

# Dynamic Network Delay Cartography

Ketan Rajawat, *Member, IEEE*, Emiliano Dall'Anese, *Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

**Abstract**—Path delays in IP networks are important metrics, required by network operators for assessment, planning, and fault diagnosis. Monitoring delays of all source-destination pairs in a large network are, however, challenging and wasteful of resources. This paper advocates a spatio-temporal Kalman filtering approach to construct network-wide delay maps using measurements on only a few paths. The proposed network cartography framework allows efficient tracking and prediction of delays by relying on both topological as well as historical data. Optimal paths for delay measurement are selected in an online fashion by leveraging the notion of submodularity. The resulting predictor is optimal in the class of linear predictors, and outperforms competing alternatives on real-world data sets.

**Index Terms**—Internet measurements, network kriging, kriged Kalman filter, delay prediction, submodularity optimization.

## I. INTRODUCTION

THE explosive growth in network traffic volumes has necessitated the development of avant-garde monitoring tools to endow network operators with a comprehensive view of the global network behavior. However, acquisition and processing of network-wide performance metrics for large networks is no easy task. For instance, monitoring path metrics such as delays or loss rates is challenging primarily because the number of paths generally grows as the square of the number of nodes in the network. Therefore, measuring and storing the delays of all possible source-destination pairs is hard in practice even for moderate-size networks.

Focus has thus shifted towards statistical means of predicting network-wide performance metrics using measurements on only a subset of nodes [1], [2]. A promising approach in this context has been the application of *kriging*, a tool for spatial prediction popular in geostatistics and environmental sciences [3], [4]. A *network kriging* approach was developed in [5], where network-wide path delays were predicted using measurements on a chosen subset of paths. The class of linear predictors introduced leverages network topology information to model the covariance among path delays. This is accomplished in [5] by assigning higher correlation between two paths if they share several links, as in this case, they are expected to incur similar delay variations.

Manuscript received March 21, 2012; revised November 11, 2012 and May 11, 2013; accepted December 12, 2013. Date of publication March 13, 2014; date of current version April 17, 2014. This work was supported by NSF-ECCS under Grant 1202135. This paper was presented at the 2012 IEEE Statistical Signal Processing Workshop.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: ketan@umn.edu; emiliano@umn.edu; georgios@umn.edu).

Communicated by F. Baccelli, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIT.2014.2311802

The present paper puts forth a *dynamic* network kriging approach capable of real-time spatio-temporal delay predictions. Specifically, a kriged Kalman filter (KKF) is employed to explicitly capture variations due to queuing delays, while retaining the topology-based kriging predictor. The resulting dynamic network kriging approach not only yields lower prediction error, but is also more flexible, allowing delay measurements to be taken on random subsets of paths. In this context, the problem of choosing the optimal paths for delay measurements is also considered. Since the KKF runs in real-time, the paths are also selected in an online fashion by minimizing the prediction error per time slot. Interestingly, the resulting combinatorial optimization problem is shown to be submodular, and is therefore solved approximately via a greedy routine. While dynamic Kriging is a well known tool in Geostatistics, the contribution here is its application in the networking context, and the development of the relevant online experiment-design framework.

Recently, a compressive sampling-based approach has also been reported for predicting network-wide performance metrics [6], [7]. For instance, diffusion wavelets were utilized in [6] to obtain a compressible representation of the delays, and account for spatial and temporal correlations. Although this allows for enhanced prediction accuracy over [5], it requires batch processing of measurements which does not scale well to large networks for real-time operation. In contrast, both the KKF and the greedy path selection algorithms entail sequential operations, and are therefore significantly faster.

Imputation of end-to-end delays has also been considered in the context of Internet geolocation. Treating end-to-end delays as distances between nodes, all-pair node distances are estimated using Euclidean embedding [8], or, matrix factorization [9]. However, these approaches do not exploit the temporal or topological information, since their focus is not on monitoring or extrapolation (that is, prediction) of delays.

The rest of the paper is organized as follows. Sec. II introduces the model and the problem statement. Sec. III deals with the KKF approach, while Sec. III-A describes techniques for estimating the relevant parameters. Finally, empirical validation of KKF and comparisons with the Kriging approach of [5] are provided in Sec. V.

**Notation.** Lower case symbols with indices, such as  $y_p$ , represent scalar variables. These variables, when stacked over their indices are denoted through their bold-faced versions  $\mathbf{y}$ . Bold-faced upper case symbols ( $\mathbf{S}$ ) represent matrices. Regular upper case symbols ( $S$ ) represent constant scalars, and typically stand for the cardinality of the set represented by corresponding calligraphic upper case symbol ( $\mathcal{S}$ ).

Identity matrix of size  $P \times P$  is denoted by  $\mathbf{I}_P$ , and its columns by  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_P$ . Matrix  $\mathbf{C}_y$  denotes the covariance matrix of the vector  $\mathbf{y}$ .

## II. MODELING AND PROBLEM STATEMENT

Consider an IP network modeled by a connected digraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with  $\mathcal{V}$  denoting the set of nodes (devices, servers, or routers), and  $\mathcal{E}$ , the communication links. The issue is to monitor path delays on a set of multi-hop paths  $\mathcal{P}$  that connect the  $P := |\mathcal{P}|$  source-destination pairs. Latency measured on path  $p \in \mathcal{P}$  at time  $t$  is denoted by  $y_p(t)$ , and all such network-wide delays are collected in the vector  $\mathbf{y}(t)$ . At any time  $t$  however, delay can only be measured on a subset of paths  $\mathcal{S}(t) \subset \mathcal{P}$ , which is represented by  $\mathbf{y}_s(t)$ . Based on such partial current and past measurements  $\mathcal{H}(t) := \{\mathbf{y}_s(\tau)\}_{\tau=1}^t$ , the goal is to predict the remaining path delays  $\mathbf{y}_{\bar{s}}(t) := \{y_p(t)\}_{p \in \mathcal{P} \setminus \mathcal{S}(t)}$  for each  $t$ .

