Compressed network monitoring
for IP and all-optical networks [*]

Mark Coates, Yvan Pointurier and Michael Rabbat
Department of Electrical and Computer Engineering
McGill University
Montreal, Quebec H3A-2A7, Canada
{mark.coates, yvan.pointurier, michael.rabbat}@mcgill.ca

ABSTRACT

We address the problem of efficient end-to-end network monitoring of path metrics in communication networks. Our goal is to minimize the number of measurements or monitors required to maintain an acceptable estimation accuracy. We present a framework based on diffusion wavelets and nonlinear estimation. Our procedure involves the development of a diffusion wavelet basis that is adapted to the monitoring problem. This basis exploits spatial and temporal correlations in the measured phenomena to provide a compressible representation of the path metrics. The framework employs nonlinear estimation techniques using ℓ_1 minimization to generate estimates for the unmeasured paths. We describe heuristic approaches for the selection of the paths that should be monitored, or equivalently, where hardware monitors should be located. We demonstrate how our estimation framework can improve the efficiency of end-to-end delay estimation in IP networks and reduce the number of hardware monitors required to track bit-error rates in all-optical networks (networks with no electrical regenerators).

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Operations— *Network monitoring*; E.4 [**Coding and Information Theory**]: Data compaction and compression

General Terms

Algorithms, Measurement, Performance

Keywords

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 [6,20]. 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 [3,5], we exploit the notion that the performance on two overlapping paths should be correlated. For example, delay statistics of two paths with at least one link in common are correlated because packets from both flows are being delayed in a common queue. Similarly, in all-optical networks, two lightpaths with at least one link in common are correlated due to crosstalk across wavelengths. For either of these examples, it is possible to predict the performance on unmeasured paths using measurements of a few paths and knowledge of routing and the network topology. In this paper we also address the topic of exploiting temporal correlation for path-level performance monitoring.

Our methodology begins with the identification of a basis, which we design using *diffusion wavelets*, that enables us to accurately approximate the vector of path metrics using only a small number of non-zero coefficients. The coefficients efficiently summarize end-to-end performance on all paths. Diffusion wavelets generalize the concept of wavelets by providing a multiscale decomposition of functions defined on a graph [7]. The diffusion wavelet framework is applicable to a wide range of monitoring scenarios and allows us to simultaneously take advantage of spatial and temporal corre-

1

lation among the monitored paths. Posing the network monitoring problem as a wavelet coefficient estimation exercise allows us to make use of recently-developed, powerful tools from the theory of compressed sensing for estimating a sparse vector using a relatively small number of measurements.

The rest of the paper is organized as follows. Section 2 formally defines the monitoring problem and framework introduced in this paper, and provides an overview of diffusion wavelets and nonlinear estimators from the theory of compressed sensing. Section 3 describes how we use diffusion wavelets to construct compressible representations of the path metric functions. Section 4 discusses heuristics for selecting the paths to monitor and deciding where to place monitoring devices. Section 5 and Section 6 provide case studies of the application of our monitoring and estimation framework. The estimation of mean end-to-end Internet queuing delays is explored in Section 5, and Section 6 focuses on monitoring bit-error rate in all-optical networks. Section 7 provides concluding remarks and indicates avenues of future research.

2. COMPRESSED NETWORK MONITORING FRAMEWORK

2.1 Problem Formulation and Notation

Our goal in this paper is to accurately monitor performance metrics (e.g., end-to-end delays in an IP network or bit-error rates in an all-optical network) on a collection of n_p end-to-end paths using measurements on a subset of these paths. The size of the subset, $n_s < n_p$, is (ideally) much smaller than the total number of paths. Let $y^{(k)} \in \mathbb{R}^{n_p}$ denote the vector of performance values at time instant k on paths indexed from 1 to n_p; i.e., the ith component, $y_i^{(k)}$, is the performance value on the ith path at time k. Let $y_s^{(k)} \in \mathbb{R}^{n_s}$ denote the performance values we observe on the subset of measured paths. Given $y_s^{(k)}$, our task is to estimate $y^{(k)}$.

The measured and complete set of performance values are related via an $n_s \times n_p$ binary-valued selection matrix, $A^{(k)}$, defined such that $A_{i,j}^{(k)} = 1$ if the ith measured path corresponds to the jth entry of $y^{(k)}$; i.e., $(y_s^{(k)})_i = (y^{(k)})_j$. Using this notation, we have $y_s^{(k)} = A^{(k)}y^{(k)}$, where each row of $A^{(k)}$ contains exactly one non-zero entry (identifying which paths we measure), and each column of $A^{(k)}$ contains at most one non-zero entry (we measure each path at most once in each time-step).

In practice, we would like to monitor path-level performance over a sequence of time-steps, $k = 1, \ldots, \tau$. Stacking the path-level performance vectors from multiple time-steps into one vector, we write $y = [y^{(1)^T}, \ldots, y^{(\tau)^T}]^T$, and similarly, for the measured performance at each time-step we write $y_s =$

$[y_s^{(1)^T}, \ldots, y_s^{(\tau)^T}]^T$. It is convenient to combine the selection matrices from each time-step into a block-diagonal matrix, A, with the selection matrices at each time-step, $A^{(1)}, \ldots, A^{(\tau)}$, along the diagonal, so that we have $y_s = Ay$ relating the observations and path-level performance values over multiple time-steps.

A special case of the monitoring framework occurs when there is a known linear relationship between the link and path metrics. Suppose we are monitoring a network with n_l links, and assume we are given a $n_p \times n_l$ binary-valued routing matrix, G, where $G_{i,j} = 1$ if link j appears in the ith path and $G_{i,j} = 0$ otherwise. Many relevant performance metrics, including mean delay and delay variance, satisfy the property that the end-to-end metric on a path is equal to the sum of the performance metric of individual links in the path, so we can write $y^{(k)} = Gx^{(k)}$, where $x^{(k)} \in \mathbb{R}^{n_l}$ are the link-level metrics. This relationship not only provides valuable information about the correlation structure, but typically $n_l \ll n_p$, so we can achieve an immediate reduction in the dimensionality of the estimation task by formulating the problem in "link-space", i.e., writing $y_s = AGx$.

2.2 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 [8] 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 [7] 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 [7] 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 ith 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 sections we propose diffusion operators 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 nonlinear 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.

2.3 Estimation of Compressible Signals

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. In the following sections we describe diffusion wavelet bases B which efficiently compress the vector of end-to-end path metrics. We can express y in terms of its wavelet coefficients as $y = B\beta$, 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 $y_s = AB\beta$. This begs the question: *can we accurately recover the vector of coefficients, β, from measurements y_s?* In particular, we would like to take advantage of the fact that β only has a few large entries, and most are very small in magnitude or even zero.

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

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

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,

$$\widehat{\beta} = \arg\min_\beta \|\beta\|_1 \text{ subject to } y_s = AB\beta, \quad (1)$$

is equivalent to the ℓ_0 problem if certain conditions on A, B, and β are satisfied [2, 11, 14]. Here, $\|\beta\|_1 = \sum_{i=1}^n |\beta_i|$. Because the ℓ_1 optimization (1) is convex, it is computationally tractable, and a solution can be obtained using linear programming.

3. 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 path metrics. We first define the graph of interest $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We are measuring a performance metric function defined on the physical paths of the network. Accordingly, the vertex set $\mathcal{V}$ for our diffusion wavelet basis has one vertex for each path at each timestep, $\mathcal{V}^{(k)} = \{v_i^{(k)}\}_{i=1}^{n_p}$, over the estimation interval $k = 1, \ldots, \tau$.

