LETTER

Low Cost Error Correction for Multi-Hop Data Aggregation Using Compressed Sensing*

Guangming CAO[†,††,†††a)], ***Student Member***, **Peter JUNG**[††††b)], **Sławomir STAŃCZAK**[††††c)], ***and*** **Fengqi YU**[†], ***Nonmembers***

SUMMARY Packet loss and energy dissipation are two major challenges of designing large–scale wireless sensor networks. Since sensing data is spatially correlated, compressed sensing (CS) is a promising reconstruction scheme to provide low–cost packet error correction and load balancing. In this letter, assuming a multi-hop network topology, we present a CS–oriented data aggregation scheme with a new measurement matrix which balances energy consumption of the nodes and allows for recovery of lost packets at fusion center without additional transmissions. Comparisons with existing methods show that the proposed scheme offers higher recovery precision and less energy consumption on TinyOS.

key words: *large–scale wireless sensor networks, compressed sensing, packet loss, energy balancing*

1. Introduction

Packet loss and energy dissipation are two major challenges when designing large–scale wireless sensor networks (WSN) operating under strict energy constraints. Such networks are often deployed to monitor the environment in some areas of interest, like forest fire monitoring.

Usually WSNs gather sensing data from a number of geographically distributed sensor nodes and transmit the data to a single fusion center (the sink) for further processing. The data is transmitted using intermediate nodes as relays in a "receive–and–forward" (or "decode–and–forward") manner. Therefore, the nodes near the sink have to carry a significantly higher load than peripheral nodes, leading to disparity in energy and power consumption. Immediate consequences of the disparity are connectivity and congestion problems due to battery depletion at sensor nodes closer to the sink. The problem is further aggravated by the interference caused by concurrent transmissions, leading to higher single–hop and end–to–end packet loss rates. Improving end-to-end packet loss rates while treating communication resources (energy and bandwidth) with care is one of the most fundamental challenges in designing WSNs, for example in [1].

This letter, while aiming at improving the end–to–end performance, shows how to beneficially utilize a sparse structure of environmental parameters in the design of network protocols and for the data recovery at the fusion center. In many WSN applications, such a property results from the spatially correlated sensing data which is essentially determined by a small number of unknown environmental parameters in highly dimensional spaces. For this reason, *compressed sensing* (CS), as an emerging area in the field of sparse recovery, provides promising reconstruction approaches to the problem of low–cost end–to–end packet error correction and hot–spot energy usage of bottleneck nodes. As the source compression at a sensor node is computationally less expensive than conventional encoding, CS is suitable for WSN hardware configurations. More importantly, due to the linearity of such compressed encoding schemes, sensor nodes can always transmit a globally fixed number of packets (fixed compressed dimension in [2]) in a hop–by–hop manner regardless of the number of nodes; this naturally avoids the unbalanced load distribution. Reference [3] first introduced CS erasure coding (CSEC) to address the problem of packet loss in WSNs through compressed sensing; the scheme offers precise data recovery under the condition of packet loss rates up to 20% while increasing the compressed dimension by the same percentage. The accurate recovery is therefore achieved by sending more data, which is not suitable to handle the worse case of serious loss rate sufficiently. Furthermore, multi-hop scenarios are not covered explicitly.

Although combining sensing networks and CS is one of the central research topics in compressed sensing, it remains a big challenge to implement the theoretical methods in WSNs to improve their packet loss tolerance, especially in the case of multi-hop tree network topologies. The main contribution of this letter is a CS–oriented data aggregation method for the multi–hop tree topology; the method is shown to precisely reconstruct lost packets at the fusion center without additional transmission costs. Moreover, the energy consumption is balanced which mitigates the hot–

Manuscript received July 12, 2013.
Manuscript revised September 5, 2013.

[†]The authors are with the Department of Integrated Electronics, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055 China.
[††]The author is also with the Chinese University of Hong Kong, Hong Kong, China.
[†††]The author is also with the Graduate University of Chinese Academy of Sciences, Beijing, 100049 China.
[††††]The authors are with the Heinrich-Hertz-Lehrstuhl für Informationstheorie und Theoretische Informationstechnik, Technische Universität Berlin, Einsteinufer 25, 10587 Berlin, Germany.

*This work is supported in part by Deutsche Forschungsgemeinschaft (DFG) grants JU 2795/2-1, STA 864/3-2, Shenzhen grants: JC201005270349A,JC201104220264A,CXB201104220033A, and Shenzhen Key Laboratory for RF Integrated Circuits.

a) E-mail: gm.cao@siat.ac.cn.
b) E-mail: peter.jung@mk.tu-berlin.de
c) E-mail: slawomir.stanczak@mk.tu-berlin.de

DOI: 10.1587/transinf.E97.D.331

spot energy usage problem. We consider peer–to–peer and end–to–end packet loss rates and compare our greedy–based and convex CS–oriented recovery methods with conventional data aggregation and CSEC reception of [3].