The per-path end-to-end delay  $y_p(t)$  consists of several independent components corresponding to contributions from each intermediate link and router. Of these, the queuing delay  $\hat{\chi}_p(t)$  is the time spent by the packets waiting in the queues of intermediate buffers, and depends on the traffic volumes in competing links. Network traffic is not only correlated spatio-temporally, but also exhibits non-stationarities, in the form of random fluctuations and bursts [10]. Indeed, it is not surprising that the statistical properties of queuing delays in large IP networks are largely unknown. In the interests of model parsimony and amenability to the tools used later, the following random-walk model is instead adopted for the latent vector of queuing delays,

$$\hat{\chi}(t) = \hat{\chi}(t-1) + \boldsymbol{\eta}(t) \quad (1)$$

where  $\boldsymbol{\eta}(t)$  denotes state noise with zero mean, and covariance matrix  $\mathbf{C}_\eta := \mathbb{E}[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)]$ . Observe that the random-walk model has very few tuning parameters, compared to say, a model which includes a non-identity state transition matrix (i.e.,  $\hat{\chi}(t) = \mathbf{B}\hat{\chi}(t-1) + \boldsymbol{\eta}(t)$ ). Further advantages of the random-walk model, including those pertaining to the computational cost, are provided in later sections.

Other components of the path delay, combined in the nonzero-mean random  $\hat{v}_p(t)$ , include the propagation, processing, and transmission delays, which are temporally uncorrelated (see [11] for details). This component of delays is however spatially correlated across paths, and the covariance matrix of the compacted vector  $\hat{\mathbf{v}}(t)$  is given by  $\mathbf{C}_v$ . Finally, the measurement of path delays using software tools such as ping and OWAMP itself introduces errors  $\epsilon_p(t)$ , which are assumed zero mean, uncorrelated over time and across paths, with covariance  $\sigma^2 := \mathbb{E}[\epsilon_p(t)\epsilon_p^T(t)]$ .

The measured delays are expressed as

$$y_p(t) = \hat{\chi}_p(t) + \hat{v}_p(t) + \epsilon_p(t) \quad p \in \mathcal{S}(t). \quad (2)$$

Letting  $\mathbf{S}(t)$  denote the  $|\mathcal{S}(t)| \times P$  selection matrix with 0-1 entries that contains the  $p$ -th row of  $\mathbf{I}_P$  if  $p \in \mathcal{S}(t)$ , the measurement equation can be compactly written as

$$\mathbf{y}_s(t) = \mathbf{S}(t)\hat{\chi}(t) + \hat{\mathbf{v}}_s(t) + \boldsymbol{\epsilon}_s(t) \quad (3)$$

where the vector  $\boldsymbol{\epsilon}_s(t)$  collects the measurement errors on paths  $p \in \mathcal{S}(t)$ , and  $\hat{\mathbf{v}}_s(t) := \mathbf{S}(t)\hat{\mathbf{v}}(t)$ .

The next section describes a KKF approach for tracking and predicting the unknown end-to-end delays  $\mathbf{y}_{\bar{s}}(t)$ , by utilizing the state-space model described by (1) and (3).

## III. DYNAMIC NETWORK KRIGING

The spatio-temporal model in (1)-(3) is widely employed in geostatistics and environmental science, where  $\hat{\chi}(t)$  is generally referred to as trend, and  $\hat{\mathbf{v}}(t)$  captures random fluctuations around  $\hat{\chi}(t)$ ; see e.g., [3, Ch. 4], [12], and [13]. Recently, a similar modeling approach was employed by [14] to describe the dynamics of wireless propagation channels, and in [15] for spatio-temporal random field estimation. In order to better relate the proposed model with the existing ones, the mean of  $\hat{\mathbf{v}}(t)$  is incorporated in the trend, and (3) is now replaced with

$$\mathbf{y}_s(t) = \mathbf{S}(t)\boldsymbol{\chi}(t) + \mathbf{v}_s(t) + \boldsymbol{\epsilon}_s(t) \quad (4)$$

where  $\mathbf{v}_s(t) := \hat{\mathbf{v}}_s(t) - \mathbb{E}[\hat{\mathbf{v}}_s(t)]$  and  $\boldsymbol{\chi}(t) := \hat{\chi}(t) + \mathbb{E}[\hat{\mathbf{v}}(t)]$ , and likewise for  $\mathbf{v}(t)$ . Next, given only first- and second-order moments of  $\boldsymbol{\eta}(t)$ ,  $\boldsymbol{\epsilon}_s(t)$ , and  $\mathbf{v}(t)$ , this section derives the best linear predictor for the unavailable path delay vector  $\mathbf{y}_{\bar{s}}(t)$ .

Suppose first that the queuing delay vector  $\boldsymbol{\chi}(t)$  is known, and let  $\bar{\mathbf{S}}(t)$  denote an  $|\bar{\mathcal{S}}(t)| \times P$  matrix containing the  $p$ -th row of  $\mathbf{I}_P$  if  $p \in \bar{\mathcal{S}}(t)$ ; that is,  $\bar{\mathbf{S}}(t)$  is a path selection matrix which returns quantities pertaining to paths in  $\bar{\mathcal{S}}(t)$ . Then, the linear minimum mean-square error (LMMSE) estimator (denoted by  $\mathbb{E}^*[\cdot]$ ) for  $\mathbf{v}_{\bar{s}}(t)$  is given by (see, e.g. [16])

$$\begin{aligned} \mathbb{E}^*[\mathbf{v}_{\bar{s}}(t)|\boldsymbol{\chi}(t), \mathbf{y}_s(t)] &= \bar{\mathbf{S}}(t)\mathbf{C}_v\mathbf{S}^T(t) \\ &\times (\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_s)^{-1}[\mathbf{y}_s(t) - \mathbf{S}(t)\boldsymbol{\chi}(t)] \end{aligned} \quad (5)$$

and is commonly referred to as kriging [4]. In practice however, the trend  $\boldsymbol{\chi}(t)$  has to be estimated from the data. In the so-termed universal kriging predictor [3],  $\boldsymbol{\chi}(t)$  is estimated using the generalized least-squares (GLS) criterion, where  $\mathbf{v}_s(t)$  is treated as noise (lumped together with  $\boldsymbol{\epsilon}_s(t)$ ). The prediction for  $\mathbf{v}_{\bar{s}}(t)$  is then obtained by replacing  $\boldsymbol{\chi}(t)$  in (5) with its estimate. This approach was proposed for network delay prediction in [5], and was referred to as network kriging. However, since the trend is estimated independently using GLS per time slot, its temporal dynamics present in (1) are not exploited.

