On the Interplay Between Routing and Signal Representation for Compressive Sensing in Wireless Sensor Networks

Giorgio Quer†, Riccardo Masiero†, Daniele Munaretto⋆,
Michele Rossi†, Joerg Widmer⋆ and Michele Zorzi†
†DEI, University of Padova, via Gradenigo 6/B – 35131, Padova, Italy.
Email: {giorgio.quer, riccardo.masiero, rossi, zorzi}@dei.unipd.it
⋆DoCoMo Euro-Labs, Landsberger Strasse 312 – 80687, Munich, Germany.
Email: {munaretto, widmer}@docomolab-euro.com

***Abstract*— Compressive Sensing (CS) shows high promise for fully distributed compression in wireless sensor networks (WSNs). In theory, CS allows the approximation of the readings from a sensor field with excellent accuracy, while collecting only a small fraction of them at a data gathering point. However, the conditions under which CS performs well are not necessarily met in practice. CS requires a suitable transformation that makes the signal sparse in its domain. Also, the transformation of the data given by the routing protocol and network topology and the sparse representation of the signal have to be incoherent, which is not straightforward to achieve in real networks. In this work we address the data gathering problem in WSNs, where routing is used in conjunction with CS to transport random projections of the data. We analyze synthetic and real data sets and compare the results against those of random sampling. In doing so, we consider a number of popular transformations and we find that, with real data sets, none of them are able to sparsify the data while being at the same time incoherent with respect to the routing matrix. The obtained performance is thus not as good as expected and finding a suitable transformation with good sparsification and incoherence properties remains an open problem for data gathering in static WSNs.**

I. Introduction

The area of communication and protocol design for Wireless Sensor Networks (WSNs) has been widely researched in the past few years. An important research topic which needs further investigation is in-network aggregation and data management to increase the efficiency of data gathering solutions (in terms of energy cost) while being able to measure large amounts of data with high accuracy. Compressive Sensing (CS) [1]–[3] is a novel data compression technique that exploits the inherent correlation in some input data set X to compress such data by means of quasi-random matrices. If the compression matrix and the original data X have certain properties, X can be reconstructed from its compressed version Y, with high probability, by minimizing a distance metric over a solution space. CS was originally developed for the efficient storage and compression of digital images, which show high spatial correlation. Recently, there has been a growing interest in these techniques by the telecommunications and signal processing communities [4]. In contrast to classical approaches, where the data is first compressed and then transmitted to a given destination, with CS the compression phase can be jointly executed with data transmission. This is important for WSNs as compressing the data before the transmission to the data gathering point (hereafter called the sink) requires to know in advance the correlation properties of the input signal over the entire network [5] (or over a large part of it) and this implies high transmission costs. With CS, the content of packets can be mixed as they are routed towards the sink. Under certain conditions, CS allows to reconstruct all sensor readings of the network using much fewer transmissions than routing or aggregation schemes. These characteristics make CS very promising for jointly acquiring and aggregating data from distributed devices in a multi-hop wireless sensor network [4].

In this paper, we address the problem of exploiting CS in WSNs taking into account network topology and routing, which is used to transport random projections of the sensed data to the sink. Thus, the main contribution of this paper is the quantification of the benefits of CS in realistic multi-hop WSNs when CS in used in conjunction with routing. In addition, we study the problem of finding good transformations to make real sensed data meet the sparsity requirements of CS and show that widely used transformations are not suitable for a large spectrum of real signals. We also provide a simulation based comparison between the commonly used random sampling (considered here in conjunction with spline interpolation) and CS based data gathering, for synthetic and real sensed data.

The paper is structured as follows. In Section II we summarize the related work on CS applied to WSNs, followed by a mathematical overview of the CS technique in Section III. In Sections IV, V and VI we describe the signals, the network model, and the data gathering protocols, respectively, which we used for the investigation of the benefits of CS applied to multi-hop WSNs. The simulation results are presented in Section VII and Section VIII concludes the paper.

II. Related Work

The problem of gathering data while jointly performing compression has been receiving increasing attention. Recently, new methods for distributed sensing and compression have been developed based on CS theory (see e.g., [2], [3], [6], [7]). An early contribution is [8], where CS is used in a distributed communication scheme for energy efficient estimation of sensed data in a WSN. Multi-hop communication and in-network data processing are not considered. Instead, data packets are directly transmitted by each node to the sink. This requires synchronization among nodes.

[9] proposes an early application involving CS for network monitoring. The considered simulation scenario is a network where a small set of nodes fails. The goal is to correctly identify these nodes through the transmission of random projections (i.e., linear combinations) indicating the status of the nodes. However, these random projections are obtained by means of a pre-distribution phase (via simple gossiping algorithms), which is very expensive in terms of number of transmissions. [10] also addresses the problem of gathering data in distributed WSNs through multi-hop routing. In detail, tree topologies are exploited for data gathering and routing, and the Wavelet transformation [11] is used for data compression. Even though CS is presented as one of the possible methods for data compression, the authors do not investigate the impact of the network topology and that of the routing scheme on the compression process. An interesting application for network monitoring exploiting CS is presented in [12], where the aim is to efficiently monitor communication metrics, such as loss or delay, over a set of end-to-end network paths by observing a subset of them. The topology is given a priori and the algorithm works in three steps: 1) compression, 2) non linear estimations and 3) suitable path selection. This last step in particular allows the selection of the best measurements for CS recovery, and therefore highly impacts the overall performance of the algorithm. In [13] and [14] an approach to distributed coding and compression in sensor networks based on CS is presented. The authors advocate the need to exploit the data both temporally and spatially. The projections of the signal measurements are performed at each source node, taking into account only the temporal correlation of the generated information. Thus, it is possible to design the best approximation of the collection of measurements for each node, since the projections can contain all the elements of this set. The spatial correlation is then exploited at the sink by means of suitable decoders through a joint sparsity model that well characterizes the different types of signals of interest. Finally, the technical report [15] follows an approach similar to ours, concluding that CS is not an effective solution when routing costs are considered.

In our work we address the joint routing and compression problem by exploiting the spatial correlation among sensor readings in a 2D WSN. The sensor nodes do not need to be aware of any correlation structure of the input signal. In particular, we only require that the sensed data has a sparse representation and that the sensor nodes can locally perform random combinations of the incoming information. The goal is to reconstruct the original signal with good accuracy from a small subset of samples using distributed CS. To the best of our knowledge, no papers so far quantified the performance of CS in multi-hop wireless networks by exploiting actual routing topologies to obtain random projections of the signal measurements, except for [15]. However, their conclusions about the effectiveness of CS for synthetic signals are different from ours and they did not address real signal analysis. The objective of our work is to fill this gap investigating the tradeoffs between energy consumption and reconstruction error for realistic scenarios. Furthermore, we analyze under which conditions CS performs well and under which conditions it fails to improve the performance.

