Scaling Network-based Spectrum Analyzer with Constant Communication Cost

Youngjune Gwon
Harvard University
gyj@eecs.harvard.edu

H. T. Kung
Harvard University
kung@harvard.edu

Abstract—We propose a spectrum analyzer that leverages many networked commodity sensor nodes, each of which samples its portion in a wideband spectrum. The sensors operate in parallel and transmit their measurements over a wireless network without performing any significant computations such as FFT. The measurements are forwarded to the backend of the system where spectrum analysis takes place. In particular, we propose a solution that compresses the raw measurements in a simple random linear projection and combines the compressed measurements from multiple sensors in-network. As a result, we achieve a substantial reduction in the network bandwidth requirement to operate the proposed system. We discover that the overall communication cost can be independent of the number of sensors and is affected only by sparsity of discretized spectrum under analysis. This principle founds the basis for a claim that our network-based spectrum analyzer can scale up the number of sensor nodes to process a very wide spectrum block potentially having a GHz bandwidth. We devise a novel recovery algorithm that systematically undoes compressive encoding and in-network combining done to the raw measurements, incorporating the least squares and ℓ_1-minimization decoding used in compressive sensing, and demonstrate that the algorithm can effectively restore an accurate estimate of the original data suitable for fine-grained spectrum analysis. We present mathematical analysis and empirical evaluation of the system with software-defined radios.

I. Introduction

A spectrum analyzer measures electrical signals present in a radio frequency (RF) channel and computes their magnitude across the channel frequency range. Spectral analysis provides characteristic information about a signal such as dominant frequency, power, bandwidth, and distortion, which are not discernible in the time domain. Conventional spectrum analyzers are specialized hardware featuring a fast RF tuner that sweeps through an analysis range or a dedicated Fast Fourier Transform (FFT) device that computes frequency response in real-time (known as FFT spectrum analyzers).

A key enabling element for emerging cognitive radio [1] and dynamic spectrum assignment (DSA) [2] is spectrum sensing. Especially, the ability for accurate, fine-grained analysis of a target spectrum will be a critical requirement. Technically, a spectrum analyzer can provide ideal assistance. A typical spectrum analyzer, however, is a bulky equipment, and its analysis is restricted to a geographical location of the equipment. Hence, it is impractical as a spectrum sensor for cognitive radio on small mobile devices. Also, it would be a prohibitively expensive solution, considering that the equipment costs tens of thousands of dollars or more.

Departed from the conventional approach, we envision a spectrum analyzer that operates over a network interconnecting many wireless sensor nodes and leverages their measurements to construct accurate power spectral estimates of a wideband spectrum, which can potentially span hundreds of MHz to a few GHz in bandwidth. The sensors are geographically distributed, work in parallel measuring their respective regions, and participate in a large-scale in-network data collection scheme. We assume that the sensors are simple, general-purpose commodity devices and need not be dedicated for our spectrum analyzer (they may be user devices). Our aim is different from contemporaries such as *SpecNet* [3] whose main contribution is to provide an API to use networked conventional spectrum analyzer equipments remotely. In this paper, we focus on the acquisition of distributed RF measurements, compressive sampling, and manipulation of the compressed data for efficient transport to the distant system backend. We describe a novel data recovery algorithm to systematically undo all compressive and in-network processing done to the original measurements.

A. Challenges

What is the communication cost of a fine-grained spectrum analysis? In digital signal processing, the Nyquist-Shannon sampling theorem [4] dictates ideal recovery of a measured signal, which needs to be sampled at least twice as frequently of its bandwidth. For example, a signal with 1-MHz bandwidth has the Nyquist rate $f_s^\star = 2 \times 1$ MHz, which means that any two successive measurements should be spaced in time $T_s^\star = 1/f_s^\star = 1/(2 \text{ MHz}) = .5 \mu\text{sec}$ to prevent a loss of information. For this case, the Nyquist sampling results in a raw data rate of $2 \times 10^6 \times sample\ size$ bits/sec.

The majority of wireless technologies today is based on digital I-Q (*in-phase* and *quadrature*) modulation that represents each sample in complex numbers. Assuming the use of a 32-bit complex float data type—which could be 64-bit long in many machines, uninterrupted plain Fourier analysis of a 1-MHz spectrum needs to keep up with a 64-Mbps data stream.

Fast Fourier Transform (FFT), an efficient numerical algorithm to compute Discrete Fourier Transform (DFT), is at the heart of spectrum analyses. FFT running on a batch of N samples in time produces N frequency components where each component can be converted into a spectral energy

around the corresponding frequency index in the spectrum. Spectrum analyzers designate *resolution bandwidth* $b = B/N$ to refer the spacing between two consecutive frequency components. Resolution bandwidth determines the granularity of an analysis—that is, the *smaller* the resolution bandwidth, the *finer* the granularity.

The computational complexity of an N-point FFT is in $O(N \log N)$ time, and a few kHz resolution bandwidth for the 1-MHz band will result N to be as large as 512 (FFT commonly performed in a power of 2). Sometimes, creating a new snapshot may not require full N measurements per every analysis cycle if some prior knowledge on a channel or signals is given. The real challenge would be analyzing a wideband spectrum of several orders of magnitude larger than the 1-MHz example. A naïve solution could lead to a network bandwidth requirement as large as hundreds of Gbps, which is much beyond what a wireless networking technology can support in the foreseeable future.

B. Problem Statement

Imagine sensor nodes in a wireless network with base station as illustrated in Fig. 1. We use a term "system backend" or simply "system" to designate a control entity responsible to run the spectrum analyzer daemon. We denote "sensor" a node that performs sampling in the time domain and provides measurements to the system, and there are P sensors employed by the system. Suppose that a spectrum under analysis has a total bandwidth B_{tot}. The system partitions the spectrum into J subchannels with bandwidths, $B_1, \ldots, B_J$, such that $B_{tot} = \sum_{i=1}^{J} B_i$. For simplicity, assume adjacent subchannels do not overlap. (It is relatively straightforward to extend our framework for overlapping scenarios.) There is a communication protocol $\mathcal{P}$ used between the system and a node to administer the node as one of the sensors. The system dispatches a measurement assignment, denoted in (f_l, B_l), to sensor node l. This means that sensor l should tune to f_l, the center frequency of the assigned subchannel, and start sampling according to its bandwidth B_l. Assuming equipartition of the spectrum (*i.e.*, $B_i = B$ for all i), sensor l yields a vector of time-domain measurements $\mathbf{x}_l$ periodically where $\dim(\mathbf{x}_i) = N$.

In this system model, the sensor measurements can be first transmitted to the base station before delivered to the system backend. If there are exactly $P = J$ sensor nodes with each measuring one of J subchannels uniquely, we have a total of $L = \sum_{i=1}^{J} \dim(\mathbf{x}_i) = J \times N$ measurements constituting $\mathbf{x}$ (all $\mathbf{x}_i$'s) for the entire spectrum. Our objective is to minimize the total number of measurements L transmitted to the system well under $J \times N$ samples per each cycle.

