COMPRESSED NETWORK MONITORING

Mark Coates, Yvan Pointurier and Michael Rabbat

Department of Electrical and Computer Engineering
McGill University
3480 University Street, Montreal, Quebec H3A 2A7, Canada

ABSTRACT

This paper describes a procedure for estimating a full set of network path metrics, such as loss or delay, from a limited number of measurements. The approach exploits the strong spatial and temporal correlation observed in path-level metric data, which arises due to shared links and stationary components of the observed phenomena. We design diffusion wavelets based on the routing matrix to generate a basis in which the signals are compressible. This allows us to exploit powerful non-linear estimation algorithms that strive for sparse solutions. We demonstrate our results using measurements of end-to-end delay in the Abilene network. Our results show that we can recover network mean end-to-end delay with 95% accuracy while monitoring only 4% of the routes.

Index Terms— network monitoring, diffusion wavelets, compressed sensing

1. INTRODUCTION

Direct monitoring of a network – either at the path level or the link level – does not scale in any practical setting. For the past decade, researchers have been actively investigating techniques for inferring network characteristics from incomplete or indirect measurements [1, 2]. This paper describes a scheme for estimating performance metrics such as delay or loss rates on many end-to-end paths in a network using measurements taken on only a few of these paths. Similar to related previous work [3, 4], we exploit the idea that, typically, each link is used by many paths. Thus, by only measuring a subset of paths, it is possible to infer the performance on other unmeasured paths using knowledge of the routing topology.

Our approach combines the use of wavelets, for efficient signal representation, with recent developments in nonlinear estimation, for recovering a sparse signal from few measurements. We use *diffusion wavelets* [5] to compress the path-level performance signal into a set of sparse coefficients (only a few coefficients have significant energy). The wavelets are designed to specifically account for anticipated correlations between different path performance values, based on knowledge of the network topology and routing policy. Then, we use ℓ_1 optimization techniques to identify the most likely set of sparse wavelet coefficients given measurements on a few paths, and these estimated wavelet coefficients are used to predict performance values on unobserved paths. In comparison to the previous state-of-the-art *network kriging* estimator [4], we find that our method makes more accurate estimates with few measurements on real data sets.

2. PROBLEM FORMULATION

Suppose we wish to monitor performance metrics on n_p paths in a network with n_l links. Call $x^{(k)} \in \mathbb{R}^{n_l}$ the vector of link-level performance metrics and $y^{(k)} \in \mathbb{R}^{n_p}$ the vector of path-level performance metrics at a particular time instant, k. We assume we are given the routing matrix, $G^{(k)} \in \{0,1\}^{n_p \times n_l}$, where $G^{(k)}_{i,j} = 1$ if and only if link j belongs to path i. We also assume that x and y are linearly related via the expression

$$y^{(k)} = G^{(k)} x^{(k)}, \tag{1}$$

i.e., the performance metric of a path is equal to the sum of the performance metrics of the links comprising that path. Examples of relevant performance metrics satisfying this assumption include mean delay, delay variance, and the (log) rate of successful transmission.

In addition to the routing matrix, G, we observe n_s metrics y_s, on a subset s of paths ($n_s = |s|$). Our goal is to infer the metrics for the remaining $n_p - n_s$ paths. Observations of end-to-end metrics are made according to a *selection matrix* $A^{(k)} \in \{0,1\}^{n_s \times n_p}$ which effectively selects the routes that are observed (sampled); each row of $A^{(k)}$ contains exactly one "1" and each column contains at most one "1". The presence of a "1" in the p^{th} column of $A^{(k)}$ indicates we observe the performance metric on path p. The observed samples (e.g., end-to-end delays) $y_s^{(k)} \in \mathbb{R}^{n_s}$ are therefore

$$y_s^{(k)} = A^{(k)} y^{(k)}. \tag{2}$$

We perform estimation of metrics over a block of measurement intervals, allowing us to exploit both spatial and

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network.

temporal correlation of the path-level metrics. We denote by $y = [{y^{(1)}}^T, \ldots, {y^{(\tau)}}^T]^T$ the vector of path-level metrics over τ timesteps. Similarly, the corresponding link metrics are $x = [{x^{(1)}}^T, \ldots, {x^{(\tau)}}^T]^T$, and we introduce block-diagonal matrices $G \in \mathbb{R}^{n_p \tau \times n_l \tau}$ and $A \in \mathbb{R}^{n_s \tau \times n_p \tau}$, with diagonal submatrices $G^{(1)}, \ldots, G^{(\tau)}$ and $A^{(1)}, \ldots, A^{(\tau)}$, respectively. With this new notation we can write $y = Gx$, and $y_s = AGx$. In general, routing in a network changes very slowly over time, and unless rerouting occurs, we have $G^{(k)} = G_0$ for all k. We focus on this special case here, although incorporating time-varying routing and load balancing is a straightforward extension.

