STCDG: An Efficient Data Gathering Algorithm Based on Matrix Completion for Wireless Sensor Networks

Jie Cheng, Qiang Ye, *Member, IEEE,* Hongbo Jiang, *Member, IEEE,* Dan Wang, *Member, IEEE,* and Chonggang Wang, *Senior Member, IEEE*

***Abstract*—Data gathering in sensor networks is required to be efficient, adaptable and robust. Recently, compressive sensing (CS) based data gathering shows promise in meeting these requirements. Existing CS-based data gathering solutions require that a transform that best sparsifies the sensor readings should be used in order to reduce the amount of data traffic in the network as much as possible. As a result, it is very likely that different transforms have to be determined for varied sensor networks, which seriously affects the adaptability of CS-based schemes. In addition, the existing schemes result in significant errors when the sampling rate of sensor data is low (equivalent to the case of high packet loss rate) because CS inherently requires that the number of measurements should exceed a certain threshold. This paper presents STCDG, an efficient data gathering scheme based on matrix completion. STCDG takes advantage of the low-rank feature instead of sparsity, thereby avoiding the problem of having to be customized for specific sensor networks. Besides, we exploit the presence of the short-term stability feature in sensor data, which further narrows down the set of feasible readings and reduces the recovery errors significantly. Furthermore, STCDG avoids the optimization problem involving empty columns by first removing the empty columns and only recovering the non-empty columns, then filling the empty columns using an optimization technique based on temporal stability. Our experimental results indicate that STCDG outperforms the state-of-the-art data gathering algorithms in terms of recovery error, power consumption, lifespan, and network capacity.**

***Index Terms*—Wireless sensor networks, data gathering, matrix completion, compressive sensing.**

I. Introduction

Wireless sensor networks (WSNs) are expected to be used in many applications such as forest fire detection and habitat monitoring. Data gathering is one of the classical problems to be tackled in WSNs. Typically, a data gathering sensor network consists of a sink and many sensor nodes. The sink serves as a gateway to connect the sensor network and the Internet. Over the Internet, users can query the network by sending an inquiry packet to the sink. After receiving a user query, the sink forwards it to the sensor nodes. Once the responses from the sensor nodes come back, the sink sends the query results back to the user.

Efficiency and adaptability are two very important issues in data gathering. With the traditional data gathering approach [19], the sink receives one data packet from each sensor node in the typical scenario mentioned previously, leading to a large amount of traffic. We call this approach “Centralized Exact” in this paper. As the sensor nodes are often battery-powered, the intensity of data traffic has a serious impact on the lifespan of WSNs. If the amount of the resulting traffic can be reduced, the lifespan of the whole network will be significantly prolonged. Recently, Compressive Data Gathering (CDG), a state-of-the-art data gathering algorithm based on compressive sensing (CS), has been proposed to extend the lifetime of WSNs in this manner [17]. Utilizing the sparsity of sensor readings, CDG only needs fewer data packets than Centralized Exact to acquire a snapshot at a high level of accuracy. However, to reduce the amount of traffic as much as possible, CDG requires that a transform that best sparsifies sensor readings should be used. As a result, it is very likely that different transforms have to be determined for varied sensor networks, which affects the adaptability of CDG seriously. Furthermore, CDG assigns the recovered data to different sensor nodes according to a predefined order, implicitly assuming that the ordering for data reconstruction is fixed all the time. That could be invalid in practice. In our research, we found that the ordering for data reconstruction should be reshuffled at a certain rate over time. Efficient Data Gathering Approach (EDCA) is another innovative data gathering scheme [6]. It takes advantage of the low-rank feature to achieve both less traffic and high-level accuracy. However, EDCA does not consider the possible empty columns in the data matrix. Practically, when the sampling rate is extremely low (or the packet loss rate is very high), the existence of empty columns will become a high probability, seriously reducing the recovery accuracy of EDCA. Finally, for both CDG and EDCA, a low sampling rate tends to result in insufficient measurements, leading to large recovery errors.

To address the problems mentioned previously, we proposed an innovative data gathering scheme based on matrix completion [5], Spatio-Temporal Compressive Data Collection (STCDG). STCDG makes use of both the low-rank and short-term stability features to reduce the amount of traffic and improve the level of recovery accuracy. Compared with CDG,

J. Cheng and H. Jiang (corresponding author) are with Department of Electronics and Information Engineering, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, 430074. China. Email: {jiecheng2009,hongbojiang2004}@gmail.com. J. Cheng is also with University of Prince Edward Island, Canada.

Q. Ye is with University of Prince Edward Island, Canada. Email: qye@upei.ca.

D. Wang is with The Hong Kong Polytechnic University, Hong Kong. Email: csdwang@comp.polyu.edu.hk.

C. Wang is with InterDigital Communications, U.S.A. Email: cgwang@ieee.org

STCDG is much more adaptable since it is independent of specific sensor networks. In addition, to achieve the same level of accuracy, STCDG only requires a smaller fraction of the sensor readings than both CDG and EDCA. Furthermore, STCDG avoids the optimization problem involving empty columns by first removing the empty columns and only recovering the non-empty columns, then filling the empty columns using an optimization technique based on temporal stability. Our experimental results indicate that STCDG outperforms Centralized Exact, CDG, and EDCA. In detail, the contributions of this paper are listed as follows.

- We set up a sensor network testbed. Through an in-depth analysis of the testbed traces, we conclude that WSNs exhibit the low-rank and short-term stability features.
- We propose an efficient data gathering scheme based on matrix completion, STCDG, that utilizes the low-rank and short-term stability features in WSNs to achieve both reduced data traffic and high level of recovery accuracy. Our experimental results indicate that STCDG outperforms Centralized Exact, CDG, and EDCA in terms of recovery errors, energy consumption, and lifespan.
- We analyze the network capacity of STCDG theoretically and validate the analysis results through ns-2 simulations.
- We prove that TDMA-based scheduling for STCDG is an NP-complete problem. Furthermore, we devise a TDMA-based scheduling algorithm that minimizes the number of required time slots when STCDG is used in WSNs.

The rest of this paper is organized as follows. Section II discusses the related work and Section III describes the details of STCDG. The experimental results based on two testbed data sets are presented in Section IV. Finally, Section V concludes this paper.

II. Related work

In data gathering sensor networks, in-network data suppression and compression are the major methods to reduce the amount of data traffic, ultimately leading to low power consumption and long lifespan. The spacial and temporal correlation of sensor readings are the foundation of the existing data suppression and compression techniques.

Traditional Source Coding is an in-network data compression method that takes advantage of the spacial correlation on the encoding side [26], [13], [8]. To achieve the best compression performance, it usually requires the coordination of sensor nodes. Yoon $et\ al.$ proposed the Clustered AGgregation (CAG) method that divides a sensor network into clusters according to sensor readings [26]. With the clusters, only one reading per cluster is forwarded to the sink and the overall error is still less than a predefined threshold. However, traditional source coding has several limitations. For example, its compression efficiency heavily depends on the routing protocol used in the network. However, the joint optimization of compression and routing has been proved to be NP-hard [8].

Distributed Source Coding (DSC) is an improvement over Source Coding in the sense that it attempts to reduce the complexity at the sensor nodes and make use of the spacial correlation at the sink [7]. The Slepian-Wolf theorem [23] is the theoretical foundation for most DSC algorithms. It indicates that when correlated readings are encoded separately, the resulting compression can be as efficient as the traditional compression when the readings are encoded jointly, as long as the separately encoded messages are decoded jointly. Despite the significant improvement, DSC still has some serious problems. First of all, DSC algorithms usually lead to very high time and space complexity. Secondly, DSC only works well when the correlation among neighboring sensors does not change over time.