We clarify that there are two types of nodes, a *sensor* or *network* node, based on functionalities in the system. Sensor nodes produce measurements in the time domain and transmit either directly to the base station, for example, as depicted by 'white' circles in Fig. 2(a) or to a network node that can relay the measurements for others as in Fig. 2(b). We assert that sensor nodes are end-nodes and *not* meant to forward
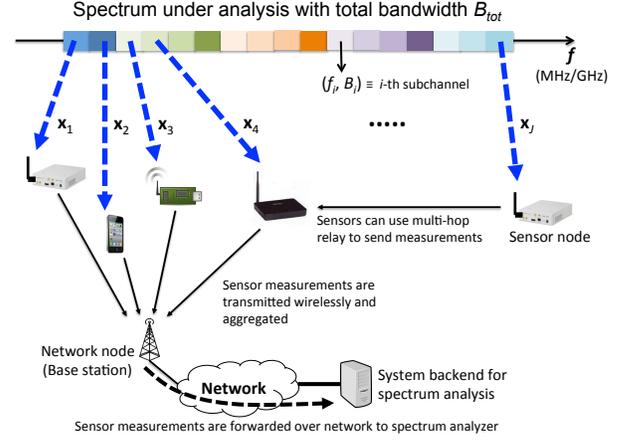
Fig. 1. A wideband spectrum (*e.g.*, $B_{tot} \approx$ a few GHz) under analysis is partitioned for collective measurement efforts by multiple sensor nodes. Spectral analysis on the measured data takes place at the potentially distant system backend.

other nodes' measurements. On the other hand, network nodes can forward the measurements received from others, using multi-hop relays to the base station as represented by 'black' circles in Fig. 2(b). Network nodes play a role that helps save the uplink bandwidth to the base station by combining or encoding multiple sets of measurements received from other nodes to eventually reduce the total measurements to the base station. Under this model, our goal is to recover *almost exact* (*i.e.*, incurring an error below some small threshold ϵ) frequency response of the spectrum under analysis as if the frequency response were constructed by running FFT on Nyquist sampling.

We model the communication cost of our network-based spectrum analyzer as the total number of measurements received by (or transmitted to) the base station, consisting of all measurements directly from sensor nodes and through network nodes. Therefore, we wish to:

$$\arg\min_{\theta} \sum_{i=1}^{J} \dim(\mathbf{y}_i = \theta(\mathbf{x}_i)) \quad \text{s.t.} \ \|X(f_k) - \hat{X}(f_k)\|_{\ell_2} \leq \epsilon,$$

where $\theta(.)$ is a size-reducing function we seek, which makes $\dim(\mathbf{y}_i) \ll \dim(\mathbf{x}_i) \ \forall i \in \{1, \cdots, J\}$. $X(.)$ is the actual frequency response of $\mathbf{x}$, and $\hat{X}(.)$ the estimate reconstructed from the compressed measurements $\mathbf{y}_i$. f_k is a frequency component index in $J \times N$-point FFT, where $0 \leq k < J \times N$. The constraint is a performance requirement stipulating the accuracy of the reconstructed frequency response such that the Euclidean (ℓ_2) norm of the frequency response error should be bound within some small constant ϵ.

C. Our Contribution

The main contribution of this paper is the practical, network-based wideband spectrum analyzer system and in-network data processing and recovery schemes. In particular, we propose the use of compressive sensing [5] in each sensor node's sampling operation with pre-distributed sensing matrices. We

have developed a method that enables a network node to *mix* compressed measurements from multiple, arbitrary sensors. Our mixing operation is just arithmetic addition, consequently imposing little computational overhead to network nodes. The data from the sensors are *summed*, and the combined data size is held *constant* while propagating up the network nodes. We have developed a special decoding algorithm for our data mixing scheme and tested that the algorithm can effectively separate the sum and recover the data approximately equivalent to the original samples.

We highlight how the algorithm works:

1) Preparation phase. Identify the location of dominant variables that correspond to the largest eigenvalues of the sample covariance matrix computed from measurements for each sensor/subchannel. In this phase, network nodes simply forward received measurements without mixing;

2) Initial approximation. Solve for an estimate of the dominant variables from an *overdetermined* system constructed by considering only the dominant variables, using the least squares;

3) Iterative refinements. Refine the estimate from the initial approximation by solving an *underdetermined* system via the ℓ_1-minimization decoding used in compressive sensing for remaining (non-dominant) variables of each sensor. This refinement is iterated for all sensors.

We find that the *locations* of dominant variables (not their *values*) in an RF subchannel can be stable over time. For such stationary subchannels, the preparation phase needs to be performed infrequently, and the network cost of preparation phase can thus be ignored. It is reasonable to assume that the total number of dominant variables is bounded by a constant and *independent* of the number of sensors employed by the system. We were able to verify this assumption empirically from our analysis on real-world RF spectrums containing modulated signals (*e.g.*, cellular, UHF/DTV broadcast channels). Network nodes are then assured to mix compressed measurements to the maximal degree in the latter two phases, and the communication cost can subsequently be held constant even if the system admits more sensor nodes. Finally, the iterative process in the algorithm reduces computational complexity by at least a factor of P, the total number of active sensor nodes in the system, and through relaxing variables for one sensor at a time.

D. Related Work

This paper addresses similar problems considered by distributed compressed sensing [6], compressive wireless sampling [7], data gathering for sensor networks [8], and collaborative spectrum sensing [9], [10]. In particular, Polo *et al.* [10] introduced an early wideband spectrum sensing framework that uses multiple, distributed cognitive radios to simultaneously sense a block of radio subchannels with compressive measurements. Wang *et al.* [11] extended the framework, describing a complete system and decoding schemes that rely on the combined use of analog-to-information conversion (AIC) hardware and the SOMP algorithm [6]. Neither work focused on manipulating compressed measurements directly
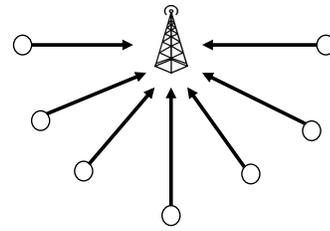
(a) Sensor nodes (in white) transmit measurements directly to base station
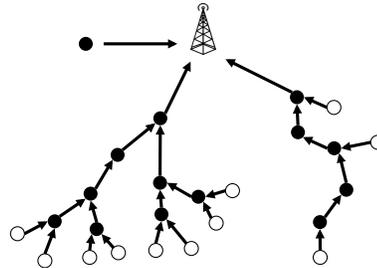
(b) Network nodes (in black) relay measurements of other network or sensor nodes

Fig. 2. In-network transmission of measurements

for further networking efficiency, which was critical for our case. The essence of our approach is to relieve sensor nodes from any significant computations other than linear projection. Similarly, network nodes operate in a care-*free* manner to combine compressed measurements by doing arithmetic additions only. Our recovery algorithm separates the mixed measurements and refine each set in an iterative procedure, which is another distinguishing remark for our work.

In recent work, Iyer *et al.* [3] proposed SpecNet, a network of conventional spectrum analyzers, promoting a means to share the expensive equipments and new ways of using them such as remote spectrum measurement and learning primary transmitter statistics in distant geographic regions. SpecNet, however, is not a spectrum analyzer *per se* but an API or user interface for remote usage, which is different from what this paper addresses.

