Efficient Data Gathering using Compressed Sparse Functions

Liwen Xu†, Xiao Qi†, Yuexuan Wang†, Thomas Moscibroda‡
†Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
‡Microsoft Research Asia, Beijing, China
{kyoi.cn, qixiao0113}@gmail.com, wangyuexuan@tsinghua.edu.cn, moscitho@microsoft.com

***Abstract*—Data gathering is one of the core algorithmic and theoretic problems in wireless sensor networks. In this paper, we propose a novel approach – Compressed Sparse Functions – to efficiently gather data through the use of highly sophisticated Compressive Sensing techniques.**

The idea of CSF is to gather a compressed version of a *satisfying function* (containing all the data) under a suitable function base, and to finally recover the original data. We show through theoretical analysis that our scheme significantly outperforms state-of-the-art methods in terms of efficiency, while matching them in terms of accuracy. For example, in a binary tree-structured network of n nodes, our solution reduces the number of packets from the best-known $O(kn \log n)$ to $O(k \log^2 n)$, where k is a parameter depending on the correlation of the underlying sensor data. Finally, we provide simulations showing that our solution can save up to 80% of communication overhead in a 100-node network. Extensive simulations further show that our solution is robust, high-capacity and low-delay.

I. Introduction & Related Works

The fast, energy-efficient, and accurate gathering of sensor data in a network is arguably the most important information processing problem in sensor networks. It is not only relevant in a variety of practical application scenarios (e.g. habitat or information monitoring), but it is also of fundamental theoretical interest. This is because the problem prototypically captures the trade-off between the need for efficiency on the one hand, and accuracy and robustness on the other hand. Not surprisingly, data gathering problems have–in various disguises and flavors–been at the center of attention for researchers in the community for years [16], [11], [10], [15].

In its most basic form of data gathering problem, n sensor nodes in the network can communicate with each other wirelessly. Each node has one sensor readings d_i, and these values have to be communicated to a *sink node*. Given that sensor nodes are naturally limited in computation power, memory, and above all energy, the data gathering process should be efficient and incur low overhead. The principal way of achieving such energy savings is through in-network *compression* which is fundamentally based on the assumption of spatial correlation between readings. On the other hand, compression can incur lack of precision. The question of how much overhead can be reduced through compression without incurring harmful losses of accuracy thus lies at the heart of the data gathering problem.

Here we briefly review previous works that is most related to ours. Early algorithmic work on data gathering was focusing mostly on using the data correlations during the *encoding phase*, e.g. [7], [8]. The other approach of dealing with compression is based on Distributed Source Coding (DSC) techniques [14]. The idea is for nodes to carry out compression encoding in a distributed fashion. Based on Slepian-Wolf coding theory, [13] showed that distributed encoding can achieve the same efficiency as joint encoding can. Afterwards, various compression encodings (e.g. Discrete Fourier Transform, Discrete Cosine Transform or Wavelet Transform [6]) are utilized to generate new data gathering schemes. [16] proposed a clustered aggregation technique that first groups sensor nodes according to their measurements and transmits similar measurements per group only once during the actual data gathering process. A most recent Major Coefficients Recovery (MCR) [15] scheme is based on the deep exploration of the energy compaction property of Discrete Cosine Transform. It only recovers the very beginning coefficients and hence reduces the number of packets to $O(kn)$. However, MCR has too strict requirements for sensing data, so it may not be an applicable method for all gathering networks.

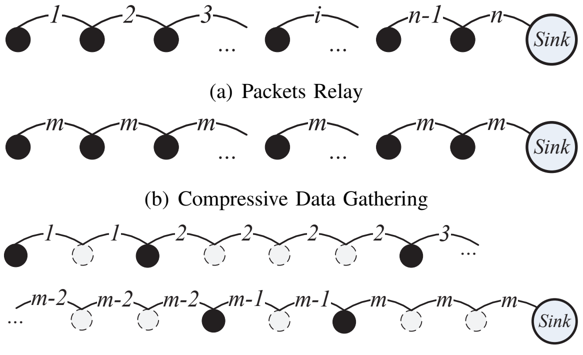
	Order of # of packets		
	Chain	Grid	Binary Tree
Packets Relay	n^2	$n\sqrt{n}$	$n \log n$
CDG [11]	$kn \log n$	$kn \log n$	$kn \log n$
CSF	$kn \log n$	$k\sqrt{n} \log n$	$k \log^2 n$