III. Preliminaries: Mathematical Background on CS

Next, we give a concise overview of the CS technique. Compressive Sensing is a recent method to represent compressible signals with significantly fewer samples than required by the sampling theorem. Reconstruction of the original data is possible with high probability through dedicated non-linear recovery algorithms without loss of information in the absence of noise and with excellent accuracy when observations are noisy [16].

For the sake of exposition, we consider signals representable through one dimensional vectors $\mathbf{x}$ in $\mathbb{R}^N$, where N is the vector length. We assume that these vectors contain the sensor readings of a network with N nodes. We further assume that these vectors are such that there exists a transformation under which they are sparse. Specifically, there must exist an invertible transformation matrix $\mathbf{\Psi}$ of size $N \times N$ such that we can write

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s} \tag{1}$$

and $\mathbf{s}$ is sparse. We say that a vector $\mathbf{s}$ is P-sparse if it has at most P non-zero entries, with $P < N$.

The compression of $\mathbf{x}$ entails a linear combination of its elements through a further *measurement matrix* $\mathbf{\Phi}$ of size $M \times N$, with $M < N$. The compressed version of $\mathbf{x}$ is thus obtained as

$$\mathbf{y} = \mathbf{\Phi x} \, . \tag{2}$$

Now, using (1) we can write

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi s} \stackrel{def}{=} \tilde{\mathbf{\Phi}} \mathbf{s} \, . \tag{3}$$

These systems are ill-posed as the number of equations M is smaller than the number of variables N. Nevertheless, if $\mathbf{s}$ is sparse, it has been shown that the above system can be inverted with high probability through the use of specialized optimization techniques [3], [17]. Once we know a sparse solution $\mathbf{s}^\star$ that verifies (3), the original data $\mathbf{x}$ can be recovered through (1).

Next, we illustrate the reconstruction process. Given a solution $\mathbf{s}_p$ of (3) such that $\tilde{\mathbf{\Phi}} \mathbf{s}_p = \mathbf{y}$ and given the null space of matrix $\tilde{\mathbf{\Phi}}$, $\mathcal{N}(\tilde{\mathbf{\Phi}})$ of dimension $N-M$, any vector $\mathbf{s}' = \mathbf{s}_p + r$, where $r \in \mathcal{N}(\tilde{\mathbf{\Phi}})$ is also a solution of (3). However, in [3] it is proved that: A1) if matrix $\tilde{\mathbf{\Phi}}$ has certain properties and A2) if $\mathbf{s}$ is P-sparse with P smaller than a given threshold, then the original $\mathbf{s}$ is the sparsest admissible solution of (3). The solution that we find in this way, that we call $\mathbf{s}^\star$, is equal to the original $\mathbf{s}$ if assumptions A1 and A2 hold. Otherwise, there will be a reconstruction error that decreases for increasing M. Of course, when $M = N$ and $\tilde{\mathbf{\Phi}}$ is full rank, the only solution of this system is $\mathbf{s}$ and it can be obtained through standard matrix inversion.

From a data gathering point of view, the signal $\mathbf{x}$ stores the data readings measured by the N nodes. These are mixed during their transmission towards the sink as explained in Section VI. Thus, each route followed by a given packet specifies the coefficients of a row of $\mathbf{\Phi}$. The data gathering point will receive the compressed vector $\mathbf{y}$ along with the coefficients of matrix $\mathbf{\Phi}$. In the following, $\mathbf{\Phi}$ is referred to as *routing matrix*. Note that the sink can obtain the coefficients of $\mathbf{\Phi}$ through different ways, e.g., these coefficients can be sent along with the information packet if the relative overhead is small, or we can use the same pseudo-random number generator at the nodes and the sink and synchronize the seeds. The problem to be solved at the receiver is thus to invert the system (3) so as to find vector $\mathbf{s}$. Note that in order to do this, the receiver should also know the transformation matrix $\mathbf{\Psi}$ that sparsifies $\mathbf{x}$.[1] We emphasize that the transformation $\mathbf{\Psi}$ is only used at the sink and not during the data gathering and routing process, that is instead captured by $\mathbf{\Phi}$. In particular, $\mathbf{\Psi}$ does not need to be known at any node but the sink.

A generalization of the CS technique for 2D signals is detailed in the Appendix.

[1]This is a reasonable assumption. For example, for image processing it has been verified that the Fourier transformation is a good tool for sparsifying real images [7]. Signals gathered by sensor fields usually show high spatial correlation [18] and can thus be sparsified as we discuss in Section IV.

IV. Considered Signals and Transformations

In this section we discuss the signals that we consider for the performance evaluation in this paper. First, we investigate synthetic signals that are sparse by construction under the DCT transformation. For these signals the degree of sparseness can be precisely controlled. As expected, when they are sufficiently sparse CS achieves substantial gains compared to plain routing schemes. Furthermore, we select a number of signals from real sensor networks measuring different physical phenomena. With such signals, we can much better characterize the performance expected for actual WSN deployments. The problem with real signals, however, is to find a good transformation that sparsifies them in some domain. This issue is discussed at the end of the section.

Synthetic signals. Here, for the input signal we use a matrix $\mathbf{X}$ that we build starting from a sparse and discrete 2D signal $\mathbf{S}$ in the frequency (DCT) domain. $\mathbf{S}$ is obtained through the following steps:

1. Let K be defined as $K = \sqrt{N}$, where N is the number of values of the 2D signal. We build a preliminary signal $\mathbf{S}_1$ of size $K \times K$ having all frequencies (i.e., all entries in the matrix) with amplitude $s_1(p,q)$, where $s_1(p,q)$ is picked uniformly at random in the interval $[0.5, 1.5]$, $\forall\, p, q = 1, 2, \ldots, K$.
2. We define a frequency mask as a 2D function that is one for entries in position (p,q) where $p + q \leq p_{\text{low}}$ or $p + q > p_{\text{high}}$ and zero otherwise. p_{low} and p_{high} are two thresholds in the value range $\{1, 2, \ldots, K\}$. This function is defined as
$$\texttt{triang}(p,q) \stackrel{def}{=} \begin{cases} 1 & p + q \leq p_{\text{low}} \text{ or} \\ & p + q > p_{\text{high}} \\ 0 & \text{otherwise} \, . \end{cases} \tag{4}$$
3. We obtain a second signal $\mathbf{S}_2$ of size $K \times K$, whose entries $s_2(p,q)$ are calculated as
$$s_2(p,q) = s_1(p,q)\texttt{triang}(p,q) \, . \tag{5}$$
4. We finally obtain $\mathbf{S}$ as follows: if $s_2(p,q) = 0$ then $s(p,q) = \xi$ where $\xi \in [0, 0.01]$ is a constant. If instead $s_2(p,q) > 0$, $s(p,q) = \xi$ with probability p_d and $s(p,q) = s_2(p,q)$ otherwise. The parameter p_d represents the fraction of entries that are on average deleted from $\mathbf{S}_2$. The case $\xi > 0$ is accounted for to mimic non ideal signals, where the significant components lie within specific regions according to

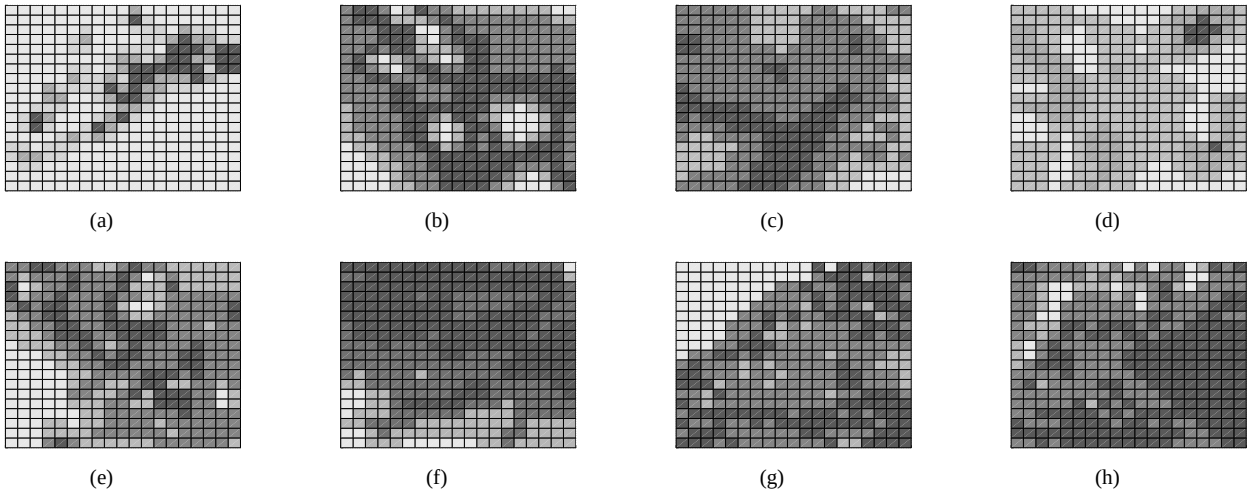

Fig. 1. Real signals: (a) Wi-Fi strength from MIT, (b) Wi-Fi strength from Stevens Institute of Technology, (c) Ambient temperature from EPFL SensorScope WSN, (d) Solar radiation from EPFL SensorScope WSN, (e) Rainfall in Texas, (f) Temperature of the ocean in California, (g) Level of pollution in Benelux and (h) in northern Italy.

(4) and some *noise floor* is also present outside these regions. In this case, with CS we would like to only retrieve the significant values, while ignoring the noise.

Therefore, the signal $\mathbf{S}$ is obtained by first applying a frequency mask, which helps to assess the reconstruction performance for low-frequency, mid-frequency, and high-frequency signals. In addition, we delete some randomly picked frequencies according to a given probability p_d. This is a simple method to control the characteristics of the signal in the DCT domain (i.e., the sparsity of the signal and its dominant frequency components) and allows to understand the effects of the signal structure on the performance of CS. For the results in Section VII synthetic signals are mapped into matrices $\mathbf{X}$ of size 20×20, which is consistent with the network topology in Section V with $N = 400$ nodes.

Real Signals. We also used real signals from different environmental phenomena, considering what is likely to be of interest for a realistic wireless sensor network in terms of size of the network (i.e., number of spatial samples) and type of phenomenon to sense. For the sensor network, we considered the topology in Section V with $N = 400$ sensor nodes.

The following real signals were utilized:

S1. Two signals representing the Wi-Fi strength of the access points in the MIT campus (Cambridge, MA) [19] and in the Stevens Institute of Technology (Hoboken, NJ) [20].

S2. Two sets of measurements from the EPFL SensorScope WSN [21], representing ambient temperature and solar radiation.

S3. Two data readings, one from the Tropical Rainfall Measuring Mission [22] concerning rain fall in Texas, and one on the temperature of the ocean off the coast of California [23].

S4. Two signals on the level of pollution in two European regions, namely, Benelux and Northern Italy [24].

These signals were quantized into five levels and rescaled in grids of 20×20 pixels. The assumption of measuring quantized signals was made as we think this is likely to be the case in actual WSN deployments, where the devices, due to communication, energy constraints or accuracy of the on-board sensor, can only sense or communicate the physical phenomena of interest according to a few discrete levels. In addition, for many signals of interest a quantized representation suffices to fully capture the needed information about the sensed phenomenon. The eight sample signals, quantized and rescaled as discussed above, are shown in Fig. 1.

Transformations. By construction, for the above synthetic signals the DCT is the right sparsification method. These signals were in fact created sparse in the DCT domain. An effective utilization of CS for real signals requires a good sparsification approach. It is not clear, however, which approach is best for a given class of signals. Here, we consider four different transformations, which are commonly used in the image processing literature:

T1. *DCT:* this is the standard 2D discrete cosine transformation, see the appendix for further details.

T2. *Haar Wavelet:* the Haar Wavelet is recognized as the first known Wavelet and is a good Wavelet transformation for the sparsification of piece-wise constant

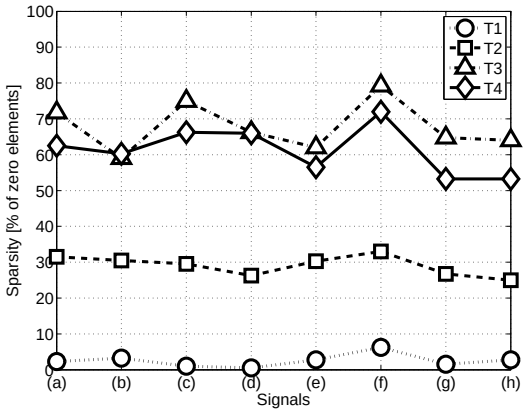

Fig. 2. Degree of sparsity for transformations T1–T4. The plot shows the percentage of zero elements of vector $\mathbf{s}$ after using transformations T1–T4.