3. DIFFUSION WAVELETS

Wavelet transforms are a staple of modern compression and signal processing methods due to their ability to efficiently represent piece-wise smooth signals (signals which are smooth everywhere, except for a few discontinuities). Traditionally, discrete wavelet transforms provide a multi-scale decomposition of functions defined on a regularly sampled interval or grid. A "mother wavelet" is dilated by powers of two and translated to obtain orthonormal wavelet bases. However, in the context of network monitoring, we seek to efficiently represent a function (performance metrics) defined on a network topology which does not, in general, have a regular structure, so standard wavelets cannot be directly applied. Crovella and Kolaczyk [6] describe one method of constructing wavelets on a graph for decomposing traffic on an arbitrary topology based on dilating and scaling a mother wavelet, similar to the traditional approach. The primary shortcoming of this approach is that it does not lead to an orthogonal basis, limiting its use as a mechanism for generating a compressible representation of a network function. More recently, Coifman and Maggioni [5] have introduced diffusion wavelets, generalizing the concept of wavelets to functions supported on a graph through the use of diffusion operators.

The construction of a diffusion wavelet basis is based on a diffusion operator, D, defined on the support of the underlying graph. For a graph with n nodes, D is an $n \times n$ matrix where $D_{i,j} > 0$ if and only if there is a link between nodes i and j. The magnitude of $D_{i,j}$ models the strength of the correlation or similarity between the function values at nodes i and j. Much like traditional wavelets, diffusion wavelets recursively split the space over which the signal is observed into smaller, orthogonal subspaces. Consider a function $f \in \mathbb{R}^n$ defined on a network of n nodes, where f_i corresponds to the value at node i; the function is initially defined on the space $V_0 = \mathbb{R}^n$. At scale $j = 1, \ldots, L$, for some pre-specified depth L, the diffusion wavelet construction recursively splits the space V_j into a scaling subspace, V_{j+1}, and a wavelet subspace, W_{j+1}, by analyzing eigenvectors of the jth dyadic power of the diffusion operator, D^{2j}. The matrix D^{2j} is, intuitively, related to averaging or smoothing over neighborhoods of radius $2j$ hops in the original graph, and the study of eigenvectors of this matrix is analogous to Fourier spectral analysis on a regular space. The ensuing orthonormal wavelet basis, adapted to the representation of the data (function values) over the graph, is obtained by concatenating bases for V_L and the wavelet subspaces, $\{W_j\}_{j=1}^{L}$. We refer the reader to [5] for the precise details of the construction.

Let $B_j \in \mathbb{R}^n$, $j = 1, \ldots, n$, denote the final collection of orthonormal wavelet basis vectors. A function on the graph can be represented as a vector $y \in \mathbb{R}^n$, where y_i is the value at the i^{th} node, and the wavelet decomposition of y is given by $y = \sum_{j=1}^{n} \beta_j B_j$, where $\beta_j = y^T B_j$ is the jth wavelet coefficient. Stacking the coefficients, β_j, into a vector, β, and concatenating the basis vectors, $\{B_j\}$ into an $n \times n$ matrix, B, we can write $y = B\beta$. In the following section we propose a diffusion operator D designed such that the corresponding wavelet representation of a path performance vector, y, is highly compressed; i.e., most of the energy in y can be captured in a few β_j. To be more precise, let us rearrange the wavelet coefficients in order of decreasing magnitude so that

$$|\beta_{(1)}| \geq |\beta_{(2)}| \geq \cdots \geq |\beta_{(n)}|,$$

and define the best m-term approximation of y in B to be $\widehat{y}^{(m)} = \sum_{j=1}^{m} \beta_{(j)} B_{(j)}$. We say that y is compressible in B when the approximation error $\|y - \widehat{y}^{(m)}\|$ decays rapidly as a function of m, meaning that y is efficiently represented using only a few basis vectors, $B_{(1)}, \ldots, B_{(m)}$. In this case, we only really need to estimate values of the few large coefficients in order to obtain a high quality estimate of end-to-end performance on many paths. Moreover, in this setting we can make use of recent breakthroughs in the area of non-linear estimation of compressible functions to quantify the number of paths that need to be measured to obtain estimates of performance at a specified level of accuracy.

