algorithm design and analysis, distributive systems, sensor networks, and wireless networks. He is a member of IEEE.

Xingfa Shen received his PhD degree and BS degree from Zhejiang University, China in 2007 and 2000, respectively. He is now an associate professor of computer science at the Hangzhou Dianzi University. His research interests include sensor networks and wireless networks. He is a member of IEEE and ACM.

Hong Zeng received his BS degree from Hangzhou Dianzi University, China in 2004. He is an assistant professor of computer science at the Hangzhou Dianzi University now. His work focuses on embedded system design.

Guojun Dai is a professor at Hangzhou Dianzi University, and he received his PhD degree and BS degree from Zhejiang University, China in 1998 and 1991, respectively. His work is on embedded system design and application. He is a member of IEEE and ACM.

Cheng Bo is a PhD student at Illinois Institute of Technology, USA from 2011. He received his BS degree from Hangzhou Dianzi University in 2011. His research interests include embedded system design and sensor networks.

DOI: 10.1002/dac

Fang Chen received his BS degree from Hangzhou Dianzi University in 2011 and works in Huawei Science and Technology Limited Company.

Changping Lv is an associated professor in Hangzhou Dianzi University. His research interests include embedded system design and energy management in wireless networks.

DOI: 10.1002/dac