2. System Model

In this section, we introduce a data gathering scheme for a multi-hop tree topology where each node always transmits a fixed number of packets in every hop, instead of sending more and more packets from hop to hop. We establish a probabilistic model for the distribution of lost packets and consider the CS–based reconstruction of the sensor readings including the recovery of the lost data packets. Let us start with the conventional aggregation.

2.1 Data Aggregation

Assume that N sensor nodes are arranged by tree routing in B branches with each branch consisting of H hops, i.e. $N = BH$. For simplicity we assume also that for each branch $b \in [0, B-1]$ the node indices are arranged in the correct order $\mathcal{B}_b := [bH+1 \ldots (b+1)H]$. In the conventional data aggregation, data is streamed in a "receive–and–forward" manner. Thus, the gathered data $y \in \mathbb{R}^M$ at the sink is then simply the sensed data $d \in \mathbb{R}^N$ of the N nodes, $y = [d_1, \ldots d_H, d_{H+1}, \ldots d_{BH}]^T = d$. Each node k in a branch has here only one child node and therefore the received data vector R_k equals the generated data vector S_{k-1} of its child node $k-1$. If linear encoding of received data R_k and sensed data d_k is not exploited, i.e. a corresponding encoding matrix $\mathcal{E}_k$ equals the identity, we obtain:

$$S_k = \mathcal{E}_k \begin{bmatrix} d_k \\ R_k \end{bmatrix} = \begin{bmatrix} d_k \\ S_{k-1} \end{bmatrix} \in \mathbb{R}^{M_k}. \tag{1}$$

The output dimension M_k increases linearly hop by hop and finally the sink receives $M = N$ packets containing the primitive data $y = d$ from the N nodes. This scheme is easy to implement without special requirements and is therefore universally used. However, as already mentioned, such a strategy does not scale well with the size of the network. It is therefore necessary to develop multi-hop methods which operate at a fixed packet load independent of the particular location of the node in a branch.

Data Aggregation in Fixed Dimension: Here, we assume that a linear combination of the sensor readings, the vector $y = \Phi d \in \mathbb{R}^M$, is observed at the sink where Φ is the measurement matrix of dimension $M \times N$. However, the data processed at node k is always an M–dimensional vector:

$$S_k = [\phi_k \ I] \begin{bmatrix} d_k \\ R_k \end{bmatrix}, \tag{2}$$

where ϕ_k is the kth column of Φ and $I \in \mathbb{R}^{M \times M}$ is a corresponding identity. Hence, the generic encoding matrix $\mathcal{E}_k$ in (1) becomes $\Phi_k = [\phi_k \ I]$. Since summations are independent of ordering, this aggregation strategy covers linear topology and tree topology with dynamic branches.

2.2 Packet loss in Data Aggregation

Before we model the effect of packet loss, it should be noted that two related types of error rate are considered in WSNs due to the multi-hop type of communications: (i) peer-to-peer and (ii) end-to-end. Peer-to-peer packet loss rate is easier to measure in practice, while end-to-end rate is more important at the system level and will be considered in this letter. We assume that the probability of packet loss in every node is independent and identically distributed (i.i.d.). Let q_k be a M_k–dimensional random vector representing the error pattern at node k. If its mth component q_{mk} for $m \in [1 \ldots M_k]$ equals 1, it means successful reception of the mth packet of node k. Otherwise it means the packet is lost. We denote a success probability by p (peer-to-peer packet loss rate is $1-p$), so that

$$\Pr\{q_{mk} = 1\} = p. \tag{3}$$

Furthermore, we use Q_k to denote the $M_k \times k$–dimensional random error matrix at node k with elements $(Q_k)_{mi} := \prod_{l=i}^{k-1} q_{ml}$.

Packet loss in Fixed S_k Dimension: Using Q_k, the effective measurement matrix is changed to $Q_k \odot \Phi_k$, where $\odot$ denotes point-wise (Hadamard) product. This means, at the sink the lost packet is counted as zero at the corresponding position in measurement matrix, i.e. the over all matrix Φ is replaced by the effective measurement matrix $\hat{\Phi} = Q \odot \Phi$, where Q is the error matrix at the sink. This method does not increase system complexity and does not consume extra energy either, in contrast to the existing solutions, e.g. CSEC. Although this method seems quite simple, it is more intuitive to use sensed data from adjacent sensors or the last sensed data of the same sensor instead of zeros. Later on, in the simulation section, we simulate the strategy where last time measurements are used due to the temporal correlations of sensed data. In our system setup, we observe a much worse performance.