3.1 Single Timestep: Spatial Diffusion

First consider the case where $\tau = 1$, that is, n_s routes are observed during one timestep k and we want to recover the metrics of the non-observed routes immediately. Let $\mathcal{G}^{(k)} = (\mathcal{V}^{(k)}, \mathcal{E}^{(k)})$ be the undirected graph over which we apply the diffusion wavelet framework. Notice that, in this work, the terms "vertex" and "edge" refer exclusively to the graph $\mathcal{G}^{(k)}$ defined in this section, while the terms "node", "link", and "path" refer to the physical nodes (e.g., routers), links and paths of the network; therefore, we can refer to "edges" be-

tween "paths". The graph $\mathcal{G}^{(k)}$ is defined as follows. The vertices $\mathcal{V}^{(k)}$ of $\mathcal{G}^{(k)}$ correspond to the paths of the network of interest, and there is an edge between the vertices v_i and v_j if and only if paths i and j share at least one link. Therefore, vertices of the graph are neighbors when their corresponding paths share a link. The function to be studied over $\mathcal{G}$ is the set of metrics associated with the vertices (paths).

We assign a weight $w_{i,j}$ to the edge between the vertices v_i and v_j to model the correlation between path metrics on routes that share the same links. The choice of these weights is problem-dependent and is determined by the anticipated relationship between link and path metrics. It effectively forms an a priori model for the correlation structure in the estimation problem.

For concreteness, we outline a methodology for choosing weights that is appropriate when there is an approximately linear relationship between path and link metrics. In this setting, it is reasonable to choose a weight that is proportional to the fraction of shared links in the two paths. Consider two paths i and j, and denote by $\mathcal{R}_i$ the set of links used by path i. We define the weight $w_{i,j}$ associated with the edge between v_i and v_j as:

$$w_{i,j} = \frac{|\mathcal{R}_i \cap \mathcal{R}_j|}{|\mathcal{R}_i \cup \mathcal{R}_j|}. \tag{2}$$

More weight is given to edges between paths that share many links, thereby emphasizing the spatial correlation intrinsic to end-to-end performance metrics — these edges are thus "spatial correlation edges", as opposed to the "time correlation edges", defined next.

The diffusion wavelet procedure in [7] requires a diffusion operator $D^{(k)}$ to generate a wavelet representation over $\mathcal{G}^{(k)}$. To obtain a diffusion operator from the construction described above, we apply Sinkhorn balancing [18] to the matrix of weights, $[w]$, to form a doubly stochastic matrix, D.

3.2 Multiple Timesteps: Incorporating Time Diffusion

When metrics change slowly with time, relative to the sampling rate, as is the case with mean end-to-end delays or BERs on lightpaths, time-correlation between the samples can be exploited to improve estimation accuracy. We account for time correlation in the diffusion operator as follows. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph with vertex set,

$$\mathcal{V} = \bigcup_{k=1}^{\tau} \mathcal{V}^{(k)} , \tag{3}$$

which is the union of the paths of the network at each timestep, and with edge set $\mathcal{E}$ such that:

$$\mathcal{E} = \left(\bigcup_{k=1}^{\tau} \mathcal{E}^{(k)} \right) \cup \left\{ \left(v_i^{(k)}, v_i^{(k+1)} \right) \right\}_{i\in\{1,...,n_p\},k\in\{1,...,\tau-1\}} .$$

In the edge set, the first term, $\bigcup_k \mathcal{E}^{(k)}$, is the union of the edge sets for each individual timestep, as introduced in the previous subsection. The second term contains time-correlation edges, which connect the subsets $\mathcal{V}^{(k)}$ of the vertex set $\mathcal{V}$ together: an edge is present between a path at timestep k and the same path at timestep $k+1$. We keep the weights already defined in the previous section for the edges between paths at the same timestep (spatial correlation edges) and we only need to define the weights for the inter-timestep edges.

With the a priori assumption that time correlation has the same strength across the network, we assign a weight of w_t to each of these new edges (time correlation edges), except for the edges between vertices of $\mathcal{V}^{(1)}$ and $\mathcal{V}^{(2)}$, and $\mathcal{V}^{(\tau-1)}$ and $\mathcal{V}^{(\tau)}$, which are assigned a weight of $2w_t$ (for balancing purposes). The specific value of w_t depends on the anticipated relative strengths of spatial and temporal correlation. We use a value of 0.5 in the examples described later in the paper, which reflects expectation of reasonably strong correlation; it is equivalent to the anticipated spatial correlation for paths that share half of their combined links.

The weight matrix $w \in \mathbb{R}^{n_p\tau \times n_p\tau}$ is then defined as the block matrix whose elements are equal to the weights defined above (I_{n_p} is the $n_p \times n_p$ identity matrix):

$$w = \begin{bmatrix} D^{(1)} & 2w_t I_{n_p} & 0 & \cdots & 0 \\ w_t I_{n_p} & D^{(2)} & w_t I_{n_p} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & w_t I_{n_p} & D^{(\tau-1)} & w_t I_{n_p} \\ 0 & \cdots & 0 & 2w_t I_{n_p} & D^{(\tau)} \end{bmatrix}. \tag{4}$$

Again, we perform Sinkhorn balancing on $[w]$ in order to obtain a doubly stochastic matrix D, which we use when constructing a diffusion wavelet basis.

3.3 Link-level Representations

A special case of our framework arises when there is a strictly linear relationship between path and link metrics. A good example is mean delay; the delay on links that share many routes is often strongly correlated because the same traffic sources are generating backlog in queues. In this case, it becomes valuable to formulate the problem in "link-space". In this formulation we have $y_s = AGB\beta$, where G is the $n_p \times n_l$ binary-valued routing matrix, B is an $n_l \times n_l$ diffusion wavelet basis, and β is now a vector of n_l coefficients. The task is still the estimation of y, and it should be emphasized that although the by-product of this formulation is an estimate of x, this is not the goal.

In general, $n_l \ll n_p$ so the link-space formulation significantly reduces the dimensionality of the problem. This formulation also means that any solution

automatically satisfies the known relationships between path and link metrics. In the path formulation outlined above, these relationships would have to be incorporated as constraints in the ℓ_1-minimization (1), making the optimization problem significantly more challenging. The derivation of a suitable diffusion operator proceeds exactly as described for the path case, but now the nodes in the graph are links in the physical network and the edges are weighted according to the fraction of shared paths.

4. PATH SELECTION

So far, we have discussed why and how it is possible to accurately estimate end-to-end metrics from a limited number of observations. However, we have not discussed how to select which routes should be observed. This problem is challenging and the appropriate approach depends on the measurement constraints or costs. We will examine two scenarios and propose heuristics. In the first scenario, we consider that the cost (or constraint) is the average number of measurements made per timestep. In this scenario, we do not constrain which paths are measured at each timestep. Rather, the set of paths measured can change from timestep to timestep. In the second scenario, we consider the case where the constraint is the number of monitoring devices and we must decide where to place them in the network. Each monitoring device can measure any incoming path on its interface.

4.1 Constraint on Number of Measurements

First we consider the case where the only constraint is on the total number of paths to be selected over a fixed number of timesteps. For this scenario, we adapt the path selection technique presented by Chua et al. in [5] to include our correlation model. The path selection procedure in [5] 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_l, where C_l is a non-singular matrix that satisfies $\Sigma_l = C_l C_l^T$. For example, $C_l = \Sigma_l^{1/2}$, and Σ_l is the covariance of x. Note that the estimation methodology in [5] is restricted to the case where path metrics are a linear combination of link-level performance values. In this case we can write $y = Gx$, 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 obtained by setting $\Sigma_l = I$. An algorithm (see Alg. 1) that implements this heuristic can be found in [13].

Algorithm 1 Path selection
Input: Matrix C_p
            Number of paths to select n_s
Output: Selection matrix A

1: Perform singular value decomposition (SVD) on C_p: $C_p = USV^T$ where S contains the singular values in descending order.
2: Perform QR decomposition with column pivoting on the first n_s columns of U: $QR = U_{(1,\dots,n_s)}^T P^T$.
3: return $A = P_{(1,\dots,n_s)}$, the matrix formed from the first n_s columns of P.

