Robust Compressive Data Gathering in Wireless Sensor Networks with Linear Topology

Mohammadreza Mahmudimanesh, Neeraj Suri
Technical University of Darmstadt, Germany
Email: {reza,suri}@cs.tu-darmstadt.de

***Abstract*—Wireless Sensor Networks (WSNs) are deployed in a variety of topologies and configurations depending on specific applications and requirements. In this paper, we study a simple and yet very important class of the WSN topologies, the linear or chain topology in which the Sensor Nodes (SNs) are connected in a series and gather the sensed data at a single base station or sink at the end of the chain. WSNs with linear topology have many practical applications, e.g., in infrastructure monitoring and surveillance of civil constructions. There is a large body of research on efficient data gathering techniques to transmit the sensed data over WSNs' limited communication bandwidth. In particular for linear topology, data collection technique has to put a balanced load on all SNs to avoid breakage of the chain at the exhausted nodes. Compressed Sensing (CS) is an efficient data collection technique for WSNs that fulfills these requirements. In a failure-free scenario, CS avoids exhausted nodes by balancing the communication and processing load on the SNs. In this paper, we examine the performance of a special implementation of CS for WSNs, called Compressive Data Gathering (CDG) when a SN in a chain WSN encounters a failure and cannot forward messages to the next hop. We propose a method to enhance the robustness of CDG in such failure scenarios by transmitting the messages to the next healthy node and excluding the failed samples from CS signal recovery mechanism. Evaluations show that our method effectively withstands the failures without sacrificing the accuracy of the collected data.**

I. Introduction

A Wireless Sensor Network (WSNs) is an interconnected set of battery powered Sensor Nodes (SNs) that is used for large scale monitoring of a physical parameter of interest [1]. The primary objective of a WSN is to deliver the sensed data to a base station or *sink*. Sink is a dedicated node with sufficient power that post-processes the data and prepares them for the end user. The SNs often possess limited computation, power and bandwidth [2]. Therefore, it is crucial to efficiently transmit the sensed data over the resource limited SNs.

WSNs are often self-configured networks and their topology depends very much on the deployment and application requirements. In a two dimensional field such as a farm or a woodland, often a tree topology is preferred for data collection. In this paper we target an important application of the WSNs, i.e, monitoring and surveillance of civil structures [3]. In particular, we study WSN linear topology that is often employed for surveillance and monitoring of constructions such as roads, railways, bridges, etc. We say that a WSN possess a *linear* or *chain* topology, when the SNs are connected in a series and transmit the data hop by hop to deliver to the sink that is positioned at the end of this chain. Larger deployments may require several segments of chain WSNs with a sink at the end of each segment.

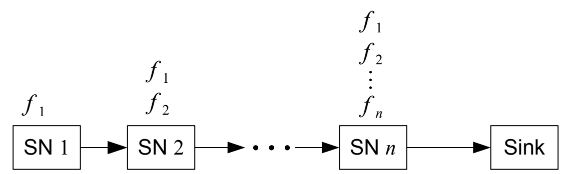

Fig. 1. Baseline data transmission in a WSN with linear topology

The problem of efficient data gathering is especially challenging in a WSN with linear topology. Figure 1 illustrate the baseline approach for data collection in a WSN with linear topology. The value sensed by SN i is denoted by f_i. Since the communication is done hop-by-hop, each node has to transmit its own data and also forward the data from the previous nodes. SN i has to transmit f_i and forward the values $\{f_1, f_2, \ldots, f_{i-1}\}$. Consequently, the nodes closer to the sink become highly overloaded. An effective solution to this problem is to reduce the amount of the transmissions by compressing the sensed data. Several studies show that, the data recorded by the SNs are highly compressible [4], [5]. Thus, the use of compression algorithms to reduce the amount of data sent to the sink is advocated.

An important requirement for the compression algorithm is to be light-weight as it runs on the resource-limited SN hardware platform [2]. In this paper we examine a simple and yet efficient network coding mechanism, called Compressed Sensing (CS) that effectively reduces in-network transmissions [6]. Remarkably, it puts a balanced communication and processing load on all SNs. This feature makes it particularly useful for WSNs with linear topology.

A. Compressed sensing

The CS theory forms the basis of several decentralized and distributed sensing methods that are especially suitable for WSN applications. CS is often implemented in two stages:

1) A set of linear combinations of the sensed data are calculated and transmitted to the sink. These linear combinations are called *measurements*
2) The originally sensed data are then reconstructed from the measurements using a *recovery* algorithm which is executed on the sink.

The sensed data are usually referred to as the *samples* and a set of samples is called a *signal*. Samples or sensor readings are different from the *measurements* that are sent to the sink.

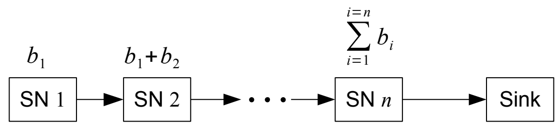

Fig. 2. CS-based data collection in a WSN with linear topology

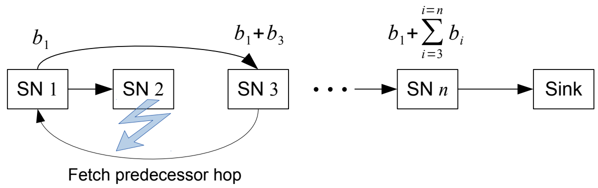

Fig. 3. Node failure and chain reconstruction

We formally define these terms in Section II-A and explain how the measurements are calculated from the sensed data.

Figure 2 shows the CS-based data collection in a WSN with linear topology. It is simplistic illustration of the Compressive Data Gathering (CDG) method that we explain in detail in Section II-B. Each node computes an encoding of its sensed value. The encoded value by SN i is denoted by b_i. Since these values are *accumulated* (arithmetically added) at each hop, the amount of transmissions at all SNs remain equal. Thus, the communication and computation load is balanced across the network. This effectively avoids occurrence of exhausted nodes.