Now, by replacing Φ with $\hat{\Phi}$, $R_k = q_{k-1} \odot S_{k-1}$ holds and (2) becomes:

$$(S_k)_m = \phi_{mk} d_k + \sum_{i=1}^{k-1} \Big(\prod_{l=i}^{k-1} q_{ml}\Big) \phi_{mi} \, d_i = \phi_{mk} d_k + (R_k)_m. \tag{4}$$

For a whole network with B branches, the data aggregated at the sink is then:

$$y_m = \sum_{b=0}^{B-1} \sum_{i=bH+1}^{(b+1)H} \Big(\prod_{l=i}^{(b+1)H} q_{ml}\Big) \phi_{mi} \, d_i. \tag{5}$$

The elements of the effective measurement matrix $\hat{\Phi} = Q \odot \Phi$ are therefore $\hat{\phi}_{mi} := Q_{mi} \cdot \phi_{mi} = (\prod_{l=i}^{(b+1)H} q_{ml}) \phi_{ml}$ in branch b. If the tree network has only one branch, i.e. a linear topology, then $\hat{\phi}_{mi} = (\prod_{l=i}^{N} q_{ml}) \phi_{ml}$.

The sink receives M data $y = \hat{\Phi} d$ of $B \cdot H$ data symbols.

The number of zeros in $\hat{\Phi} = Q \odot \Phi$ determines the number of lost data contributions (corresponding elements in Φ are multiplied by zeros) in y and it is a random variable. We then define the *end–to–end data loss rate*:

$$\begin{aligned} 1-P &= \frac{\mathbb{E}(\#\text{ zeros in } Q)}{MN} = \frac{\mathbb{E}(\#\text{ zeros in a branch of } Q)}{MH} \\ &\overset{(a)}{=} \frac{1}{H}\sum_{h=1}^{H} h \Pr\{q_{mh}=0|(q_{m,h+1}\dots q_{m,H})=1\} \\ &= \frac{1}{H}(1-p)p^H \sum_{h=1}^{H} h\, p^{-h}, \qquad (6) \end{aligned}$$

where the expectation is taken the average over the statistics of Q. Step (a) follows from the properties of Q since its zeros occur as a series which always starts independently for each compressed dimension $m \in [1\dots M]$ and for each branch $[0\dots B-1]$ in its first hop. We call P in (6) the *end–to–end data success rate* since it indicates the averaged proportion of d contained in the observation y. P depends on the *peer–to–peer packet success rate* p. For example, taking $H=5$ and $p=77.7\%$, P is up to 50%.

2.3 Data Reconstruction with Compressed Sensing

Usually, the spatial behavior of environmental parameters, for example temperature, is determined only by a small number of unknown parameters. In the language of CS, this means that the parameters are sparse (or compressible) in a certain domain, for example in the wavelet domain as it has been observed in [2]. Thus, the sensed data $d=(d_i)_{i=1}^N$ can be expressed as $d=\Psi x \in \mathbb{R}^N$ where Ψ is a unitary matrix and x is a K–sparse vector, i.e. x contains most K non–zero elements: $\|x\|_{\ell_0} := |\{i : x_i \neq 0\}| \leq K \ll N$. Then from (5), the aggregated data at the sink is $y=(Q\odot\Phi)\Psi x =: Ax \in \mathbb{R}^M$. It is known that the stable reconstruction of K–sparse $x \in \mathbb{R}^N$ (and therefore the reconstruction of the sensor readings d) requires only $M=O(K\ln N) \ll O(N)$ randomized observations. An efficient solution to the following underdetermined and combinatorial problem:

$$\min_{z\in\mathbb{R}^N} \|z\|_{\ell_0} \quad \text{s.t.} \quad y = Az, \qquad (7)$$

has been extensively studied in the field of compressed sensing [4]. If any $2K$ columns of Φ are linear independent, x is uniquely determined by y, i.e. $M=2K$ measurements are sufficient. But, proving this condition is as hard as solving (7), which is known to be NP complete. However, in the case of i.i.d. Gaussian measurements Φ, the *orthogonal matching pursuit* (OMP) accurately recovers x with probability greater than $1-2\epsilon$ if $M \geq CK\ln(N/\epsilon)$ for some $\epsilon \in (0, 0.36)$ where $C \leq 20$ (C can be improved for large K) [5]. This greedy algorithm is quite simple and fast, but it is not always finds the correct solution.