II. PRELIMINARIES

A. Notation

A boldfaced variable indicates that it is a vector or matrix. We take the notational convention used in compressive sensing to denote a sampled, uncompressed signal $\mathbf{x} \in \mathbb{C}^N$, and its compressive measurements $\mathbf{y} \in \mathbb{C}^M$, where $M \ll N$. $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is a measurement or projection matrix used in compressive sensing. $\mathbf{x}$ has a sparse representation $\mathbf{s} \in \mathbb{C}^N$ in a basis $\mathbf{\Psi} \in \mathbb{C}^{N \times N}$ such that $\mathbf{x} = \mathbf{\Psi} \mathbf{s}$. We use $\mathbf{Q} \in \mathbb{C}^{N \times N}$ to represent the optimal sparsifying basis for $\mathbf{x}$. Superscripted H means Hermitian (conjugate) transpose. $\mathbb{E}[\cdot]$ is the expectation operator.

Subscripted variables represent that they belong to a group of same kinds. For example, $\mathbf{x}_i$ is the measurement of a signal

by sampling i-th subchannel, and $\mathbf{x}_k$ from k-th subchannel. A parenthesized subscripted index means a sorted position. For example, $\mathbf{s}_{(i)}$ is an i-th sorted vector among all $\mathbf{s}_i$'s according to their magnitudes. Tilde '~' above a vector variable means that the values of some elements in the vector are replaced by zeros. Tilde '~' above a matrix variable also designates the replacement by zeros. For a matrix, all elements in some columns or rows are replaced by zeros. Hat '^' above a variable means that it is an approximate (solution) of the original variable.

B. Compressive Sensing

Compressive sensing [5] is a recent signal processing technique that compresses data into a random linear combination stated in Definition 1. The recovery assumes a sparse structure present in the data. The exploitation of sparsity is a common theme behind many compression schemes, but compressive sensing leverages it differently by directly capturing K significant components of size-N data in M measurements without analysis (note $K < M \ll N$). Conventional compression schemes, however, would require an analysis of the data commensurate with N or greater in computational complexity before compression. Theorem 2 provides a lower bound of M, the number of measurements required for the ℓ_1-minimum compressive sensing decoder [12], [13] to achieve the exact data recovery with high probability. While there are other decoding strategies (*e.g.*, [14]) and they also work under our framework, we limit our discussion on sparse recovery techniques to the ℓ_1-minimization throughout this paper.

Definition 1: (Compressive Measurements) A linear coding scheme with an $M \times N$ measurement matrix $\boldsymbol{\Phi}$ for $M \ll N$ on an $N \times 1$ vector $\boldsymbol{x}$ produces $M \times 1$ compressive measurements, $\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{x}$.

Theorem 2: An $N \times 1$ $\boldsymbol{x}$ is K-sparse if there exists an $N \times N$ basis $\boldsymbol{\Psi}$ such that $\boldsymbol{x} = \boldsymbol{\Psi}\,\boldsymbol{s}$, and $\boldsymbol{s}$ has only $K < N$ nonzero elements. Provided that the matrix $\boldsymbol{\Phi}$ satisfies the Restricted Isometry Property and $M \geq c\,K\,\log\,(N/K)$ for some small constant c, an exact recovery of $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{x}$ can be achieved with high probability by ℓ_1-minimization decoding that chooses the solution minimizing ℓ_1-norm of $\boldsymbol{s}$:

$$min_{\boldsymbol{s} \in \mathbb{R}^N} \|\boldsymbol{s}\|_{\ell_1} \quad subject\ to\ \boldsymbol{y} = (\boldsymbol{\Phi}\boldsymbol{\Psi})\,\boldsymbol{s} \tag{1}$$

Proof: See Candès [12]. ■

C. Optimal Basis Decoding

For given N and K, the quality of compressive sensing decoding depends on M. More interestingly, the decoding is *incremental*—using a smaller M (than required for exact recovery) does not disqualify the decoded result entirely. In other words, the largest some number (corresponding to the smaller M) of decoded values should still be accurate. If desired accuracy is not met, one can increase M. However, this is not the only way to improve the decoding accuracy. If there exists another basis that can sparsify the data with *fewer* nonzero magnitudes, using such basis will also improve

the accuracy. Using such an optimally designed basis is the approach we take in this paper.

But how can one systematically find such an optimal basis? Consider $\mathbf{x}$ an $N \times 1$ random variable corresponding to a zero-mean (*e.g.*, sinusoids), complex-valued, and wide-sense stationary discrete random process. Its autocorrelation matrix is $\mathbf{R}_\mathbf{x} = \mathbb{E}[\mathbf{x}\,\mathbf{x}^\mathrm{H}]$:

$$\mathbf{R}_\mathbf{x} = \mathbb{E}\left[\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} [x_1^*\,x_2^*\,\cdots\,x_N^*] \right] \tag{2}$$

$\mathbf{R}_\mathbf{x}$ is real and symmetric, and the eigen-decomposition gives $\mathbf{R}_\mathbf{x} = \mathbf{Q}\,\boldsymbol{\Lambda}\,\mathbf{Q}^\mathrm{H}$, where columns of $\mathbf{Q}$ are eigenvectors of $\mathbf{R}_\mathbf{x}$, and $\boldsymbol{\Lambda}$ a diagonal matrix of eigenvalues. $\mathbf{Q}$ is orthogonal, thus $\mathbf{Q}^{-1} = \mathbf{Q}^\mathrm{H}$. What intrigues us is the representation $\mathbf{s} = \mathbf{Q}^\mathrm{H}\,\mathbf{x}$, because $\mathbf{R}_\mathbf{s} = \mathbb{E}[\mathbf{s}\,\mathbf{s}^\mathrm{H}] = \mathbb{E}[\mathbf{Q}^\mathrm{H}\,\mathbf{x}\,\mathbf{x}^\mathrm{H}\,\mathbf{Q}] = \mathbf{Q}^\mathrm{H}\,\mathbf{R}_\mathbf{x}\,\mathbf{Q} = \boldsymbol{\Lambda}$. In other words, the correlation matrix of $\mathbf{s}$ is diagonal, making all cross-correlation terms *zero*, which signifies that $\mathbf{s}$ is an *uncorrelated* representation of $\mathbf{x}$. So, there should be no statistical redundancy in the elements of $\mathbf{s}$, or $\mathbf{s}$ is the most compact, non-lossy representation for $\mathbf{x}$. This process is known as Kahrunen-Loève Transform (KLT) [15], and we call $\mathbf{Q}$ a KLT basis. We note that there is a computationally more efficient algorithm by Coifman and Wickerhauser [16], which can be used *online* for estimating or incrementally updating an optimal basis.