Compressive Sensing (CS) is a method for finding sparse solutions to underdetermined linear systems [9]. It has led to a completely different approach to distributed data compression in WSNs. Compared with traditional DSC, CS-based data compression moves most computation from sensor nodes to the sink, which makes it a good fit for in-network data suppression and compression. Over the past years, a variety of CS-based methods have been devised to solve the data gathering problem in WSNs [14], [15], [16]. Haupt $et\ al.$ summarizes the potential of applying CS to the data gathering problem in multi-hop WSNs [14]. Duarte $et\ al.$ exploited both intra- and inter-signal sparsity to lower the sampling ratio and proposed two joint sparsity models for distributed compressed sensing [10]. Wu $et\ al.$ focused on the soil moisture data collection using compressive sensing [25]. They defined a pair of well-designed measurement matrix and representation basis in order to achieve incoherence and sparsity at the same time. Wang $et\ al.$ devised a method to recover the missing data more precisely by considering the linear correlation between the data from different nodes [24]. Zheng $et\ al.$ discussed the energy and latency performance of varied data collection algorithms using compressive sensing [28]. Baron $et\ al.$ studied joint sparsity models and joint data recovery methods based on CS [4]. However, multi-hop communication was not taken into consideration. As mentioned previously, Luo $et\ al.$ proposed CS-based CDG to reduce communication cost and prolong network lifespan [17], [18]. Despite that CDG leads to significantly less traffic and longer lifetime than Centralized Exact, there is still much room for improvement.

Matrix completion is a technique that takes advantage of the low-rank feature to recover the missing entries in a matrix [5], [22]. As an extension of compressive sensing, it has been used in various research areas. Keshavan $et\ al.$ compared three recovery methods based on low-rank matrix completion with noisy observations [15]. Based on nuclear norm minimization, Zhang $et\ al.$ presented a novel approach to estimating the missing values in traffic matrices [27]. We also proposed an efficient data recovery method for data gathering based on matrix completion in 2010 [6]. However, the proposed method only utilizes the low-rank feature of the data matrix. In this paper, we present an innovative recovery algorithm for data gathering that takes advantage of both the low-rank and short-term stability features in WSNs. To our knowledge, this is the first data gathering algorithm based on matrix completion that makes full use of the low-rank and short-term stability features.

III. The STCDG Scheme

In this section, we describe our approach to efficient data gathering in WSNs. Because our approach utilizes the matrix completion technique inspired by compressive sensing and takes advantage of the low-rank and short-term stability features resulting from the spatial/temporal correlation in WSNs, the proposed mechanism is named Spatio-Temporal Compressive Data Gathering (STCDG). The details of STCDG are presented as follows.

A. Preliminaries

In our research, we consider a sensor network consisting of N nodes. Each node is assigned an integer ID, n, which is in the range of 1 to N. We assume that all the readings generated by sensor nodes are positive real numbers. We also assume that time is divided into equal-sized time slots. With the Centralized Exact algorithm, during each time slot, every sensor node probes the environment and forwards the reading to the sink through a multi-hop path. As a result, N readings can be collected at the sink for each time slot. For T time slots, $N \times T$ readings can be gathered. These readings can be organized into an $N \times T$ matrix X ($X \in \mathbb{R}^{N \times T}$), where the row and column number correspond to the node ID and time slot number respectively.

With STCDG, each sensor node only forwards its readings to the sink according to a preset probability (i.e., a preset sampling ratio). As a result, only a fraction of the readings from each node are transmitted to the sink, leading to a variety of different benefits such as reduced traffic and prolonged lifespan. Of course, this also leaves some entries in X empty. In our research, when an entry in X is missing, we use zero as a placeholder to replace the entry. In addition, we use B to denote this modified matrix. Note that B is the matrix available at the sink when STCDG is used to collect the readings. Furthermore, we define a special $N \times T$ matrix, Q, using Eq. (1):

$$Q(n,t) = \begin{cases} 0, & \text{if } X(n,t) \text{ is unavailable.} \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

where t is the sequence number of the time slot within the T-slot time window. Obviously, we have:

$$B = X. * Q \tag{2}$$

where $.*$ represents a scalar product (or dot product) of two matrices. Namely, $B(n,t) = X(n,t)Q(n,t)$.

Using the matrix completion technique inspired by compressive sensing, STCDG attempts to recover the missing entries efficiently. Namely, STCDG tries to use the incomplete data matrix B to generate an approximation matrix, $\tilde{X}$, each entry of which is sufficiently close to the corresponding entry in X quantitatively.

B. Low rank and short-term stability

To deeply understand the low-rank and short-term stability features in WSNs, we thoroughly analyzed two sets of data from two independent sensor network testbeds. The first set of data was collected from 54 sensors at the Intel Berkeley Research Lab between February 28, 2004 and April 5, 2004 [2]. This set of data contains four different traces that correspond to temperature, humidity, light, and voltage, respectively. To confirm the low-rank and short-term stability features existing in the first set of data, we set up a sensor network testbed in a two-story residential building located in Charlottetown, Canada. This testbed consists of 1 sink (corresponding to sensor node No. 1) and 24 sensor nodes. The sensor location and floor plan details are included in Fig. 1. The second set of data was collected from this testbed during the period of August 24, 2012 to August 27, 2012. It also includes four traces corresponding to temperature, light, humidity, and voltage, respectively. We found that, for the traces under investigation, the data matrix X always exhibits both the low-rank and short-term stability features. The details of our experimental results are presented as follows.

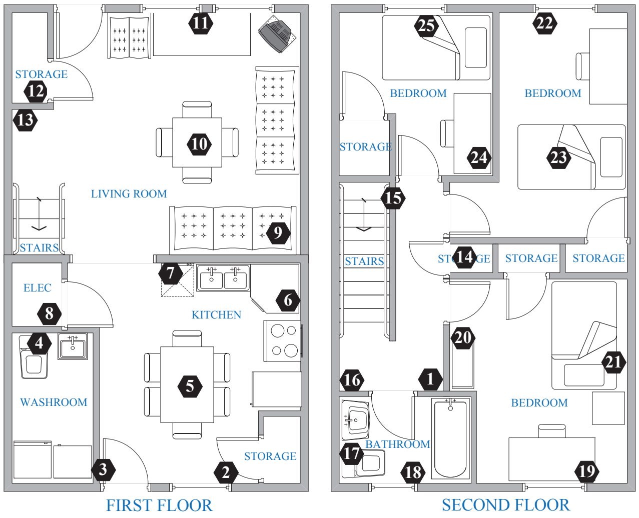


Fig. 1. Sensor location details

To check whether the data matrix X has a good low-rank approximation, we used singular value decomposition (SVD). Specifically, X, an $N \times T$ matrix, can be decomposed using SVD according to Eq. (3):

$$X = U \Sigma V^T \tag{3}$$

where U is an $N \times N$ unitary matrix, V is a $T \times T$ unitary matrix, and Σ is a $N \times T$ diagonal matrix with the diagonal elements (i.e. the singular values) $\sigma_1, \sigma_2, \cdots, \sigma_r$ organized in a decreasing order (here r denotes the rank of X). The metric that we used to determine whether X has a good low-rank approximation is the fraction of the nuclear form captured by the top d singular values. Formally, the fraction is defined using Eq. (4):

$$g(d) = \frac{\sum_{i=1}^{d} \sigma_i}{\|X\|_*} = \frac{\sum_{i=1}^{d} \sigma_i}{\sum_{i=1}^{r} \sigma_i} \tag{4}$$

where σ_i is the i-th largest single value and $\| \cdot \|_*$ denotes the nuclear norm of a matrix.

The fraction of the nuclear norm captured by top d singular values in the case of the testbed traces are presented in Fig. 2.

We found that the top 5 singular values capture $82\% - 99\%$ of the nuclear norm. These results indicate that the data matrix X has a good low-rank approximation in all the scenarios under investigation.