Another main insight of sparse optimization and compressed sensing is that under certain conditions on A the solution to (8) can be obtained by solving the convex relaxation [6] where the ℓ_0–term is replaced by $\|x\|_{\ell_1} = \sum_i |x_i|$. It is also known as the *basis pursuit* (BP):

$$\min_{z\in\mathbb{R}^N} \|z\|_{\ell_1} \quad \text{s.t.} \quad y = Az. \qquad (8)$$

Conditions for correct reconstruction of x using the program (8) instead of (7), have been established for *fixed* matrices A, for example, in terms of the *restricted isometry property* (RIP) †. Matrices with independent sub–Gaussian rows or columns are with overwhelming probability sufficiently close to restricted isometries once $M=O(K\ln N) \ll O(N)$ [8]. Meanwhile, the conditions have been relaxed substantially once the observations are *randomized* and incoherent to the domain where x is sparse (the RIPless theory, see [9]).

A key assumption in all known CS guarantees for the programs (8) and (7) is that the random matrix A has to be known at the decoding stage. Thus, in our application the corresponding positions of packet errors have to be known at the sink. Fortunately, this can be achieved with little cost in WSNs. In most data collection scenarios, every packet has a certain sequence number based on its node ID such that the sink can identify it. Therefore, the sink knows the status of all packets. In CS, the following should be handled appropriately. Sensor readings are always encoded into M packets which need be numbered for identification. The positions of zeros are known at the sink. Therefore, sensor nodes should record lost packets once they are detected.

3. Simulation Results

In our simulations, we use MATLAB and TinyOS with TOSSIM which is a discrete event-based simulator. The sensor data d are temperature values from field environment, which are spatially and temporally correlated. Thus, we assume that the vector $d=\Psi x$ is K–sparse in a fixed basis Ψ. To illustrate the result we have repeated the experiment 1000 times for each fixed M and random measurements. We observed that for $N=100$ and $K=9$ in the tree topology with $B=10$ branches, the sink can recover the original data at $M=40$ with sufficient precision, i.e. the reconstruction error $\|\hat{d}-d\|_{\ell_2}/\|d\|_{\ell_2}$ is less than 1% where $\hat{d}$ is our recovery.

Recovery of lost data using CS: In Fig. 1 we show the performance of our method for end–to–end packet loss rate

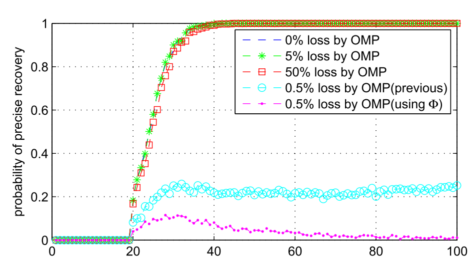

Fig. 1 Data recovery by OMP with $\hat{\Phi}$ for data loss rates $(1-P)$ of 0%, 5%, 50%, comparing with OMP using Φ instead of $\hat{\Phi}$ or OMP with previous readings ($N=100$, $K=9$).

† A satisfies the RIP of order K if there exists $\delta_K \in (0,1)$ such that $|\|Ax\|_{\ell_2}^2 - \|x\|_{\ell_2}^2| \leq \delta_K$ holds for all K-sparse vectors x [7].

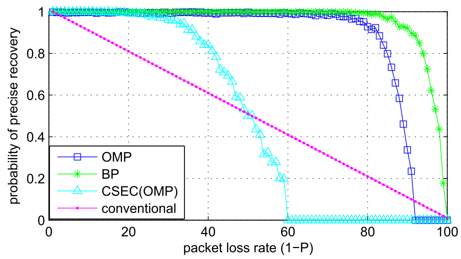

Fig. 2 Compare our proposed method with two different data recovery approaches. (1) CS with OMP and BP respectively ($M = 40$, $K = 9$), (2) CSEC [3] with OMP ($M = 48$, $K = 9$), (3) conventional aggregation.

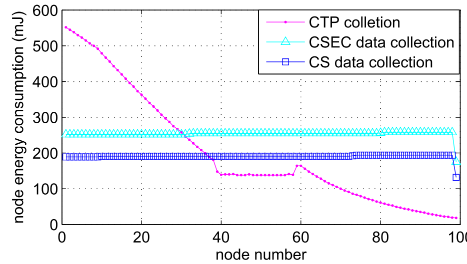

Fig. 3 Energy consumption of every sensor node in the network under 20% packet loss. CS data collection is more balanced than CTP, and CSEC consumes more energy than our CS.