TABLE I
# OF MESSAGES TRANSMITTED

In recent years, the understanding of the possibilities of efficient data gathering has been revolutionized by the adoption of *Compressive Sensing* techniques. A key breakthrough in this regard was the development of *Compressive Data Gathering* (CDG) [11], which was the first work to show that data can be encoded in a distributed fashion in the network by projecting it using random coefficients, and then decoding it at the source using compressive sensing techniques. In contrast to existing work, instead of applying compression on the data itself, we compress data in the form of *sparse functions*, which we send to the source. The source can recover the function using techniques from polynomial approximation/interpolation theory, and use it to *compute* data values that were not reported. This approach allows us to very significantly sparsify the network, which can substantially reduce message overhead as well as increase network lifetime.

Table I gives us a direct feel for the order of improvement of our novel *Compressed Sparse Functions* data gathering scheme over others. Figure 1(c) is a simplest example demonstrating

978-1-4673-5946-7/13/$31.00 ©2013 IEEE



(a) Packets Relay

(b) Compressive Data Gathering

(c) Compressed Sparse Functions

Fig. 1. 1D data gathering networks

CSF in a canonical chain network of n nodes where one node can only communicate with the nearest neighbors. Most surprisingly, we reduce the message complexity from the previously known, e.g in binary tree topology, $O(kn \log n)$ to a *sub-linear* function, $O(k \log^2 n)$.

II. Compressed Sparse Functions

A. Model & Assumptions

Assumptions: Similarly to other papers on data gathering, we assume that sensor data is spatially correlated. To simplify the discussion, we assume that each message transmission can contain only one or a fixed number of sensor readings. We also assume that initially, each node has the same energy level.

Notations: For the 1D network scenario, we consider n nodes with a unique identifier $1 \leq i \leq n$, increasing from left to right. Each node holds a reading denoted by d_i, which is to be gathered. Thus each node is labeled as a pair of numbers (i, d_i), and the n pairs form a mapping from $[n] = \{1, 2, \cdots, n\}$ to $D = \{d_1, d_2, \cdots, d_n\}$, $f : [n] \to D$. We further assume that $d_i \in \mathbb{R}$, and denote by f a function that maps integers to real numbers.

B. 1D Networks

The key idea underlying our Compressed Spare Functions approach is, *if* we can find an suitable function that contains all the n sensor's readings (all sensor readings can be computed by the function provided identifiers), and *if* we can express this function in a very sparse way, then it should in principle be possible to 1) only communicate the sparse functions to the sink node, 2) recover the function at the sink, and then 3) use the recovered function to generate *all* the n readings.

Moreover, determining the function may be more efficient than using other compression schemes. It is for this reason, the sending of a highly compressed version of sparse functions, that we call our scheme *Compressed Sparse Functions-based data gathering*. While in principle, the above approach seems intriguing, many questions remain. First, do such sparse functions exist, and if so, how can we quickly and efficiently compute, encode, and decode them? In this section, we answer these questions affirmatively in three steps, before putting all results together in Section II-B4.

1) Function Basis and its Sparse Representation: A function $f : \mathbb{Z} \to \mathbb{R}$ such that $f(i) = d_i, 1 \leq i \leq n$, is called a *satisfying function* for $d_i, 1 \leq i \leq n$. It is trivial to know for any set of d_i satisfying functions exist. We use $F : \mathbb{Z} \to \mathbb{R}$ to denote the set of satisfying functions.

According to the definition, if we can find one satisfying function $f \in F$ for a set of data to be gathered, we can extract *all* the data from it. Hence, we can consider the data gathering problem as the problem of *identifying a suitable satisfying function*.

Definition 2.1: We define a *function base* as a set of functions $P = \{p_1(x), p_2(x), \cdots, p_m(x)\}$, and let $m = |P|$ denote the number of functions in this base. We say $\mathbf{a} = (a_1, a_2, \cdots, a_m) \in \mathbb{R}^m$ is a representation of f under the function base P if $f = \sum_{i=1}^{m} a_i p_i(x)$. Furthermore, if $\mathbf{a}$ contains only $k \ll n$ non-zero elements, we say this is a *k-sparse representation*.