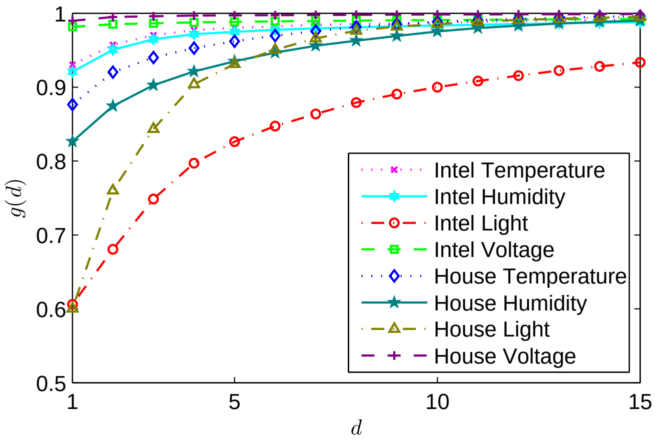


Fig. 2. Fraction captured by top d singular values

To study the short-term stability of X, we calculated the gap between each pair of adjacent readings for each sensor node and compared the difference between each pair of adjacent gaps. Specifically, the gap between each pair of adjacent readings is equal to $gap(n,t) = (X(n,t) - X(n,t-1))$, where $1 \leq n \leq N$ and $2 \leq t \leq T$. Consequently, the difference between each pair of adjacent gaps is equal to $dif(n,t) = ((X(n,t+1) - X(n,t)) - (X(n,t) - X(n,t-1)))$ $= X(n,t+1) + X(n,t-1) - 2 \cdot X(n,t)$, where $1 \leq n \leq N$ and $2 \leq t \leq T-1$. The smaller the resulting $dif(n,t)$, the stabler the sensor readings for node n around the time slot t. To compare the short-term stability feature of varied traces, we calculated the normalized difference for each entry in X using Eq. (5):

$$\begin{aligned} h(n,t) &= \frac{|dif(n,t)|}{mean\ \ gap} \\ &= \frac{|X(n,t+1) + X(n,t-1) - 2 \cdot X(n,t)|}{\frac{\Sigma_{1 \leq n' \leq N, 2 \leq t' \leq T}(|X(n',t') - X(n',t'-1)|)}{N \cdot (T-1)}}, \\ & 1 \leq n \leq N, 2 \leq t \leq T-1 \end{aligned} \tag{5}$$

where $|\cdot|$ represents the absolute value of a quantity, n' denotes a node ID, and t' is the sequence number of a time slot. Furthermore, we define $f(H)$ as the cumulative distribution function of $\{h(n,t)\}$, where H is in the range of 0 to 2. For a small H, if the resulting $f(H)$ is large, then we can conclude that, overall, the sensor readings do not change much in the short term.

Fig. 3 includes the fraction captured by the $h(n,t)$ quantities that are less than H. The results indicate that, for the testbed traces, when H is equal to 0.6, the resulting fraction $f(H)$ is in the range of $46\% \sim 84\%$; when H is set to 0.8, the fraction $f(H)$ is always greater than 58%. Overall, all the traces under investigation exhibit the feature of short-term stability.

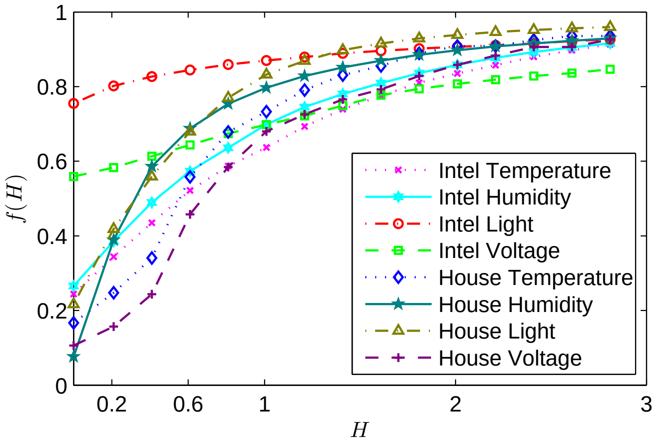


Fig. 3. Fraction captured by the $h(n,t)$ satisfying $h(n,t) \leq H$

C. STCDG details

After finding that the data matrix X exhibits the low-rank and short-term stability features, we devised an innovative data gathering scheme, STCDG, that takes advantage of these features to recover X using partial sensor readings. This leads to a variety of benefits, including low power consumption, long lifespan, and large network capacity. The details of the proposed scheme are described as follows.

Low Rank: Candès *et al.*'s recent work on matrix completion has proved that it is highly possible to recover a low-rank matrix using a subset of its entries [5]. In our research, the recovery problem can be formulated as the following rank minimization problem:

$$\begin{aligned} &\text{minimize} \quad \text{rank}(X), \\ &\text{subject to} \quad \mathcal{A}(X) = B. \end{aligned} \tag{6}$$

where $rank(\cdot)$ denotes the rank of a matrix, $\mathcal{A}(\cdot)$ is a known affine transformation, and B is the transformed matrix obtained by the sink when STCDG is used. However, solving this rank minimization problem is often impractical because it is NP-hard. The time complexity of existing solutions is at least doubly exponential in the dimension of the matrix. We solved this optimization problem using the nuclear norm heuristic [5]. Furthermore, in our research, we used a specific type of affine transformation: scalar product operation. As a result, the previous problem can be changed to the following nuclear norm minimization problem:

$$\begin{aligned} &\text{minimize} \quad \|X\|_*, \\ &\text{subject to} \quad X.*Q = B. \end{aligned} \tag{7}$$

There have been a few effective solutions to the nuclear norm minimization problem [22]. Our approach attempts to find a suitable X whose rank is r to satisfy $X = LR^T$, where L and R are $N \times r$ and $T \times r$ matrices respectively. Since there could be more than one pair of L and R that meet this condition, we try to find a pair of L and R so that $\|L\|_F^2 + \|R\|_F^2$ is minimal (note that $\|\cdot\|_F$ denotes the Frobenius norm of a matrix). Then we can arrive at the following minimization problem:

$$\begin{aligned} &\text{minimize} \quad \|L\|_F^2 + \|R\|_F^2, \\ &\text{subject to} \quad (LR^T).*Q = B. \end{aligned} \tag{8}$$

Furthermore, because the real data matrix X is not exactly low-rank, looking for a low-rank solution that strictly satisfies the sampling equation $(LR^T).*Q = B$ might not work well. So we introduced a regularization parameter ζ which allows a tunable tradeoff between a precise fit to the collected data and the goal of achieving low rank. This led to the following optimization problem:

$$\text{minimize } \|(LR^T).*Q - B\|_F^2 + \zeta(\|L\|_F^2 + \|R\|_F^2) \qquad (9)$$

Short-Term Stability: As mentioned previously, the data matrix also exhibits the feature of short-term stability. To further reduce the recovery error, we introduced another constraint about short-term stability, $\|(LR^T)S^T\|_F^2$. Note that this constraint is the sum of all $dif(n,t)$ values for X. Minimizing this quantity guarantees that the short-term stability feature is preserved. Finally, we arrived at the following minimization problem:

$$\begin{aligned}\text{minimize } &\|(LR^T).*Q - B\|_F^2 + \zeta(\|L\|_F^2 + \|R\|_F^2)\\ &+\eta\|(LR^T)S^T\|_F^2,\end{aligned} \qquad (10)$$

where η is another tuning parameter and $S = Toeplitz$(0, 1, -2, 1), which denotes the Toeplitz matrix with central diagonal given by ones, the first upper diagonal given by negative twos, and the second upper diagonal given by ones. In detail, S can be defined using Eq. (11):

$$S = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ 0 & 0 & 1 & -2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{T\times T} \qquad (11)$$

In this final solution, the tuning parameters ζ and η allow a tradeoff among the optimization targets: satisfying the sampling equation, maintaining the low-rank feature, and achieving short-term stability. Since the sensor readings received by the sink are always 100% precise while real WSNs often exhibit the low-rank and short-term stability features approximately, we assigned a weight of “1” to the first term in Notation (10) and set $\zeta = \eta = 0.1$.

The final solution is a convex optimization problem that could be solved using the alternating least squares procedure proposed by Zhang *et al.* [27]. Specifically, L and R are first initialized randomly. Then we fix one of L and R, and make the other one the optimization variable. In this manner, the problem is converted to a standard linear least squares problem. After this, we swap the roles of L and R, and continue alternating towards a solution till convergence. In our research, STCDG often converges after a moderate number of iterations and results in an acceptable recovery error. The details of the performance of STCDG will be presented in Section V.

Empty Columns: As mentioned previously, with STCDG, each sensor node only forwards its readings to the sink according to a preset probability. Consequently, it is possible that the sink receives no readings during some time slots. In the case that the sampling ratio is low, it is very likely that some empty columns will exist. When this event takes place, some columns in the matrix B will be completely empty, ultimately leading to catastrophic recovery errors. To avoid the serious errors, when there are some empty columns in B, STCDG first ignores the empty columns and only recovers the non-empty columns, then uses the short-term stability feature to fill the empty columns. Of course, when there does not exist any empty-column in B, the previous solution is enough.