From the spatio-temporal model set forth in Sec. II, it is clear that estimating the trend  $\boldsymbol{\chi}(t)$  can benefit from processing both present and past measurements jointly. Towards this end, the Kalman filtering (KF) machinery offers a viable option for tracking the evolution of  $\boldsymbol{\chi}(t)$  from the set of historical data  $\mathcal{H}(t)$ . At each time  $t$ , the KF finds the LMMSE estimate  $\hat{\boldsymbol{\chi}}(t) := \mathbb{E}^*[\boldsymbol{\chi}(t)|\mathcal{H}(t)]$ , and its error covariance matrix  $\mathbf{M}(t) := \mathbb{E}[(\boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t))(\boldsymbol{\chi}(t) - \hat{\boldsymbol{\chi}}(t))^T]$  using the following set of recursions (see e.g., [16, Ch. 3])

$$\hat{\boldsymbol{\chi}}(t) = \hat{\boldsymbol{\chi}}(t-1) + \mathbf{K}(t)(\mathbf{y}_s(t) - \mathbf{S}(t)\hat{\boldsymbol{\chi}}(t-1)) \quad (6a)$$

$$\mathbf{M}(t) = (\mathbf{I}_P - \mathbf{K}(t)\mathbf{S}(t))(\mathbf{M}(t-1) + \mathbf{C}_\eta) \quad (6b)$$

where the so-termed Kalman gain  $\mathbf{K}(t)$  is given by

$$\mathbf{K}(t) := (\mathbf{M}(t-1) + \mathbf{C}_\eta) \mathbf{S}^T(t) \times [\mathbf{S}(t)(\mathbf{C}_v + \mathbf{C}_\eta + \mathbf{M}(t-1)) \mathbf{S}^T(t) + \sigma^2 \mathbf{I}_S]^{-1}. \quad (7)$$

Once  $\hat{\chi}(t)$  has been estimated via KF,  $\mathbf{v}_{\bar{s}}(t)$  can be readily obtained via kriging as in (5), yielding the predictor

$$\hat{\mathbf{y}}_{\bar{s}}(t) = \bar{\mathbf{S}}(t) \hat{\chi}(t) + \bar{\mathbf{S}}(t) \mathbf{C}_v \mathbf{S}^T(t) (\mathbf{S}(t) \mathbf{C}_v \mathbf{S}^T(t) + \sigma^2 \mathbf{I}_S)^{-1} \times [\mathbf{y}_s(t) - \mathbf{S}(t) \hat{\chi}(t)]. \quad (8)$$

The predictor in (8) constitutes what is also referred to as the kriged Kalman filter [12], [13]. The LMMSE framework employed here yields the best linear predictor even for non-Gaussian distributed noise. The prediction error of the KKF is characterized in the following proposition, whose proof is provided in Appendix A. A similar result appears in [12], but for the Gaussian noise case.

**Proposition 1.** *The prediction error covariance matrix at time  $t$  is given by*

$$\mathbf{M}_{\bar{s}}^y(t) := \mathbb{E}\{(\mathbf{y}_{\bar{s}}(t) - \hat{\mathbf{y}}_{\bar{s}}(t))(\mathbf{y}_{\bar{s}}(t) - \hat{\mathbf{y}}_{\bar{s}}(t))^T\} \quad (9a)$$

$$= \sigma^2 \mathbf{I}_{\bar{S}} + \bar{\mathbf{S}}(t) \left[ (\mathbf{M}(t-1) + \mathbf{C}_v + \mathbf{C}_\eta)^{-1} + \frac{1}{\sigma^2} \mathbf{S}^T(t) \mathbf{S}(t) \right]^{-1} \bar{\mathbf{S}}^T(t). \quad (9b)$$

Having a closed-form expression for the prediction error will come handy for selecting the matrix  $\mathbf{S}(t)$ , as shown later in Sec. IV.

The KF step also allows  $\tau$ -step prediction for  $\tau \geq 1$ , which is given by  $\hat{\mathbf{y}}(t + \tau) = \hat{\chi}(t)$ , since the kriging term is temporally white. In the present context, this can be useful in preemptive routing and congestion control algorithms, as well as for extrapolating missing measurements. In the latter case, the covariance matrix is updated simply as  $\mathbf{M}(t) = \mathbf{M}(t-1) + \mathbf{C}_\eta$ . Before concluding the description of the KKF, the following remarks are due.

**Remark 1.** The random walk model adopted in (1) may result in an unstable filter. Operationally, if the KKF is unstable, an incorrect initialization of  $\mathbf{M}(0)$  or  $\chi(0)$  may result in poor prediction performance even as  $t \rightarrow \infty$ . This can be remedied by adopting a damped model  $\chi(t) = b\chi(t-1) + \eta(t)$  with  $b < 1$ . Here,  $\chi(t)$  is a zero-mean random process which does not incorporate the mean of  $\hat{\mathbf{v}}(t)$ . The mean delay of all paths should instead be estimated a priori, and subtracted from the measurements themselves, so that each component of the path delay in (4) is zero-mean. With this modification, the results in this paper can be generalized to the damped case. The random walk model is nevertheless used here since no instability issues were observed in the two data sets considered in Sec. V. An alternative formulation, developed along the lines of [17], can also be used in the AR(1) case. This technique may however increase the number of state-space parameters, and considerably complicate the expressions developed in Sec. IV.

**Remark 2.** A distributed implementation of the KKF may be desirable for enhancing the robustness and scalability of delay monitoring. In large-scale networks, a distributed algorithm also mitigates the message passing overhead required

to collect all measurements at a fusion center. If the model covariances  $\mathbf{C}_v$  and  $\mathbf{C}_\eta$  are globally known, and the selection matrix  $\mathbf{S}(t)$  is constant for all  $t$ , a distributed implementation of (6) can be derived along the lines of [18]. To this end, notice that substituting (6a) in (8), one can re-write the KKF predictor as

$$\hat{\mathbf{y}}_{\bar{s}}(t) = \bar{\mathbf{S}}(t) [\mathbf{F}(t) - \mathbf{F}(t) \mathbf{S}(t) \mathbf{K}(t) + \mathbf{K}(t)] \mathbf{y}_s(t) + \bar{\mathbf{S}}(t) [\mathbf{F}(t) \mathbf{S}(t) \mathbf{K}(t) - \mathbf{K}(t) - \mathbf{F}(t)] \mathbf{S}(t) \hat{\chi}(t-1) + \bar{\mathbf{S}}(t) \hat{\chi}(t-1) \quad (10)$$