B. Problem statement

WSNs are usually deployed in uncontrolled operational environments, and hence, it can happen that some SNs get damaged and cannot continue their function. This paper studies the problem of CS-based data collection in a chain WSN in presence of node failures. When a node encounters a failure, it stops accumulating and forwarding the measurements to the next hop. This failure causes a breakage of the chain at the position of the failing SN.

Figure 3 illustrates a failure scenario in which SN 2 stops sending to the next hop. We assume that SN 3 detects this failure since it does not receive any messages from SN 2. Consequently, SN 3 fetches the measurement from the first healthy predecessor node, i.e., SN 1. Looking at the accumulated measurement that is received by the sink, we observe that only maintaining the chain connectivity is not sufficient to cancel the effect of node failure. The value of b_2 is missing in the accumulated measurement that is received by the sink. Consequently, the sink also has to exclude the missing samples when it wants to recover the original data. Therefore, a list of failed SNs must be communicated to the sink which in turn requires more communication overhead. Moreover, maintaining the consistency of such a list is a cumbersome task especially in a large-scale WSN.

The objectives and related contributions of this paper are:

- Maintaining the connectivity of a chain WSN that performs CS-based data collection by making auxiliary links and isolating the failing nodes.
- Detecting the location of the failures without modifying the measurement mechanism at the SN level and without sending health-monitoring messages to the sink.
- Minimizing the effect of faulty or missing sensor readings on accuracy of the CS recovery algorithm.

This paper presents a simple and effective enhancement to CS-based data gathering for WSNs with chain topology that is resilient to node and link failures as well as communication noise. After detailing our system model in Section III we present our solution in Section IV. Our evaluations in Section V show the superior performance of our method in handling node and link failures compared with the state of the art CS-based data collection methods for WSNs which we review shortly in Section II.

II. Background and Related Work

In this section, we first shortly review the fundamentals of the CS theory. Then, we review an adaptation of the CS theory to data collection in WSNs with chain topology.

A. Compressed Sensing

For a WSN of size n, we give each SN an integer id in the range $[1, n]$ and denote the value sensed by SN i by f_i. The full set of the sensed data can be represented by vector $\mathbf{f} = [f_1\ f_2\ \dots\ f_n]^T$. Vector $\mathbf{f}$ is called a *spatial signal* and each f_i is called a *sample* of this signal. In CS, the signal is not sampled and compressed directly. Instead, a set of linear measurements of the samples are acquired. The original signal is then reconstructed from these linear measurements.

We call a vector $\boldsymbol{\phi} \in \mathbb{R}^n$, a *sensing vector*, and the inner product of a sensing vector with $\mathbf{f}$ is called a *measurement*. We assume that the measurements can be contaminated with some additive white Gaussian noise:

$$y_i = \boldsymbol{\phi}_i^T \mathbf{f} + z_i \quad , \quad i \in \{1, 2, \dots, m\} \tag{1}$$

where y_i are the measurements, $\boldsymbol{\phi}_i$ are the sensing vectors and z_i is the additive noise. This can be also written using matrix notation:

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{f} + \mathbf{z} \tag{2}$$

where $\mathbf{y} = [y_1\ y_2\ \dots\ y_m]^T$ is called the *measurement vector*, $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \dots\ \boldsymbol{\phi}_m]^T$ is the *measurement matrix* and $\mathbf{z} = [z_1\ z_2\ \dots\ z_m]^T$ is the noise vector.

CS allows accurate reconstruction of $\mathbf{f}$ from a few measurements, when $\mathbf{f}$ is sufficiently compressible. We say that $\mathbf{f}$ is *compressible* under the Ψ-transform, when $\mathbf{f} = \boldsymbol{\Psi}\mathbf{x}$ for an orthonormal $n \times n$ matrix $\boldsymbol{\Psi}$ and $\mathbf{x}$ is sparse [7]. Vector $\mathbf{x}$ is called *sparse* when it has $s \ll n$ non-zero components and all its remaining $(n - s)$ components are zero. The signals recorded by WSNs are reported to be highly compressible under a suitably chosen compressive basis such as Discrete Cosine Transform (DCT) or Haar wavelet [8]–[10].

Candes et al. prove necessary conditions for the measurement matrix $\boldsymbol{\Phi}$ and the compressive matrix $\boldsymbol{\Psi}$ such that it is possible to recover $\mathbf{f}$ from $O(s \log n)$ measurements [7], [11], [12]. They show that when the elements of the measurement matrix $\boldsymbol{\Phi}$ are drawn from a normal random distribution and $\boldsymbol{\Psi}$

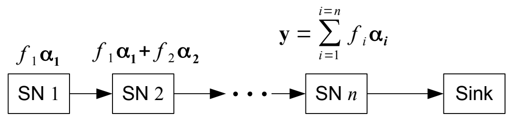

Fig. 4. CDG in a WSN with linear topology

is a dense orthonormal matrix, $\mathbf{f}$ can be accurately recovered from $O(s \log n)$ measurements [11]. Randomized sampling brings a key benefit for WSNs by eliminating the need for centralized coordination [6], [10].

In order to recover $\mathbf{f}$ from $\mathbf{y}$, first we need to solve the following convex optimization problem [11].

$$\underset{\tilde{\mathbf{x}}\in\mathbb{R}^n}{\text{minimize}} \; \|\tilde{\mathbf{x}}\|_1 \quad , \quad \text{subject to} \quad \|\mathbf{y}-\boldsymbol{\Phi}\boldsymbol{\Psi}\tilde{\mathbf{x}}\|_2^2 \;\leq\; \epsilon \tag{3}$$

where ϵ is the upper bound of error magnitude, $\|\cdot\|_1$ is the norm-1 operator and $\|\cdot\|_2$ is the norm-2 operator[1] [7]. The parameter ϵ is often set to the magnitude of the additive noise. This problem is solved at the sink using an efficient convex optimization solver [13]–[15]. If $\hat{\mathbf{x}}$ is the solution to the convex optimization problem in Equation 3, then $\hat{\mathbf{f}} = \boldsymbol{\Psi}\hat{\mathbf{x}}$ will estimate the original signal $\mathbf{f}$ with an error bounded by the measurement noise [7].