Specifically, we assume that B has K empty columns, which implies that B has $T-K$ non-empty columns. When $K \neq 0$, STCDG first generates an $N \times (T-K)$ matrix B', which contains the non-empty columns in B. The sequence of the non-empty columns in B is preserved in B'. Obviously, there is a correspondence between the non-empty columns in B and the columns in B'. Using the solution corresponding to Notation (10), STCDG can arrive at a recovered data matrix X', which has N rows and $(T-K)$ columns. Then STCDG generates a $N \times T$ matrix X'' that also has K empty columns and $T-K$ non-empty columns. The K empty columns in X'' correspond to the K empty columns in B. In addition, the $T-K$ non-empty columns in X'' come from X', but they are placed in the proper locations in X'' according to the correspondence between the non-empty columns in B and the columns in B'. After this, each entry in the empty columns of X'' is filled with a placeholder “0”. Finally, STCDG uses the short-term stability feature to fill the empty columns in X''. Formally, the filling problem is converted into the following minimization problem. This problem can be solved using semidefinite programming (SDP) easily.

$$\text{minimize } \|X - X''\|_F^2 + \|XS^T\|_F^2. \qquad (12)$$

In summary, Notation (10) is used to generate the recovered matrix when there is no empty column in B while Notation (12) is adopted to construct the recovered matrix when there are some empty columns. We use X''' to denote the matrix generated by either Notation (10) or Notation (12). Note that the non-empty entries in B are the precise readings from sensor nodes. Generally speaking, they are more precise than the corresponding entries in the recovered matrix X'''. Hence, these entries in X''' should be replaced by the corresponding entries in B in order to reduce the recover error. Formally, the final approximation matrix $\tilde{X}$ can be calculated using Eq. (13):

$$\tilde{X} = X''' - X'''.*Q + B. \qquad (13)$$

In sensor networks, there could be some abnormal readings when there is a short-term change in the environment (e.g. the light in a room is turned on for a short period of time at night). If these readings are not forwarded the sink, the recovery error will be significantly enlarged. Thus, abnormal readings should be forwarded to the sink regardless of the sampling ratio. To maintain the preset sampling ratio in the long term, after forwarding an abnormal reading, the sensor suspends its normal forwarding operations until the preset sampling ratio allows it again. Note that the suspending rule only applies to normal readings, abnormal readings are always forwarded to the sink immediately.

IV. Network Analysis of STCDG

In this section, we analyze the network capacity of STCDG under protocol model and discuss the TDMA-based scheduling algorithm for STCDG. The details are described as follows.

A. Network capacity under protocol model

In the application of data gathering, many sensor nodes forward their readings to the single sink node. Namely, a many-to-one communication model is used. In this section, we present the network capacity of STCDG under the many-to-one model. Formally, network capacity is defined using Definition 1.

Definition *1:* The network capacity λ in a data gathering sensor network is the achievable rate at which λT bits of data from each sensor node is received by the sink during the T-slot time window.

To further discuss the network capacity λ of STCDG, we have the following assumptions. First of all, the sensor network consists of N static sensor nodes, each of which is equipped with one omni-directional antenna. The sensor nodes share a single-frequency radio channel. As a result, they cannot transmit and receive at the same time. Secondly, the sensor nodes are deployed with a uniform distribution over an experimental field of area A. For simplicity, we assume that the sink is located at the center of the field. Thirdly, all the nodes have the same communication capacity of W bits/slot. Namely, each sensor can transmit or receive at most W bits per time slot. Fourthly, the protocol model is used as the interference model. We use r to denote the transmission range of sensor nodes. The transmission from node v_i to v_j is successful under the protocol model if and only if the following condition is satisfied:

$$\|v_i - v_j\| \leq r \text{ and } \|v_i - v_k\| > r + \delta, \delta \geq 0 \qquad (14)$$

where $\|\cdot\|$ denotes the distance between two nodes and v_k represents any other node node in the network. Finally, we assume that the routing protocol uses a tree structure to forward the readings to the sink. Specifically, the sink node is the root of the routing tree and it has several child nodes. Each of the child nodes leads a subtree. Since the sensor nodes are uniformly deployed, for simplicity, the state-of-the-art data collection schemes based on compressive sensing (especially Luo $et\ al.$'s CDG [17]) assume that all subtrees are roughly of the same size. In addition, they assume that the number of the measurements from each subtree is approximately equal to the total number of required measurements divided by the number of subtrees all the time. To compare STCDG with these recent data collection schemes, we also use these two assumptions in our experiments.

Marco $et\ al.$ presented the following lemma in 2003 [21]. In the lemma, A_r is used to denote the area of a circle with the radius equal to r. Obviously, $A_r = \pi r^2$. The circle is located inside the experimental field.

Lemma *1:* In a randomly deployed network with N nodes,

$$Prob\left\{\frac{NA_r}{A} - \sqrt{\alpha_N N} \leq n_{A_r} \leq \frac{NA_r}{A} + \sqrt{\alpha_N N}\right\} \longrightarrow 1, \quad \text{as } N \longrightarrow \infty$$

where n_{A_r} is the number of nodes in A_r and α_N is such a quantity that $\lim\limits_{N\to\infty} \frac{\alpha_N}{N} = \epsilon$, ϵ is positive but arbitrarily small.

Using Lemma 1, we analyzed the network capacity of STCDG when it is used to collect sensor readings in WSNs. The theoretical result is summarized in Theorem 1.

Theorem *1:* A uniformly deployed network using STCDG can achieve a network capacity of $\lambda \geq \frac{W}{NP}\frac{\pi r^2 - \sqrt{\epsilon}}{\pi(2r+\delta)^2+\sqrt{\epsilon}}$ with high probability as $N \longrightarrow \infty$, where P is the sampling ratio in STCDG and ϵ is arbitrarily close to 0.

Proof: Consider a transmission source that is at least $(2r + \delta)$ from the border of the experimental field. For this source, the area of interference is a circle with the radius equal to $(2r + \delta)$. The source is located at the center of the circle. According to Lemma 1, the number of interfering neighbors, n_1, is limited by the following inequality with high probability:

$$\frac{NA_{(2r+\delta)}}{A} - \sqrt{\alpha_N N} \leq n_1 \leq \frac{NA_{(2r+\delta)}}{A} + \sqrt{\alpha_N N} \qquad (15)$$

We use a graph $G_1(V_1, E_1)$ to denote the network consisting of the n_1 interfering nodes in the circle, where V_1 represents the set of nodes and E_1 is the set of edges corresponding to the communication links. Obviously, the highest degree of this graph is $n_1 - 1$. It is known in graph theory that the chromaticity of such a graph is upper bounded by the highest degree plus one, namely, $n_1 - 1 + 1 = n_1$. Hence, there exists a schedule of at most $l \leq n_1$ time slots that would allow all nodes to transmit at least once during this schedule. Therefore, the transmission rate of each node is:

$$\gamma = \frac{W}{l} \qquad (16)$$

On the other hand, the nodes one hop away from the sink which can communicate with sink directly are the roots of the subtrees. The number of these nodes, n_2, is also bounded with high probability according to Lemma 1:

$$\frac{NA_r}{A} - \sqrt{\alpha_N N} \leq n_2 \leq \frac{NA_r}{N} + \sqrt{\alpha_N N} \qquad (17)$$

Note that the root of the routing tree has n_2 subtrees and the total number of readings received by the sink is NP when STCDG is used. Therefore, the number of readings from each subtree is equal to NP/n_2. Thus, the achievable network capacity λ satisfies the following condition:

$$\frac{W}{l} = \frac{NP\lambda}{n_2} \qquad (18)$$

Combining Notation (15), (17), and (18), we have:

$$\begin{aligned}\lambda = \frac{W}{NP}\frac{n_2}{l} \geq \frac{W}{NP}\frac{n_2}{n_1} \geq \frac{W}{NP}\frac{\frac{NA_r}{A} - \sqrt{\alpha_N N}}{\frac{NA_{(2r+\delta)}}{A} + \sqrt{\alpha_N N}} \\ = \frac{W}{NP}\frac{\pi r^2 - \sqrt{\epsilon}}{\pi(2r+\delta)^2 + \sqrt{\epsilon}}\end{aligned} \qquad (19)$$

■

Theorem *2:* A uniformly deployed sensor network using STCDG can achieve a network capacity gain of $1/P$ over the baseline transmission (i.e. Centralizd Exact), where P is the sampling ratio in STCDG.