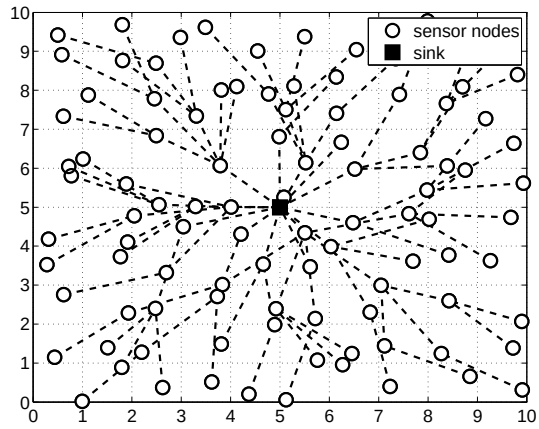

Fig. 3. Example of the considered multi-hop topology.

signals as the ones in S1–S4, see [25].

T3. *Horz-diff:* this is a transformation that we propose here to exploit the spatial correlation of our signals. First, the 2D signal matrix $\mathbf{X}$ is written in vector form as follows:

$$\begin{aligned}\mathrm{svec}(\mathbf{X}) \quad = \quad & (x(1,1), x(1,2), \ldots, x(1,K), \\ & x(2,K), x(2,K-1), \ldots, x(2,1), \\ & x(3,1), x(3,2), \ldots, x(3,K), \\ & x(4,K), x(4,K-1), \ldots, x(4,1), \\ & \ldots) \end{aligned} \tag{6}$$

At this point we obtained the sparse vector $\mathbf{s}$ from $\mathrm{svec}(\mathbf{X})$ by pair-wise subtraction of its elements.

T4. *HorzVer-diff:* according to this transformation the input signal $\mathbf{X}$ is processed by: 1) pair-wise subtraction of the elements along the columns of $\mathbf{X}$ and then 2) pair-wise subtraction of the elements of the resulting matrix, along its rows.

In Fig. 2 we show the degree of sparseness achievable using the above transformations T1–T4 with the considered real signals (a)–(h). Notably, DCT (T1) and Haar Wavelet (T2) are not effective, whereas T3 and T4 perform best.

DCT and Wavelet transformations in this case have poor performance as, even though the sampled input signals $\mathbf{X}$ are quite large ($N = 400$ data points) for typical sensor deployments (where each node gathers a single data point), their size is still too small for T1 and T2 to perform satisfactorily. T3 and T4 perform best since they exploit the characteristics of piece-wise constant signals, even if the sparsity obtained is not sufficient for CS to work properly. Since standard techniques as T1–T4 are not sactisfactory, a more fundamental approach, i.e., via estimation of the correlation $\mathbf{X}$ and Karhunen-Loève expansion, may be needed. We leave this for future research.

V. Network Model

The concern of this paper is about data gathering in 2D WSNs. Hence, for the rest of the paper we consider sensor grids of N nodes as follows. We consider N nodes to be deployed in a square area with side length L. This area is split into a grid with N square cells and we place each of the N nodes uniformly within a given cell so that each cell contains exactly one node. For the transmission range R of the nodes we adopt a unit disk model, i.e., nodes can only communicate with all other nodes placed at a distance less than or equal to R.[2] We use $R = \sqrt{5}L/\sqrt{N}$ as this guarantees that the structure is fully connected under any deployment of the nodes. A further node, the data gathering point or sink node, is placed in the center of the deployment area. We consider geographic routing to forward the data towards the sink, where each node considers as its next hop the node within range that provides the largest geographical advancement towards the sink. In Fig. 3, we show an example topology; as per the above construction process, each cell has a node and the network is always connected. The tree in this figure is obtained through the above geographic routing approach, and is used by the data aggregation protocols to route data towards the sink.

According to this network scenario, the input signal is a square matrix $\mathbf{X}$ with N elements, where element (i, j) (referred to as $x(i, j)$) is the value sampled by the sensor placed in cell (i, j) of the sensor grid.

[2]The unit disk graph model is used here for simplicity of explanation and topology representation. However, the presented methodology can be readily applied to more realistic propagation models, e.g., fading channels.

Despite its simplicity and the assumption that each cell contains a sensor node, this scenario captures the characteristic features (multi-hop routing and all to one transmission paradigm) of actual WSN deployments and allows to study the interplay between data gathering and compressive sensing.

VI. Data Gathering Protocols

In this section we present the data gathering protocols that will be considered for the investigation of the benefits of CS when used in multi-hop WSNs. As pointed out in Section II, there is a well studied line of research on the application of CS to data gathering in wireless networks. Previous studies however adapted the routing technique or the data transmission phase so as to take full advantage of CS. What we do here is different as we pick a distributed WSN and consider the usual data gathering paradigm where sensors forward the packet(s) they receive along shortest paths towards the sink. This occurs in a completely unsynchronized and distributed manner, without knowledge about the correlation structure of the data and without knowing how it is processed at the sink through CS. Thus, our aim is to assess whether CS provides performance benefits with respect to standard schemes even in such distributed and unsynchronized network scenarios.

In what follows we present two schemes: the first is a standard geographical routing protocol, whereas the second is the same protocol in terms of routing, but it exploits CS for data recovery at the sink. We then characterize the structure of the $\mathbf{\Phi}$ matrix (see Section III) which is determined by the routing policy.

Data gathering protocols. To simplify the investigation and to pinpoint the fundamental performance tradeoffs, in this first study we neglect channel access considerations (i.e., collisions, transmission times, etc.). Also, we assume a unit cost for each packet transmission and we ignore processing overhead at the nodes, as it is expected to be cheap compared to the cost of packet transmission.

P1. *Random sampling (RS):* this is the simplest protocol that we consider. In this case, each node becomes a source with probability $P_T = M/N$, which was varied in the simulations to obtain tradeoff curves for an increasing transmission overhead. On average, M nodes transmit a packet containing their own sensor reading. Each packet is routed to the sink following the path that minimizes the number of transmissions (as defined by our geographical routing approach). Along this path, the packet is not processed but simply forwarded. The cost of delivering a single packet to the sink is given by the number of hops that connect the originating node to the data gathering point. The signal is reconstructed by interpolation of the collected values according to the method in [26].