For compressive sensing, $\mathbf{Q}$ is an ideal candidate for $\boldsymbol{\Psi}$ used in decoding. To make this point clear, let us first consider the time-domain representation of a real-world wireless spectrum illustrated in Fig. 3. Carrier-modulated RF signals (sinusoids) do not appear sparse in time domain, thus it is customary to look for their frequency response by taking Fourier Transform. Discrete Fourier Transform (DFT) takes N samples in time domain to N samples in the frequency, computing the inner products of $\mathbf{x}$ with sampled complex sinusoids. In matrix form, it is $\mathbf{X} = \mathcal{F}\mathbf{x}$, where the DFT basis $\mathcal{F}$ has coefficients $f_{nk} = e^{-j2\pi nk/N}\ \forall n, k \in \{0, 1, \ldots, N-1\}$. $\mathcal{F}$ is *invertible*, so we also have $\mathbf{x} = \mathcal{F}^{-1}\mathbf{X}$.

In Fig. 4, we use the DFT basis to represent the time samples from Fig. 3, whereas in Fig. 5 we use the adaptive KLT basis. The KLT basis was estimated by averaging 10 prior measurement intervals—such estimation is the cost of using KLT than a *fixed* basis like DFT. In this particular example, KLT took approximately 10.5 times fewer nonzero components to describe the time samples than DFT (*i.e.*, $10.5\times$ sparsity gain). As a result, KLT basis can achieve a better accuracy of decoding despite taking substantially fewer measurements.

III. MINIMIZING NETWORK COMMUNICATION COST

The underlying principle for reducing the communication cost is to take advantage of a measured signal's sparsity and simple encoding of multiple, parallel measurements by the
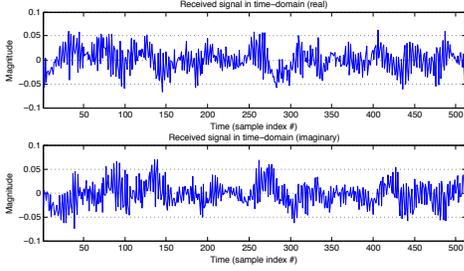
Fig. 3. Time-domain representation of a real wireless spectrum (source: 512 samples from 6-MHz bandwidth UHF Ch. 21 at f_c = 515 MHz measured in Cambridge, MA on Saturday May 21, 2011 11:03pm EST, using USRP2)
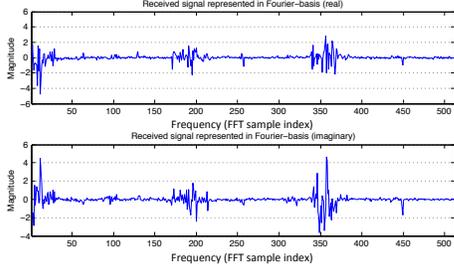
Fig. 4. Same data from Fig. 3 represented in Fourier basis

sensors. This section explains two approaches integrated in our system.

A. Exploiting Signal Sparsity with Compressive Measurements

Compressive sensing enables us to exploit a concealed sparse structure of the measured signal *blindly*, provided that there will be a suitable basis available for recovery. In the exploitation of sparsity, compressive sensing provides a key advantage for cases such as data collection in a mesh-styled wireless sensor network [8]. As discussed in the next section, multi-hop relays can gracefully embrace compressive measurements with a substantially lower communication cost for large-scale data collection. In contrast, conventional compression schemes such as joint entropy coding [17] require solving a complex optimization problem at nodes.
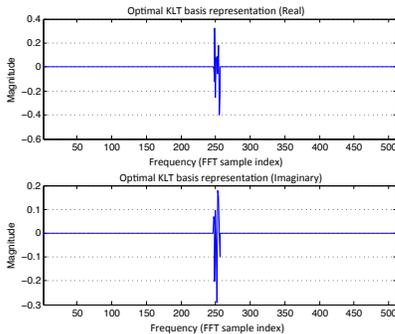
Fig. 5. Same data from Fig. 3 represented in estimated KLT basis
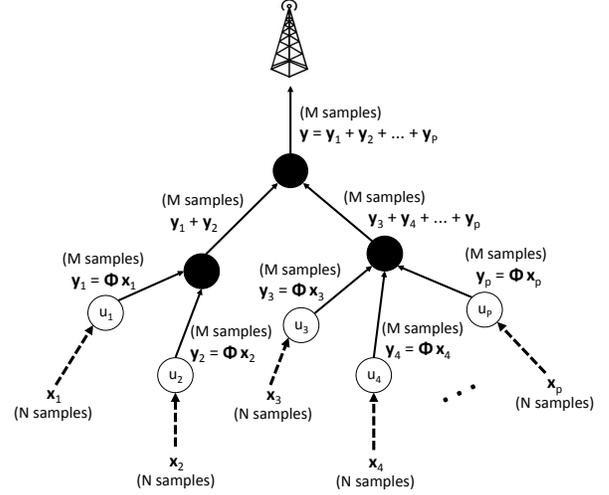
Fig. 6. An example for P-way mixing of compressive measurements (note: $\mathbf{x}_i \in \mathbb{C}^N$, $\mathbf{y}_i \in \mathbb{C}^M$, and the P-way sum $\mathbf{y} \in \mathbb{C}^M$)

B. Combining Multiple Measurements

In Fig. 6, we illustrate an example how compressive measurements independently captured by P sensors can be combined over the network. We note that the combing operations are purely arithmetic, keeping only track of which sensor measurements are added. We define combined compressive measurements from P different sensors.

Definition 3: (P-way Combined Compressive Measurements) For P sensors $u_1, \ldots, u_P$, the sum $\boldsymbol{y} = \boldsymbol{y}_1 + \ldots + \boldsymbol{y}_P = \sum_{i=1}^{P} \boldsymbol{y}_i$ mixes the sensor measurements P-way, where $\boldsymbol{y}_i = \boldsymbol{\Phi} \boldsymbol{x}_i$ contains M compressive measurements of node u_i.

The top black node in Fig. 6 will finally produce the sum $\mathbf{y}$ and transmit it to the base station. The additive mixing *preserves* the number of measurements, despite passing through intermediate network nodes and aggregating more data. The resulting save in bandwidth is roughly by a factor of P, the number of sensor nodes whose measurements are combined in the sum.

IV. SYSTEM OPERATION

We describe in detail the three phases outlined in Section I.C.

A. Preparation Phase

In Section II.C, we mentioned the use of optimal basis for decoding, which can significantly reduce the number of compressive measurements without compromising the accuracy of recovered results. Optimal bases, however, are *data-dependent* and therefore need to be estimated (and re-estimated). *Stationarity* of the data determines how frequently or infrequently a basis should be updated.

During Preparation Phase, the system estimates optimal bases to be used for each subchannel (a partition of the spectrum) by performing Kahrunen-Loève expansion on the measurements received from the sensors. Each sensor node samples its assigned subchannel in some unit intervals. During

the basis estimation, the system directs sensor nodes with a longer measurement period. Preparation Phase needs not be scheduled too frequently for analyzing modulated RF signals as we can expect that the *locations* of dominant eigenvectors (*i.e.*, leading principal components) in the covariance matrix of a subchannel change slowly or remain relatively stationary over time.