Assuming f is a satisfying function, and a function base $P = \{p_1, p_2, \cdots, p_n\}$ is give. Under P, f has a representation $\mathbf{a} = (a_1, a_2, \cdots, a_n)$. Without loss of generality, $n = |P| = |\mathbf{a}|$ is also the number of sensor nodes.

The critical equations $f(i) = d_i$ can be expressed as

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} p_1(1) & p_2(1) & \cdots & p_n(1) \\ p_1(2) & p_2(2) & \cdots & p_n(2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(n) & p_2(n) & \cdots & p_n(n) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \tag{1}$$

where $\mathbf{a}$ is to be determined. It typically requires n equations to solve $\mathbf{a}$, however if $\mathbf{a}$ is *sparse* (recall Definition 2.1, by this it means f has a sparse representation under P), we can solve this system via *Compressive Sensing* (CS).

2) Compressive Sensing: Knowing $\mathbf{A}_{m \times n}$, and $\mathbf{y}_{m \times 1}$, the linear system of equations $\mathbf{A}\mathbf{x}_{n \times 1} = \mathbf{y}$ can hardly be solved ($\mathbf{x}$ is unknown) because $m < n$ makes it underdetermined. However if $\mathbf{x}$ is k-sparse ($\mathbf{x}$ contains only k non-zero elements), CS shows this system can be solved by optimizing

$$\underset{\hat{\mathbf{x}} \in \mathbb{R}^n}{argmin} \|\hat{\mathbf{x}}\|_1, \text{ such that } \mathbf{A}\hat{\mathbf{x}} = \mathbf{y}, \tag{2}$$

if $\mathbf{A}$ satisfies Restricted Isometry Property (RIP) and

$$m \geq ck \log n, \tag{3}$$

where c is a positive constant [4], [5]. Practically, it is sufficient that $1 \leq c \leq 4$.

Recalling equation (1), if f has a k-sparse representation under P, i.e. $\mathbf{a}$ is k-sparse, we can determine $\mathbf{a}$ using only as few as $m \geq ck \log n$ equations. We *randomly* pick up m equations from (1) indexed by $(i_1, i_2, \cdots, i_m)$, and rewrite the system in the form

$$\begin{pmatrix} d_{i_1} \\ d_{i_2} \\ \vdots \\ d_{i_m} \end{pmatrix} = \begin{pmatrix} p_1(i_1) & p_2(i_1) & \cdots & p_n(i_1) \\ p_1(i_2) & p_2(i_2) & \cdots & p_n(i_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(i_m) & p_2(i_m) & \cdots & p_n(i_m) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}. \tag{4}$$

The left side contains m readings, and at the right side we have to know P thus to know the matrix. So firstly we gather readings from m nodes which are $(d_{i_1}, d_{i_2}, \cdots, d_{i_m})$. Secondly we have to know a suitable base P that ensures $\mathbf{a}$ is sparse. Then by applying a CS recovery algorithm, $\mathbf{a}$ can be calculated, thus the satisfying function f is identified and all the other readings can be computed.

3) Cosine Basis and Discrete Cosine Transform: As mentioned above, another essential point is that whether we *can* actually find a suitable function base that ensures a sparse representation of f. For this purpose, we employ the Discrete Cosine Transform (DCT) as a tool to construct a function base. Particularly, we use Type-IV DCT because of its symmetry and orthogonality properties.

The Type-IV DCT matrix of length n is defined as

$$\mathbf{\Psi} = (\psi_{ij}), 1 \leq i, j \leq n, \tag{5}$$

$$\psi_{ij} = \sqrt{\frac{2}{n}} \cos\left[\frac{\pi}{n}\left(i-1+\frac{1}{2}\right)\left(j-1+\frac{1}{2}\right)\right]. \tag{6}$$

$$\mathbf{\Psi} = \mathbf{\Psi}^T = \mathbf{\Psi}^{-1} \tag{7}$$

The most important properties of DCT are *decorrelation* and *energy compaction* which means that the coefficients in the DCT domain of correlated data are usually sparse [9].