Having studied the foundations of the CS theory, we see how this theory is applied in practice to WSNs.

B. Compressive Data Gathering

Compressive Data Gathering (CDG) is an adaptation of CS for WSNs that suits the linear topology very well [10]. Basic operation of CDG in a chain WSN is depicted in Figure 4. Given the measurement matrix $\boldsymbol{\Phi}$ as

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \cdots & \phi_{1,n} \\ \phi_{2,1} & \phi_{2,2} & \cdots & \phi_{2,n} \\ \vdots & & \ddots & \vdots \\ \phi_{m,1} & \phi_{m,2} & \cdots & \phi_{m,n} \end{bmatrix} \tag{4}$$

we define the column vector $\boldsymbol{\alpha}_i$ as $\boldsymbol{\alpha}_i = [\phi_{1,i} \; \phi_{2,i} \; \ldots \; \phi_{m,i}]^T$.

Each SN is given a unique id and runs a pseudo-random number generator algorithm seeded by its id to produce $\boldsymbol{\alpha}_i$. All of the SNs run the same pseudo-random number generator algorithm, though with different seeds. SN i senses the value f_i and multiplies this real number by the column vector $\boldsymbol{\alpha}_i$. If applicable, it accumulates the measurements received from the previous hop and forwards the result to the next hop. The same process is repeated by every SN till the measurement vector $\mathbf{y}$ is delivered to the sink, see Figure 4.

The measurement matrix $\boldsymbol{\Phi}$ can be easily reproduced at the sink by executing the same pseudo-random number generator seeded by the SN id's. Therefore, $\boldsymbol{\Phi}$ does not need to be communicated between the SNs and the sink. Having $\boldsymbol{\Phi}$ and $\mathbf{y}$, the sink can recover $\mathbf{f}$ from $\mathbf{y}$ after solving Equation 3. This process forms the main building block of many distributed data gathering techniques based on CS [8], [10], [16].

[1]For a real vector $v \in \mathbb{R}^n$, norm-1 of v is defined as $\|v\|_1 = \sum_{i=1}^{n} |v_i|$ and norm-2 of v is defined as $\|v\|_2 = \sqrt{\sum_{i=1}^{n} |v_i|^2}$

C. Improvement over the state of the art

In failure-free scenarios, the measurement vector $\mathbf{y}$ is received by the sink after running the network coding of CDG. When a node fails, what is received by the sink is different from what is expected to be received. As we show in Section IV, this causes anomalies in the signal that degrades the accuracy of the recovered signal.

The advantage of our work over CDG is that it handles node failures. By node failure, we mean the situation where a node gets damaged and cannot transmit or forward the measurements to the next hop. We propose a method to maintain the connectivity of the chain and rebuild the connection at the position of the failing node. Then we present our solution that applies a post-processing phase to detect the location of the failures and exclude the missing samples from the recovery process. If our connectivity restoring technique does not succeed due to a heavy damage to a burst of SNs, our detection method still precisely determines the location of chain breakage and also excludes the missing segment of the signal from signal recovery.

III. System Model

We consider a WSN with linear topology consisting of static SNs and a single static sink at the end of the chain. The goal is to collect all of the sensed data at the sink.

A. Communication cost

The communication cost depends on both communication range and the number of messages. Sending more data consumes more battery power. Also achieving a more distant receiver requires to transmit with higher radio power. While the energy consumption grows linearly with the size of the transmitted data, it grows quadratically with the communication range. Total communication cost to transmit m messages to a receiver in the distance of d is $O(md^2)$.

For simplicity we assume that the nodes are placed on equal distances of each other. For a chain WSN consisting of n nodes, the distance between each two consecutive nodes is the same and equal to d. Let P_0 be the radio power required for communicating a unit of data between nodes i and $i+1$. Thus the radio power required for communication between nodes i and $i+2$ is $4P_0$, and in general, the radio power for communication between nodes i and $i+k$ is k^2P_0.

B. Sensor validation criteria

It is required that the range of valid sensor readings are known. A SN calculates and transmits the measurements only when the sensor reading is within a finite range. For example, when a temperature sensor which is designed to measure values between -50 degrees to +500 degrees Celsius reports a value of -1000 or +2000, then the SN regards this value as invalid and does not compute the measurement, and consequently, no message will be sent by this SN.

Note that the term *measurement* is formally defined in Equation 1. It differs from sample or the value recorded by the sensor.

If the range of floating point numbers that a SN can store in its memory is $[B_L, B_U]$ and the range of valid sensor values is $[s_l, s_u]$, we require that

$$|s_l - s_u| \ll |B_L - B_U|. \quad (5)$$

The range of valid sensor readings must be bounded and the length of this range must be a small fraction of the range of numbers that can be stored in the memory of the SN.

IV. Detecting and isolating failures

In this Section we describe our connectivity maintenance and failure detection technique in three steps using an illustrative example. First, we describe our technique for restoring the connectivity of the WSN when one or more nodes fail in the network. Second, we propose a method that detects the exact location of failing nodes without transmitting any health monitoring messages. Finally, we show that our failure detection technique withstands the extreme failures in which the connectivity of the chain is not recoverable.

A. Restoring the connectivity of the chain

In a normal operation of the WSN, all nodes are accumulating and transmitting their measurements hop by hop to deliver the measurement vector $\mathbf{y}$ to the sink. As described in Section III-A, the nodes are regularly located in a series with distance d from each other. All nodes are transmitting with the radio power P_0 to communicate with their direct neighboring nodes.