How KLT basis can be estimated: We adopt a technique that integrates compressive sensing with using optimal KLT basis [18]. Here, we summarize the technique. The KLT basis $\mathbf{Q}$ is computed from a sample autocorrelation matrix $\mathbf{R_x} = \mathbb{E}[\mathbf{xx}^{\mathrm{H}}]$, where $\mathbf{x}$ contains discrete channel measurements over time. Similarly, the autocorrelation matrix of the compressive measurements $\mathbf{y}$ is: $\mathbf{R_y} = \mathbb{E}[\mathbf{yy}^{\mathrm{H}}]$. By Definition 1, $\mathbf{R_y} = \mathbb{E}[\mathbf{\Phi xx}^{\mathrm{H}}\mathbf{\Phi}^{\mathrm{T}}] = \mathbf{\Phi}\mathbb{E}[\mathbf{xx}^{\mathrm{H}}]\mathbf{\Phi}^{\mathrm{T}}$. So, $\mathbf{R_y} = \mathbf{\Phi R_x}\mathbf{\Phi}^{\mathrm{T}}$. Since $\mathbf{\Phi}^{\mathrm{T}}$ is not a square matrix, using its pseudo-inverse $(\mathbf{\Phi}^{\mathrm{T}})^{\dagger}$, we obtain:

$$\mathbf{R_y}(\mathbf{\Phi}^{\mathrm{T}})^{\dagger} = \mathbf{\Phi R_x} \qquad (3)$$

Eq. (3) suggests that we have been compressively sensing $\mathbf{R_x}$ in $\mathbf{R_y}(\mathbf{\Phi}^{\mathrm{T}})^{\dagger}$, which is already captured by $\mathbf{y} = \mathbf{\Phi x}$ to encode the original channel measurements $\mathbf{x}$.

Following the above reasoning, we conclude that a KLT basis $\mathbf{Q}$ can be recovered via compressive sensing, comprised in four steps:

1) Decode $\mathbf{X}$ (DFT of $\mathbf{x}$) from $\mathbf{y} = (\mathbf{\Phi}\mathcal{F}^{-1})\mathbf{X}$ where $\mathcal{F}^{-1}$ is the inverse DFT matrix;
2) Recover $\mathbf{x}$ by computing $\mathbf{x} = \mathcal{F}^{-1}\mathbf{X}$;
3) Repeat the previous steps l times to numerically compute: $\mathbf{R_x} = \mathbb{E}[\mathbf{xx}^{\mathrm{H}}] = \frac{1}{l}\sum_{i=1}^{l}\mathbf{x}_i\mathbf{x}_i^{\mathrm{H}}$;
4) Obtain $\mathbf{Q}$ from the eigenvalue decomposition $\mathbf{R_x} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{H}}$.

B. Initial Approximation via Least Squares Recovery

The KLT basis estimation can identify dominant *eigen-components* by examining $\mathbf{\Lambda}$ whose diagonals are the eigenvalues of the channel autocorrelation $\mathbf{R_x}$. When compressive measurements arrive (as a P-way sum), the system first sets up an *overdetermined* system of equations specified by Definition 4, using only several dominant components per each subchannel mixed in the P-way sum. To set up the overdetermined equations, the system leverages its knowledge on the locations of the leading eigen-components of the subchannels. We compute the initial approximation by the least squares, a method well-known for overdetermined equations.

Definition 4: The initial approximation determines a solution by the least squares method to an overdetermined system of equations:

$$\boldsymbol{y} = \boldsymbol{\Phi}\,[\tilde{\boldsymbol{Q}}_1\,\tilde{\boldsymbol{Q}}_2\,\cdots\,\tilde{\boldsymbol{Q}}_P]\begin{bmatrix}\tilde{\boldsymbol{s}}_1\\\tilde{\boldsymbol{s}}_2\\\vdots\\\tilde{\boldsymbol{s}}_P\end{bmatrix} \qquad (4)$$

where $\boldsymbol{y}$ is the P-way sum of compressive measurements, $\tilde{\boldsymbol{Q}}_i$ is zero-filled truncated from the i-th KLT basis $\boldsymbol{Q}_i$ such that $\tilde{\boldsymbol{Q}}_i$

contains only α_i nonzero columns of dominant eigenvectors corresponding to the first α_i leading eigenvalues.

C. Iterative Refinement via Compressive Sensing Decoding

We refine the initial approximation through iterative relaxation(s) based on compressive sensing. We will relax only *one* $\mathbf{s}_i$ at a time by systematically eliminating $(P-1)$ $\mathbf{y}_i$'s with back-substituting $(P-1)$ $\tilde{\mathbf{s}}_i$'s from using the result of the initial approximation. For example, we relax $\mathbf{s}_1$ by solving:

$$\mathbf{y} = \mathbf{\Phi}\,[\mathbf{Q}_1\,\tilde{\mathbf{Q}}_2\,\cdots\,\tilde{\mathbf{Q}}_P]\begin{bmatrix}\boldsymbol{s}_1\\\hat{\tilde{\boldsymbol{s}}}_2\\\vdots\\\hat{\tilde{\boldsymbol{s}}}_P\end{bmatrix} \qquad (5)$$

where the rest $\hat{\tilde{s}}_i$'s are the back-substituted initial approximate. It is important to distinguish this system of equations, which is *underdetermined*, from the overdetermined system in Definition 4. We use the ℓ_1-minimization decoding, which is popular in compressive sensing, and relax all N unknowns in $\mathbf{s}_1$. After relaxing $\mathbf{s}_1$, we relax the next, say $\mathbf{s}_2$. For $\mathbf{s}_2$, we back-substitute $\mathbf{s}_1$ with $\hat{\mathbf{s}}_1$ (which was just relaxed) and $\tilde{\mathbf{s}}_3, \ldots, \tilde{\mathbf{s}}_P$ with $\hat{\tilde{\mathbf{s}}}_3, \ldots, \hat{\tilde{\mathbf{s}}}_P$ from the initial approximation:

$$\mathbf{y} = \mathbf{\Phi}\,[\mathbf{Q}_1\,\mathbf{Q}_2\,\tilde{\mathbf{Q}}_3\,\cdots\,\tilde{\mathbf{Q}}_P]\begin{bmatrix}\hat{\boldsymbol{s}}_1\\\boldsymbol{s}_2\\\hat{\tilde{\boldsymbol{s}}}_3\\\vdots\\\hat{\tilde{\boldsymbol{s}}}_P\end{bmatrix} \qquad (6)$$

This process iterates for all P $\mathbf{s}_i$'s. Note that another stage of P relaxations can take place. The new stage uses the result of the relaxations from the previous stage.

V. ALGORITHMS

In this section, we first look into a joint decoding algorithm that can recover the original signals from the sum of compressive measurements. Next, we present our algorithms for initial approximation and iterative refinement, which we described in the previous section. We argue that the latter algorithms can achieve better accuracy than the joint decoding while using fewer measurements.

A. Joint Decoding Algorithm

For simplicity, consider 2-way combined compressive measurements $\mathbf{y} = \mathbf{y}_1 + \mathbf{y}_2 = \mathbf{\Phi}\,\mathbf{x}_1 + \mathbf{\Phi}\,\mathbf{x}_2$. Joint decoding is a technique to recover $\mathbf{x_1}$ and $\mathbf{x_2}$ from $\mathbf{y}$ in one shot (hence, the term *joint*). Decoding one of the signals, say $\mathbf{x}_1$, from $\mathbf{y}$ is trivial if $\mathbf{y}_2$ is known, because we can decode $\mathbf{y} - \mathbf{y}_2 = \mathbf{\Phi}\,\mathbf{x}_1$. But how can we recover the original signals without explicit availability of additional measurements to separate one signal from another?