The intuition behind this heuristic is that most of the energy of the path metric signal should reside in the space 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 do not have access to the link covariance matrix, but the diffusion operator provides a model of the path-level covariance, Σ_p. We therefore set $\Sigma_p = D^\tau$ (recall D is the diffusion operator and τ the number of timesteps used to account for time-correlation). We then strive to identify a subset of rows of G that approximately spans the same space as the first n_s left singular vectors of the matrix $C_p = \Sigma_p^{1/2}$. In the case of a link-level representation, we set $\Sigma_l = D^\tau$ and use $C_p = GC_l$, with C_l defined as above. Path selection is performed using Algorithm 1.

4.2 Constraint on Number of Monitors

In this scenario, there is a more restrictive constraint. We have a limited number of monitoring devices, M, and we must choose where to place them. Our intuitive goal is the same as for the previous scenario: to approximately span the space where most signal energy is expected to lie. The same algorithm is applicable, but there is not a direct mapping from n_s to M, because the number of monitors required to measure n_s paths varies according to how many of the paths terminate at the same interface. We therefore iterate, running Algorithm 1 repeatedly for increasing values of n_s until M monitors are used. The resultant procedure is described by Algorithm 2. The output of the algorithm is a selection matrix A and a set of monitoring locations $\mathcal{E}_s$.

5. MEAN END-TO-END DELAYS

To illustrate the estimation technique presented in this paper, we use experimental delay data collected on the Abilene 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 400 five-minute intervals. There are thus 121 path metrics to be estimated at each time step. Owing to the large scale of the Abilene network,

Algorithm 2 Monitor location selection

Input: routing matrix G, covariance matrix C_p
            number of monitors to select M

Output: selection matrix A, monitor set $\mathcal{E}_s$.

1: $M' = M$.
2: Perform SVD on C_p: $C_p = USV^T$ where S contains the singular values in descending order.
3: **repeat**
4:     Perform QR decomposition with column pivoting on the first M' columns of U: $QR = U^T_{(1,\dots,M')}P^T$.
5:     Let $P' = P_{(1,\dots,M')}$ be the matrix formed of the first M' columns of P.
6:     $\mathcal{E}_s = \emptyset$.
7:     **for** i=1 ... M' **do**
8:         Interpret P' as a routing matrix and determine the last link ℓ on the path described by the ith row of P'.
9:         $\mathcal{E}_s = \mathcal{E}_s \cup \{\ell\}$.
10:    **end for**
11:    $M' = M' + 1$.
12: **until** $|\mathcal{E}_s| = M$.
13: return $A = P_{(1,\dots,M)}$, the matrix formed of the first M columns of P.

the experimental Abilene end-to-end delays are dominated by the propagation delays; those delays can be determined accurately using fiber maps. Therefore, we apply our estimation framework to (end-to-end) queuing delays, which are more variable. To obtain the end-to-end queuing delays from the end-to-end delays, we assume that the propagation delay for any path is the minimum end-to-end delay over the duration of the experiment for this path and subtract off the minimum for each path. In the remainder of this section, we designate by "end-to-end delay" the end-to-end queuing delays.

Fig. 2(a) provides a visualization of one of the wavelet basis vectors, which allows us to assess how the energy of wavelets is spatially distributed. In this figure, the size of each vertex of $\mathcal{G}$ is scaled according to the magnitude of corresponding wavelet coefficient. Visualization (vertex layout) of the graph is achieved through the application of Isomap [19], where the distances between vertices is set to the inverse of the weight matrix. The figure provides a clear depiction of the clustering induced by the routing matrix. There are two primary clusters of vertices (on the left and the right) corresponding approximately to links appearing primarily in east-west and west-east paths respectively. Additionally, nodes 14 and 17 (corresponding to links 14 and 17 in Figure 1) are separate from the clusters. These are two of the more "vertical" links in the network which are used in both east-to-west and west-to-east paths
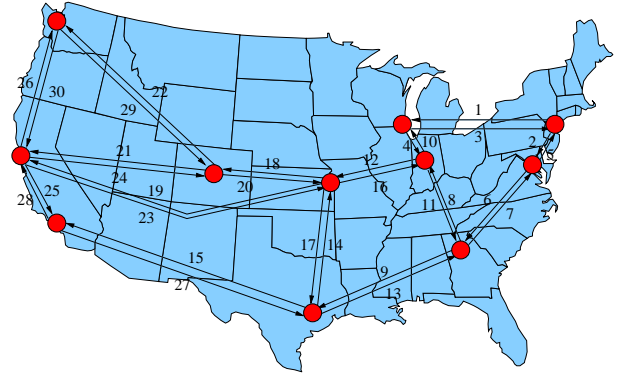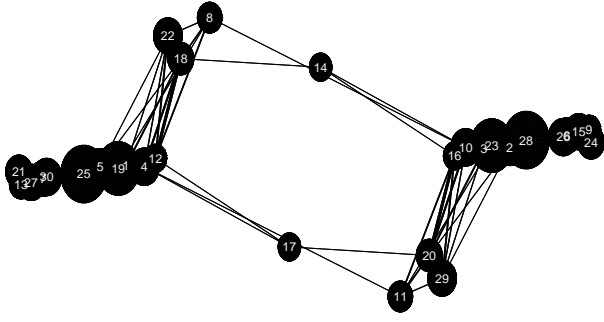

Figure 1: Abilene backbone: 11 nodes, 30 (unidirectional) links. The numbers are link identifiers.
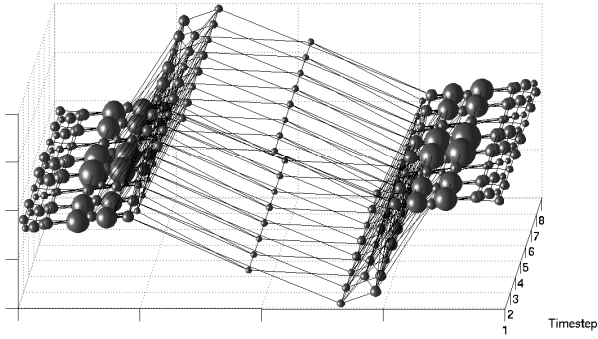
across the network.

Such graphical representation of wavelet basis vectors can be extended to the multi-timestep case. In the multi-timestep case, the time dimension is represented using a third dimension, as is shown in Fig. 2(b) for 8 timesteps. Vertical slices of the plot represent per-timestep network state while the sequence in time is represented over the labeled axis. Again, such visualization allows one to study where wavelet energy lies in space and time. In this case for example, the wavelet energy is concentrated on four links in the network most of the time.

We now verify the compressibility of the data. Fig. 3 shows the delays for all links and the absolute values of the diffusion wavelet coefficients, over $\tau = 1$ (top panel) and $\tau = 8$ (bottom panel) timesteps, sorted in descending order. The decay of the delays expressed in the original basis is very slow and exhibits a heavy tail. In the diffusion wavelet basis however, the link delays exhibit a power law-like decay, as can be seen by comparison to the reference functions $k \mapsto \alpha k^{-p}$. Moreover, the coefficients decay at a much faster rate in the 8-timestep case than in the 1-timestep case, indicating the value of incorporating time-correlation.

We have seen how, in the diffusion wavelet construction step (Section 2), deeper scales correspond to finer granularity. In the Fourier domain, deeper scales correspond to higher frequencies. Queuing delays are relatively low frequency signals, especially in the time dimension. During the nonlinear estimation step (1), estimated coefficients corresponding to high frequencies should be encouraged to be small; indeed, in case of estimation errors (unavoidable when the number of observations is low compared to the total number of coefficients to estimate), assigning high values to high-frequency coefficients leads to poor signal reconstruction. We make use of the knowledge that the signal to be estimated has a mainly low frequency spectrum as

(a) Single timestep graph.