All nodes are placed in a series arranged from node 1 to n as depicted in Figure 4. When node $i \in \{1, \ldots, n\}$ fails to transmit to node $i{+}1$, node $i{+}1$ detects this failure since it does not receive any messages from its previous hop. Consequently, it tries to contact the node $i-1$ and fetch its measurements. According to our system model, this requires $4P_0$ radio power since the nodes $i-1$ and $i+1$ are placed in a distance of $2d$.

Definition 1. Step-back count *is defined as the number k when node $i+1$ successfully fetches the measurements from node $i-k$ in case that the nodes $i-k+1, i-k+2, \ldots, i$ fail to deliver their encoded values to node $i+1$.*

A step-back of size k requires the nodes $i-k$ and $i+1$ to transmit with radio power $(k+1)^2 P_0$. Note that node $i+1$ sequentially tries to fetch data from nodes $i-1, \ldots, i-k+1$ until it reaches the first healthy predecessor $i{-}k$. Respectively, these trials has a cost of $4P_0, 9P_0, \ldots, k^2 P_0$ before reaching the healthy node $i-k$.

The maximum allowed step-back count is obviously not unlimited. Depending on the capabilities of SN's radio module, there is a limit for the maximum communication range.

Definition 2. Step-back limit k_{max} *is the maximum step-back count k that is allowed by the communication capabilities of the radio module of a sensor node.*

Depending on the success of the connectivity maintenance phase, one of the following cases may occur:

- **Successful network restoration**: The information flow continues by stepping back by $k \leq k_{max}$ hops. In this case, the connectivity of network is restored, though the samples $i{-}k{+}1, i{-}k{+}2, \ldots, i$ are missing due to node failures.
- **Unrecoverable chain breakage**: Restoration mechanism cannot rebuild the chain because even the node $i{-}k_{max}$ does not respond to the measurement fetching request that is sent by node $i+1$. All of the samples $1, 2, \ldots, i$ will be missed because of chain breakage.

In Section IV-B we show that when the recovery algorithm at the sink does not detect and isolate the missing samples from the recovery process, the overall accuracy of the reconstructed signal decreases significantly. In Section IV-C we propose a simple method that detects the exact location of the failing nodes. Next in Section IV-D we present a modified version of Equation 3 that excludes the missing samples caused by failing nodes. Finally, Section IV-E studies the case where the chain breakage is not restorable. We show that our method can also handle this case and isolate the missing part of the signal from the recovery, thus preservers the accuracy of the genuine part of the signal.

B. Degrading effect of missing samples on recovery phase

Here, we consider an illustrative synthesized spatial temperature signal that is compressible under DCT. Our discussion can easily extend to sensing any other physical parameter that is compressible in some compressive basis. Consider a WSN consisting of temperature sensors with linear topology consisting of $n = 256$ SNs. The values of the samples sensed by the SNs are represented as a spatial signal vector $\mathbf{f}$ of size 256.

In failure-free operation of the WSN, all of the SNs are transmitting their $f_i \boldsymbol{\alpha}_i$, and thus, the measurement vector $\mathbf{y}$ is correctly received by the sink. Suppose that when none of the samples of $\mathbf{f}$ are missing, the vector $\mathbf{f}$ is compressible under DCT. More precisely, $\mathbf{f} = \boldsymbol{\Psi}_D \mathbf{x}$ where $\boldsymbol{\Psi}_D$ is the $n{\times}n$ inverse DCT matrix and $\mathbf{x}$ is sparse. We set the sparsity of $\mathbf{x}$ to 10 in the synthesized signal of our example. To make it more realistic, we add a white Gaussian noise to the measurement vector $\mathbf{y}$. We have set the power of the noise to be 5% of the signal power.

According to our setup, to estimate the original signal $\mathbf{f}$ from the measurement vector $\mathbf{y}$, we have to solve the following convex optimization problem.

$$\underset{\tilde{\mathbf{x}} \in \mathbb{R}^n}{\text{minimize}} \ \|\tilde{\mathbf{x}}\|_1 \quad , \quad \text{subject to} \quad \|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_D \tilde{\mathbf{x}}\|_2^2 \ \leq \ \epsilon \quad (6)$$

If $\hat{\mathbf{x}}$ is the solution to Equation 6, then the original signal is estimated by $\hat{\mathbf{f}} = \boldsymbol{\Psi}_D \hat{\mathbf{x}}$.

In the failure-free case, the recovered signal $\hat{\mathbf{f}}$ shows a good accuracy compared to the original signal $\mathbf{f}$, see Figure 5.a. When the connectivity of the chain is maintained, node failures are equivalent to missing samples in the signal vector $\mathbf{f}$. When some of the samples are missing due to node failures, the corresponding elements of vector $\mathbf{f}$ suddenly drop to zero. When this happens, there is no guarantee that $\mathbf{f}$ is compressible under DCT anymore, and hence, we cannot accurately recover the original vector $\mathbf{f}$ by solving Equation 6. Looking at Figure 5.b we observe that the accuracy of signal recovery

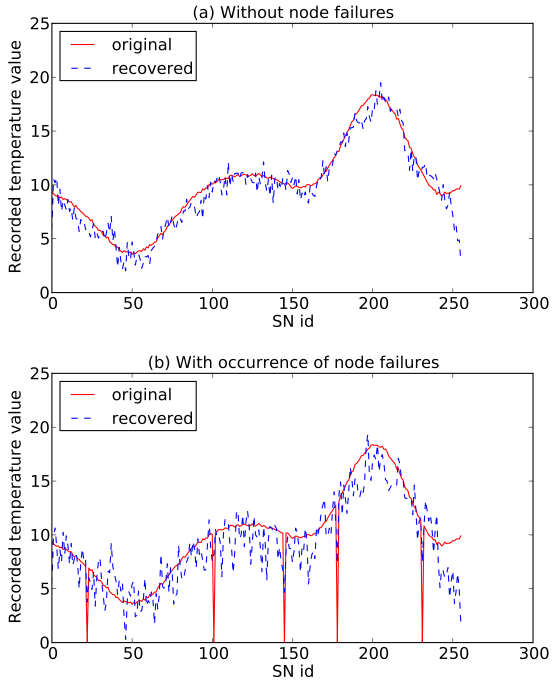

Fig. 5. Degraded signal recovery due to missing samples

significantly decreases. Furthermore, it is not possible to detect which nodes are failing. Next, we present a method that exactly detects the location of failures and excludes the failing SNs from the signal recovery process.