where  $\mathbf{F}(t) := \mathbf{C}_v \mathbf{S}^T(t) (\mathbf{S}(t) \mathbf{C}_v \mathbf{S}^T(t) + \sigma^2 \mathbf{I}_S)^{-1}$ . With  $\hat{\chi}(t-1)$  available from the previous iteration, it is clear from (10) that if  $\mathbf{d}(t) := [\mathbf{F}(t) - \mathbf{F}(t) \mathbf{S}(t) \mathbf{K}(t) + \mathbf{K}(t)] \mathbf{y}_s(t)$  were available at each node of the network, the KKF predictor (8) could be performed locally at each node. Assume that measurements are collected at a sub-set of nodes  $\mathcal{V}_s \subset \mathcal{V}$ , and node  $v \in \mathcal{V}_s$  measures delays of the set of paths  $\mathcal{S}_v \subset \mathcal{S}$ ; that is,  $v$  is the end-node of all the paths in  $\mathcal{S}_v$ . Then, to compute  $\mathbf{d}(t)$  in a distributed manner, consider rewriting it as a sum of  $|\mathcal{V}_s|$  terms, each involving only the local measurements  $\mathbf{y}_{s,v}(t) := \{[y_p | p \in \mathcal{S}_v]\}^T$ . Next, collect in the  $P \times |\mathcal{S}_v|$  matrix  $\Phi_v(t)$ , the columns of matrix  $\mathbf{F}(t) - \mathbf{F}(t) \mathbf{S}(t) \mathbf{K}(t) + \mathbf{K}(t)$  corresponding to the paths in  $\mathcal{S}_v$ . Then,  $\mathbf{d}(t)$  can be expressed as  $\mathbf{d}(t) = \sum_{v \in \mathcal{V}_s} \Phi_v(t) \mathbf{y}_{s,v}(t)$ , which is equivalent to [19]

$$\{\mathbf{d}_v(t)\}_{v \in \mathcal{V}_s} = \arg \min_{\{\mathbf{d}_v\}} \sum_{v \in \mathcal{V}_s} \|\mathbf{d}_v - |\mathcal{V}_s| \Phi_v(t) \mathbf{y}_{s,v}(t)\|_2^2 \quad (11a)$$

$$\text{s.t. } \mathbf{d}_v = \mathbf{d}_{v'}, \quad v' \in \bar{\mathcal{V}}_s, v \in \mathcal{V}_s \quad (11b)$$

where  $\mathbf{d}_v(t)$  represents a local copy of  $\mathbf{d}(t)$  at node  $v$ , and  $\bar{\mathcal{V}}_s \subset \mathcal{V}_s$  is the set of nodes communicating with  $v$ . Building on (11), an iterative consensus algorithm whereby each node  $v$  exchanges its local copy  $\mathbf{d}_v(t)$  only with nodes in  $\bar{\mathcal{V}}_s$ , can be derived by employing the so called alternating direction method of multipliers as detailed in [18] and [19]. Notice that, since the model covariances are globally known, recursions (9) can be performed locally at each node.

#### A. Estimating Model Parameters

The LMMSE-optimal dynamic kriging framework described in Sec. III requires knowledge of model covariance matrices  $\mathbf{C}_v$ ,  $\sigma^2 \mathbf{I}_S$ , and  $\mathbf{C}_\eta$ , to operate. Of these,  $\sigma^2$  depends on the precision offered by the measurement software, and can be safely assumed known a priori.

The structure of  $\mathbf{C}_v$  is motivated by the modeling assumptions and utilizes topological information. Intuitively, propagation, transmission, and processing delays over paths  $p, q \in \mathcal{P}$  should be highly correlated if these paths share many links. This relationship can be modeled by utilizing the Gramian matrix  $\mathbf{G} := \mathbf{R} \mathbf{R}^T$ , where  $\mathbf{R}$  is the  $P \times |\mathcal{E}|$  path-link routing matrix; that is, the  $(p, l)$ th element of  $\mathbf{R}$  is 1 if path  $p \in \mathcal{P}$  traverses link  $l \in \mathcal{E}$ , and 0 otherwise. Each off-diagonal entry  $(p, q)$  of  $\mathbf{G}$  represents the number of links common to the paths  $p, q \in \mathcal{P}$ . On the other hand, the elements on the main diagonal of  $\mathbf{G}$  count the number of constituent links per path. The covariance matrix of  $\mathbf{v}(t)$  can therefore be modeled as  $\mathbf{C}_v = \gamma \mathbf{G}$ .

A similar model for  $\mathbf{C}_v$  was adopted by [5], where it was motivated from the property that path delays are sum of link delays, that is,  $\mathbf{v}(t) = \mathbf{R}\mathbf{x}(t)$ , where vector  $\mathbf{x}(t)$  collects the link delays. Under this assumption, it holds that  $\mathbf{C}_v = \gamma \mathbf{G}$  if the link delays are uncorrelated across links, and have covariance matrix  $\gamma \mathbf{I}_{|\mathcal{E}|}$ . Note that the KKF and path-selection techniques also work with a generic link-delay covariance matrix  $\Sigma$ , i.e.,  $\mathbf{C}_v = \mathbf{R}\Sigma\mathbf{R}^T$ . Unfortunately however, in most IP networks the link delays cannot be directly observed, which makes estimation of  $\Sigma$  difficult, if not impossible. For example, consider a network (1–2–3) where two end terminals (nodes 1 and 3) are connected via an intermediate router (node 2). Clearly, the delays incurred by the individual links (1–2 and 2–3) cannot be discerned from each other, no matter how accurately the end-to-end delays (between 1 and 3) are measured. The same reasoning applies to the corresponding covariance matrices, irrespective of the estimation technique used.