4. COMPRESSIBLE REPRESENTATIONS

In order to construct a compressible representation, we develop a diffusion wavelet basis where the diffusion operator is related to the anticipated correlation between link metrics. We first define the graph of interest $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We are measuring (weighted sums of) a performance metric function defined on the physical links of the network. Accordingly, the vertex set $\mathcal{V}$ for our diffusion wavelet basis has one vertex for each link at each timestep: $\mathcal{V}^{(k)} = \{v_i^{(k)}\}$, over the estimation interval $k = 1, \ldots, \tau$.

The edges in the graph model correlation and their associated weights define the diffusion operator. We form the edge set $\mathcal{E}$ by inserting two types of edges into the graph. We add a *spatial-correlation* edge between the vertices $v_i^{(k)}$ and $v_j^{(k)}$ if there is at least one route in the network that uses both links i

and j during time interval k. We assign a weight $w(v_i^{(k)}, v_j^{(k)})$ to this edge to model the spatial correlation. Denote by $\mathcal{R}_i$ the set of paths that use link i. We set

$$w(v_i^{(k)}, v_j^{(k)}) = \frac{|\mathcal{R}_i \cap \mathcal{R}_j|}{|\mathcal{R}_i \cup \mathcal{R}_j|}, \tag{3}$$

thereby assigning greater weights to edges that join network links which have more routes in common. We also add a *temporal-correlation* edge between vertices $v_i^{(k)}$ and $v_i^{(k+1)}$ for $k = 1, \ldots, \tau - 1$, reflecting the idea that performance of a link at one timestep is a good predictor for the same link at the next timestep. We assign a weight $w(v_i^{(k)}, v_i^{(k+1)}) = c$ to these edges, where $c > 0$ is a constant that is adjusted according to the (anticipated) relative strength of temporal correlation. In this paper, we set $c = 0.5$, because we anticipate strong correlation.

The diffusion wavelet procedure in [5] requires a diffusion operator D to generate a wavelet representation over $\mathcal{G}$. To obtain a diffusion operator D from the construction described above, we apply Sinkhorn balancing [7] to the matrix of weights, $[w]$, to form a doubly stochastic matrix.

5. NON-LINEAR ESTIMATION

Now, suppose we have made observations y_s of the end-to-end performance for a subset of the paths we are interested in, and we wish to estimate y. We have $y_s = Ay$, where A is the selection matrix, indicating which paths we observe directly.

Our estimation procedure is based on the belief that the link metrics are highly compressible when represented using the diffusion wavelet basis B. Call β the link metrics coefficients in the diffusion wavelet basis B, such that $x = B\beta$.

We can express y in terms of its wavelet coefficients as $y = GB\beta$, where most of the energy in y is captured in a few entries of β. Combining this expression with the expression for y_s above leads to $y_s = AGB\beta$. In this formulation, we have a strong prior belief that β has only a few large entries, with most being very small in magnitude or even zero. This motivates the adoption of estimation strategies that produce sparse estimates.

A straightforward approach to obtaining a sparse estimate of β is to solve an ℓ_0 optimization of the form:

$$\widehat{\beta} \;=\; \arg\min_{\beta} \|\beta\|_0 \text{ subject to } y_s = AGB\beta,$$

where $\|\beta\|_0$ counts the number of non-zero entries of β. It is well known that this problem is NP-hard, requiring one to enumerate all possible subsets of non-zero coefficients.