C. Signal elevation during measurement

From our system model description we recall that the values recorded by the SNs are bounded between a lower and upper bound, namely s_l and s_u respectively. In our exemplified WSN, we assume that the temperature values are bounded between 0 and 25. Note that s_l and s_u can be any positive or negative real numbers. We take 0 and 25 just as an example here. This discussion also applies to signals recorded from any other physical parameter other than temperature.

Suppose that each SN *elevates* its recorded value by an *offset* before applying the measurement mechanism illustrated in Figure 4. Formally speaking, when SN i senses the value f_i, it first adds f_i by a positive real number c that we call it *offset* and then multiplies $\boldsymbol{\alpha}_i$ by $f_i + c$.

We choose c to be two or three order of magnitudes larger than $|s_l - s_u|$. The reason is that when node failures occur, the missing samples are better distinguished from the genuine samples. In fact, the offset elevates the range of valid values to a higher level. In our current example, if we select $c = 1000$, then the range of offsetted values will be $[1000, 1025]$ instead of $[0, 25]$.

Note that when all nodes perform the offsetting, the whole signal vector $\mathbf{f}$ is elevated by the offset c. Let $\mathbf{g} \in \mathbb{R}^n$ be the elevated version of $\mathbf{f}$, i.e.,

$$\mathbf{g} = \mathbf{f} + \mathbf{c}_n \tag{7}$$

where $\mathbf{c}_n$ is a column vector of size n with all of its elements being equal to c.

D. Detection and exclusion of the missing samples

When some nodes fail, their corresponding values in $\mathbf{g}$ will also drop to zero, since they cannot transmit their measurements. In this case, the compressibility of $\mathbf{f}$ (and also $\mathbf{g}$) under DCT decreases. However, $\mathbf{g}$ shows high compressibility under Haar wavelet transform [17]. In order to find out which nodes are missing, we try to recover $\mathbf{g}$ by solving the following convex optimization problem.

$$\underset{\tilde{\mathbf{u}} \in \mathbb{R}^n}{\text{minimize}} \ \|\tilde{\mathbf{u}}\|_1 \quad , \quad \text{subject to} \ \ \|\mathbf{y} - \mathbf{\Phi}\mathbf{\Psi}_H \tilde{\mathbf{u}}\|_2^2 \ \leq \ \epsilon \tag{8}$$

where $\mathbf{\Psi}_H$ is the inverse Haar transformation matrix.

If $\hat{\mathbf{u}}$ is the solution to Equation 8, then $\hat{\mathbf{g}} = \mathbf{\Psi}_H \hat{\mathbf{u}}$ estimates the original elevated signal $\mathbf{g}$. The size of the signal must be a power of two when applying the Haar transform. In this example, the size of the signal is equal to the number of SNs, i.e., $n = 256$. When the number of nodes is not a power of two, one can pad sufficient number of pseudo-samples with a predetermined value. Pseudo-samples with a value of c is a suitable choice here.

Recovery result using Haar wavelet is depicted in Figure 6.a. Here, $s_l = 0$, $s_u = 25$ and $c = 1000$. While recovery using DCT cannot distinguish the failing SNs, using Haar wavelet as our compressive basis, we can exactly detect the location of the failing nodes. The estimated signal $\hat{\mathbf{g}}$ recovered by solving Equation 8 shows significantly lower values at the failing nodes, see Figure 6.a. By employing signal elevation and performing the reconstruction using Haar wavelet, the exact location of the failures in a chain WSN is determined.

After detecting the position of the missing samples, we exclude those samples and rerun the signal recovery on that part of the signal that actually contains valid sensor readings.

Using matrix representations, the measurement vector received by the sink is given by

$$\mathbf{y} = \mathbf{\Phi}\mathbf{g} + \mathbf{z} \tag{9}$$

where $\mathbf{z}$ is the additive noise.

Let M be the set of failing nodes and k be the number of failing nodes, i.e., $k = |M|$. We define an $m \times (n-k)$ matrix $\mathbf{\Phi}'$ by removing the columns $m_1, m_2, \ldots, m_k$ from $\mathbf{\Phi}$ where $\{m_1, m_2, \ldots, m_k\} = M$.

$$\mathbf{\Phi}' := [\Phi_{i,j}] \quad i \in \{1, \ldots, n\} \ , \ j \in \{1, \ldots, n\} - M \tag{10}$$

We define the vector $\mathbf{g}'$ of size $(n-k)$ by removing the elements g_i from $\mathbf{g}$ where $i \in M$.

$$\mathbf{g}' := [g_i] \quad i \in \{1, \ldots, n\} - M \tag{11}$$

It is easy to show that:

$$\mathbf{\Phi}\mathbf{g} = \mathbf{\Phi}'\mathbf{g}'. \tag{12}$$

We also define the column vector $\mathbf{f}'$ by removing the elements f_i from $\mathbf{f}$ where $i \in M$.