For the remaining parameters, namely  $\gamma$  and  $\mathbf{C}_\eta$ , an empirical approach is described next. It entails a training phase, and a set of measurements  $\{\mathbf{y}_s(t)\}_{t=1}^{t_L}$  collected at time slots  $t = 1, \dots, t_L$ . During the KKF operation,  $t_L - 1$  time slots can be periodically devoted to updating model covariances, while predicting the networks-wide delays  $\mathbf{y}_s(t)$  for  $t = 1, \dots, t_L$ . Let  $\widehat{\mathbf{C}}_v(t) := \widehat{\gamma}(t)\mathbf{G}$  and  $\widehat{\mathbf{C}}_\eta(t)$  denote the estimates of  $\mathbf{C}_v$  and  $\mathbf{C}_\eta$ , respectively, at time  $t$ . Estimating the covariance matrix of the state noise is well known to be a challenging task, primarily because  $\chi(t)$  and  $\chi(t-1)$  are not directly observable. Furthermore, methods such as those in [20] are not applicable in the present context, as they require the KF to be time-invariant and stationary. As shown in [21], a viable means of estimating  $\mathbf{C}_\eta$  from  $\{\mathbf{y}_s(t)\}_{t=1}^{t_L}$  relies on approximating the noise  $\eta(t)$  as  $\mathbf{q}(t) := \widehat{\chi}(t) - \widehat{\chi}(t-1)$ . Then, upon noticing that the resultant process  $\{\mathbf{q}(t)\}$  is temporally-white, the sample mean and covariance of  $\mathbf{q}$  can be obtained as

$$\widehat{\mathbf{m}}_q(t_L) = \frac{\sum_{t=2}^{t_L} \mathbf{q}(t)}{t_L - 1} \quad (12)$$

$$\widehat{\mathbf{C}}_q(t_L) = \frac{\sum_{t=2}^{t_L} (\mathbf{q}(t) - \widehat{\mathbf{m}}_q(t))(\mathbf{q}(t) - \widehat{\mathbf{m}}_q(t))^T}{t_L - 2}. \quad (13)$$

Using (13), and exploiting the equality  $\mathbb{E}\{\widehat{\mathbf{C}}_q\} = (t_L - 1)^{-1} \sum_t (\mathbf{M}(t-1) - \mathbf{M}(t)) + \mathbf{C}_\eta$ , it follows that an unbiased estimate of  $\mathbf{C}_\eta$  can be obtained as

$$\widehat{\mathbf{C}}_\eta(t_L) = \widehat{\mathbf{C}}_q(t_L) + \frac{1}{t_L - 1} \sum_{t=2}^{t_L} (\mathbf{M}(t) - \mathbf{M}(t-1)). \quad (14)$$

Finally, in order to obtain  $\widehat{\gamma}$ , consider the innovations at time  $t$  as  $\iota_p(t) := y_p(t) - \widehat{\chi}_p(t-1)$ , and notice that if the model covariances are correct, then  $\iota_p(t)$  is temporally white and zero-mean [20]. Indeed, it is possible to show that  $\mathbb{E}[\iota_p(t)\iota_q(t)] = [\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v]_{pq} + \sigma^2$  for any  $p, q \in \mathcal{S}(t)$  [21]. Further, let  $\mathcal{T}_{pq} := \{t | 1 \leq t \leq t_L, p, q \in \mathcal{S}(t)\}$  be the set of time slots for which paths  $p$  and  $q$  are both measured. Then, the sample covariance between  $\iota_p(t)$  and  $\iota_q(t)$  is given by  $\widehat{\mathbf{C}}_{pq} := |\mathcal{T}_{pq}|^{-1} \sum_{t \in \mathcal{T}_{pq}} \iota_p(t)\iota_q(t)$  for all pairs  $p, q \in \mathcal{P}$ . Given  $\mathbf{M}(t-1)$  and  $\sigma^2$ , this observation yields the

following estimate

$$[\widehat{\mathbf{C}}_v(t)]_{pq} = \frac{\sum_{t \in \mathcal{T}_{pq}} \iota_p(t)\iota_q(t) - \sigma^2 - [\mathbf{M}(t-1) + \widehat{\mathbf{C}}_\eta(t)]_{pq}}{|\mathcal{T}_{pq}|}. \quad (15)$$

Indeed, entries of  $\widehat{\mathbf{C}}_v(t)$  can be updated recursively using  $\widehat{\mathbf{C}}_v(t-1)$  in (15). At each time, only a few entries are updated, depending on which paths are observed (cf.  $\mathcal{S}(t)$ ).

Finally,  $\widehat{\gamma}(t)$  can be obtained by fitting  $\widehat{\mathbf{C}}_v(t)$  to  $\gamma \mathbf{G}$  in the least-squares sense, which yields

$$\widehat{\gamma}(t_L) = \frac{\sum_{p, q \in \mathcal{P}} [\mathbf{G}]_{pq} [\widehat{\mathbf{C}}_v(t_L)]_{pq}}{\|\mathbf{G}\|_F^2}. \quad (16)$$

As further justification for the random-walk model, it is remarked that a model of the form  $\chi(t) = \mathbf{B}\chi(t-1) + \eta(t)$  requires learning the entries of  $\mathbf{B}$ . Since the state vector is not directly observed, estimation of  $\mathbf{B}$  is usually significantly more difficult [16], [22], [23]. Such a model would also need a longer training phase, and may exhibit poor *generalization performance* if the amount of training data is limited [24]. This problem also arises when trying to use the model  $\mathbf{C}_v = \mathbf{R}\Sigma\mathbf{R}^T$ , where additionally,  $\Sigma$  is not uniquely identifiable, as explained earlier. Finally, it is remarked that if the delays exhibit frequent abrupt changes, it is quite possible that the random walk model is too simple to capture their dynamics. The performance will then be dominated by the data (namely, the innovations) rather than the queuing delay model, and the KKF will be unable to properly capture the temporal correlation present in the delays.

#### IV. ONLINE EXPERIMENTAL DESIGN

This section considers the problem of optimally choosing the set of paths  $\mathcal{S}(t)$  (equivalently, the matrix  $\mathbf{S}(t)$ ) so as to minimize the prediction error. To begin with, a simple case is considered where the set  $\mathcal{S}(t)$  is allowed to contain any  $S$  paths. Operational requirements may however impose further constraints on  $\mathcal{S}(t)$ , and these are discussed later.

The prediction error can be characterized by using a scalar function of  $\mathbf{M}_s^y(t)$ ; see e.g., [25]. To this end, the so called D-optimal design is considered, where the goal is to minimize the function  $f(\mathcal{S}(t)) := \log \det(\mathbf{M}_s^y(t))$ . The paths selected at time  $t$  are therefore given by the solution of the following optimization problem

$$\mathcal{S}^*(t) = \arg \min_{\mathcal{S} \in \mathcal{P}} f(\mathcal{S}) \quad (17)$$

$$\text{s. t. } |\mathcal{S}| = S. \quad (18)$$

Clearly, tackling (17) incurs combinatorial complexity and is challenging to solve exactly, even for moderate-size networks. Indeed, (17) is an example of the so called subset selection problem, which is NP-complete in general; see e.g., [26] and references therein.