Simply put, joint decoding leverages the overcomplete representation used in, for example, blind source separation [19]. Using an overcomplete basis $\mathbf{\Psi} = [\mathbf{\Psi}_1\mathbf{\Psi}_2]$ for the

joint representation $\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$, we can rewrite the sum $\mathbf{y} = \boldsymbol{\Phi} \begin{bmatrix} \boldsymbol{\Psi}_1 \ \boldsymbol{\Psi}_2 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$, which is in the standard compressive sensing format $\mathbf{y} = (\boldsymbol{\Phi}\boldsymbol{\Psi})\,\mathbf{s}$. This can be decoded, and we split the joint solution $\mathbf{s}$ by the dimensions of $\mathbf{s}_1$ and $\mathbf{s}_2$. However, there is a drawback for joint decoding—it needs to solve for an increased number of variables (*i.e.*, $2N$ rather than N). We state the recoverability of P-way mixed compressive measurements by joint decoding in Theorem 5.

Theorem 5: (Joint Decoding of Mixed Compressive Measurements) For P-way combined compressive measurements of Definition 3, signals $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P$ with sparsifying bases $\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_P$ can be recovered by first decoding an overcomplete representation $\boldsymbol{y} = \boldsymbol{\Phi} \begin{bmatrix} \boldsymbol{\Psi}_1 & \cdots & \boldsymbol{\Psi}_P \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_P \end{bmatrix}$, then restoring $\boldsymbol{x}_i$ from $\boldsymbol{s}_i$, using $\boldsymbol{x}_i = \boldsymbol{\Psi}_i \boldsymbol{s}_i$. The recovery is exact with high probability if M, the number of measurements in $\boldsymbol{y}$, is sufficiently large for $\boldsymbol{s}_i$'s. That is, given total sparsity $K_{tot} = \sum_{i=1}^{P} K_i$ where K_i is the number of nonzero elements in $\boldsymbol{s}_i$, if $M \geq c\, K_{tot} \log(\frac{PN}{K_{tot}})$ for some small constant c. Otherwise, the recovery will be lossy.

Proof: For $\mathbf{y}_i = \boldsymbol{\Phi}\mathbf{x}_i$, $M_i \geq c\, K_i \log(\frac{N}{K_i})$. So, $\sum_{i=1}^{P} M_i \geq \sum_{i=1}^{P} c\, K_i \log(\frac{N}{K_i})$, and we want to examine if M is at least $\sum_{i=1}^{P} c\, K_i \log(\frac{N}{K_i})$ or larger. For $K_1 = K_2 = \ldots = K_P$, $\sum_{i=1}^{P} c\, K_i \log(\frac{N}{K_i}) = c\, K_{tot} \log(\frac{PN}{K_{tot}})$. For distinct K_i values, $c\, K_{tot} \log(\frac{PN}{K_{tot}}) > \sum_{i=1}^{P} c\, K_i \log(\frac{N}{K_i})$. Therefore, taking M measurements suffices exact recovery of $\mathbf{x}_i$'s with high probability. ∎

B. New Decoding Algorithm

Joint decoding can be implemented easily, but because the number of unknowns increases with the number of sensors in the P-way sum, admitting new sensor nodes will add extra variables in its one-shot decoding. This can be cumbersome for the system, as it would occasionally require to monitor and interfere on-going data collection paths that exceed some max-hop limit. Joint decoding is a mismatch to in-network data collection/compression schemes, and we need more efficient decoding algorithm that fits to our P-way model.

Initial Approximation Algorithm. Fig. 7 depicts initial approximation on the left.
1) Obtain $\tilde{\mathbf{Q}}_i$'s by keeping only the first α_i leading eigenvectors of $\mathbf{Q}_i$'s and replacing other columns with zeros;
2) Construct $\tilde{\mathbf{Q}} = \begin{bmatrix} \tilde{\mathbf{Q}}_1 \| \tilde{\mathbf{Q}}_2 \| \ldots \| \tilde{\mathbf{Q}}_P \end{bmatrix}$;
3) Obtain $\hat{\tilde{\mathbf{s}}}$ by solving the system $\mathbf{y} = \boldsymbol{\Phi}\tilde{\mathbf{Q}}\tilde{\mathbf{s}}$ for $\tilde{\mathbf{s}}$ via the least squares;
4) Let $\hat{\mathbf{s}}_i^0 = \hat{\tilde{\mathbf{s}}}_i$;
5) Compute $\hat{\mathbf{y}}_i^0 = \boldsymbol{\Phi}\,\mathbf{Q}_i\,\hat{\mathbf{s}}_i^0$, and pass $\langle \hat{\mathbf{y}}_i^0, \hat{\mathbf{s}}_i^0 \rangle$ to iterative refinement.

Iterative Refinement Algorithm. In the rest of Fig. 7, iterative refinement algorithm is explained. The algorithm requires
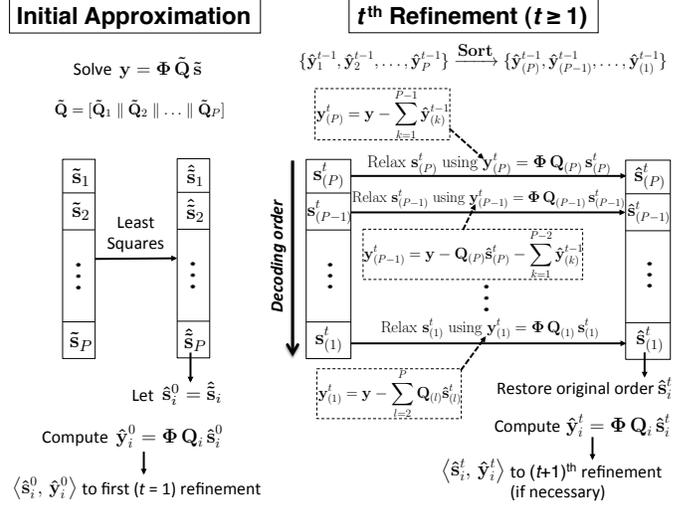
Fig. 7.   Initial approximation and iterative refinement decoding