(b) Multiple timesteps graph (The labeled axis is the time axis.)

Figure 2: Representation of an example wavelet basis vector on $\mathcal{G}$. Each vertex depicted here corresponds to a link of the network, and the thickness of the edge between vertices i and j increases relative to the weight w_{ij}. Each vertex is scaled according to the magnitude of the wavelet basis vector at the vertex. Vertex layout is determined by application of Isomap [19].

follows: we penalize in (1) the coefficients associated to deeper scales of the diffusion wavelet basis by them assigning weights ω_i, such that (1) becomes:

$$\widehat{\beta} \;=\; \arg\min_{\beta} \|\beta\|_1 \text{ subject to } y_s = AB\Omega\beta, \quad (5)$$

where Ω is the diagonal matrix such that $\Omega_{i,i} = \omega_i$. Recall each β_i is a coefficient in the diffusion wavelet basis. Following the discussion above, the weights should increase with the depth of the scale associated to β_i. Here, we chose a geometric increase in the weights: denoting by k the scale associated to a diffusion wavelet coefficient β_i, then $\omega_i = \alpha^k$ where α is a parameter that is fixed to 2 in the remainder of this section.

We show how our estimation techniques performs in Figure 4, which plots performance as the number of
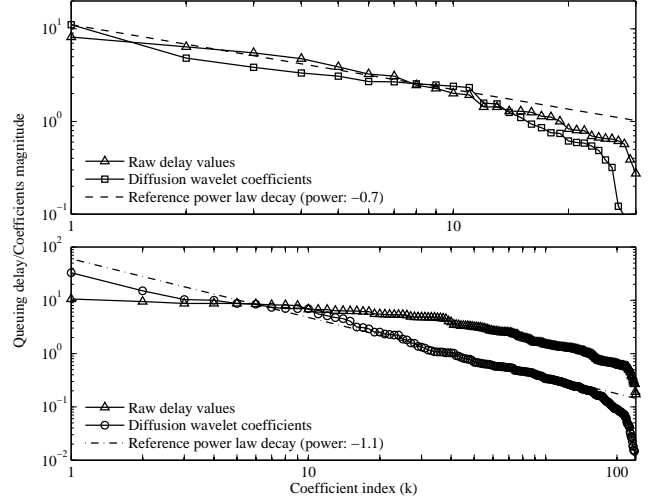

Figure 3: Link delays for the complete network over 1 (top panel) and 8 timesteps (bottom panel), sorted by magnitude, in the original basis (raw delay values) and in the diffusion wavelet basis (wavelet coefficients magnitudes). For comparative purposes, we also show the power-law decay functions $k \mapsto \alpha k^{-p}$ (for constant α, for $p = 0.7$ in the 1-timestep case, and $p = 1.1$ in the 8-timestep case).

paths-per-timestep varies from 1 to 30, with the block-size set to $\tau = 1$ and $\tau = 8$ timesteps. For the $\tau = 8$ case, end-to-end delays are estimated by blocks. Our dataset includes measurements for all paths in the network, so we can verify the accuracy of our estimation procedure against ground truth. We assess performance in terms of the relative end-to-end mean delay error, $|\langle y - y_{est}\rangle|/\langle y\rangle$, and the relative ℓ_2 error, $\|y - y_{est}\|_2/\|y\|_2$, where $\|y\|_2 = \sqrt{\sum_i y_i^2}$. Performance is averaged over 400 timesteps.

First consider the single timestep case, $\tau = 1$ in Fig. 4. We verify that the rank of G is equal to the number of links n_l and thus the observation of n_l paths ensures exact recovery of all end-to-end link delays with any estimation technique we are presenting. Our technique outperforms linear estimation (network kriging), with the performance improvement being most substantial when there are few measurements per timestep.

However, to fully harness the power of the nonlinear estimator, we need to consider data (and its diffusion, via the diffusion operator) over several timesteps. Now consider the block estimation case, $\tau = 8$, in Fig. 4; when less than 10 samples per timestep are collected, nonlinear estimation in a wavelet basis exhibits much lower estimation error than the linear estimator in terms of average end-to-end delay; ℓ_2 error is also lowered when nonlinear estimation in a wavelet basis is used.
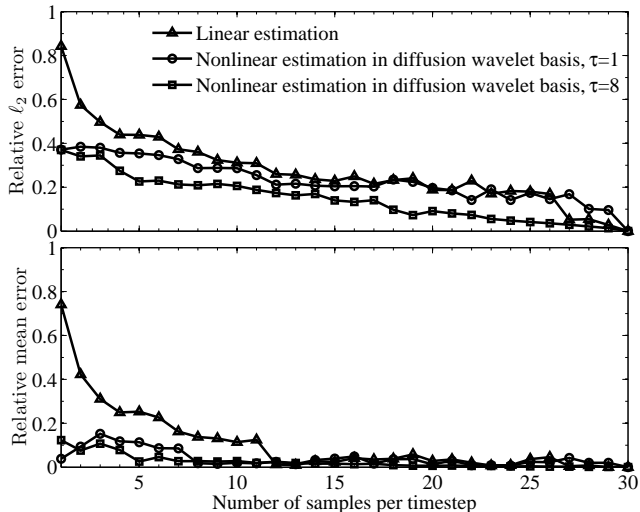
Figure 4: Relative ℓ_2 end-to-end delay error (top) and relative mean error (bottom) as functions of the average number of measurements per timestep (for $\tau = 1$ and $\tau = 8$), for our nonlinear estimation framework and the linear estimator [5].

In terms of mean end-to-end delay (Fig. 4, bottom panel), our results suggest that by making only 3 measurements per timestep we can hope to recover the mean network end-to-end delay with an error of less than 10%. The error stabilizes for larger number of samples per timestep and decreases to 0 as the number of samples per timestep approaches 21.

In the last figure (see Fig. 5), we provide a more detailed insight into the nature of our end-to-end delay recovery techniques. We show the recovered end-to-end delay (original data, estimation via nonlinear estimation in the wavelet basis and path selection accounting for time-correlation, and linear estimation) over time for 2 different paths. We used $\tau = 8$ and 10 samples per timestep in the estimation procedure. In Fig 5 (bottom panel), for example, we see that linear estimation severely underestimates the end-to-end delay for the chosen path. In general, the linear estimation exhibits substantial bias. In contrast, the nonlinear estimator exhibits much less bias but more variability. It is possible to estimate the bias if we are provided measurements of all link-level queueing delays (or can make sufficient estimates to form unbiased estimates.) However, such observations are not always available and we focus here on the case where estimating the bias is not possible. In the following section, we study an application to our technique where physical constraints on the observations prohibit the utilization of full-ranked observations to precompute the bias
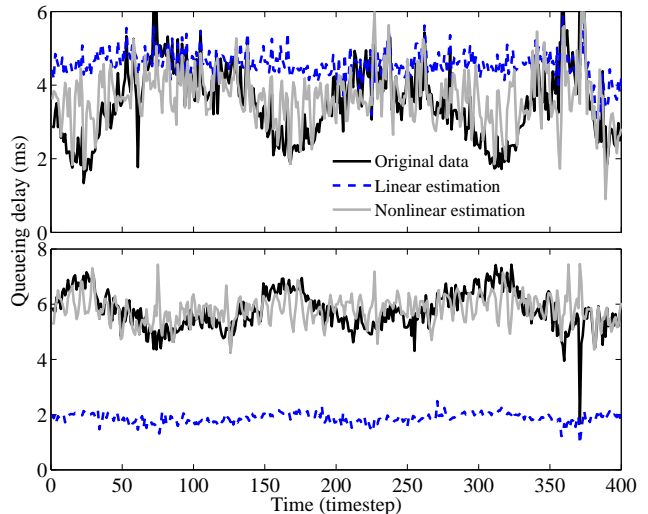
The presented nonlinear estimation technique is out-

Figure 5: Comparison between nonlinear estimation and linear estimation of path delays for two example paths.

performs the standard linear estimation technique. In terms of computation time, on standard hardware, most of the time is spent computing the basis B (seconds to minutes depending on τ for the Abilene topology). This is a one-time cost since B only depends on the network topology. The nonlinear estimation part is typically an order of magnitude slower than linear estimation, however it only takes tens to hundreds of milliseconds, depending on τ, to estimate a block of end-to-end delays for all paths (110τ end-to-end delays), making the technique deployable for real-time monitoring in networks with tens of nodes.