Theorem 1: Given the function basis $P = \{p_i(x)\}, 1 \leq i \leq n$, where

$$p_i(x) = \sqrt{\frac{2}{n}} \cos\left[\frac{\pi}{n}\left(i-1+\frac{1}{2}\right)\left(x-1+\frac{1}{2}\right)\right], \tag{8}$$

There exists a satisfying function $f \in F$, such that f has a representation $\mathbf{a}$ of small sparsity k.

Proof: Define $\mathbf{a} = (a_1, a_2, \cdots, a_n)^T = \mathbf{\Psi}(d_1, d_2, \cdots, d_n)^T$, where $\mathbf{\Psi}$ is defined as in (5).

Therefore define

$$\begin{aligned} f(x) &= a_1p_1(x) + a_2p_2(x) + \cdots + a_np_n(x) \\ &= \mathbf{a}^T(p_1(x), p_2(x), \cdots, p_n(x))^T. \end{aligned}$$

Then we calculate

$$\begin{aligned} f(i) &= \mathbf{a}^T(p_1(i), p_2(i), \cdots, p_n(i))^T \\ &= (d_1, d_2, \cdots, d_n)\mathbf{\Psi}^T(\psi_{1i}, \psi_{2i}, \cdots, \psi_{ni})^T \\ &= (d_1, d_2, \cdots, d_n)(0, 0, \cdots, 1, \cdots, 0)^T = d_i \end{aligned}$$

Thus we have $f(i) = d_i, 1 \leq i \leq n$, i.e., f is a satisfying function.

On the other hand, $\mathbf{a}$ is also the transformed vector of $(d_1, d_2, \cdots, d_n)$ in DCT domain. Due to the correlation of data, $\mathbf{a}$ should be k-sparse, where k is a small number. ■

4) Putting Things Together: Now we summarize a novel compressed sparse functions based data gathering scheme in the following steps.

- **Step 1.** Randomly activate m sensor nodes out of a total n nodes where $m \geq ck \log n$.
- **Step 2.** Collect the m readings from the active nodes.
- **Step 3.** Compute the corresponding DCT matrix, and run a recovery algorithm by providing the collected readings. The algorithm can determine the function base, and therefore the function f. Using f, the sink can simply compute the remaining $n - m$ readings of those nodes are not activated.

C. 2D Networks

So far, we have described the scheme in 1D networks, but generalization to 2D is straightforward. Without loss of generality, we consider two types of 2D topologies, *grid* topology in a rectangle sensing field and tree-structured topology in randomly deployed networks.

For grids, we first map the sensor nodes on 2D surface into a 1D sequence, and use a unique integer to identify each node as we do in 1D cases. Then a 2D scenario is converted into a 1D scenario and we can apply CSF by following the steps listed in Section II-B. At last, by performing an inverse mapping from the 1D sequence we recover the 2D topology. The only thing we need to take care is that the mapping of the 2D topology should try its best to maintain the correlation between adjacent nodes. For this reason, we employ *Hilbert Curves* to map grid topologies[12]. By doing this, we can also avoid periodical values in 1D sequences.

In random tree-structured topology with the sink being the root, we also apply Hilbert Curve to convert a grid which covers the whole sensing field as well as in the resolution high enough to distinguish every node in the grid. Figure 2 illustrates this re-ordering process. By such a pretreatment of ordering, CSF *uncovers* the correlation between data of random nodes and therefore maintains highly optimized energy efficiency. Along the re-arranged 1D vector, the same CSF steps can be carried out.

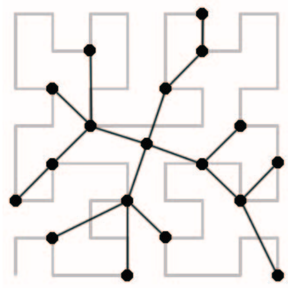


Fig. 2. An example of random 2D network

D. Theoretical Analysis of CSF

Comparing with other schemes, CSF collects data from much fewer nodes. This improvement dramatically reduces the transmission overhead in the network. In the following discussion we use *the number of packets transmitted* ($\#_{method}$ for short) to briefly compare CSF with the baseline Packets Relay (PR) and Compressive Data Gathering (CDG) in energy efficiency in different topologies.