P2. *Random sampling with CS (RS-CS):* this protocol is similar to RS. As above each node becomes a source with probability $P_T = M/N$. Again, each of these source nodes transmits a packet containing the reading of its own sensor. As this packet travels towards the sink, we combine the value contained therein with that of any other node that is encountered along the path. Specifically, let v_i^m with $i = 1, 2, ..., \ell_m$ be the readings of the sensors along the path from node m to the sink, where v_1^m is the reading of the node itself and ℓ_m is the length of the path. Node m sends a packet containing the value $y_1^m = \alpha_1 v_1^m$ as well as the combination coefficient α_1, where α_1 is a value chosen uniformly at random either from $(0, 1]$ or from the set $\{-1, +1\}$.[3] The next node along the path will update the transmitted value and send out $y_2^m = y_1^m + \alpha_2 v_2^m$ where α_2 is again a random value. Also the coefficient α_2 is included in the data packet along with with α_1. We proceed with these random combinations, where in general node $i+1$ sends out

$$y_{i+1}^m = y_i^m + \alpha_{i+1} v_{i+1}^m \,, \tag{7}$$

until the packet finally reaches the sink. The sink extracts $y_{\ell_m}^m = \sum_{i=1}^{\ell_m} \alpha_i v_i^m$, together with the vector of α coefficients that were used along the route. These coefficients, according to the CS formalism in Section III, form the mth row of matrix $\mathbf{\Phi}$, referred to as φ_m. Note that some optimizations are possible. First, if we know in advance the network topology, we can assign combination coefficients at setup time to all nodes, rather than including them in the packets. We can further use the same pseudo-random number generator at the nodes and the sink and synchronize the seeds. However, all of this goes beyond the scope of this paper and we do not focus on how to optimize the control overhead of CS.

A few observations are in order. When we use CS at the sink, we receive packets carrying more valuable information than in the plain forwarding case. The received values are *linear random* combinations of the readings of several sensor nodes. For example, considering RS-CS, when the sink receives the mth packet it can build a system of the form

$$\mathbf{y} \stackrel{def}{=} \begin{pmatrix} y_{\ell_1}^1 \\ y_{\ell_2}^2 \\ \vdots \\ y_{\ell_m}^m \end{pmatrix} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_m \end{pmatrix} \mathtt{vec}(\mathbf{X}) = \mathbf{\Phi}\mathtt{vec}(\mathbf{X}) \,, \tag{8}$$

[3]The implications of the selection of the set to use are discussed in Section VII.

where the $y^r_{\ell_r}$ with $r = 1, 2, \ldots, m$ are the combined values that were received by the sink in the packet that traversed the rth path, $\mathbf{X}$ is the input 2D signal, $\mathbf{\Phi}$ is an $m \times N$ matrix whose generic row r, φ_r, contains the vector of coefficients α included in the packet. Note that, in general, some of these coefficients might be equal to zero. Specifically, the node in cell (i, j) of the 2D grid can only contribute to entry $(i-1)K + j$ of vector φ_r (see also the ordering shown in the appendix in (13)). Thus, the combination coefficient in position $(i-1)K+j$ of φ_r, with $i, j = 1, 2, \ldots, K$, is non-zero if and only if node (i, j) was included in the path followed by the rth packet and is set to zero otherwise.[4] Hence, matrix $\mathbf{\Phi}$ highly depends on the *network topology* and on the *selected routing rules* as each of its rows will have non-zero elements only in those positions representing nodes that were included in the path followed by the corresponding packet.

Note that (8) is a system of linear equations that is in general ill-posed (as $m \leq M$ and M is expected to be smaller than N). At the sink, we know vector $\mathbf{y}$ and matrix $\mathbf{\Phi}$ and we need to find the 2D input signal $\mathbf{X}$. We can now use the derivations in the Appendix and rewrite $\mathbf{y} = \tilde{\mathbf{\Phi}}\mathtt{vec}(\mathbf{S})$ which is solved for $\mathtt{vec}(\mathbf{S})$ using standard compressive sensing tools for the 1D case [27], thus finding the sparsest $\mathtt{vec}(\mathbf{S})$ that verifies the system, referred to here as $\mathbf{S}^\star$. $\mathbf{S}^\star$ is finally used to reconstruct $\mathbf{X}$, i.e., $\mathbf{X}^\star = \mathbf{\Psi}\mathbf{S}^\star\mathbf{\Psi}^T$ (see also (12) in the appendix).

Characterization of the routing matrix $\mathbf{\Phi}$. According to our network model, the nodes that transmit their packet to the sink are chosen at random. As said above, every row φ_j of $\mathbf{\Phi}$ represents a path from a given sensor to the sink and each forwarding node in this path contributes with a non zero coefficient. We characterize the sparsity ν_j of φ_j counting the number of elements in this row that differ from zero: $\nu_j = \sum_{i=1}^{N} \mathbb{1}\{\alpha_i^j \neq 0\}$, where α_i^j is the ith entry of vector φ_j and $\mathbb{1}\{E\}$ is the indicator function, which is 1 when event E is true and zero otherwise. ν_j is the cost, in terms of number of transmissions, for sending the jth packet to the sink. With the network scenario in Section V it is easy to see that, for any source node in the network, the number of transmissions required for its packet to reach the sink is $O(\sqrt{N})$. Hence, the total cost for the transmission of M packets is $O(M\sqrt{N})$. As an example, for a network with $N = 400$ nodes the cost of delivering a packet to the sink is $\simeq 4.5$ transmissions, which is close to $\sqrt{N}/4$. The sparsity of φ_j directly translates into the sparsity of $\mathbf{\Phi}$ that, in turn, affects the *coherence* between the matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$. In the literature, the concept of coherence (or its dual, called *incoherence*) between these two matrices is directly related to the effectiveness of the CS recovery phase and is well defined when they are orthonormal. Specifically, the routing matrix $\mathbf{\Phi}$ and $\mathbf{\Psi}$ must be incoherent for CS to work properly [2].