It has recently been shown that the solution to a simpler ℓ_1 optimization problem,

$$\widehat{\beta} \;=\; \arg\min_{\beta} \|\beta\|_1 \text{ subject to } y_s = AGB\beta, \tag{4}$$

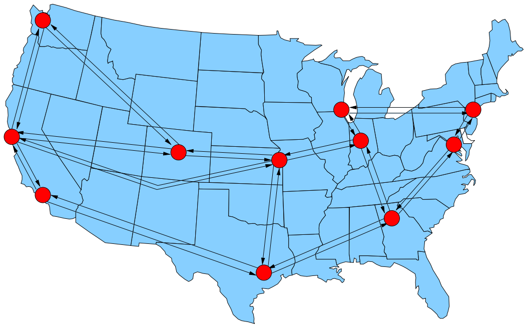

Fig. 1. Abilene backbone: 11 nodes, 30 (unidirectional) links.

is equivalent to the ℓ_0 problem if certain conditions on A, GB, and β are satisfied [8–10]. Here, $\|\beta\|_1 = \sum_{i=1}^{n} |\beta_i|$. Because the ℓ_1 optimization (4) is convex, it is computationally tractable, and a solution can be obtained using linear programming.

6. PATH SELECTION

So far, we have discussed why and how it is possible to estimate accurately end-to-end metrics from a limited number of observations. However, we have not discussed how to choose the path selection matric A, that is, select the n_s routes which should be observed. This problem is challenging and the appropriate approach depends on the measurement constraints or costs. Here, we consider that the cost (or constraint) is the number of measurements made per timestep. It thus does not matter here which paths we measure. For example, if we are allowed one observation per timestep, then we can measure the same route every timestep or we can measure a different route every timestep.

We adapt the path selection technique presented by Chua et al. in [4] to include our correlation model. The path selection procedure in [4] strives to minimize the mean square of the prediction error of a linear end-to-end delay estimator. The exact minimization procedure is NP-complete (it amounts to the problem of subset selection) and hence heuristics are needed.

Chua et al. propose a heuristic that consists of finding the rows of the routing matrix G that approximate the span of the first n_s left singular vectors of GC, where C is a nonsingular matrix that satisfies $\Sigma = GC$ and Σ is the covariance of x. Note that the estimation methodology in [4] is restricted to metrics of the form $y = Gx$, which leads to the incorporation of a link-level covariance matrix in this path selection procedure. In the case where this covariance matrix is not known, reasonable results can be often by obtained by setting $\Sigma = I$. An algorithm (see Alg. 1) that implements this heuristic can be found in [11].

The intuition behind this heuristic is that most of the energy of the path metric signal should reside in the space

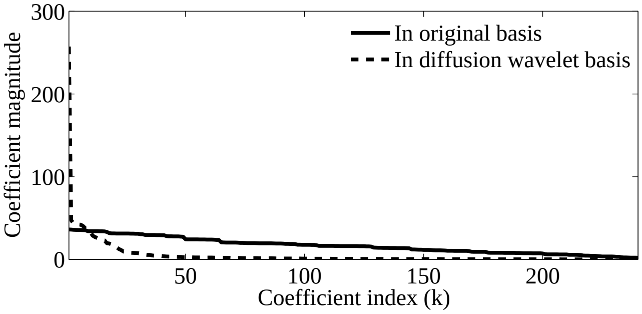

Fig. 2. Link delays for the complete network over 8 timesteps, sorted by magnitude, in the original basis and in the diffusion wavelet basis.

spanned by the n_s left singular vectors of GC. Identifying a set of paths that approximately span this space is thus a desirable goal.

Here, we use as the covariance $\Sigma = D^{\tau}$ (recall D is the diffusion operator and τ the number of timesteps used to account for time-correlation), and thus selection is performed by Alg. 1 where the input matrix M_i is equal to GC.

```
Algorithm 1 path selection
Input:   Matrix $M_i$
         Number of paths to select $n_s$
Output: Selection matrix $A$
  Perform SVD on $M_i$: $M_i = USV^T$ where $S$ contains
    the singular values of $G$, in descending order.
  Perform QR decomposition with column pivoting on the
    first $n_s$ columns of $U$: $QR = U^T_{(1,\ldots,n_s)}P^T$.
  Return $A = P_{(1,\ldots,n_s)}$, the matrix formed from the first
    $n_s$ columns of $P$.
```

7. NUMERICAL RESULTS

To illustrate the estimation technique presented in this paper, we use experimental delay data collected on the Abilene backbone network depicted in Fig. 1. The network consists of 11 nodes and 30 unidirectional links. Mean end-to-end delay measurements are collected between every pair of nodes over 48 five-minute intervals. There are thus 121 path metrics to be estimated at each time step.