Interestingly, it is possible to solve (17) approximately by utilizing the notion of *submodularity*. Consider a function  $g(\mathcal{S})$ , which takes as input sets  $\mathcal{S} \subset \mathcal{P}$ . Given a set  $\mathcal{A} \in \mathcal{P}$  and an element  $p \in \mathcal{P} \setminus \mathcal{A}$ , the increment function is defined as  $\delta_{\mathcal{A}}^g(p) := g(\mathcal{A} \cup \{p\}) - g(\mathcal{A})$ . Function  $g(\cdot)$  is submodular if its

increments are monotonically decreasing, meaning  $\delta_{\mathcal{A}}^g(p) \geq \delta_{\mathcal{B}}^g(p)$  for all  $\mathcal{A} \subset \mathcal{B} \in \mathcal{P}$ . Likewise,  $g(\cdot)$  is supermodular if  $\delta_{\mathcal{A}}^g(p) \leq \delta_{\mathcal{B}}^g(p)$  for all  $\mathcal{A} \subset \mathcal{B} \in \mathcal{P}$ . In the present case, the following proposition holds.

**Proposition 2.** *The function  $f(\mathcal{S})$  is monotonic and super-modular in  $\mathcal{S}$ .*

The proof of Proposition 2 is provided in Appendix B, and relies on related results from [25].

An important implication of Proposition 2 is that a greedy forward selection algorithm can be developed to solve (17) approximately [27]. Upon defining the shifted function  $h(\mathcal{S}) := f(\mathcal{S}) - \log \det(\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v + \sigma^2 \mathbf{I}_P)$ , a result from [27] ensures that the solution of the greedy algorithm  $\mathcal{S}^g(t)$  satisfies the inequality

$$h(\mathcal{S}^g(t)) \leq \left(1 - \frac{1}{e}\right) h(\mathcal{S}^*(t)). \quad (19)$$

While performance of the greedy algorithm is usually much better in practice, this bound ensures that it does not break down for pathological inputs.

The greedy algorithm involves repeatedly performing the updates  $\mathcal{S} \leftarrow \mathcal{S} \cup \arg \min_{p \notin \mathcal{S}} \delta_{\mathcal{S}}^f(p)$  until  $|\mathcal{S}| = S$ . This is useful in the present case, since the increments can be evaluated efficiently using determinant update rules. Specifically, the updates are given by

$$\delta_{\mathcal{S}}^f(p) = -\log \left(1 + [\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v]_{p,p}\right) \quad \forall p \in \mathcal{P} \quad (20a)$$

$$\begin{aligned} \delta_{\mathcal{S}}^f(p) &= -\log \left(1 + \left[ (\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)^{-1} + \mathbf{S}^T \mathbf{S} \right]_{p,p}^{-1}\right) \\ &\quad \forall p \in \mathcal{P} \setminus \mathcal{S}. \end{aligned} \quad (20b)$$

Further, each iteration requires a rank-one update to the matrix inverse in (20b), which can also be performed efficiently. The full greedy approach is summarized in Algorithm 1, where  $\Phi := (\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)/\sigma^2$ . Algorithm 1 involves only basic operations, and it is easy to verify that its worst case complexity is  $O(PS^3)$ . Further, the final value of the matrix  $\mathbf{V}$  evaluated in the last iteration (Algorithm 1, line 11) is exactly the inverse term required for evaluating the Kalman gain in (7). In fact, the operational complexity can be further reduced using lazy updates [28]. Finally, it is worth mentioning that the low-complexity of Algorithm 1 is also a result of the random-walk model used here. In particular, the state space model  $\chi(t) = \mathbf{B}\chi(t-1) + \eta(t)$  would result in significantly more complicated expressions.

Next, consider a more practical scenario, where the software installed at each end-node can measure delays on all paths originating at that node. At any time  $t$  however, delays are measured from only  $N$  end-nodes. Let  $\mathcal{V}_e$  denote the set of all end-nodes, and  $\mathcal{P}_v$ , the set of paths which have the node  $v \in \mathcal{V}_e$  as their origin (likewise,  $\mathcal{P}_{\mathcal{N}} := \bigcup_{v \in \mathcal{N}} \mathcal{P}_v$  for  $\mathcal{N} \subset \mathcal{V}_e$ ). For any subset  $\mathcal{N}$  (and its complement  $\bar{\mathcal{N}} := \mathcal{V} \setminus \mathcal{N}$ ), define the selection matrix  $\mathbf{N}$  ( $\bar{\mathbf{N}}$ ) consisting of canonical vectors  $\mathbf{e}_p^T$  as rows, for all  $p \in \mathcal{P}_{\mathcal{N}}$  ( $p \in \mathcal{P}_{\bar{\mathcal{N}}}$ ). Defining the cost function  $f_n(\mathcal{N}) := f(\mathcal{P}_{\mathcal{N}})$ , the online optimal design problem for this

---

**Algorithm 1** Greedy Algorithm for Solving (17)

---

```

1: function GREEDY( $\Phi, S$ )
2:    $s \leftarrow \arg \max_{1 \leq p \leq P} [\Phi]_{p,p}$ 
3:    $\mathbf{V} := [1/([\Phi]_{s,s} + 1)]$ 
4:    $\mathcal{S} \leftarrow \{s\}$ 
5:   for  $k = 2 : S$  do
6:      $\mathbf{w}_p \leftarrow \Phi_{\mathcal{S},p}$  for all  $p \in \mathcal{P} \setminus \mathcal{S}$  //  $\mathbf{w}_p$ 
       has entries  $[\Phi]_{s,p}$  for all  $s \in \mathcal{S}$ 
7:      $s \leftarrow \arg \max_{p \notin \mathcal{S}} [\Phi]_{p,p} - \mathbf{w}_p^T \mathbf{V} \mathbf{w}_p$ 
8:      $\mathcal{S} \leftarrow \mathcal{S} \cup \{s\}$ 
9:      $d \leftarrow [\Phi]_{s,s} - \mathbf{w}_s^T \mathbf{V} \mathbf{w}_s + 1$ 
10:     $\mathbf{u} \leftarrow -\mathbf{V} \mathbf{w}_s$ 
11:     $\mathbf{V} \leftarrow \begin{bmatrix} \mathbf{V} + \mathbf{u} \mathbf{u}^T / d & \mathbf{u} / d \\ \mathbf{u}^T / d & 1/d \end{bmatrix}$ 
12:  return  $\mathcal{S}$ 