6. ALL-OPTICAL NETWORK MONITORING

In this section, we apply the compressive network monitoring framework derived in this paper to the particular case of bit-error rate (BER) monitoring in all-optical networks. We address the problem of monitoring circuit-switched all-optical networks with no wavelength conversion subject to a variety of physical impairments. More specifically, we tackle the specific case where signal statistics (which are used to determine signals' bit-error rates) can only be measured at certain locations. This is a key issue in all-optical networks since the equipment needed to take measurements at one location is extremely costly. The problem is then two-fold: given fixed BER monitors and hence the BER of observed lightpaths, what is the best estimate of the BERs of unobserved lightpaths? Then, how should BER monitors be placed to facilitate the estimation problem?

All-optical networks are high-speed, optical networks where OEO (optical-electrical-optical) conversion, which takes place at the nodes in traditional optical networks (e.g., SONET networks), is removed [21]. In all-optical networks, signal are transmitted in the optical domain with no electrical regeneration from end to end. In the nodes, which are called optical cross-connects (OXCs), signals are switched spatially in the optical domain [10]. The absence of OEO conversion allows (among other benefits) all-optical networks to bypass the capacity bottleneck incurred by the relatively low speed of electronic components, as data processing at 40 Gbit/s and above requires expensive devices. However, removing OEO conversion results in two main practical issues for all-optical network operation and management.

First, signals are propagated over very long distances without electrical regeneration and physical impairments accumulate as signals propagate in optical fiber and OXCs. Recently, network-layer techniques, namely, Routing and Wavelength Assignment (RWA) techniques, have been harnessed to counter these physical impairments. Assuming circuit-switched networks with no wavelength conversion[1], a RWA algorithm chooses a route and a wavelength (the combination of which is called "lightpath" [4]) to accommodate each incoming call at call admission time. It is possible to increase the quality of transmission in optical networks by using appropriate RWA techniques [15, 16]. In this paper, we make no assumption regarding the particular RWA used in the network.

The second major issue in all-optical networks is the absence of OEO converters, which makes monitoring difficult. Indeed, in traditional, non all-optical networks, signals are detected at each node, allowing error detection and correction. For example, SONET frames carry parity bits to detect errors [17]. Monitoring in all-optical networks is therefore restricted, both in terms of what can be measured and where it can be measured. Since electrical signals are not available at intermediate nodes of a lightpath, only a few optical quantities such as the optical power of the signal are measurable, and obtaining such intermediate measurements requires expensive optical spectrometers. Error detection can only be performed at the edge of the network, since that is the only place where electrical conversion is performed.

6.1   All-Optical Network Model

We consider circuit-switched all-optical networks where data is carried over *lightpaths*, that is, the combination of a route (assumed to be fixed for the duration of the call) and a wavelength, fixed from start to end of the route. Opaque networks, which allow wavelength conversion within a route, are beyond the scope of this paper. Links are assumed to be unidirectional and each link can carry C channels (wavelengths) simultaneously. A model for a lightpath is depicted in Fig. 6. The figure represents the lightpath and the sources of physical impairments considered in this paper. Other physical devices such as dispersion compensators and multiplexers/demultiplexers, which are assumed not to further degrade the signal's SNR, are not represented here. At the source of a call, a transmitter, located at an OXC, modulates data and sends it over optical fiber as an on-off keyed signal over a given wavelength. As it is transmitted over the optical fiber, the signal sustains chromatic dispersion and self-phase modulation which combine and contribute to intersymbol interference (ISI).

The transmitted signal is also subject to nonlinear crosstalk, that is, the nonlinear interaction with other signals that are transmitted simultaneously over the same fiber spans: cross-phase modulation and four-wave mixing. Optical amplifiers inject amplifier spontaneous emission (ASE) noise, and the signal is also subject to node crosstalk, which refers to signal leakages caused, for instance, by imperfect filtering at the nodes [12]. We refer the reader to [23], [22], [9], [15] for more details regarding the models of ASE noise, nonlinear crosstalk, node crosstalk, and their combined effects, respectively. Note that we ignore here a number of physical impairments such as receiver noise and polarization mode dispersion, but these effects can be incorporated easily in our model as additional noise variances, as will be seen shortly. Fig. 6 also illustrates the physical degradation of the transmitted signal in terms of an eye diagram; at the receiver, the eye diagram gets closed, thereby indicating a degraded SNR.

We denote by μ_0 and μ_1 the means of the distributions of the "0" and "1" samples, respectively, and by σ_0 and σ_1 their standard deviations. Let

$$Q = \frac{\mu_1 - \mu_0}{\sigma_0 + \sigma_1} \qquad (6)$$

be the *Q-factor* associated with the considered lightpath. The Q-factor can be interpreted as a signal-to-noise ratio, from which we can derive the bit-error rare, using a Gaussian assumption [1]:

$$BER = \frac{1}{2}\text{erfc}(\text{Q}/\sqrt{2}). \qquad (7)$$

We model each of the physical impairments described above by a noise variance in the SNR of the signal. Assuming these effects are statistically independent, these variances due to these effects are additive. Let σ_{isi}^2 be the noise variance caused by ISI, σ_{ase}^2 the noise variance caused by ASE noise, σ_{nl}^2 the noise variance caused by nonlinear crosstalk, and σ_{oxc}^2 the noise variance caused

---

[1]All-optical packet switched networks and wavelength conversion devices are currently at the experimental stage and are not ready for industrialization.
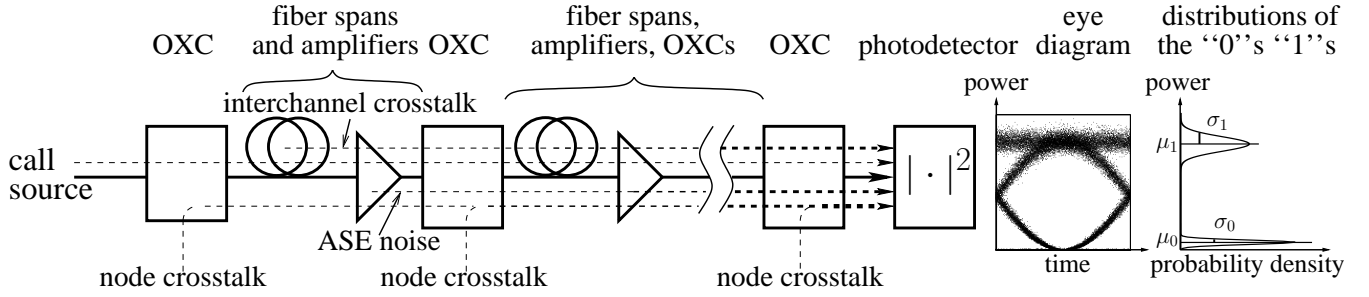
Figure 6: Model for a lightpath in an all-optical network, and sources of physical impairment. The signal traverses nodes (OXCs), spans of optical fiber and optical amplifiers before reaching destination, where it is detected by a photodetector — represented here by a square law device. Each device degrades the SNR of the signal: nodes inject node crosstalk, fiber spans injects nonlinear crosstalk, and amplifiers inject amplifier (ASE) noise. The BER associated to the signal can be computed from the distributions of the received "0" and "1" samples, and is related to the appearance of the eye diagram of the signal.

by node crosstalk, then we have $\sigma_1^2 = \sigma_{isi}^2 + \sigma_{ase}^2 + \sigma_{nl}^2 + \sigma_{oxc}^2$. Therefore, determining the BER of a lightpath boils down to determining four quantities, which can be measured at receivers using adapted equipment: μ_0, μ_1, σ_0, σ_1. In the remainder of this section, the BER estimation for a lightpath designates the simultaneous estimation of these four quantities.