We first verify the compressibility of the data. Fig. 2 shows the delays for all links and the absolute values of the diffusion wavelet coefficients, over $\tau = 8$ timesteps, sorted in descending order. The decay of the delays expressed in the original basis is very slow and exhibits a heavy tail, whereas in the diffusion wavelet basis, the decay is much faster.

Figure 3 compares the performance of the proposed technique (nonlinear estimation in the diffusion wavelet basis) with network kriging, the current state-of-the-art estimation technique for network end-to-end metrics [4]; network kriging uses an eigenbasis formed using the routing matrix and linear (MMSE) estimation. We estimate the average end-to-end delay over the entire network, $\langle y \rangle = \frac{1}{n_p \tau}\sum_{i=1}^{n_p} y_i$, and the individual delays of all paths. Estimation is performed for $\tau = 1$ and $\tau = 8$ for a total of 48 timesteps and 5% confidence intervals are depicted. We assess performance based on the relative mean end-to-end delay error $\langle |y - y_{est}| \rangle / \langle y \rangle$ and the relative ℓ_2-norm error $\|y - y_{est}\|_{\ell_2} / \|y\|_{\ell_2}$. We provide results for the single timestep case ($\tau = 1$), where only spatial correlation between the routes is accounted for, and the multi timestep case ($\tau = 8$) where both spatial and temporal correlations are accounted for. In each case, we built a diffusion wavelet basis with the maximum depth allowed by the numerical approximations used in the construction procedure: $L = 10$ for $\tau = 1$ and $L = 11$ for $\tau = 8$.

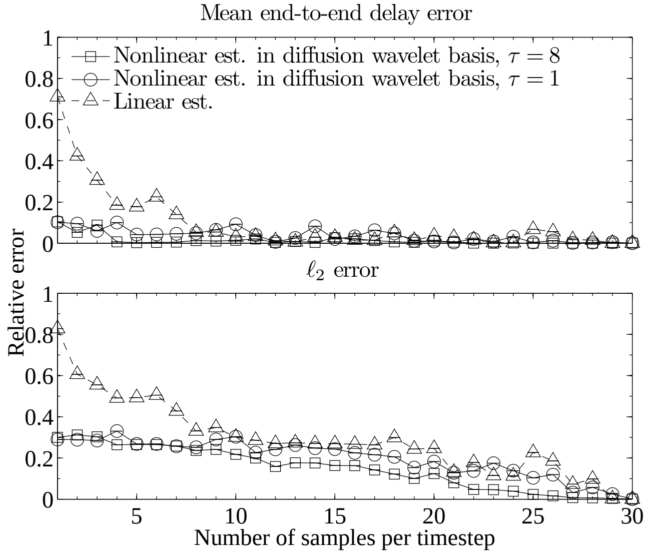

Fig. 3. Top panel: Relative mean end-to-end delay error (top) and relative ℓ_2-norm error (bottom) as functions of the average number of measurements per timestep.

In general, accounting for time correlation decreases the estimation error, but the improvement is marginal and not universal, Since there is substantial temporal correlation in the data, this suggests that there is room for improvement in how our methodology incorporates temporal correlation information. When less than 10 samples per timestep are collected, non-linear estimation in a diffusion wavelet basis using temporal correlation information exhibits much lower estimation error than the linear estimator. Our results suggest that by making only 5 measurements per timestep (4% of the network paths) we can hope to recover the mean network end-to-end delay with an error of less than 5%. Fig. 4 shows the recovered end-to-end delay over time for two example paths. We see that linear estimation exhibits substantial bias. In contrast, the nonlinear estimator exhibits much less bias but more variability, an artifact induced by the utilization of diffusion wavelet bases.