In our settings, however, $\mathbf{\Phi}$ is built on the fly according to the routing topology, whereas $\mathbf{\Psi}$ is obtained according to any of the transformations T1–T4 that we discussed in Section IV. In the literature the concept of coherence is not defined for non-orthogonal matrices. However, according to the rationale in [2], [13] a quantity that is strictly related to the incoherence can be computed as follows. Roughly speaking, incoherence between two matrices means that none of the elements of one matrix has a sparse representation in terms of the columns of the other matrix (if used as a basis). Put differently, two matrices are highly coherent when each element of the first can be represented linearly combining a small number of columns of the second. Hence, to characterize the incoherence we first project each row of $\mathbf{\Phi}$ into the space generated by the columns of $\mathbf{\Psi}$. After this, we take the sparsest projections obtained in this space as an indication of the incoherence. Formally, we have:

$$\zeta_j = (\mathbf{\Psi}^T\mathbf{\Psi})^{-1}\mathbf{\Psi}^T\varphi_j^T \ , \tag{9}$$

where φ_j is the jth row of $\mathbf{\Phi}$ and ζ_j is the vector of coefficients corresponding to its projection on the space generated by the columns of $\mathbf{\Psi}$. A measure of the incoherence is then obtained as

$$I(\mathbf{\Phi}, \mathbf{\Psi}) = \min_{j=1,\ldots,N} \left[\sum_{i=1}^{N} \mathbb{1}\{\beta_i^j \neq 0\} \right] \in \left[1, N\right] \ , \tag{10}$$

where β_i^j is the ith entry of vector ζ_j.

In Fig. 4 we show the incoherence, obtained from (10), for the four transformation methods T1–T4 and for the following matrices $\mathbf{\Phi}$: R1) $\mathbf{\Phi}$ is built according to the CS routing protocol that we explained above, picking random coefficients in $\{-1, +1\}$, R2) $\mathbf{\Phi}$ is built as in case R1, picking random coefficients in $(0, 1]$, R3) $\mathbf{\Phi}$ has all coefficients randomly picked in $\{-1, +1\}$ and R4) $\mathbf{\Phi}$ has coefficients uniformly and randomly picked in $(0, 1]$. As can be deduced from the results of [9], cases R3 and R4 are near optimal in terms of projections of the measurements and can be built through a pre-distribution of the data (that in a multi-hop WSN is in general demanding in terms of number of transmissions).

From this plot we see that the DCT transformation (T1) has a high incoherence with respect to all of the considered routing matrices. The remaining transformations T2–T4 all perform similarly and give satisfactory performance only for cases R3 and R4, whereas for random projections

[4] Given this, we see that setting an entire column of the matrix to zero, say column $c = (i-1)K + j$ for given i and j, means that we completely ignore the contribution of the node placed in cell (i, j). This happens when none of the m received packets passes through this node while being routed to the sink.

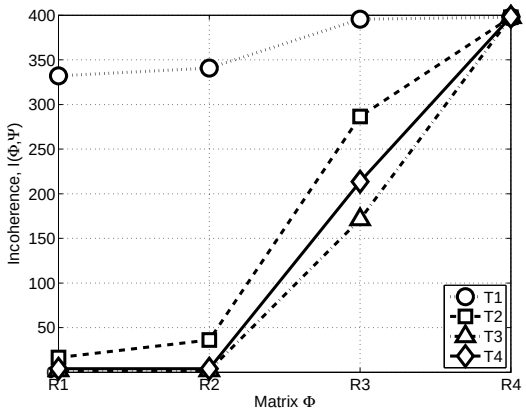

Fig. 4. Incoherence $I(\mathbf{\Phi}, \mathbf{\Psi})$ between the routing matrix $\mathbf{\Phi}$, cases R1–R4, and the transformation matrix $\mathbf{\Psi}$, transformations T1–T4. The maximum value for $I(\mathbf{\Phi}, \mathbf{\Psi})$ equals the number of nodes in the network, $N = 400$.

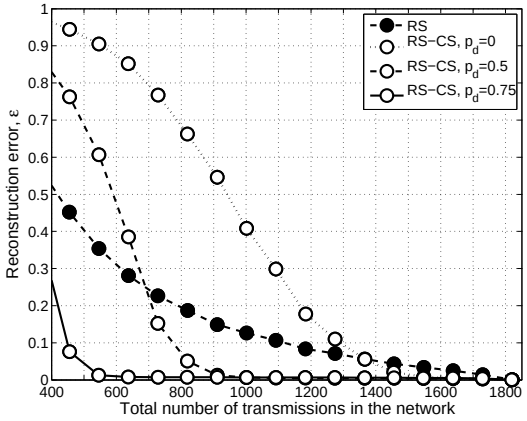

Fig. 5. Reconstruction quality ε as a function of the total number of packets transmitted in the network: comparison between RS and RS-CS for synthetic signals and different values of p_d.

obtained through the actual routing scheme they are highly coherent to $\mathbf{\Phi}$. This has strong negative implications on the CS recovery performance and will be discussed in the following section.

VII. Results

In this section we discuss the results we obtained by simulating the RS and RS-CS data gathering schemes for synthetic and real signals. The metric of interest is the reconstruction quality at the sink, which is defined as follows. Given a 2D input signal $\mathbf{X}$, a matrix $\mathbf{\Phi}$ and a vector $\mathbf{y}$ (containing the received values that are linear combinations of the sensor readings in the network) we have that $\mathbf{y} = \mathbf{\Phi}\mathtt{vec}(\mathbf{X})$. This system, that in general is ill-posed (as $M \leq N$), is solved for $\mathtt{vec}(\mathbf{X}) = \mathtt{vec}(\mathbf{\Psi S \Psi}^T)$ either through norm one [2] or smoothed zero norm [17] minimization. These methods efficiently find the sparsest $\mathbf{S}$, referred to as $\mathbf{S}^\star$, that verifies the previous system.[5] If $\mathbf{X}^\star = \mathbf{\Psi S}^\star \mathbf{\Psi}^T$ is the solution found for this system and $\mathbf{X}$ is the true input signal, the *reconstruction error* is defined as

$$\varepsilon = \frac{\|\mathtt{vec}(\mathbf{X}) - \mathtt{vec}(\mathbf{X}^\star)\|_2}{\|\mathtt{vec}(\mathbf{X})\|_2} \,. \tag{11}$$

A. Results for Synthetic signals

In Fig. 5 we show the reconstruction error ε as a function of the total number of packets sent in the network for RS and RS-CS. For this plot we considered a low-pass signal with $p_{\text{low}} = \sqrt{N}/2 + 1$ and $p_{\text{high}} = \sqrt{N}$, with $N = 400$. Also, we considered three values of $p_d \in \{0, 0.5, 0.75\}$ so as to vary the sparseness of the signal. As a first observation, random sampling performs nicely for low-pass signals. Nevertheless, a perfect reconstruction of the sensed signal at the sink requires the transmission of a large number of packets (up to 1800). When the signal is sufficiently sparse ($p_d \geq 0.5$) CS outperforms standard data gathering schemes, requiring less than half the packet transmissions (about 900) to achieve the same recovery performance. We noticed that values of ε larger than 0.3 always led to very inaccurate reconstructions of the original signal. Fig. 5 was obtained using L_1 minimization and combination coefficients in the set $\{-1, +1\}$. However, we obtained similar performance using smoothed L_0 norm and/or coefficients in the set $(0, 1]$. Note that using the set $\{-1, +1\}$ allows for reduced overhead as, in practical implementations, a single bit suffices to transmit each coefficient.