In this work, we consider that the BER of a lightpath depends only on the network state, that is, on the network topology and on the lightpaths that are already established in the network. Indeed, in our model a Q factor depends on the topology via μ_0, μ_1, σ_0, σ_{isi}, σ_{ase}, and the crosstalk injected by other lightpaths via σ_{oxc} and σ_{nl}. In particular, we consider that the network is an event-driven system where events are lightpath establishment and tear-down. A timestep here thus consists in the arrival or the termination of new call. Cases where BERs vary between calls arrivals and departure, e.g., because of link failures, can be easily dealt with by sampling the BER measurements on a regular basis. Therefore, in this section, we denote by k a timestep (equivalently, a network state) and call $y_{\mu_0}^{(k)}$, $y_{\mu_1}^{(k)}$, $y_{\sigma_0}^{(k)}$, and $y_{\sigma_0}^{(k)}$ the vectors of the quantities we want to estimate, respectively, μ_0, μ_1, σ_0 and σ_1 for all lightpaths established at time k. We denote by $G^{(k)} \in \mathbb{R}^{n_p^{(k)} \times n_l}$ the routing matrix at time k, where $n_p^{(k)}$ is the number of established lightpaths at time k and n_l is the number of (unidirectional) links in the network. Each row of $G^{(k)}$ corresponds to an established lightpath and $G_{i,j}^{(k)} = 1$ when lightpath i uses link j. Contrary to the network delay case, here $G^{(k)}$ varies with k — in particular, the routing matrices at two different timesteps may not even have the same number of rows.

Our goal here is to estimate the BER of all lightpaths, at all timesteps, given a reduced number of lightpaths have actually been observed. To do so, we are using the spatial and time correlation between lightpaths. The spatial correlation is induced by the physical behavior of the network: physical impairments are caused at the link level and thus the BERs of two different lightpaths (on different wavelengths) sharing links are correlated. The time correlation is induced by the stationarity of the BERs with time; between two timesteps, only one lightpath can be established or torn down, thus the BER of a given lightpath between times k and $k+1$ varies little. Before we turn to the estimation problem, which will again be expressed in the diffusion wavelet framework, we first address the problem of sample (lightpath) selection, which we recast as the problem of physically placing BER monitor devices in an all-optical network.

6.2 BER Monitor Placement

In the context of all-optical networks, it is not possible to observe samples (that is, to measure the BER of lightpaths) independently from one time step to the next. Monitors are physical devices that cannot be moved from one site to another. Each monitor is located at a node, at the end of a link, and all lightpaths that terminate at this link can be observed — each monitor can thus observe up to C lightpaths simultaneously. However, lightpaths traversing but not ending at a monitored link cannot be observed by the BER monitor since those lightpaths' signals remain in the optical domain. If we could equip all links with a BER monitor, then the BER of all lightpaths in the network would be known at all times. However, this brute force monitoring scheme is very expensive and does not scale. In this section, we consider the scenario where we are given a fixed budget, or, equivalently, a number M of BER monitors. The problem is thus to select links where the monitors should be placed so as to facilitate the estimation of BERs of the lightpaths that are not

directly observed. This corresponds to the second path selection scenario described in Section 4.

Note that the physical constraint that BER monitors are fixed is actually very restrictive. BER monitors are fixed before the network starts operating. The number of observed lightpaths varies with time and it is possible that no lightpath is observed at all if no established lightpath ends at a link where a monitor is placed. The freedom to observe different (light)paths at different timesteps is lost. The situation is made substantially more complex if alternate or adaptive routing is used. In alternate routing, $K > 1$ shortest paths are pre-computed between any two nodes, such that if no wavelength is available on the shortest path between two nodes to accommodate some call, another route can be chosen to accommodate the call; adaptive routing can be viewed as the case $K = \infty$. Indeed, with fixed non-alternate routing ($K = 1$), we can exploit foreknowledge of the routes used by lightpaths to place monitors. This is not possible with alternate or adaptive routing. For the purpose of path selection (but not estimation), we assume that routing is fixed (non-alternate, non-adaptive) and $K = 1$. We then compute the shortest path routing matrix and use this in Algorithm 2, together with C_p derived from the diffusion wavelet basis, to determine the locations of the monitors.

6.3   Numerical Results

We apply the estimation framework described in Section 2 to the bit-error case, estimating in turn μ_1, μ_0, σ_1 and σ_0 for all lightpaths, at all times. We simulate the operation of an all-optical network where BERs are computed according to the model described in Section 6.1. Physical-layer parameters for the network are described in [15]. We simulated the arrival and departure of 350 calls in a downscaled version[2] of the NSF network, depicted in Fig. 7. This topology contains 14 nodes and 42 unidirectional links. We used $C = 8$ wavelengths in the simulations and adaptive routing. When a network starts operating, there is no lightpath yet established in the network, and when a sufficient number of calls have arrived, the number of lightpaths in the network ceases to increase and the network operates in steady state. Our simulation results only account for the steady-state operation of the network, not for the initial period where calls keep arriving without departing.

We illustrate the compressibility of each of the four metrics μ_0, μ_1, σ_0 and σ_1 for $\tau = 8$ timesteps in Fig. 8. All metrics are highly compressible in the diffusion wavelet basis, allowing for the utilization of the nonlinear esti-

---

[2]It is currently not possible to build a continental-sized all-optical network; we modeled a regional-sized network, based on the NSF topology.
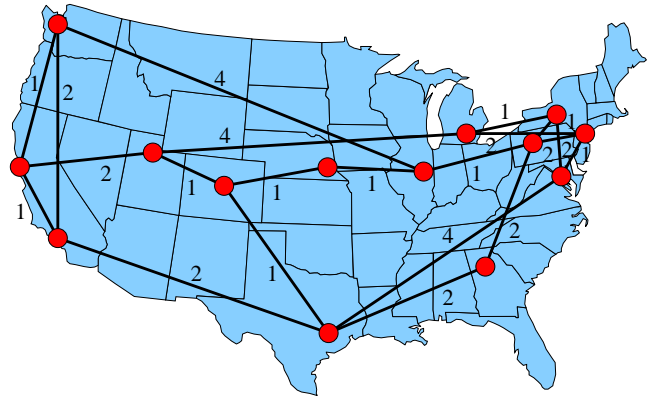

Figure 7: **Down-scaled version of the NSF topology (scaling factor: 1/10) used to perform the simulations. On the figure, the weights represent the number of 70-km spans for the links. Each link is bidirectional.**
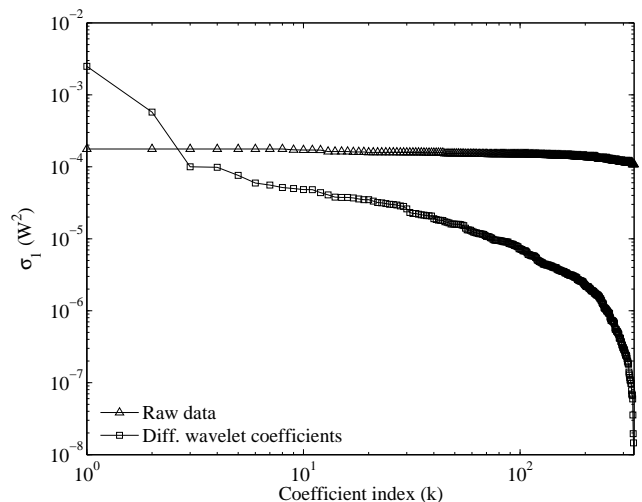

Figure 8: **Compressibility of σ_1 for $\tau = 8$ timesteps and $L = 10$ diffusion wavelet scales. All three other metrics μ_0, μ_1 and σ_0 exhibit a similarly fast decay in the diffusion wavelet basis.**

mation framework.