Proof: When Centralized Exact is used in the data gathering network, the total number of readings received by the sink is N and thus the number of readings from each subtree is equal to N/n_2. As a result, the achievable network capacity $\lambda' = \frac{W}{N}\frac{n_2}{l}$. Note that the achievable capacity of STCDG is λ=$\frac{W}{NP}\frac{n_2}{l}$. So STCDG can achieve a capacity gain of $\frac{\lambda}{\lambda'} = \frac{1}{P}$ over baseline transmission. ∎

B. TDMA-based scheduling for STCDG

TDMA-based scheduling algorithms that minimize the number of time slots have been proved to be NP-complete [3], [11]. In this section, we prove that TDMA-based scheduling for STCDG is an NP-complete problem. We also present an efficient TDMA-based scheduling algorithm for STCDG.

Theorem *3:* TDMA-based scheduling for STCDG is an NP-complete problem.

Proof: The problem is clearly in NP since an assignment can be verified in polynomial time. To prove that the scheduling problem for STCDG is NP-complete, we can transform the problem to a scheduling problem for baseline transmission, which has been proved to be NP-complete [3], [11]. Specifically, we use $G_2(V_2, E_2)$ to denote the routing tree used for baseline transmission, where V_2 consists of the N nodes and E_2 includes the edges in the routing tree. With STCDG, only part of the nodes in the routing tree forward their readings to the sink during each snapshot. We use D'_i to denote the set of nodes that are randomly selected to forward their data during the i-th snapshot, where i is in the range of 1 to T. Note that T is polynomial of N. Without loss of generality, we assume that $T \sim O(N)$. To forward the data from the nodes in D'_i to the sink, part of the original routing tree is required. We use $G'_i(V'_i, E'_i)$ to denote the partial routing tree. Obviously, $D'_i \subset V'_i \subset V_2$ and $E'_i \subset E_2$. Let $B'_i = V'_i \setminus D'_i$. Note that B'_i contains the bridge nodes that help establish a path from the nodes in D'_i to the sink. Let t_i be a schedule for $G'_i(V'_i, E'_i)$ in the scenario of baseline transmission. Apparently, t_i can also be used as a schedule for the nodes in D'_i in the case of STCDG. Of course, this might lead to some empty time slots because not all nodes in V'_i need to send a packet. To improve the performance of the schedule, we can trim the empty time slots before using it in STCDG. Grandham $et\ al.$ proved that the upper bound of the required time slots in the case of baseline transmission is $3N$ [12]. Note that we can use $O(N)$ operations to trim the empty time slots if necessary. As a result, it takes $O(N)$ operations to generate the schedule for D'_i. To produce T schedules (note that we assumed that $T \sim O(N)$ previously), we need $O(N^2)$ operations at most. This indicates that the transformation can be completed in polynomial time. Therefore, the scheduling problem for STCDG is NP-complete. ∎

In our research, we also devised a collision-free TDMA-based scheduling algorithm for STCDG. The algorithm only uses interference information to minimize the number of required time slots. As mentioned previously, protocol model is used as the interference model in this paper.

Specifically, we use $G(V, E)$ to denote the network under investigation. V consists of the N nodes and E includes all the edges in the network. In our research, we defined the concept of Maximum Non-Interference Set (MNIS) for each time slot. MNIS contains the maximum number of nodes that can transmit data simultaneously without collision during one time slot. Basically, the nodes in MNIS are far enough from each other so that no simultaneous transmissions will collide. In addition, because it is the maximum set, adding one more node to it will lead to some collision.

To acquire a snapshot, a number of time slots are required. Since there is only one common wireless channel in the network, the sink can receive at most one packet during each time slot. To make the data collection delay as short as possible, our scheduling algorithm attempts to establish a schedule with which a packet is sent to the sink during each time slot whenever possible. During each time slot, the randomly-selected node that can send its packet to the sink using the least number of time slots is called the“start point”. Once the start point for a time slot is selected, it is scheduled to send a packet to the sink (or to its parent if it is not a neighbor of the sink) during that slot. Note that, during the time slot, other nodes that have been randomly selected should also forward their packets to their parents as long as these transmissions do not result in collision. Actually, our algorithm is used to find the MNIS for the time slot (note that the MNIS always contains the start point). When the MNIS for the slot is finalized, all the nodes in the MNIS can send their packets during the same slot without collision. Once the data transmissions scheduled for the current time slot is completed, the data collection tree is updated and a new MNIS is generated for the next time slot. This process goes on until all the readings from the randomly selected nodes for the current snapshot have been received by the sink. The performance of this scheduling algorithm is discussed in Section V-E.

V. Implementation and evaluation

In our research, we implemented the proposed data gathering scheme STCDG using MATLAB. Our implementation is composed of three phases. During the initialization phase, the sink broadcasts a fixed probability to the sensor nodes in the network. In our research, this probability is called the *sampling ratio*. The sampling ratio is in the range of 5% to 90%. Note that (1 − *sampling ratio*) indicates the *dumping ratio* of all sensor readings. In the second phase, each node forwards their readings to the sink according to the preset sampling ratio. In the final phase, after collecting the randomly-selected readings over a T-slot time window, the sink recovers the missing readings using the method summarized in Section III-C.

After implementing STCDG, we carried out extensive experiments to evaluate its performance. First of all, we set up a sensor network testbed in a two-story residential building in Charlottetown, Canada and collected the sensing readings from August 24, 2012 to August 27, 2012. Then the gathered data and the traces from the Intel Berkeley Research Lab [2] were used to evaluate the performance of STCDG from the perspectives of recovery error, power consumption, and lifespan. Furthermore, through ns-2 simulations [1], we analyzed the network capacity of STCDG in practical scenarios and studied the performance of the TDMA-based scheduling algorithm presented in Section IV-B.

To compare the recovery performance of STCDG to that of the existing schemes quantitatively, we use the following two definitions in this paper. The error matrix E is defined as the difference between the original matrix and the recovered matrix. Formally, E is defined using Eq. (20):

$$E = X - \tilde{X} \tag{20}$$

Furthermore, we define the concept of Normalized Mean Absolute Error (NMAE) using Eq. (21):

$$NMAE = \frac{\sum_{i,j,Q(i,j)=0} |E(i,j)|}{\sum_{i,j,Q(i,j)=0} |X(i,j)|}, \quad 1 \le i \le N \ and \ 1 \le j \le T \tag{21}$$

Note that NMAE only takes the recovery errors about the missing entries in X into consideration.

For CDG, the sampling ratio is defined as the ratio of the number of measurements that the sink intends to receive, M, to the number of nodes in the network, N. For comparison purposes, the range of the sampling ratio for CDG is also set to 5% to 90%.

A. Testbed configuration

In our research, we used two sets of testbed traces to evaluate the performance of STCDG. The first set contains the real-world traces collected at Intel Berkeley Research Lab in 2004. It involves 54 sensor nodes deployed in the single-floor lab. The details of the sensor location information are available in [2].

To evaluate the recovery performance of STCDG thoroughly, we set up a sensor network testbed in a two-story residential building located in Charlottetown, Canada. The sensor location and floor plan details are included in Fig. 1. Specifically, this testbed involves 25 TelosB sensors deployed on the two floors in the building. Among the 25 sensors, sensor No. 1 is used as the sink and does not sense the environment. The remaining 24 sensors are deployed in such a manner that there exists at least 1 sensor in each independent room and there are multiple sensors in each large room (e.g. living room or kitchen). From this testbed, we collected the second set of testbed traces from August 24, 2012 to August 27, 2012.

In both data sets, each packet from the sensor nodes contains multiple types of sensing information: temperature, humidity, light, and voltage. In our research, we used the temperature and light data to compare STCDG with other data collection schemes. Specifically, we chose the temperature/light traces gathered on March 1, 2004 from the first trace set (we call them the Intel Temperature and Intel Light trace respectively) and those collected on August 25, 2012 from the second set (we call them the House Temperature and House Light trace respectively). In all of these traces, the sink receives one packet from each sensor node once every thirty seconds. This leads to 2880 packets from each sensor node everyday. Furthermore, T was set to 120 in our experiments. Thus, for the Intel Temperature/Light traces, X is a 54 × 120 matrix. For the House Temperature/Light traces, X is a 25 × 120 matrix.

B. Recovery Performance

Using the Intel Temperature/Light and House Temperature/Light traces, we studied the recovery performance of CDG, EDCA, and STCDG thoroughly. For CDG, the random projection used in our experiments is the same as the one adopted by Luo $et\ al.$ [17]; the transform used in our research is wavelet transform.