For high-pass signals the performance of CS is unvaried for the same degree of sparseness. This is expected as CS recovery operates in the frequency domain and is only affected by the number of non-zero frequency components and not by their position. Clearly, RS with the considered interpolation technique is not appropriate for high-pass signals, in which case it shows poor recovery performance.

As a consequence, CS-RS shows good recovery performance for synthetic signals as, by construction, the DCT transformation effectively sparsifies the signal and this transformation is incoherent with respect to the routing matrix $\mathbf{\Phi}$ (see Fig. 4).

B. Results for Real Signals

In Fig. 6 we show the reconstruction error ε as a function of the total number of packets sent in the network

[5] We found that these two methods are nearly equivalent in terms of quality of the solution, although the zero norm is simplest and faster. This might be important for practical implementations.

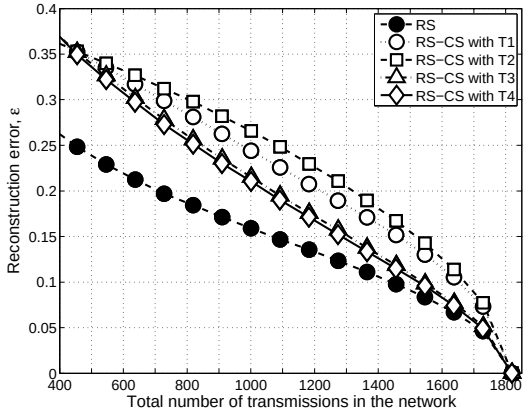

Fig. 6. Reconstruction error ε *vs* total number of packets transmitted in the network: comparison between RS and RS-CS (for transformations T1–T4) for the real signals in Section IV.

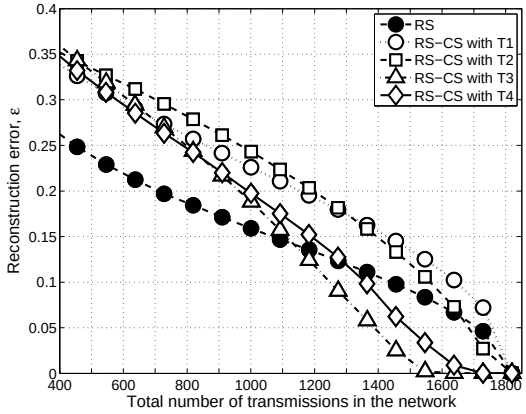

Fig. 7. Reconstruction error ε *vs* total number of packets transmitted in the network: comparison between RS and RS-CS (for transformations T1–T4) when a pre-distribution of the data is allowed so that the routing matrix $\mathbf{\Phi}$ approaches that of case R4 of Section VI.

for RS and RS-CS. The sensed signals belong to the data sets presented in Section IV. In this case, differently from the case of synthetic signals, RS-CS does not outperform RS, even though the performance of the two methods is very close. The reason for this is twofold. First, the considered transformations T1–T4 sparsify the real signals only up to 70% (see Section IV). This is mainly due to the characteristics of the signals and to the small size of the sample set. Second, the transformations with the best performance in terms of sparsification have a high coherence with respect to the routing matrix of CS-RS. Hence, while the sparsification performance may suffice, matrix $\mathbf{\Phi}$ (routing) does not have the required properties in terms of coherence for CS to perform satisfactorily.

In fact, for good recovery performance CS needs a good transformation in terms of sparsification. Also, transformation and routing matrices must be incoherent. From Figs. 2, 4 and 6 we see that transformations T3 and T4 are the most suitable to sparsify the considered real signals and this allows them to perform better than T1 and T2 (even though they perform poorly in terms of incoherence, see Section VI). In addition, although T2 can sparsify real signals better than T1 (Fig. 2), the latter performs better than T2 in terms of transmission cost *vs* error reconstruction (Fig. 6), since it has better incoherence properties $I(\mathbf{\Phi}, \mathbf{\Psi})$ (Fig. 4).

Finally, in Fig. 7 we accounted for a pre-distribution phase of the data so that matrix $\mathbf{\Phi}$ is as close as possible to that of case R4 of Section VI (we verified that case R3 gives similar performance). In this case, CS-RS outperforms RS as T1 and T2 provide a sparse representation of the signal and the routing matrix is sufficiently incoherent with respect to these transformations. However, this pre-distribution phase (which is similar to that proposed in [9]) has a high transmission cost for static networks, which is ignored in Fig. 7. In mobile networks, the pre-distribution could take advantage of the nodes' mobility so as to decrease the cost associated with the construction of $\mathbf{\Phi}$. This is not dealt with in this paper and is left for future research.

VIII. Conclusions

In this work we studied the behavior of CS when used jointly with a routing scheme for recovering two types of signals: synthetic ones and real sensor data. We showed that for the synthetic signal the reconstruction at the sink node is enhanced when applying CS, whereas the application of CS for real sensor data is not straightforward. Thus, as a next step of our ongoing research, we intend to further investigate which signal representation and routing allows CS to outperform random sampling in realistic WSN deployments. This requires to jointly investigate the design of the two matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$, since the sparsity requirements and the incoherence between routing and signal representation have to be met.

References

[1] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 4036–4048, 2006.

[2] E. Candès 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.

[3] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[4] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, "Compressive Sensing for Networked Data: a Different Approach to Decentralized Compression," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, 2008.

[5] J. Liu, M. Adler, D. Towsley, and C. Zhang, "On optimal communication cost for gathering correlated data through wireless sensor networks," in *MOBICOM*, 2006.