$$\mathbf{f}' := [f_i] \quad i \in \{1, \ldots, n\} - M \tag{13}$$

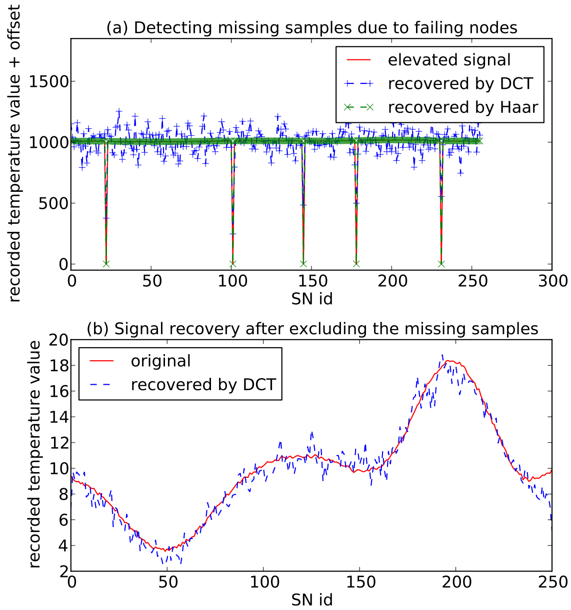

Fig. 6. Detecting and excluding the failing SNs

According to the definitions of $\mathbf{g}$ and $\mathbf{g}'$, we know that:

$$\mathbf{g}' = \mathbf{f}' + \mathbf{c}_{(n-k)} \tag{14}$$

where $\mathbf{c}_{(n-k)}$ is a column vector of size $(n-k)$ with all of its elements being equal to c.

Since $\mathbf{f}$ is compressible under DCT and $\mathbf{f}'$ is equal to $\mathbf{f}$ except some missing samples, it is still expected to be sufficiently compressible under DCT, i.e.,

$$\mathbf{f}' = \mathbf{\Psi}'_D \mathbf{x}' \tag{15}$$

such that $\mathbf{\Psi}'_D$ is the $(n-k) \times (n-k)$ inverse DCT matrix and $\mathbf{x}' \in \mathbb{R}^{(n-k)}$ is the (nearly) sparse projection of $\mathbf{f}'$ under DCT. Note that any compressive basic other than DCT can also apply here. Choosing DCT is independent of our offsetting and detection mechanism using Haar wavelets. We use DCT as a an example and this discussion can be generalized to any other compressive basis Ψ when $\mathbf{f} = \mathbf{\Psi x}$ and $\mathbf{x}$ is sparse or nearly sparse.

Putting Equation 12 in Equation 9 we have:

$$\mathbf{y} = \mathbf{\Phi}'\mathbf{g}' + \mathbf{z} \tag{16}$$

and by substituting Equation 14 we have:

$$\begin{aligned} \mathbf{y} &= \mathbf{\Phi}'(\mathbf{f}' + \mathbf{c}_{(n-k)}) + \mathbf{z} \\ &= \mathbf{\Phi}'\mathbf{f}' + \mathbf{\Phi}'\mathbf{c}_{(n-k)} + \mathbf{z} \end{aligned} \tag{17}$$

We define a vector $\mathbf{w} \in \mathbb{R}^m$ as:

$$\mathbf{w} := \mathbf{y} - \mathbf{\Phi}'\mathbf{c}_{(n-k)} \tag{18}$$

According to the definition of $\mathbf{w}$ and Equation 17, we have:

$$\mathbf{w} = \mathbf{\Phi}'\mathbf{f}' + \mathbf{z}. \tag{19}$$

Therefore, we can recover $\mathbf{x}'$ by solving a modified version of Equation 3 as follows.

$$\underset{\tilde{\mathbf{x}}' \in \mathbb{R}^n}{\text{minimize}} \ \|\tilde{\mathbf{x}}'\|_1 \quad , \quad \text{subject to} \ \ \|\mathbf{w} - \mathbf{\Phi}'\mathbf{\Psi}'_D\tilde{\mathbf{x}}'\|_2^2 \ \leq \ \epsilon \tag{20}$$

If $\hat{\mathbf{x}}'$ is the solution to Equation 20, the original signal excluding the missing samples, i.e., $\mathbf{f}'$ is then estimated by $\hat{\mathbf{f}}' = \mathbf{\Psi}'_D\hat{\mathbf{x}}'$. The recovery result is shown in Figure 6.b for our current example.

By comparing Figures 6.b and 5.b to each other, we see that, a much more accurate signal recovery is possible after excluding the missing samples. Figure 5.b shows the recovery when some of the samples are missing due to node failures. Figure 6.a shows how our technique can exactly detect the samples that are lost due to failing SNs. Figure 6.b shows the recovery of the same signal after excluding the missing samples. We observe a significantly more accurate signal reconstruction after exclusion of the missing samples.

Note that in Figure 6.b the size of the recovered signal is 251 instead of 256, because 5 samples corresponding to the failing SNs are excluded.

E. *Detecting unrecoverable chain breakage*

In this section we study the situation where the chains breaks at some node and our chain rebuilding procedure as described in section IV-A does not restore the connectivity. We do not want the next hops after the failing nodes to wait for the retransmission of the lost encoded values. In particular, when the failing nodes face an unrecoverable error, those encoded values may never be retransmitted and the next hop will wait forever for the missing part of the measurement vector. We apply the same technique as discussed in the previous section. The SNs elevate their sensed values by an offset c. Then, we first perform signal recovery using Haar wavelet to detect the position of the failure. Finally, we exclude the missing portion of the spatial signal and rerun the recovery on the valid part of the signal using the compressive basis under which the genuine data are compressible.

The conditions are the same as our example in the previous section. The signal $\mathbf{f}$ is compressible under DCT and records temperature values between 0 and 25 degrees which are elevated by $c = 1000$. We inject a failure in SNs $\{50 - k_{max}, \dots, 50\}$. Thus, the sensed values from nodes $\{1, \dots, 50\}$ will be lost as the chain restoration mechanism at node 50 is unsuccessful.

Recovery of the elevated signal after solving Equation 8 is depicted in Figure 7.a. We observe that recovery using DCT cannot exactly determine the location of the failing node. It gives only a rough estimation of the location where the chain is broken. On the other hand, recovery using Haar wavelet exactly distinguishes the missing segment of the signal and determines the position of the failure.