We compare the performance of the nonlinear estimator in a diffusion wavelet basis with the linear estimation framework presented in [5]. Contrary to the nonlinear estimation framework where correlation between end-to-end metrics is accounted for via the diffusion operator, the linear estimation framework requires that there exist a linear relation between the link-level and the end-to-end delay metrics. Although such a linear relation follows directly from the physics of the problem in the end-to-end delay case, such is not the case here. However, we can show that, after appropriate transformations, each end-to-end metric can be approximated

by a linear combination of link-level metrics.

Recall that $G^{(k)}$ is the routing matrix of the network at timestep k. In Section 2.1, the linear relation $y^{(k)} = G^{(k)}x^{(k)}$ where $y^{(k)}$ is a per-path metric and $x^{(k)}$ a per-link metric was induced by the additivity of the link delays to form end-to-end delays. In the bit-error rate estimation case, the per-link variances for (node and nonlinear) crosstalk are additive by construction, suggesting that in the linear estimation framework we should estimate the variance σ_1^2 rather than estimating the standard deviation σ_1. This insight is verified in Fig. 9 (bottom-right plot), where we plot the (end-to-end) variances σ_{isi}^2 and σ_{nl}^2 as a function of the length of a lightpath. The variances σ_i^2 and σ_{ase}^2 are approximately linear with respect to the lightpath length; therefore the per-link variances $x_{\sigma_1^2}$ are approximately additive (accounting for an offset which can be precomputed, and which is due to ISI induced by the filtering process at the receiver), leading to:

$$y_{\sigma_1^2} = Gx_{\sigma_1^2} + \sigma_{1_0}^2, \tag{8}$$

where $\sigma_{1_0}^2$ is a hardware and signal modulation format-dependent constant that can be pre-computed and subtracted away.

Similarly, we show in Fig. 9 that μ_1^2, μ_0^2, and σ_0^2 are also (essentially) linearly related to the number of links, thereby suggesting the approximations

$$y_{\mu_1^2} = Gx_{\mu_1^2} + \mu_{1_0}^2, \tag{9}$$
$$y_{\mu_0^2} = Gx_{\mu_0^2} + \mu_{0_0}^2, \tag{10}$$
$$y_{\sigma_0^2} = Gx_{\sigma_0^2} + \sigma_{0_0}^2 \tag{11}$$

where μ_{1_0}, μ_{0_0}, σ_{0_0} are pre-computed constants that can also be subtracted from the measurements.

Let $y^{(k)}$ denote the column-vector containing one of the metrics of interest ($y_{\mu_1^2}$, $y_{\mu_0^2}$, $y_{\sigma_1^2}$, or $y_{\sigma_0^2}$) for all lightpaths established in the network at timestep k. We denote by $x^{(k)}$ the corresponding per-link metrics. Since the constants $\sigma_{1_0}^2$, $\mu_{1_0}^2$, $\sigma_{1_0}^2$ and $\mu_{1_0}^2$ can be precomputed and subtracted from $y_{\sigma_1}^2$, $y_{\mu_1}^2$, $y_{\sigma_0}^2$ and $y_{\mu_1}^2$, respectively, we have a problem of the general form:

$$y^{(k)} = G^{(k)}x^{(k)}. \tag{12}$$

Consequently, in the following, the nonlinear estimation framework is used to estimate the quantities μ_0, μ_1, σ_0 and σ_1. The linear framework is used to estimate the transformed versions of μ_0, μ_1, σ_0 and σ_1, then μ_0, μ_1, σ_0 and σ_1 are determined by inverting (8)-(11). In both cases, BERs are determined via Q factors using (6) and (7).

We now compare the performance of nonlinear estimation in diffusion wavelet bases with that of the linear estimator presented in [5]. Furthermore, we provide the following lower bound for the nonlinear estimate. Assume that we were given the possibility to observe lightpaths directly in the diffusion wavelet basis. Then, the
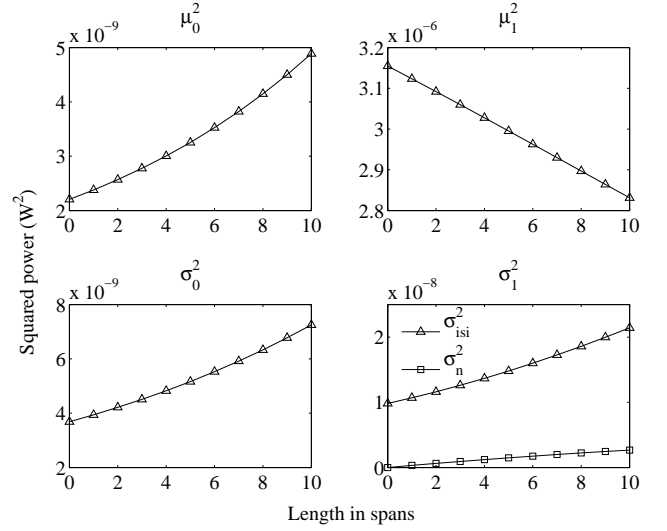

Figure 9: Linearity of the physical impairments metrics with respect to distance (number of spans on a lightpath): after a simple transformation, all metrics are approximately linear, thereby allowing the utilization of linear estimation.

best (unachievable) n_s-term estimator would involve selecting the coefficients with the highest magnitudes in the diffusion wavelet basis and setting all other coefficients to zero. The performance of this estimator provides a lower bound; we cannot directly observe projections onto the wavelet basis functions and we do not know, a priori, which n_s terms have the highest magnitude, so this bound is very loose in general.

Note that, because monitors are fixed, it is possible that some lightpaths established in the network are left completely unobserved, that is, that none of the links of these lightpaths are measured at any time in the monitoring period. These completely unobserved lightpaths are ignored in our evaluations (since BER estimates for these lightpaths are based on no information and can only be arbitrary). If a lightpath uses a link that is part of another lightpath which is directly observed, we call this lightpath an "estimable" lightpath. It is possible that an estimator returns a physically meaningless estimate (BER) for an estimable lightpath (e.g., a negative BER). We call lightpaths for which physically meaningful estimates are returned "estimated lightpaths". We show in Fig. 10 the proportion of estimable lightpaths to the total number of lightpaths in the network over the simulation time, and the proportion of estimated lightpaths (again, with respect to the total number of lightpaths in the network over the simulation time) for each of the estimators we tested. The monitor placement algorithm is seen to perform adequately: even if only 5 monitors are used, more than 60% of the light-
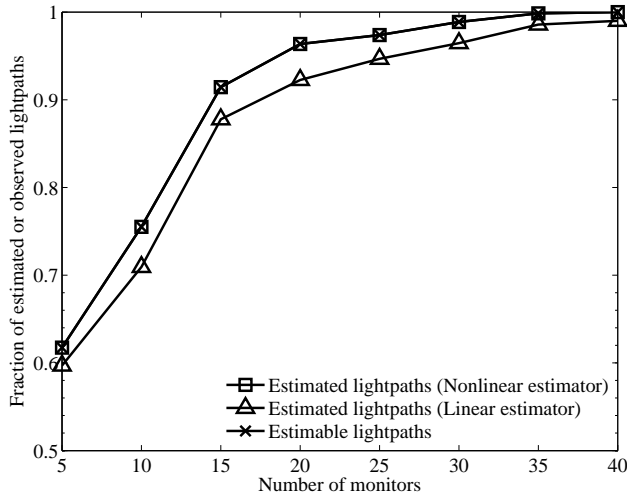
Figure 10: Fraction of *estimable* and estimated lightpaths. The BERs of some lightpaths are not *estimable* because none of their links is observed through any other lightpath; among estimable lightpaths, some are not *estimated* at all because the estimates returned by the estimator were physically meaningless (e.g., negative BER).