Table 1 The energy model of sensor node.

Component	Status	Current
MCU (Atmega128)	Active	8.93 mA
	Idle	4.93 mA
	Power save	8 uA
	Power down	0.3 uA
	Transmit (power: 0 dBm)	10.4 mA
	Receive(sensitivity: -110dBm)	9.3 mA
RADIO(CC1000)	Power down	0.2 uA
LED	Lighting	3 mA
SENSOR (SHT10)	Active	4 mA

$1 - P = 5\%$ and 50%. It shows that our method recovers the sensor data in both cases very well. The performance is almost the same as that without loss. Figure 1 also shows other methods using CS, as mentioned before. It might be more intuitive, for example, keeping Φ instead of $\hat{\Phi}$ or using last time measurements instead once the current packet is lost (previous OMP). We can see in both cases, CS doesn't work even $1 - P = 0.5\%$.

Comparison with other methods: Fig. 2 compares our greedy–based (OMP) and convex CS–oriented (BP) recovery methods with CSEC (using OMP) and conventional data aggregation. We can see that for all methods the recovery accuracy gets worse with increasing packet loss rate, where BP has the best performance. Contrarily, the whole row of measurement matrix is dropped off by CSEC, when one packet loss happens during the sampling. CSEC precisely recovers the sensor data for packet loss rate up to 20% through increasing the compression dimension by 20%, i.e. $M = 48$ instead of $M = 40$. The conventional aggregation appears on a straight line in Fig. 2 since each packet loss causes a corresponding data vacancy at the sink.

Energy dissipation: Our method also keeps the benefit of balanced energy dissipation that CS promises because at every hop the fixed M packets are sent in the whole network. In order to visualize the energy efficiency gains of different methods, we decrease the energy consumption for channel detection to minimum. It is because this part of energy is decided by low-level MAC protocol and is the same for all data recovery algorithms with/without using CS. Here, we compare two CS data collections with traditional tree collection protocol (CTP) in TinyOS. All data are sampled every 10 seconds during 1000 seconds in a linear topology as $B = 1$ where the sink is node 0 as the starting point of this network. Using the energy model shown in Table 1, the statistic of the energy consumption of every node is analyzed. For $1 - P = 20\%$ in Fig. 3 we can see that CTP is unbalanced and the nodes near the sink exhaust earlier. Although in CS scheme the peripheral nodes send more packets than in the traditional one, i.e. M packets instead of 1, both two CS scenarios are more balanced and energy-efficient over the whole network, while CSEC consumes more energy due to its overhead as explained above.

4. Conclusions

We have described a low cost error correction method for multi–hop data aggregation using compressed sensing. We have shown that this method has low–cost implementation and excellent performance due to the sparse nature of sensor data. Our method is stable even at the packet loss rate of 50%. Furthermore, it has been verified, using TinyOS, that the energy dissipation is also efficient and balanced.

References

[1] R. Bista, Y.K. Kim, M.S. Song, and J.W. Chang, "Improving data confidentiality and integrity for data aggregation in wireless sensor networks," IEICE Trans. Inf. & Syst., vol.E95-D, no.1, pp.67–77, Jan. 2012.

[2] C. Luo, F. Wu, J. Sun, and C. Chen, "Compressive data gathering for large-scale wireless sensor networks," Proc. 15th International Conference on Mobile Computing and Networking (MobiCom), pp.145–156, 2009.

[3] Z. Charbiwala, S. Chakraborty, S. Zahedi, Y. Kim, M. Srivastava, T. He, and C. Bisdikian, "Compressive oversampling for robust data transmission in sensor networks," Proc. IEEE, INFOCOM, 2010, pp.1–9, March 2010.

[4] D. Donoho, "Compressed sensing," IEEE Trans., Inf. Theory, vol.52, no.4, pp.1289–1306, April 2006.

[5] J.A. Tropp, Anna, and C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Trans. Inf. Theory, vol.53, pp.4655–4666, 2007.

[6] E.J. Candès, J.K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," Communications on Pure and Applied Mathematics, vol.59, no.8, pp.1207–1223, 2006.

[7] E. Candes and T. Tao, "Decoding by linear programming," IEEE Trans., Inf. Theory, vol.51, no.12, pp.4203–4215, Dec. 2005.

[8] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," in Compressed Sensing, Theory and Applications, ch. 5, Cambridge University Press, 2012.

[9] R. Kueng and D. Gross, "Ripless compressed sensing from anisotropic measurements," arXiv preprint arXiv:1205.1423, 2012.