Signal recovery using DCT without excluding the missing segment still gives us a good accuracy. However, it erroneously detects low temperatures for SNs 1-50, see the dotted curve in Figure 7.b. The recovered signal after excluding the missing segment retains its accuracy and also distinguishes the missing samples for nodes 1-50, see the dashed curve in Figure 7.b.

V. Evaluation

Throughout the paper we have tested our technique on different simulated WSNs with our illustrative synthesized compressible signals. In this section, we apply our method

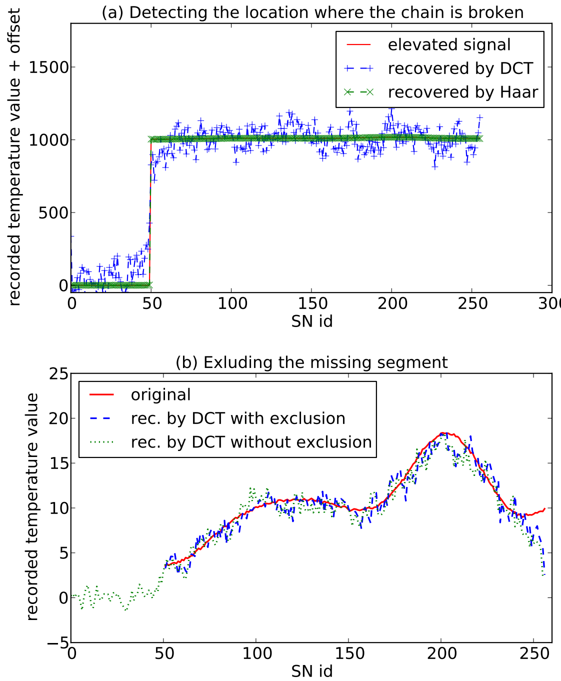

Fig. 7. Detecting the location of unrecoverable chain breakage

on real-world datasets and put the WSN under stress tests to evaluate the performance of our failure detection technique.

The simulations are performed on a real-world dataset form the Sensorscope deployment [18]. Since not all of the SNs in the testbed were sampling synchronously, we selected 64 SNs with the most amount of synchronously sampled data. The sensed data from LUCE dataset is applied in our simulated network that possesses a linear topology. We take the data from the real-world dataset while the network topology is determined in our simulation program to be a linear topology. We employ our implementation of the recovery algorithm which uses CVXOPT software package [19]. The simulation is implemented in Python using the NumPy/SciPy scientific programming libraries [20].

First, we evaluate the effectiveness of our step-back method introduced in Section IV-A. In particular, we want to analyze the behavior of the step-back method for different values of k_{max} and different number of failure occurrences. We assume that each node has an independent probability p of failure. We simulate the chain WSN of size 64 with different values for $k_{max} \in \{1, 2, 3\}$ and different values of node failure probability $p \in [0.01, 0.15]$. Looking at Figure 8.a, we observe that the step-back method effectively reduces the amount of lost samples. Without the step-back method, i.e. when $k_{max} = 0$, we loose a lot of samples whenever a single node fails to transmit its values. This happens because when $k_{max} = 0$ and no step-back takes place, any node failures lead to a chain breakage. The step-back method tries to maintain the connectivity of the network when a node failure occurs. With higher k_{max} each node tries to fetch the values from farther predecessor nodes when their direct predecessor does not send them any data.

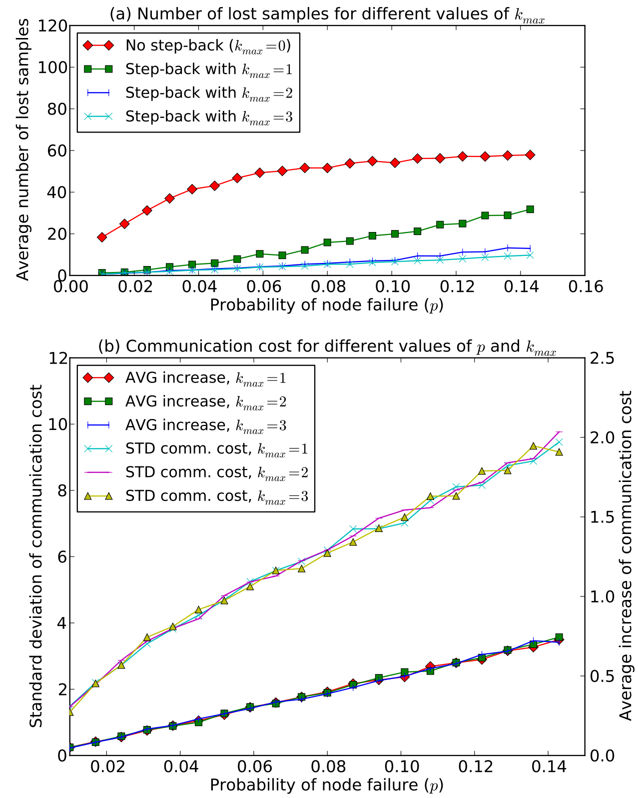

Fig. 8. Analysis of the step-back method

Observation 1. *The step-back method significantly reduces the lost samples when restoring the connectivity of the chain succeeds.*

Another interesting observation is that the average amount of increase in power consumption and the balance of load on the SNs mainly depends on the probability failure p. Figure 8.b illustrates the increase in communication cost and its standard deviation across the network after applying the step-back method for different values of p and k_{max}. We observe a moderate increase in average power consumption when applying the step-back method. The standard deviation shows how the load is distributed on the network. Higher standard deviation indicates that the nodes that are restoring the chain connectivity are overloaded.

Observation 2. *Higher step-back limit k_{max} increases the ability to restore the chain connectivity. Changes in power consumption of the WSN is prevalently determined by the failure probability p rather than step-back limit k_{max}.*

According to Observation 2 it is recommended to use a higher step-back limit k_{max} as long as the hardware capabilities of the SN allows it.