[6] R. Baraniuk, "Compressive Sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[7] J. Romberg, "Imaging via Compressive Sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 14–20, 2008.

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

[9] M. Rabbat, J. Haupt, A. Singh, and R. Novak, "Decentralized Compression and Predistribution via Randomized Gossiping," in *IPSN*, 2006.

[10] G. Shen, S. Y. Lee, S. Lee, S. Pattem, A. Tu, B. Krishnamachari, A. Ortega, M. Cheng, S. Dolinar, A. Kiely, M. Klimesh, and H. Xie, "Novel distributed wavelet transforms and routing algorithms for efficient data gathering in sensor webs," in *ESTC*, 2008.

[11] M. Duarte, M. Wakin, and R. Baraniuk, "Wavelet-domain compressive signal reconstruction using a Hidden Markov Tree model," in *ICASSP*, 2008.

[12] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed Network Monitoring for IP and All-optical Networks," in *IMC*, 2007.

[13] M. Duarte, M. Wakin, D. Baron, and R. Baraniuk, "Universal Distributed Sensing via Random Projections," in *IPSN*, 2006.

[14] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk, "Distributed Compressed Sensing of Jointly Sparse Signals," in *Thirty-Ninth Asilomar Conf. on Signals, Systems and Computers*, 2005.

[15] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari and A. Ortega, *Compressed Sensing and Routing in Multi-Hop Networks*, Technical Report, University of Southern California, 2009.

[16] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[17] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L0 norm," *IEEE Trans. on Signal Processing*, 2009, Accepted.

[18] A. Jindal and K. Psounis, "Modeling Spatially Correlated Data in Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 466–499, 2006.

[19] "MIT Wireless Network Coverage," Last time accessed: January 2009. [Online]. Available: http://nie.chicagotribune.com/activities_120505.htm

[20] T. Kamakaris and J. V. Nickerson, "Connectivity maps: Measurements and applications," in *Hawaii International Conf. on System Sciences*, 2005.

[21] "EPFL LUCE SensorScope WSN," Last time accessed: January 2009. [Online]. Available: http://sensorscope.epfl.ch/

[22] "Tropical Rainfall Measuring Mission," Last time accessed: January 2009. [Online]. Available: http://trmm.gsfc.nasa.gov

[23] "Partnership for Interdisciplinary Studies of Coastal Oceans," Last time accessed: January 2009. [Online]. Available: www.piscoweb.org

[24] "ENEA: Ente Nuove Tecnologie e l'Ambiente," Last time accessed: January 2009. [Online]. Available: www.enea.it

[25] Napler Addison, *The Illustrated Wavelet Transform Handbook*. Taylor & Francis, 2002.

[26] D. T. Sandwell, "Geophysical Research Letters," *Biharmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data*, vol. 14, no. 2, pp. 139–142, 1987.

[27] E. J. Candes, M. B. Wakin, and S. Boyd, "Enhancing Sparsity by Reweighted ℓ_1 Minimization," *Journal of Fourier Analysis and Applications*, 2008, Accepted.

[28] G. R. Belitskii and Yu. I. Lyubich, *Matrix norms and their applications*. Birkhäuser, 1988.

APPENDIX

In this appendix, we review a known method from image processing to generalize the CS theory in Section III to 2D signals, as those gathered by the WSN of Section V. Accordingly, the input signal is a $K \times K$ square matrix $\mathbf{X}$ with $N = K^2$ elements. Element (i, j) of this matrix, $x(i, j)$, is the value sampled by the sensor placed in cell (i, j) of the sensor grid. We assume that the 2D signal $\mathbf{X}$ is sparse under a given transformation. Thus, $\mathbf{X}$ can be written as

$$\mathbf{X} = \mathbf{BSA} \ , \tag{12}$$

where $\mathbf{B}$ and $\mathbf{A}$ are two non singular matrices and $\mathbf{S}$ is a $K \times K$ matrix representing the signal in the transformation domain.

In what follows, we use tools from linear algebra to reformulate the 2D problem as an equivalent 1D problem. It is worth noting that this transformation does not lose any information and preserves the correlation among sensed values in the 2D space.

Now we define a $\mathtt{vec}(\cdot)$ function, transforming a $K \times K$ matrix into a vector of length N (through a reordering of the matrix elements)

$$\begin{aligned} \mathtt{vec}(\mathbf{X}) \quad = \quad & \big(x(1,1), \ldots, x(k,1), x(1,2), \ldots, x(k,2), \\ & \ldots, x(1,k), \ldots, x(k,k)\big)^T \ . \end{aligned} \tag{13}$$

As explained in Section VI, the values that we collect at the sink can be represented through a vector $\mathbf{y}$ of $M < N$ elements. They are linear combinations of the sensor readings represented by the matrix $\mathbf{X}$ of size $K \times K$, and thus $\mathbf{y} = \mathbf{\Phi}\mathtt{vec}(\mathbf{X})$. The $M \times N$ matrix $\mathbf{\Phi}$ contains the combination coefficients that are picked at random according to a given distribution. From linear algebra we know that the vector form of a given product among three matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{S}$ can be rewritten as [28]

$$\mathtt{vec}(\mathbf{BSA}) = (\mathbf{A}^T \otimes \mathbf{B})\mathtt{vec}(\mathbf{S}) \ , \tag{14}$$

where $\otimes$ is the Kronecker product. Hence, using (12) and (14) we can write $\mathtt{vec}(\mathbf{X}) = (\mathbf{A}^T \otimes \mathbf{B})\mathtt{vec}(\mathbf{S})$. Using $\mathbf{y} = \mathbf{\Phi}\mathtt{vec}(\mathbf{X})$ we obtain $\mathbf{y} = \mathbf{\Phi}(\mathbf{A}^T \otimes \mathbf{B})\mathtt{vec}(\mathbf{S})$ that, defining $\tilde{\mathbf{\Phi}} = \mathbf{\Phi}(\mathbf{A}^T \otimes \mathbf{B})$, can be rewritten as

$$\mathbf{y} = \tilde{\mathbf{\Phi}}\mathtt{vec}(\mathbf{S}) \ , \tag{15}$$

where $\mathbf{y}$ is the vector containing the received (combined) values and $\mathtt{vec}(\mathbf{S})$ is a column vector of length N containing the input signal in the transformation domain. Given (15) we can recover the sparse signal $\mathtt{vec}(\mathbf{S})$ using the solvers developed for standard CS theory in 1D.