```

---

scenario is expressed as

$$\mathcal{N}^*(t) = \arg \min_{\mathcal{N} \subset \mathcal{V}_e} f_n(\mathcal{N}) \quad (21a)$$

$$\text{s. t. } |\mathcal{N}| = N. \quad (21b)$$

It follows from the properties of submodular functions that the cost function  $f_n(\mathcal{N})$  is also monotonic and supermodular in  $\mathcal{N}$ . In particular, observe that the increments  $\delta_{\mathcal{N}}^n(v) = f_n(\mathcal{N} \cup \{v\}) - f_n(\mathcal{N}) = f(\mathcal{P}_{\mathcal{N} \cup \{v\}}) - f(\mathcal{P}_{\mathcal{N}})$  for  $v \notin \mathcal{N}$  satisfy the non-increasing property, i.e.,  $\delta_{\mathcal{A}}^n(v) \leq \delta_{\mathcal{B}}^n(v)$  for all  $\mathcal{A} \subset \mathcal{B} \subset \mathcal{V}_e$  and  $v \notin \mathcal{B}$ . A greedy algorithm similar to Algorithm 1 can therefore be developed to obtain an approximate solution with the same  $(1 - 1/e)$  guarantee as in (19). Complexity of the greedy algorithm in this case would be however higher, since evaluating  $\delta_{\mathcal{N}}(v)$  now requires rank- $|\mathcal{P}_v|$  updates in the determinant and inverses. Nevertheless, the algorithm would still be efficient as long as  $|\mathcal{P}_v| \ll P$  for all  $v \in \mathcal{V}_e$ . In the special case when delay measurements are performed by only one node per time slot ( $N = 1$ ), the solution of (21a) is simply given by

$$\mathcal{N}^*(t) = \arg \min_{v \in \mathcal{V}_e} \log \det (\mathbf{I}_{|\mathcal{P}_v|} + [\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v]_{vv}) \quad (22)$$

where  $[\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v]_{vv}$  is the  $|\mathcal{P}_v| \times |\mathcal{P}_v|$  submatrix containing the rows and columns of  $\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v$  corresponding to the paths in  $\mathcal{P}_v$ .

In some networks, it may be relatively straightforward to install delay measurement software on every end-node, while allowing each end-node to measure delay on only one path per time slot. This amounts to replacing the budget-constraint (18) in (17) with

$$|\mathcal{S} \cap \mathcal{P}_v| = 1 \quad \forall v \in \mathcal{V}_e. \quad (23)$$

Interestingly, constraints of this form can also be handled using the greedy approach by simply imposing (23) while searching for the best increment at every iteration. Specifically, the search space of path  $p$  [cf. Algorithm 1, line 7] now becomes  $p \in \mathcal{P} \setminus \mathcal{P}_{\mathcal{N}}$ , where  $\mathcal{N} = \{v : \mathcal{S} \cap \mathcal{P}_v \neq \emptyset\}$ . More general constraints of the form  $|\mathcal{S} \cap \mathcal{P}_v| \leq S_v$  can

similarly be incorporated. Constraints of this form are referred to as partition matroid constraints, under which the greedy algorithm provides an approximation ratio of  $1/2$  [29].

## V. EMPIRICAL VALIDATION

Performance of the proposed network-wide latency prediction schemes is validated using two different datasets, which include delays measured on:

- Internet2 backbone network,<sup>1</sup> a lightly loaded network that exhibits low delay variability; and,
- New Zealand Active Measurement Project (NZ-AMP),<sup>2</sup> a network deployed across several universities and ISPs in New Zealand, characterized by comparatively higher variability in delays.

The delays in the Internet2 are measured using one-way ping (OWAMP), while those in NZ-AMP are measured using ping. Using the aforementioned datasets, the performance of KKF is also compared against that of competing alternatives in [5] and [6]. It is remarked that for the sake of comparison and generality of conclusions, dependence on factors such as probe-sizes, header fields, network policies, and traceroute errors are ignored. Also omitted are the physical constraints that might limit the delays from taking arbitrary values on the real line. Indeed, none of the three algorithms prevent the delay predictions from taking values that are either too large or too small. In practice however, these predicted delays should be clipped to a reasonable interval, e.g., negative delay predictions should be read as zeros.

Before proceeding, a brief description of the nonlinear estimation technique in [6] is provided. The approach hinges on a sparse representation of the network-wide delays, and employs  $\ell_1$ -norm minimization to recover the sparse basis coefficient vector. Specifically, the path delays adhere to the postulated linear model  $\mathbf{y}(t) = \mathbf{H}\boldsymbol{\beta}(t)$ , where  $\|\boldsymbol{\beta}(t)\|_0 \ll P$ , and the matrix  $\mathbf{H} \in \mathbb{R}^{P \times P}$  is constructed using diffusion wavelets [30]. The diffusion matrix used for computing the wavelet basis is obtained by applying Sinkhorn balancing [31] to the matrix  $\mathbf{W} \in \mathbb{R}^{P \times P}$ , whose  $(p, q)$ -th element is defined as

$$[\mathbf{W}]_{p,q} = \frac{[\mathbf{G}]_{pq}}{[\mathbf{G}]_{pp} + [\mathbf{G}]_{qq} - [\mathbf{G}]_{pq}} \quad (24)$$

where  $\mathbf{G}$  is the Gramian defined in Sec. III-A. The overall algorithm amounts to solving the following minimization problem

$$\hat{\boldsymbol{\beta}}'(t) = \arg \min_{\boldsymbol{\beta}'} \|\boldsymbol{\beta}'\|_1 \quad (25a)$$

$$\text{s. t. } \mathbf{y}_s(t) = \mathbf{S}(t)\mathbf{H}\mathbf{L}\boldsymbol{\beta}' \quad (25b)$$

where  $\mathbf{L}$  is a diagonal matrix whose  $(n, n)$ -th entry is given by  $[\mathbf{L}]_{n,n} = 2^k$ , with  $k \in \mathbb{N}$  denoting the scale corresponding to the diffusion wavelet coefficient  $\beta_n$  [6]. Subsequently,  $\mathbf{y}_s(t)$  is predicted as  $\hat{\mathbf{y}}_s(t) = \hat{\mathbf{S}}(t)\mathbf{H}\mathbf{L}\hat{\boldsymbol{\beta}}'(t)$ .