The detailed experimental results are included in Fig. 4 to 7. Overall, the recovery capability ranking is STCDG, EDCA, and CDG. Namely, to achieve the same NMAE, CDG needs a lower dumping ratio (i.e. higher sampling ratio) than EDCA and STCDG requires the highest dumping ratio (i.e. the lowest sampling ratio). Specifically, for the Intel Temperature trace, CDG, EDCA, and STCDG can achieve very low NMAE (less than 0.02) until the their dumping ratios reach 60%, 85%, and 91% respectively. After these critical ratios, the performance of all the schemes under investigation deteriorates quickly. This indicates that when they are used for data gathering in WSNs, the dumping ratios should not exceed these critical values. The details of the experimental results in this case are summarized in Fig. 4. We believe that the reason why STCDG and EDCA outperform CDG is that CDG requires that the sensor readings should be sparse enough while the real-world networks cannot always meet this requirement. In a small-to-medium scale sensor network where sensor nodes are not densely deployed, it is likely that the resulting sparsity is not low enough. When this takes place, CDG will requires a low dumping ratio in order to recover the missing data successfully. Different than CDG, EDCA only requires the low-rank feature while STCDG needs both the low-rank and short-term stability features. These requirements are more easily to meet in real-world sensor networks.

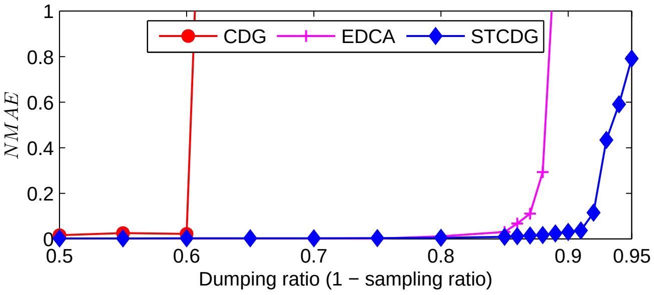


Fig. 4. Recovery errors: Intel temperature

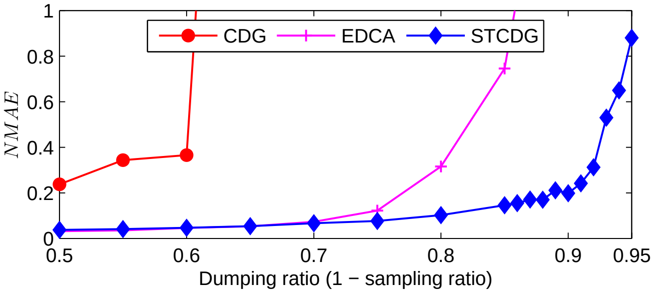


Fig. 5. Recovery errors: Intel light

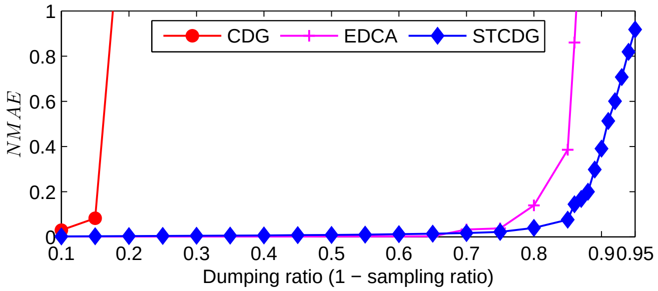


Fig. 6. Recovery errors: house temperature

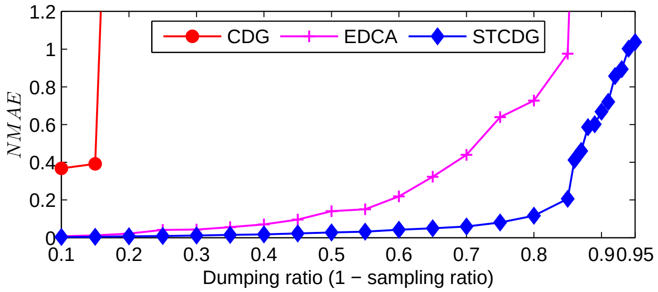


Fig. 7. Recovery errors: house light

For the Intel Light trace, the recovery capability ranking of CDG, EDCA, and STCDG stays unchanged. However, all the schemes under investigation perform worse than in the case of Intel Temperature. We believe that the reason lies in the fact that the spacial and temporal correlation of the light readings tends to be looser than that of the temperature readings. For example, the temperature readings from different rooms tend to go up/down simultaneously. However, the light readings can be affected by individual sources (e.g. the light in a room is turned on/off at night while the lights in other rooms stay off all the time). The details of the experimental results in this scenario are included in Fig. 5.

In the case of House Temperature/Light, the performance tendency of CDG, EDCA, and STCDG is similar to that in the scenario of Intel Temperature/Light. The only difference is that all the schemes under investigation require lower dumping ratios (higher sampling ratios) to achieve the same level of NMAE. The details of the experimental results in these cases are presented in Fig. 6 to 7. We believe that the reason behind the difference is that the smaller number of sensor nodes in the residential building have a negative impact on the sparsity, low-rank and short-term stability feature of the sensor readings.

C. Power consumption and lifespan

The power consumption model adopted in our study is similar to that used by Mainwaring $et\ al.$ [20]. Specifically, we assume that the transmission of 32 bits consumes 1 unit of power and the reception of 32 bits uses 0.4 unit of power. Although multiple types of sensing information could be forwarded to the sink using one packet, without loss of generality, we assume that only the temperature data is required and thus each packet only includes the temperature information. We further assume that each packet consists of 64 bits. Among the 64 bits, 32 bits are used to store the temperature information, 16 bits are assigned for node ID, and the remaining 16 bits are reserved for time stamp. Consequently, the transmission and reception of a packet consume 2 units and 0.8 units of power respectively.

In our research, we used the Intel Temperature trace to evaluate the power consumption and lifespan performance of STCDG. We first calculated the total units of power required by Centralized Exact, CDG, and STCDG in the case of Intel Temperature. The details of the total power consumption results are included in Fig. 8. The total power consumption of Centralized Exact stays unchanged in all experimental scenarios because all sensor readings are forwarded to the sink. For CDG and STCDG, as the dumping ratio increases, the total power consumption goes down. This is because the higher the dumping ratio, the smaller the number of packets to be forwarded to the sink. Note that Centralized Exact outperforms CDG in terms of total power consumption when the dumping ratio is less than 70%. The reason behind this phenomenon is that, with Centralized Exact, the sensor nodes far from the the sink only need to transmit few packets during each snapshot (in the extreme case, each leaf node in the routing tree only needs to transmit one packet containing its own reading); with CDG, every node needs to transmit M packets. When the dumping ratio is low enough (i.e. the sampling ratio is high enough), CDG requires more packet transmissions than Centralized Exact. Different than CDG, STCDG outperforms CDG and Centralized Exact in all experimental scenarios.

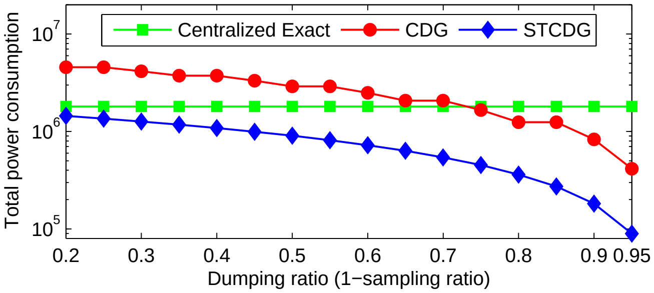


Fig. 8. Total power consumption

In WSNs, the lifetime of the first sensor node that runs out of power determines the lifespan of the network. In this paper, we use “relative lifespan” to quantify the performance of STCDG. To calculate the relative lifespan of a data gathering scheme, we first find out the node that consumes the highest number of power units in the case of Centralized Exact. This node is usually a neighbor of the sink. The number of power units consumed by this node is denoted as M_0. Then for a given dumping ratio, we find out the node that consumes the highest number of power units in the case of another data gathering scheme (i.e. CDG or STCDG). We use M_{max} to denote the number of power units consumed by this node. Finally, the relative lifespan is defined as the ratio of $(1/M_{max})$ to $(1/M_0)$.

The lifespan results of Centralized Exact, CDG, and STCDG are summarized in Fig. 9. We noticed that, as the dumping ratio increases, the relative lifespan of both CDG and STCDG go up in all experimental scenarios. However, CDG increases at a much faster rate than STCDG. These results indicate that STCDG outperforms both Centralized Exact and CDG significantly in terms of power usage. Note that the performance of EDCA is not included in Fig. 9. This is because, in this paper, the power consumption and lifespan of a data gathering scheme is only affected by the number of packets to be transmitted. For a given dumping ratio, EDCA and STCDG lead to the same number of packets to be transmitted, resulting in the same power consumption and lifespan performance. For clarity, only the results of STCDG are included.