the results $\langle \hat{\mathbf{s}}_i^{t-1}, \hat{\mathbf{y}}_i^{t-1} \rangle$ passed from the previous stage. (Note that Stage 0, *i.e.*, $t = 0$, refers to initial approximation.)
1) Sort $\hat{\mathbf{s}}_i^{t-1}$'s by their ℓ_1-norm to obtain $\hat{\mathbf{s}}_{(i)}^{t-1}$'s such that $\|\hat{\mathbf{s}}_{(P)}^{t-1}\|_{\ell_1} \geq \|\hat{\mathbf{s}}_{(P-1)}^{t-1}\|_{\ell_1} \geq \ldots \geq \|\hat{\mathbf{s}}_{(1)}^{t-1}\|_{\ell_1}$;
2) Sort $\hat{\mathbf{y}}_i^{t-1}$'s according to the indices found in the previous step from sorting $\hat{\mathbf{s}}_i^{t-1}$'s to obtain $\hat{\mathbf{y}}_{(i)}^{t-1}$'s;
3) Relax $\mathbf{s}_{(i)}^t$ and obtain $\hat{\mathbf{s}}_{(i)}^t$ in the *descending* order by the ℓ_1-norm computed in Step 1, *i.e.*, in the order of $\mathbf{s}_{(P)}^t \to \mathbf{s}_{(P-1)}^t \to \ldots \to \mathbf{s}_{(1)}^t$:
   i) Compute $\mathbf{y}_{(i)}^t = \mathbf{y} - \sum_{l=i+1}^{P} \mathbf{Q}_{(l)} \hat{\mathbf{s}}_{(l)}^t - \sum_{k=1}^{i-1} \hat{\mathbf{y}}_{(k)}^{t-1}$;
   ii) Do ℓ_1-minimization decoding on $\mathbf{y}_{(i)}^t = \boldsymbol{\Phi}\,\mathbf{Q}_{(i)}\,\mathbf{s}_{(i)}^t$ and obtain $\hat{\mathbf{s}}_{(i)}^t$;
4) Restore the original order $\hat{\mathbf{s}}_i^t$;
5) If this is the final refinement stage, stop and return the decoded results $\hat{\mathbf{s}}_i^t$, else compute $\hat{\mathbf{y}}_i^t = \boldsymbol{\Phi}\,\mathbf{Q}_i\,\hat{\mathbf{s}}_i^t$ and pass $\langle \hat{\mathbf{s}}_i^t, \hat{\mathbf{y}}_i^t \rangle$ to the $(t+1)$-th refinement and continue.

VI. EVALUATION

A. Methodology

We empirically evaluated the performance of our spectrum analyzer. We used USRP2 and N200 nodes [20] that run GNU radio [21], an open-source software radio framework. We set up 4 USRPs in an indoor lab and analyzed a 200-MHz UHF spectrum[1]. The spectrum was partitioned into J = 8 subchannels (*i.e.*, each with 25-MHz bandwidth) whose center frequencies were $f_c \in \{512.5, 537.5, 562.5, 587.5, 612.5, 637.5, 662.5, 687.5\}$ MHz. We configured each USRP to cycle between two subchannels, responsible for taking measurements at both frequencies. Note that the Nyquist sampling rate of a subchannel is 25 MHz $\times$ 2 = 50×10^6 samples/sec. For each instance of measurements, we used $N = 512$ per subchannel (*i.e.*, the resolution bandwidth of 25 MHz/512 $\approx$ 48.8 kHz). This corresponds to $L = J \times N = 4096$-point

---

[1]It spans about 33 UHF channels (6 MHz each) between Ch. 19 and Ch. 51.

frequency response for the entire spectrum contributed by all USRPs.

For compressive sensing, we pre-generated a set of $\boldsymbol{\Phi}$ matrices and stored in each USRP. For each subchannel, the USRPs could take compressive measurements using configurable M, which we varied from 26 ($M/N = 5\%$ or 20x compression) to 308 (60% or 1.67x). USRPs used the same $\boldsymbol{\Phi}$ under each configuration. We also saved uncompressed original data for evaluative purposes. The uplink transmission to the base station from an USRP was simulated over a fixed Ethernet, as precise wireless uplink behaviors (assuming no packet losses) should hardly affect what we examined (*i.e.*, number of total measurements and decoding accuracy).

The USRPs were coarsely synchronized, and each USRP's measurement start and end times cannot be determined exactly. This is similar to a swept-tuned spectrum analyzer that measures a narrow subchannel one at a time, staying for a short duration before moving to next and cycling the entire spectrum. The USRPs (unfortunately) incur a 2-second, hardware-related delay when changing radio frequency, resulted from driving a voltage controlled oscillator (VCO).

B. Description of Experiments

We evaluated the following schemes:

1) *No mixing.* USRPs perform compressive sensing on their assigned subchannels and transmit the compressive measurements directly to the base station, *i.e.*, there are 8 transmissions to the base station per one complete cycle of measurements for the entire spectrum;

2) *P-way combined compressive measurements.* Varying $P = 2, 4, 8$, USRPs mix their compressive measurements, thus there are $8/P$ transmissions per one complete cycle.

We compared the decoding performance of the proposed algorithm to joint decoding for $P = 8$. Note that for no mixing case the proposed algorithm needs not be used, as each subchannel can be recovered individually by the standard ℓ_1-minimum decoder.

C. Error Metric

We use the following error metric to evaluate the accuracy of our decoding algorithm:

$$\xi = \frac{1}{L} \sum_{k=1}^{L} \frac{\|X(f_k) - \hat{X}(f_k)\|_{\ell_2}}{\|X(f_k)\|_{\ell_2}} \tag{7}$$

where $X(f_k)$ is the frequency response of the spectrum under analysis from the Nyquist sampling (no compression and no manipulation of original data), and $\hat{X}(f_k)$ the recovered frequency response from a scheme we mentioned in Section VI.B. Note frequency indices $f_k \in [500, 700)$ MHz with $k = 0, \ldots, L - 1$, following the discrete Fourier analysis convention. Note also that the error metric ξ is *normalized* (by the true value) per-sample mean.

D. Decoding Accuracy and Complexity

With varying number of compressive measurements for each scheme, we counted the total number of measurements received by the base station while computing the error metric. Fig. 8 presents the number of measurements plotted against the error metric ξ reflecting the accuracy of reconstructed frequency responses. The 8-way combining scheme achieved a 5-fold saving in communication bandwidth for the same accuracy by no combining scheme. When we used the proposed decoding algorithm for P-way combined schemes, we applied 2 stages of iterative refinements. The proposed algorithm proved to be better than the joint decoding in accuracy. In Fig. 9, we plot frequency responses (*i.e.* spectrum analyzer display) of the spectrum constructed from uncompressed original time samples (4096 measurements), compressed but no combining scheme (832 measurements, 4.9x compression), and 8-way combined (208 measurements, 20x compression) for visual comparison.

The proposed algorithm was also better in computational complexity. To decode 8-way combined measurements, the joint decoding requires to operate on $8 \times 512 = 4096$ variables at once, whereas the proposed algorithm operates on 40 unknowns (*i.e.*, each $\alpha_i = 5$) with the least squares and 2 stages of $J = 8$ relaxations (each decodes $N = 512$ variables). The ℓ_1-minimum decoding has complexity of $O(N^3)$ [13], and the joint decoding requires $O(P^3 N^3)$ while the proposed algorithm has only $O(2PN^3)$ excluding the least squares in the initial approximation. The least squares solve for $\sum_{i=1}^{P} \alpha_i$ unknowns, which is about orders of magnitude fewer than PN and therefore not a contributing factor.