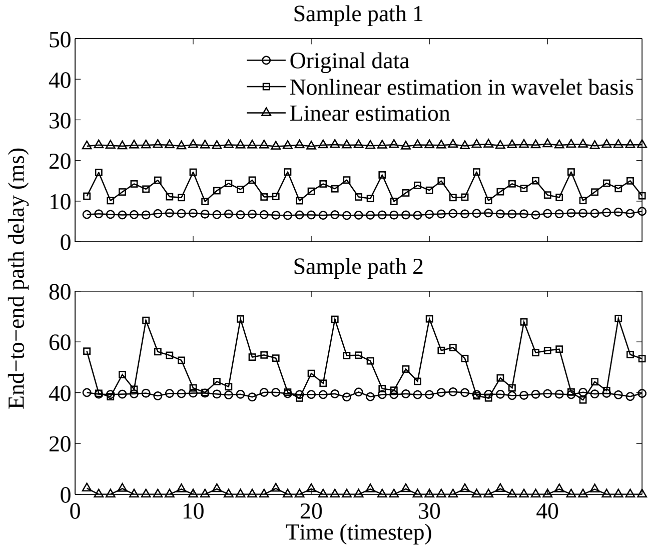

Fig. 4. Comparison between non-linear estimation and linear estimation of path delays for two example paths.

8. DISCUSSIONS AND FUTURE WORK

In this paper we focused on developing a compressible representation for the case where we know that performance on a path is the sum of performances on links, but we do not necessarily need to make this assumption. When this is not true, we can still apply the same methodology. As long as we have some prior information about how performance on different paths is correlated, we can design a diffusion wavelet basis to compress the path performance signal. We have applied our framework to a specific example (end-to-end delay estimation in a network whose topology is known) and showed that the observation of only a few sample end-to-end delays was sufficient to recover most of the information regarding the unobserved end-to-end delays. In particular, we showed that our technique outperformed network kriging, which uses linear estimation. Our technique is very general and can be applied to many different setups. For instance, see [12] for more on applying the methodology described in this paper to estimate quality of service metrics in all-optical networks.

There are three key components to the network monitoring framework described in this article: 1) a compressing transformation to sparsify the target signal, 2) a nonlinear estimation scheme which favors sparse solutions, and 3) a path selection algorithm for determining the optimal monitoring strategy. Our ongoing work involves refining each of these components. More specifically, we report here that diffusion wavelets provide a compressing transformation, but the resulting estimates have oscillatory artifacts due to the nature of the diffusion wavelet basis. We are currently investigating the use of alternative sparsifying transformations such as graph wavelets [6] and a novel multi-scale decomposition on graphs inspired by Haar wavelets. Similarly, we are experimenting with alternative nonlinear estimators and approaches to path selection. Last, our framework was validated using data from the Abilene backbone network. Investigating theoretical performance bounds for our framework is a interesting and challenging extension to this work.

Acknowledgment

The authors would like to thank Dr. David Chua (Department of Mathematics and Statistics, Boston University) who provided the Abilene data set used in [4].

9. REFERENCES

[1] Y. Vardi, “Network tomography: Estimating source-destination traffic intensities from link data,” *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, March 1996.

[2] M. Coates, A. Hero, R. Nowak, and B. Yu, “Internet tomography,” *IEEE Signal Processing Mag.*, pp. 47–65, May 2002.

[3] Y. Chen, D. Bindel, H. Song, and R. Katz, “An algebraic approach to practical and scalable overlay network monitoring,” in *Proc. ACM SIGCOMM*, Portland, Oregon, August 2004.

[4] D. Chua, E. Kolaczyk, and M. Crovella, “Efficient monitoring of end-to-end network properties,” in *Proc. Infocom*, vol. 3, Miami, FL, USA, Mar. 2005, pp. 1701–1711.

[5] R. Coifman and M. Maggioni, “Diffusion wavelets,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 53–94, July 2006.

[6] M. Crovella and E. Kolaczyk, “Graph wavelets for spatial traffic analysis,” in *Proc. IEEE Infocom*, San Francisco, USA, March 2003.

[7] R. Sinkhorn, “A relationship between arbitrary positive matrices and double stochastic matrices,” *Ann. Mathematical Statistics*, vol. 35, no. 2, pp. 876–879, June 1964.

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

[9] D. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[10] J. Haupt and R. Nowak, “Signal reconstruction from noisy random projections,” *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 4036–4048, September 2006.

[11] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore: The Johns Hopkins University Press, 1996.

[12] M. Coates, Y. Pointurier, and M. Rabbat, “Compressed network monitoring,” in *IMC’07*, in submission.