Assuming the distance between nodes are the same in chain topology, we define the *transmission range* r of a node to an integer representing the number of nodes one can communicate with in one direction.

When $r_{max} = n$, every node can directly deliver its data to the sink node. So $\#_{PR} = \#_{CDG} = n$, and $\#_{CSF} = m$. When $r_{min} = 1$, every node can only communicate with its adjacent neighbors. This situation is well studied in previous works, and the conclusion is $\#_{PR} = O(n^2)$, $\#_{CDG} = O(kn \log n)$. In CSF, m nodes are randomly activated, thus $\#_{CSF} = O(kn \log n)$ in the worst cases.

In general cases when $1 < r < n$, the most efficient strategy for each scheme to do the gathering is to separate all the nodes into several groups according to r. After carrying out the scheme over each group, $\#_{PR} = O(\frac{n^2}{r})$, $\#_{CDG} = O(kn \log n)$, and $\#_{CSF} = O(\frac{mn}{r})$.

In grid topology, we simply consider the cases when $r = 1$. Through the similar discussion as in chain topology, we can see in a $\sqrt{n} \times \sqrt{n}$ grid, $\#_{PR} = O(n\sqrt{n})$, $\#_{CDG} = O(kn \log n)$ and $\#_{CSF} = O(k\sqrt{n} \log n)$. The cases of tree-structured topology is almost the same where $\#_{PR} = O(n \log n)$, $\#_{CDG} = O(kn \log n)$ and $\#_{CSF} = O(k \log n^2)$.

In 2D topologies, CSF performs much more efficiently than the other schemes meanwhile the collection accuracy can also be maintained at a good level. More detailed analysis will appear in the following extended paper.

III. Evaluation & Simulation

In this section we evaluate the performance of CSF through simulations in comparison with PR and CDG. All the evaluations are tested for networks of three different types of topologies, which are chain, grid and random 2D topology.

- **Chain** Each node can only communicate with the adjacent neighbors.
- **Grid** We assume that one node can only communicate with the four adjacent neighbors and the sink is deployed at one corner. A spanning tree of the grid forms the routing pathes.
- **Random 2D** Within a certain sensing range, several nodes are deployed at uniformly random. The sink is located at the center, and shortest-hop routing pathes are generated towards the sink. Therefore the topology in this scenario usually looks like a tree that appears *more spread* than the spanning trees in grid topologies.

In the ideal experiments we first explore the recovery properties. Through these experiments we setup reasonable values for simulation parameters. In the following simulations, we mainly evaluate the network performance and show the better network capacity, energy efficiency and network delay of CSF than the other schemes.

A. Recovery Property & Parameter Setup

In the discussion about recovery accuracy, *Signal to Noise Ration* (SNR) is employed to represent the recovery level between an original signal $\mathbf{s}$ and its approximation $\hat{\mathbf{s}}$.

Assuming data are gathered ideally through networks, the experiment is carried out based on two real data sets, CTD (Conductivity, Temperature and Depth) data from NOAA [2] for chain topologies and temperature data from NASA [3] for random 2D as well as grid topologies.

By setting $\frac{m}{k}$-ratio at different level, we explore how much extra accuracy CSF can achieve when m is greater than the necessary $k \log n$. In Figure 3, the recovery accuracy curves indicate 1) CSF achieves better recovers accuracy than CDG provided the same $\frac{m}{k}$ ratio, and 2) $\frac{m}{k}$ ratio's lower bound $k \log n$ and its corresponding recoveries are highlighted with circles in the figures, which clearly shows a threshold. Therefore it is reasonable to set the positive constant $c = 1$ (in Equation (3)) in the following and simulations.