E. Effect of Iterative Refinements

Fig. 10 depicts the error improvement versus the number of refinement stages applied for 8-way combining (*i.e.*, $P = 8$). The total number of measurements received by the base station per one complete cycle, M_{tot}, were 50, 100, and 200. Note that zero iterative (refinement) stage means the initial approximation only. Error improvement becomes more significant for smaller M_{tot}, which suggests that more computations in decoding could compensate insufficient measurements to some degree but could not overcome completely. There is a diminishing return on the error improvements, and the return is saturated faster for larger M_{tot}.

VII. Conclusion

We have described a network-based spectrum analyzer that operates over distant sensor nodes providing the measurements to construct fine-grained spectral information. Overcoming the network communication cost as we scale up the number of sensor nodes has been critical to our approach. To address this, we have devised a recovery algorithm that accompanies a simple, additive in-network combining scheme for compressed measurements from multiple sensors. Our approach makes an important assumption that discrete measurements obtained by sensors bring out sparsity in the frequency domain or in a custom basis. Designing sparsity-preserving discretization
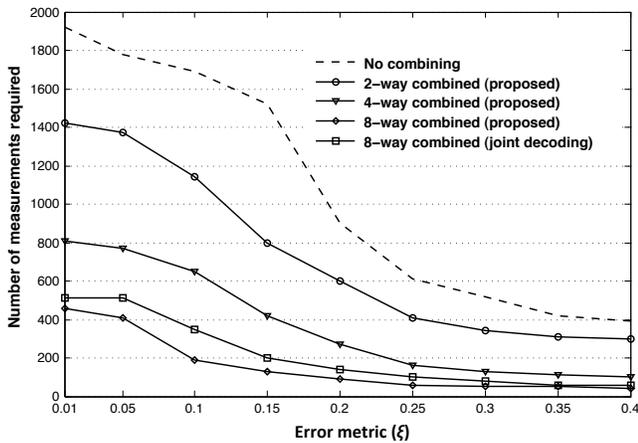
Fig. 8. Total number of measurements used in spectrum analysis vs. accuracy of decoding measured by error metric ξ. The proposed methods run 2 refinement stages. Joint decoding for 8-way combined uses a comparable number of measurements to the proposed 8-way combined for similar error performance, but joint decoding is much harder computationally and takes significantly longer decoding time (see Section V.A).
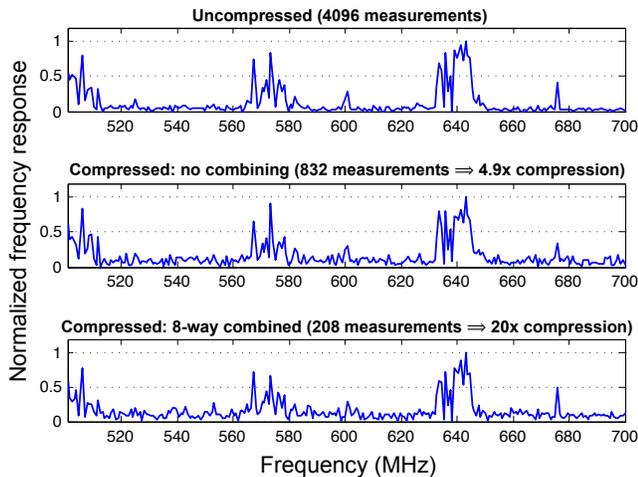
Fig. 9. Snapshots of recovered spectrum under uncompressed, compressed without combining, and compressed with 8-way combined methods
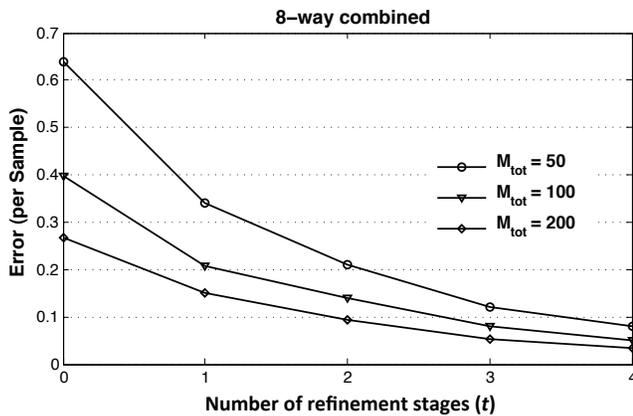
Fig. 10. Error improvement over number of refinement stages performed

schemes is a challenging, ongoing research effort that applications of our work can benefit from, but orthogonal to the main considerations of this paper. Although it may sound surprising, we conclude that it is feasible to scale our network-based spectrum analyzer with constant communication cost.

REFERENCES

[1] J. Mitola and G. Maguire, "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
[2] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79–89, May 2007.
[3] A. Iyer, K. Chintalapudi, V. Navda, R. Ramjee, V. N. Padmanabhan, and C. R. Murthy, "SpecNet: Spectrum Sensing Sans Frontières," in *USENIX NSDI*, 2011.
[4] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Trans. of American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, Apr. 1928.
[5] D. L. Donoho, "Compressed Sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
[6] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed Compressed Sensing," Rice University, Tech. Rep. ECE-0612, 2005.
[7] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive Wireless Sensing," in *IPSN*, 2006.
[8] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive Data Gathering for Large-scale Wireless Sensor Networks," in *ACM MOBICOM*, 2009.
[9] F. Zeng, C. Li, and Z. Tian, "Distributed Compressive Spectrum Sensing in Cooperative Multihop Cognitive Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 37 –48, 2011.
[10] Y. L. Polo, Y. Wang, A. Pandharipande, and G. Leus, "Compressive Wideband Spectrum Sensing," in *IEEE ICASSP*, 2009.
[11] Y. Wang, A. Pandharipande, Y. L. Polo, and G. Leus, "Distributed Compressive Wide-band Spectrum Sensing," in *Information Theory and Applications Workshop*, 2009.
[12] E. J. Candès, "Compressive Sampling," in *Proc. of International Congress of Mathematicians*, 2006.
[13] R. G. Baraniuk, "Compressive Sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, Jul. 2007.
[14] D. Needell and J. A. Tropp, "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples," *CACM*, vol. 53, pp. 93–100, Dec. 2010.
[15] K. Karhunen, "Über Lineare Methoden in der Wahrscheinlichkeitsrechnung (On Linear Methods in Probability and Statistics)," PhD Dissertation, University of Helsinki, 1947.
[16] R. R. Coifman and M. V. Wickerhauser, "Entropy-based Algorithms for Best Basis Selection," *IEEE Trans. on Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.
[17] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed Source Coding for Sensor Networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, Sep. 2004.
[18] Y. Gwon, H. T. Kung, and D. Vlah, "Compressive Sensing with Optimal Sparsifying Basis and Applications in Spectrum Sensing," in *IEEE GLOBECOM*, 2012.
[19] T.-W. Lee, M. S. Lewicki, M. Girolami, and T. Sejnowski, "Blind Source Separation of More Sources Than Mixtures Using Overcomplete Representations," *IEEE Signal Processing Letters*, vol. 6, no. 4, pp. 87–90, Apr. 1999.
[20] "The Universal Software Radio Peripheral (USRP) Products, Ettus Research LLC," http://www.ettus.com/.
[21] "GNU Radio," http://gnuradio.org.