Figure 11: Relative ℓ_2 (top panel) and mean error (bottom panel) for the two estimators, for log(BER). Also depicted is the lower bound on the nonlinear estimation described in the text.

paths are estimable. This proportion rises to 90% if 15 monitors are used. The nonlinear estimator estimates the BER of all of the estimable lightpaths, whereas the linear estimator consistently leaves the BER of a small proportion (5–10%) of estimable lightpaths unestimated unless a very high number (35 and more) monitors are installed in the network.

In optical networks, only the order of magnitude of the BER is relevant and hence we work solely with log(BER) to evaluate the performance of the estimators. We compare in Fig. 11 the performance of the linear and the nonlinear estimators in the diffusion wavelet framework for the relative ℓ_2 error $||(y - y_{est})||_2/||y||_2$ (top panel) and the relative mean error $|\langle y - y_{est}\rangle|/\langle y\rangle$ (bottom panel), where y is the vector containing the log of the BER for each lightpath, at each time instant. We also give 5% confidence intervals. The performance improvement achieved by the nonlinear estimation technique is largest when few monitors are available. In particular, when 15 monitors or less are placed in the network (out of a maximum of 42 monitors), corresponding to a maximum of 90% of estimated lightpaths, the nonlinear estimation technique exhibits a significant advantage over the linear estimator in terms of ℓ_2 norm. In terms of mean BER, the nonlinear estimator is able to predict the true mean BER over the network even with a very small number of monitors (less than 1% error in mean on log(BER) with 5 monitors), while the linear estimator requires 25 monitors to achieve the same
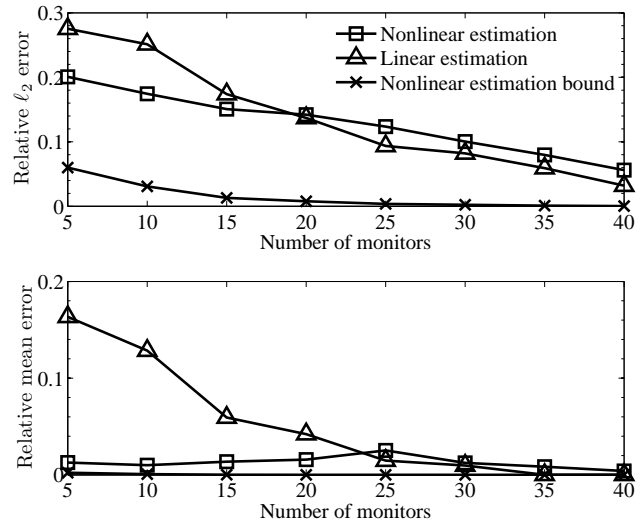
performance. As was the case for end-to-end delays, the nonlinear estimator has a very low bias. When the number of estimators increase the gap between the nonlinear and linear estimation techniques closes and linear estimation actually performs slightly better than the nonlinear estimation. We emphasize that practical situations are really those where the number of monitors is small, which is when our nonlinear framework applies best and performs best.

Moreover, the nonlinear estimation technique applies to more general situations than the linear estimation framework. Indeed, for the linear estimation framework to apply, we need to identify a linear relationship between link-level (x) and lightpath-level metrics (y). For the case of lightpath BER estimation, we were able to define an approximately linear relationship for transformed metrics. This artificial construct is unnecessary in our nonlinear estimation framework, since correlation between lightpaths is naturally modeled through the diffusion operator. Finally, we give in Fig. 11 lower bounds on the performance of the nonlinear estimator. These lower bounds are substantially lower than what is achieved by our nonlinear estimator, which is expected given we picked coefficients directly in the diffusion wavelet basis to construct the bound.

7. CONCLUSION

We have presented a framework for monitoring path metrics based on incomplete end-to-end measurements. The core of the framework is the development of a basis

in which the path metric signal is compressible, which allows us to use powerful nonlinear estimators from the theory of compressed sensing. Diffusion wavelets provide an appealing mechanism for developing the basis, because the specification of a diffusion operator allows us to create very general models for the correlations between metrics on different paths. Case studies involving the estimation of mean end-to-end delays and the monitoring of lightpath BERs in all-optical networks indicate the promise of our framework. Currently we are investigating the development of alternate bases which can better capture spatial localization of signal changes. We are also developing theoretical bounds on the number of paths that need to be measured to achieve a specified accuracy.

8. REFERENCES

[1] G. Agrawal. *Fiber-Optic Communications Systems*. John Wiley & Sons, Inc., third edition, 2002.

[2] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, Feb. 2006.

[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, USA, Aug. 2004.

[4] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: a novel approach to high bandwidth optical WANs. *IEEE Trans. Commun.*, 40(7):1171–1182, July 1992.

[5] D. Chua, E. Kolaczyk, and M. Crovella. Efficient monitoring of end-to-end network properties. In *Proc. Infocom*, Miami, USA, Mar. 2005.

[6] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Mag.*, May 2002.

[7] R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, July 2006.

[8] M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In *Proc. IEEE Infocom*, San Francisco, USA, Mar. 2003.

[9] T. Deng, S. Subramaniam, and J. Xu. Crosstalk-aware wavelength assignment in dynamic wavelength-routed optical networks. In *Proc. Broadnets*, Oct. 2004.

[10] P. D. Dobbelaere, K. Falta, L. Fan, S. Gloekner, and S. Patra. Digital MEMS for optical switching. *IEEE Commun. Mag.*, pages 88–95, Mar. 2002.

[11] D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, Apr. 2006.

[12] E. Goldstein and L. Eskildsen. Scaling limitations in transparent optical networks due to low-level crosstalk. *IEEE Photon. Technol. Lett.*, 7(1):93–94, Jan. 1995.

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

[14] J. Haupt and R. Nowak. Signal reconstruction from noisy random projections. *IEEE Trans. Inform. Theory*, 52(9):4036–4048, Sept. 2006.

[15] Y. Pointurier, M. Brandt-Pearce, T. Deng, and S. Subramaniam. Fair QoS-aware adaptive Routing and Wavelength Assignment in all-optical networks. In *Proc. IEEE ICC*, June 2006.

[16] B. Ramamurthy, D. Datta, H. Feng, J. Heritage, and B. Mukherjee. Impact of transmission impairments on the teletraffic performance of wavelength-routed optical networks. *J. Lightwave Technol.*, 17(10):1713–1723, Oct. 1999.

[17] R. Ramaswami and K. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, second edition, 2002.

[18] R. Sinkhorn. A relationship between arbitrary positive matrices and double stochastic matrices. *Ann. Mathematical Statistics*, 35(2):876–879, June 1964.

[19] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 5500(290):2319–2323, Dec. 2000.

[20] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *J. American Statistical Assosciation*, 91(433):365–377, Mar. 1996.

[21] A. Willner, M. Cardakli, O. Adamczyk, Y.-W. Song, and D. Gurkan. Key building blocks for all-optical networks. *IEICE Trans. Commun.*, E83-B:2166–2177, Oct. 2000.

[22] B. Xu and M. Brandt-Pearce. Comparison of FWM- and XPM-induced crosstalk using the Volterra Series Transfer Function method. *J. Lightwave Technol.*, 21(1):40–53, Jan. 2003.

[23] B. Xu and M. Brandt-Pearce. Analysis of noise amplification by a CW pump signal due to fiber nonlinearity. *IEEE Photon. Technol. Lett.*, 16(4):1062–1064, Apr. 2004.