B. Simulation

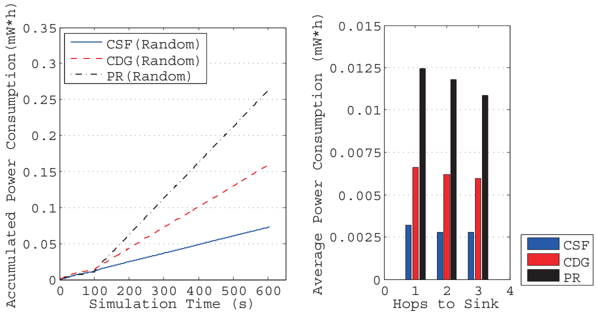
We implement CSF, PR and CDG on Contiki OS, and four hardware simulations are carried out on Cooja platform [1] to focus on power consumption, network capacity, packet loss rate and network delay respectively. All these simulations are run for three types of network topology, chain, grid and random 2D as mentioned above. The specific settings are shown in Table II.

Radio component	cc2420
Transmission range	50 meters
Interference range	100 meters
MAC protocol	CSMA
MAC buffer size	6
MAC transmission retry	10
Payload size	23 Bytes
k/n percentage	4%
m/k ratio	$\log n$

TABLE II
Simulation Parameters

1) Power Consumption: The power consumption simulations are carried out on 25-node grid and random networks separately. Figure 4(a) describes the accumulation of network's total power consumption during the first several hundred seconds in the random topology scenario. During the first 100 seconds of routing generation, three schemes consume nearly the same level of power. Afterwards, when networks begin to gather and send data, power consumption accumulates almost linearly with time elapsing. CSF consumes even less than half of CDG's power consumption, nearly 25% of PR. The similar result can be found in simulations of grid topology, where the accumulated power consumption of CSF, CDG and PR is 0.089, 0.129 and 0.280 (mW·h) respectively after 600 seconds from the beginning.

Figure 4(b) shows the distribution of power consumption of nodes at different hop distance from the sink. CSF and CDG both perform better balance than PR.



(a) Accumulated power consump. (b) Power consump. distribution

Fig. 4. Power consumed in random topology

2) Gathering Capacity: In this simulation we evaluate the throughput of network a scheme could achieve in term of gathered *information*. CSF and CDG both require only m packets in order to collect n pieces of data as fully discussed before. We consider that sink (the network system) actually gathers n pieces of data after collecting the required m pieces

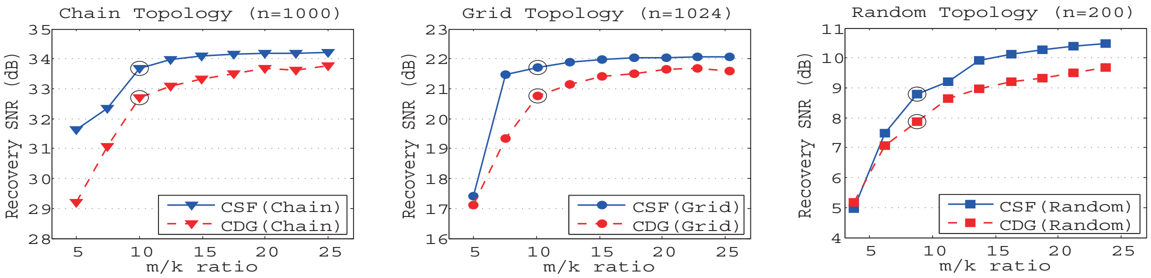


Fig. 3. Recovery accuracy according to different $\frac{m}{k}$ ratio

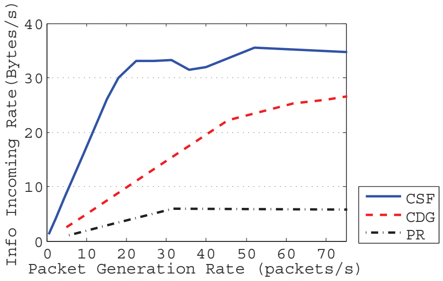


Fig. 5. Gathering capacity in random topology

Network Scale (# of nodes)	50	75	100
CSF	1.25 s	2.79 s	4.19 s
CDG	12.47 s	45.86 s	60.14 s

TABLE III
GATHERING DELAY

because from which n items can be extracted. So we define *information incoming rate at sink* as

Definition 3.1: The amount of information sink (the network system) could extract from the received packets per second is called the *Information Incoming Rate at Sink* (Bytes/s).