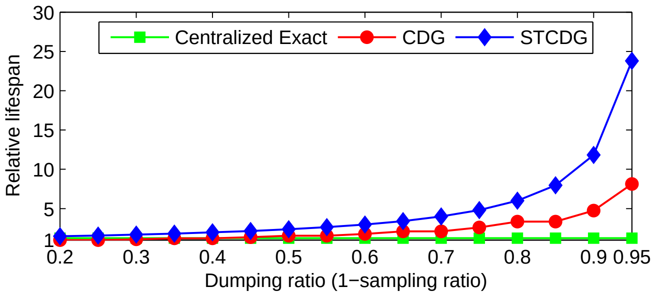


Fig. 9. Relative lifespan

D. *Network capacity*

In Section IV-B, we analyzed the network capacity of STCDG theoretically. The analysis was based on scheduled medium access control (MAC). Practically, the cost of scheduled MAC is relatively high in terms of computation and communication overhead. In our research, we also studied the network capacity of STCDG in a more practical scenario through ns-2 simulations. Specifically, IEEE 802.11 was used as the MAC protocol. The data rate of the transmission link was 2Mbps and the payload size of each packet was 20 bytes.

The topology used in our simulations was similar to that adopted by Luo $et\ al.$ [17]. In detail, we considered a sensor grid with 1089 nodes organized into 33 rows and 33 columns. The node at the center of the grid plays the role of the sink. The distance between adjacent nodes in the same row or column is 14 meters. The transmission and interference range of the sensor nodes are 15 meters and 25 meters respectively.

With the grid topology, a routing tree needs to be established before packets can be forwarded to the sink. Fig. 10 includes a typical routing tree for the grid topology. The routing tree has 4 subtrees, each of which contains a similar number of nodes. The performance of the data gathering schemes under investigation varies sightly for different routing trees. In our experiments, we generated 10 different routing trees randomly, repeated the simulations with these trees, and calculated the average performance results.

In the simulations, the sampling ratio is fixed at 20% for both CDG and STCDG. This leads to roughly 220 measurements for CDG and 220 randomly selected readings for STCDG. Furthermore, we assume that the data from each subtree can be recovered using 55 random measurements in the case of CDG. For Centralized Exact, all the readings from the sensor nodes are forwarded to the sink.

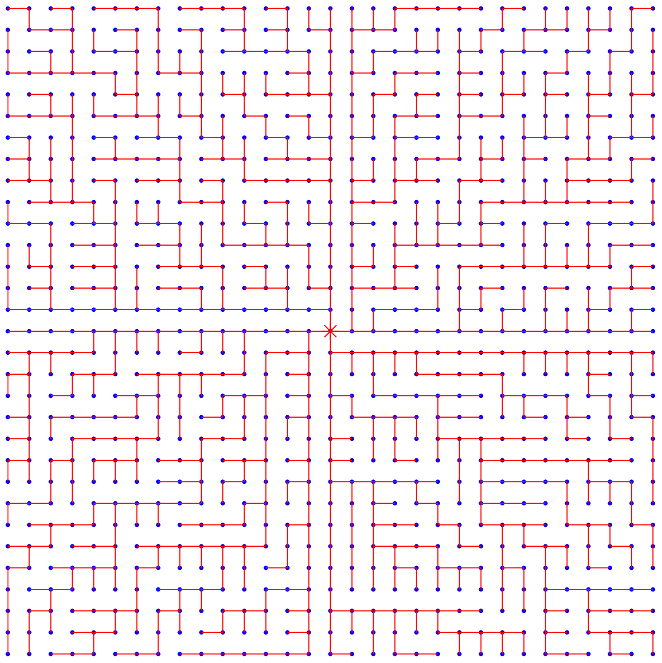


Fig. 10. The routing tree for the grid topology

In our research, we define the input interval as the gap between the timepoint when sensor nodes start to collect their readings for a snapshot to the timepoint when they begin to gather the readings for the next snapshot. The output interval is defined as the period from the moment when the last packet of a snapshot is received by the sink to the moment when the sink receives the last packet of the next snapshot. In our experiments, we tune the input interval and study how output interval behaves accordingly. At the beginning of each input interval, a packet is generated by each sensor node that should send a reading to the sink. In the case of STCDG, the nodes that are not randomly selected do not need to generate any packets. Generally speaking, the longer the input interval, the longer the output interval. As the input interval decreases, the output interval goes down. However, if the input interval is not achievable, the output interval will start to go up as the input interval decreases due to the phenomenon of congestion collapse. The output interval corresponding to this turning point (i.e. the minimum output interval) can be used to infer the network capacity.

Our output interval results are included in Fig. 11. For Centralized Exact, the minimum output interval, 9.54 seconds per snapshot, is achieved when the input interval is 6.40 seconds per snapshot. In the case of CDG, the minimum output interval, 4.72 seconds per snapshot, appears when the input interval is equal to 4.50 seconds per snapshot. This indicates that CDG can roughly achieve a capacity gain of 2 in the experimental scenario. For STCDG, the minimum output interval, 2.02 seconds per snapshot, can be achieved when input interval is set to 1.60 seconds per snapshot. This means that STCDG leads to a capacity gain of 4.7 in the experimental scenario. The reason why STCDG achieves a higher capacity gain than CDG is that, with STCDG, the nodes far from the sink only need to forward few packets per snapshot (in the extreme case, each leaf node in the routing tree only needs to send one packet per snapshot); with CDG, all nodes need to send M packets to the sink. Namely, more traffic has to be forwarded to the sink in the case of CDG, resulting in lower network capacity. Note that the network capacity results of EDCA are not included in Fig. 11 because they are the same as the results of STCDG due to the reason mentioned previously.

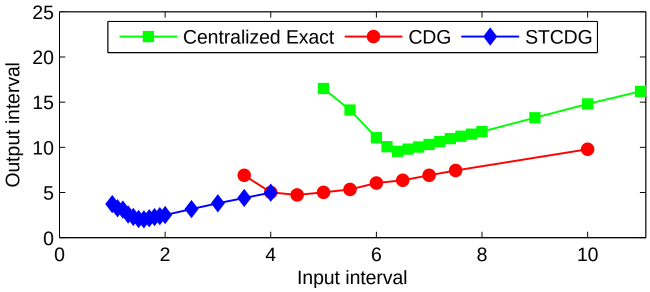


Fig. 11. Output interval

E. Performance of TDMA-based scheduling

We proposed a TDMA-based scheduling algorithm in Section IV-B. The algorithm can actually be used by all the data gathering schemes under investigation. In this section, we evaluate the performance of these data collection schemes when the proposed TDMA-based scheduling algorithm is used to guide packet transmission. The metric adopted for comparison purposes is the number of time slots required to gather one snapshot.

The topology used is similar to the one used in Section V-D. Specifically, we used a grid topology with a large amount of sensors whose transmission and interference range are 15 meters and 25 meters respectively. The distance between adjacent nodes in the same row or column is 14 meters. The number of the sensor nodes varies from 31×31=961 to 41×41=1681. These nodes are deployed at the grid intersections and the scale of the grids is in the range of 31 *rows* $\times$ 31 *columns* to 41 *rows* $\times$ 41 *columns*.

In our experiments, the sampling ratio is fixed at 20% for both CDG and STCDG. The node at the center of the grid plays the role of the sink. For each scale of the grids, 10 random routing trees are generated. For each routing tree, the TDMA-based scheduling algorithm is used to find out the number of time slots required to collect a snapshot. Then the average of the number of time slots is used for evaluation purposes.

Our experimental results are summarized in Fig. 12. We found that, in all experimental scenarios, both CDG and STCDG require substantially less time slots than Centralized Exact although CDG needs slightly more slots than STCDG. In addition, as the number of nodes in the grid goes up, the number of required time slots in the case of Centralized Exact increases at a much faster rate than that in the scenarios of CDG and STCDG. The reasons behind these phenomena are twofold. First of all, both CDG and STCDG require that only part of the sensor nodes send their packets to the sink (due to the 20% sampling ratio) while, with Centralized Exact, all nodes should forward their packets. Secondly, at the same sampling ratio, CDG leads to more transmissions than STCDG.