Now, we put the WSN under *stress test* and measure the accuracy of the recovered signal. The stress test deliberately fails some of the SNs, i.e., zeros their corresponding samples in the spatial signal. We measure the accuracy of signal recovery when the number of failures increases. The accuracy of the

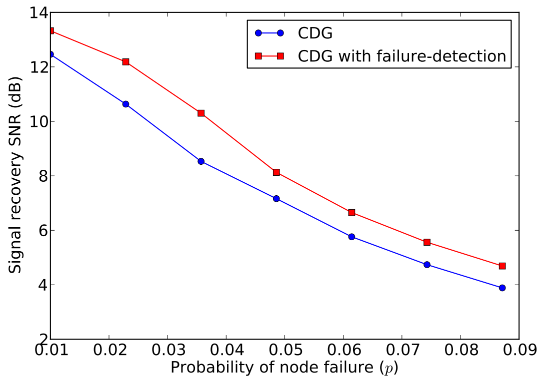

Fig. 9. More accurate recovery after isolation of the missing samples

recovered signal is given by Signal-to-Noise Ratio (SNR) which is measured in decibels (dB). In our simulated network, setting $k_{max} = 5$ effectively maintains the connectivity of the chain. Figure 9 illustrates the signal reconstruction accuracy of our method and compares it to CDG for chain topology [10]. Equipping CDG with out failure detection and isolation mechanism improves the accuracy of the recovered signal. Note that each unit dB increment of SNR roughly corresponds to 25% higher accuracy of the reconstructed signal as we measure the SNR by a logarithmic scale.

Observation 3. *Detection and isolation of the missing samples improves the accuracy of the recovered signal in Compressive Data Gathering.*

VI. Conclusion

This paper provides an enhancement to Compressive Data Gathering (CDG) in chain Wireless Sensor Networks (WSNs). CDG is based on the theory of Compressed Sensing (CS) that allows an efficient and robust data collection method for large-scale WSNs and especially WSN with linear topology. CS-based data collection methods for WSNs like CDG are inherently robust to additive communication noise. In addition to communication noise, a WSN also faces another source of erroneous data collection. The WSNs are usually deployed in harsh operational environments, and thus, it is likely that some SNs gets damaged and cannot continue their function. Thus, the SNs of a WSN are at the risk of temporary or permanent defects. In this paper, we studied the performance of CDG in WSNs with linear topology under circumstances where a SN encounters a failure and cannot transmit its measurements or forward the accumulated measurements from other SNs.

We proposed a simple and effective method based on a best-effort technique to maintain the connectivity of the chain topology. We also introduce an enhancement to CS measurement and recovery that excludes the missing samples due to node failures without transmitting health monitoring messages. Our proposed technique, first determines the location of the missing samples that are caused by node failures. Then, the recovery algorithm is executed on the remaining part of the signal that contains genuine data. Our evaluations prove that exclusion of the missing samples significantly improves the accuracy of the recovered signal.

Acknowledgment

The authors are partly supported by the Research Training Group (Graduiertenkolleg) 1362 of the German Research Foundation (Deutsche Forschungsgemeinschaft - DFG).

References

[1] J. Elson and D. Estrin, "Sensor networks: A bridge to the physical world," in *Wireless Sensor Networks*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Springer US, 2004, pp. 3–20.

[2] J. Polastre *et al.*, "Telos: enabling ultra-low power wireless research," in *ACM IPSN'05*, 2005, pp. 364–369.

[3] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26–34, 2006.

[4] A. O. Marco F. Duarte, Godwin Shen and R. G. Baraniuk, "Signal compression in wireless sensor networks," *Philosophical Trans. of the Royal Society*, vol. 370, no. 1958, pp. 118–135, 2012.

[5] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 37 – 59, 2012.

[6] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Joint source-channel communication for distributed estimation in sensor networks," *IEEE Trans. on Information Theory*, vol. 53, no. 10, pp. 3629–3653, 2007.

[7] E. Candes *et al.*, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489 – 509, 2006.

[8] M. Mahmudimanesh, A. Khelil, and N. Suri, "Balanced spatio-temporal compressive sensing for multi-hop wireless sensor networks," in *IEEE MASS'12*, 2012, pp. 389–397.

[9] ——, "Reordering for better compressibility: Efficient spatial sampling in wireless sensor networks," *Proc. of SUTC'10*, pp. 50–57, 2010.

[10] C. Luo *et al.*, "Compressive data gathering for large-scale wireless sensor networks," in *Proc. of the 15th annual international conference on Mobile computing and networking (Mobicom)*, 2009, pp. 145–156.

[11] E. Candes and Y. Plan, "A probabilistic and ripless theory of compressed sensing," *IEEE Trans. on Information Theory*, vol. 57, no. 11, pp. 7235–7254, Nov.

[12] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5406 –5425, 2006.

[13] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–617, 2007.

[14] S. Becker, J. Bobin, and E. J. Candès, "Nesta: A fast and accurate first-order method for sparse recovery," *SIAM J. Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011.

[15] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[16] W. Bajwa *et al.*, "Compressive wireless sensing," in *Proc. of the 5th international conference on Information processing in sensor networks (IPSN)*, 2006, pp. 134–142.

[17] A. Graps, "An introduction to wavelets," *IEEE Compututer Science Engineering.*, vol. 2, pp. 50–61, June 1995.

[18] "Sensorscope LUCE deployment at École Polytechnique Fédérale de Lausanne," 2008. [Online]. Available: http://lcav.epfl.ch/op/edit/sensorscope-en

[19] J. Dahl and L. Vandenberghe, "CVXOPT: A python package for convex optimization, http://www.abel.ee.ucla.edu/cvxopt," 2006.

[20] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python, http://www.scipy.org," 2001–.