Assuming one piece of data contains one byte, we run the simulations in 100-node grid and random networks. Figure 5 shows that CSF has a much more rapid increase in the information incoming rate than the other two, which means under the same level of network load our scheme can gather data more frequently. For CSF, when packet generation rate is greater than 20packets/s, it reaches a capacity limit of about 30Bytes/s. Meanwhile this limit for CDG is only 25Bytes/s which comes at around 50packets/s. Clearly CSF has a higher capacity than CDG, meanwhile PR has a much lower capacity.

3) Gathering Delay: The third performance we evaluate is *gathering delay*, by which we mean the time duration from the moment the first packet is sent to the moment sink gathers all required packets. We run simulations in networks of different scale to find out the shortest gathering delay the scheme could possibly achieve without losing a single packet. Several rounds of gathering are executed in each setting and the average results of CSF and CDG are listed in Table III. The gap between CSF and CDG in this evaluation is substantial that CSF experiences only less than 10% of the delay that CDG suffers, i.e. our scheme gathers data much faster.

IV. Conclusion

We propose the novel Compressed Sparse Function based data gathering scheme for large-scale wireless sensor networks in this paper, which utilizes Compressive Sensing techniques to greatly reduce the transmission overhead. Leading to a breakthrough in energy efficiency that in tree structured networks a sub-linear order of transmission overhead could be obtained, CSF also provides better recovery accuracy when comparing with previous schemes. Extensive simulations on Contiki system support the theoretic analysis of energy saving and demonstrate the high-capacity and low-delay properties that CSF exhibits.

Acknowledgment

We appreciate Prof. Jian Li for the generous help. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61073174, 61033001, 61061130540, and the Hi-Tech research and Development Program of China Grant 2006AA10Z216.

References

[1] Contiki. http://www.contiki-os.org/.
[2] CTD Data, NOAA. http://tao.noaa.gov/tao/data_deliv/.
[3] GES DISC, NASA. http://mirador.gsfc.nasa.gov/.
[4] E. Candès, T. Tao. *Decoding by linear programming*. IEEE Trans. Inf. Theory, Vol. 51.
[5] E. Candès, J. Romberg, T. Tao. *Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information*. IEEE Trans. Inf. Theory, Vol. 52(2).
[6] A. Ciancio, S. Pattem, A. Ortega, B. Krishnamachari. *Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm*. In Proc. of IPSN, 2006.
[7] R. Cristescu, B. Beferull-Lozano, M. Vetterli. *On network correlated data gathering*. In Proc. of IEEE Infocom, Vol. 4, 2004.
[8] R. Cristescu, B. Beferull-Lozano, M. Vetterli, R. Wattenhofer. *Network correlated data gathering with explicit communication: Np-completeness and algorithms*. IEEE/ACM Trans. on Networking, 14(1).
[9] S. Khayam. *The discrete cosine transform (DCT): theory and application*. Michigan State Univ.: East Lansing (USA), 2003.
[10] . Li, Y. Wang, Y. Wang, *Complexity of Data Collection, Aggregation, and Selection for Wireless Sensor Networks*, IEEE Tran. on Computers, vol. 60, 2010.
[11] C Luo, F Wu, J Sun, *Compressive Data Gathering for Large-scale Wireless Sensor Networks*, in Proc. ACM Mobicom'09.
[12] H. Sagan. *Space-filling curves*. NewYork: Springer-Verlag, 1994.
[13] D. Slepian, J. Wolf. *Noiseless encoding of correlated information sources*. 19:471-480, 1973.
[14] Z. Xiong, A. Liveris, S. Cheng, *Distributed source coding for sensor networks*, IEEE Signal Processing Mag., vol. 21, 2004.
[15] L. Xu, Y. Wang, Yongcai Wang. *Major Coefficients Recovery: a Compressed Data Gathering Scheme for Wireless Sensor Network*. 2011 IEEE Global Communications Conference.
[16] S. Yoon, C. Shahabi. *The Clustered Aggregation (CAG) Technique Leveraging Spatial and Temporal Correlations in Wireless Sensor Networks*. ACM Trans. on Sensor Networks, 3, 2007.