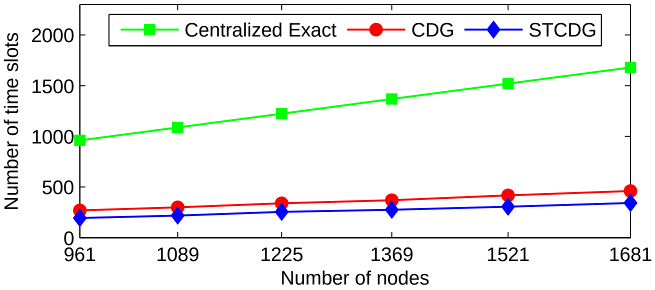


Fig. 12. Number of time slots required for one snapshot

VI. Conclusions

In this paper, we propose an efficient data gathering mechanism for WSNs, STCDG. It takes advantage of the low-rank and short-term stability features in WSNs to reduce the amount of traffic in WSNs and improve the recovery accuracy, ultimately leading to prolonged lifespan and lowered power consumption. In addition, STCDG avoids the optimization problem involving empty columns by first removing the empty columns and recovering the non-empty columns, then filling the empty columns using an optimization technique based on temporal stability. Our experimental results show that STCDG outperforms Centralized Exact, CDG, and EDCA in terms of recovery error, power consumption, lifespan, and network capacity. Our future work will involve an in-depth analysis of the spatial and temporal correlation in WSNs in order to further reduce the number of samples required for effective data recovery. The routing and topology information can also be utilized to reduce data traffic in WSNs.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61073147, Grant 61173120, Grand 61271226, and Grand 61272410; by the National Natural Science Foundation of China and Microsoft Research Asia under Grant 60933012; by the Fundamental Research Funds for the Central Universities under Grant 2011QN014 and Grant 2012QN078; by the National Natural Science Foundation of Hubei Province under Grant 2011CDB044; by the Fok Ying Tung Education Foundation under Grant 132036; and by the Program for New Century Excellent Talents in University under Grant NCET-10-408 (State Education Ministry). The corresponding author is Hongbo Jiang.

References

[1] ns-2, http://www.isi.edu/nsnam/ns/.

[2] Sensor data from intel berkeley research lab, http://db.lcs.mit.edu/labdata/labdata.html.

[3] E. Arikan. Some complexity results about packet radio networks (Corresp.). *IEEE Transactions on Information Theory*, 30(4):681–685, Jul 1984.

[4] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk. An information-theoretic approach to distributed compressed sensing. In *43rd Allerton Conference on Communication, Control, and Computing*, pages 13–18, Sep 2005.

[5] E. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, Jun. 2009.

[6] J. Cheng, H. Jiang, X. Ma, L. Liu, L. Qian, C. Tian, and W. Liu. Efficient Data Collection with Sampling in WSNs: Making Use of Matrix Completion Techniques. In *Proc. of IEEE GLOBECOM*, pages 1 –5, Dec 2010.

[7] J. Chou, D. Petrovic, and K. Ramachandran. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. *Proc. of IEEE INFOCOM*, 2(1):1054–1062, Mar 2009.

[8] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer. Network correlated data gathering with explicit communication: Np-completeness and algorithms. *IEEE/ACM Transactions on Networking*, 14(1):41 – 54, Feb. 2006.

[9] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr 2006.

[10] M. Duarte, S. Sarvotham, M. Wakin, D. Baron, and R. Baraniuk. Joint sparsity models for distributed compressed sensing. In *Online Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, Nov 2005.

[11] A. Ephremides and T. Truong. Scheduling broadcasts in multihop radio networks. *IEEE Transactions on Communications*, 38(4):456–460, Apr 1990.

[12] S. Gandham, Y. Zhang, and Q. Huang. Distributed minimal time convergecast scheduling in wireless sensor networks. In *Proc. of ICDCS*, Jun. 2006.

[13] H. Gupta, V. Navda, S. Das, and V. Chowdhary. Efficient gathering of correlated data in sensor networks. *ACM Transactions on Senor Networks*, 3(3):31–43, Mar. 2007.

[14] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak. Compressed sensing for networked data. *IEEE Signal Processing Magazine*, 25(2):92–101, Mar 2008.

[15] R. Keshavan, A. Montanari, and S. Oh. Low-rank matrix completion with noisy observations: a quantitative comparison. *Proc. of the 47th annual Allerton conference on Communication, control, and computing*, pages 1216–1222, Sep 2009.

[16] S. Lee, S. Patterm, M. Sathiamoorthy, B. Krishanamachari, and A. Ortega. Compressed sensing and routing in multi-hop networks. *USC CENG Technical Report*, 2009.

[17] C. Luo, F. Wu, J. Sun, and C. W. Chen. Compressive data gathering for large-scale wireless sensor networks. In *Proc. of IEEE/ACM MOBICOM*, pages 145–156, Sep 2009.

[18] C. Luo, F. Wu, J. Sun, and C. W. Chen. Efficient measurement generation and pervasive sparsity for compressive data gathering. *IEEE Transactions on Wireless Communications*, 9(12):3728 –3738, Dec 2010.

[19] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1):122–173, Mar 2005.

[20] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, Sep 2002.

[21] D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff. On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data. In *Proc. of IEEE IPSN*, pages 1–16, Apr 2003.

[22] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *Society for Industrial and Applied Mathematics Review*, 52(3):471–501, Aug 2010.

[23] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4):471–480, Jul 1973.

[24] J. Wang, S. Tang, B. Yin, and X.-Y. Li. Data gathering in wireless sensor networks through intelligent compressive sensing. In *Proc. of IEEE INFOCOM*, pages 603 –611, Mar 2012.

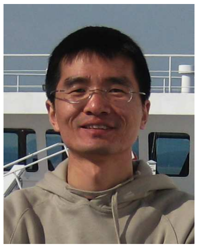
[25] X. Wu and M. Liu. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In *Proc. of ACM IPSN*, pages 1–12, Apr 2012.

[26] S. Yoon and C. Shahabi. The clustered aggregation (cag) technique leveraging spatial and temporal correlations in wireless sensor networks. *ACM Transactions on Senor Networks*, 3(1), Mar 2007.

[27] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. *ACM SIGCOMM Computer Communication Review*, 39(4):267–278, Oct 2009.

[28] H. Zheng, S. Xiao, X. Wang, and X. Tian. Energy and latency analysis for in-network computation with compressive sensing in wireless sensor networks. In *Proc. of IEEE INFOCOM*, pages 2811–2815, Mar 2012.

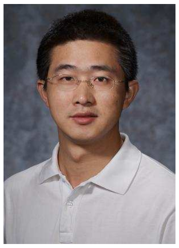
Jie Cheng received the B.S. and M.S. degrees from National University of Defense Technology, and Ph.D. from Huazhong University of Science and Technology in 2011. For now His research concerns wireless networking, especially algorithms and architectures for sensor networks.



Qiang Ye received his PhD degree in computer science from University of Alberta. He is an associate professor in the Department of Computer Science and Information Technology at the University of Prince Edward Island. His research interests lie in the area of Communication Networks in general. Specifically, he is interested in Wireless Sensor/Ad Hoc Networks, Network Reliability and Security, and Performance Evaluation.

Hongbo Jiang received the B.S. and M.S. degrees from Huazhong University of Science and Technology, China. He received his Ph.D. from Case Western Reserve University in 2008. After that he joined the faculty of Huazhong University of Science and Technology as an associate professor. His research concerns computer networking, especially algorithms and architectures for high-performance networks and wireless networks. He is a member of IEEE.

Dan Wang received the B. Sc degree from Peking University, Beijing, China, in 2000, the M. Sc degree from Case Western Reserve University, Cleveland, Ohio, USA, in 2004, and the Ph. D. degree from Simon Fraser University, Burnaby, B.C., Canada, in 2007; all in computer science. He is currently an assistant professor at the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless sensor networks, Internet routing, and peer-to-peer networks. He is a member of the IEEE.



Chonggang Wang received his PhD degree in computer science from Beijing University of Posts and Telecommunications. He has conducted research with NEC Laboratories America, AT&T Labs Research, and University of Arkansas, and Hong Kong University of Science and Technology. His research interests include future Internet, machine-to-machine (M2M) communications, and cognitive and wireless networks. He has published more than 80 journal/conference articles and book chapters. He is a senior member of the IEEE.