Under the premise that delays change slowly with time, the described algorithm can be used to estimate  $\mathbf{y}_s(t)$  over

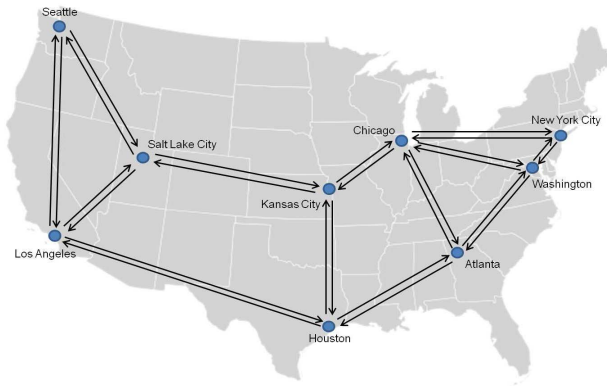


Fig. 1. Internet2 IP backbone network.

a sequence of  $\tau > 1$  contiguous time-steps jointly. In this case, problem (25) is solved by replacing  $\mathbf{y}_s(t)$  with  $\bar{\mathbf{y}}_s(t) := [\mathbf{y}_s^T(t-\tau+1), \mathbf{y}_s^T(t-\tau+2), \dots, \mathbf{y}_s^T(t)]^T$ , and by computing the  $P\tau \times P\tau$  diffusion wavelet matrix based on  $\mathbf{W}$  and temporal correlations as shown in [6]. Although this is a viable way to capture temporal correlations of delays, observe that it requires solving  $\ell_1$ -norm minimization problems with  $P\tau$  variables every  $\tau$  time slots. This increase in complexity prohibits the use of a large value of  $\tau$ , and the simulations here only report performance with  $\tau = 5$ . It is also worth mentioning that such a batch solution also does not compare favorably to a real-time implementation, such as that provided by the KKF where delay predictions become available every time new measurements arrive.

### A. Internet2 Delay Data

The One Way Active Measurement Project (OWAMP) collects one way delays on the Internet2 backbone network.<sup>3</sup> The network has 9 end-nodes and 26 directional links as depicted in Fig. 1. Delays are measured on the 72 paths among the end-nodes every minute. The data  $\{\mathbf{y}(t)\}$  is collected over  $t_P = 4500$  minutes (about three days) in July 2011.

The model KKF covariances  $\mathbf{C}_v$  and  $\mathbf{C}_\eta$  are estimated using data from the initial 1,000 time slots. In this phase, 50 paths are randomly selected per time slot. The KKF is initialized by setting  $\gamma = 1$ ,  $\mathbf{C}_\eta = \mathbf{C}_v$ , and run for 500 time slots. Next,  $\hat{\gamma}(t)$  and  $\hat{\mathbf{C}}_\eta(t)$  are updated in an online fashion, as outlined in Sec. III-A. The final values are obtained at the conclusion of the training phase at  $t = 1,000$ .

Pictorially, the performance of different algorithms can be assessed through delay maps shown in 2. Such maps can succinctly represent the network health, and are especially useful for networks which otherwise have low delay variability, such as the Internet2. The map in Fig. 2(a) corresponds to the true delays, whereas maps (b), (c), and (d) depict the predicted values obtained from the network kriging, wavelet-based approach, and KKF respectively. Predictions are performed using measurements over an interval of

<sup>1</sup>[Online] <http://www.internet2.edu/network>

<sup>2</sup>[Online] <http://erg.cs.waikato.ac.nz/amp>

<sup>3</sup>[Online] <http://ndbl.net.internet2.edu/cgi-bin/owamp.cgi>

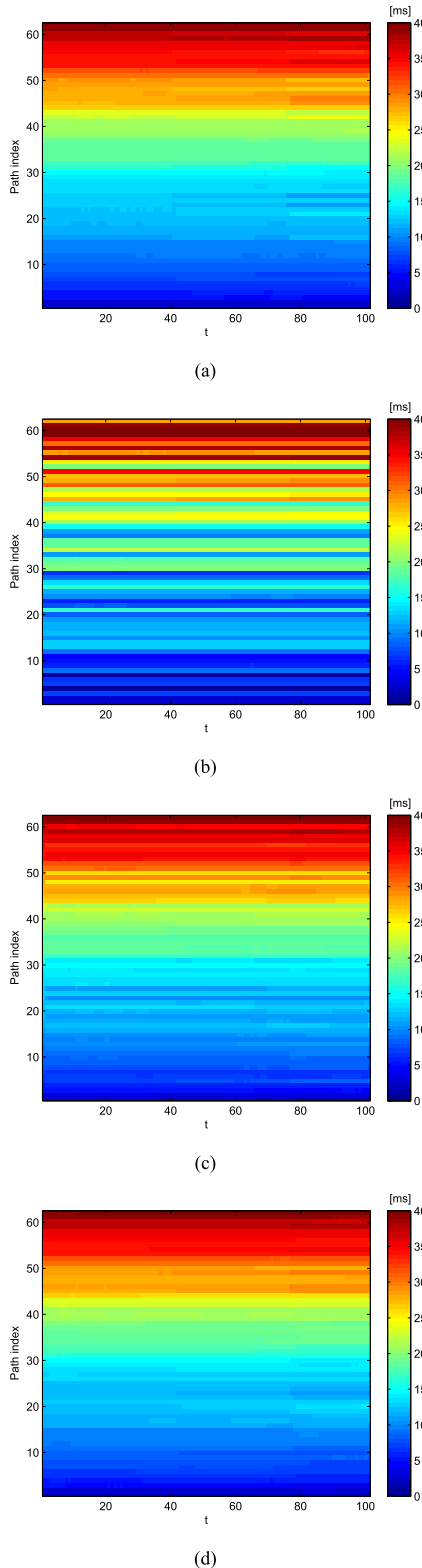


Fig. 2. True and predicted delay map for 62 paths in the Internet2 network over in interval of 100 minutes. (a) True map. (b) Kriging. (c) Wavelets. (d) KKF.

100 minutes on 10 random paths (same paths are used throughout the considered interval), and the delays are predicted on the remaining 62 paths are reported. In these maps, paths are arranged in increasing order according to the true delay at time  $t = 1$ . It can be seen that the map produced by the

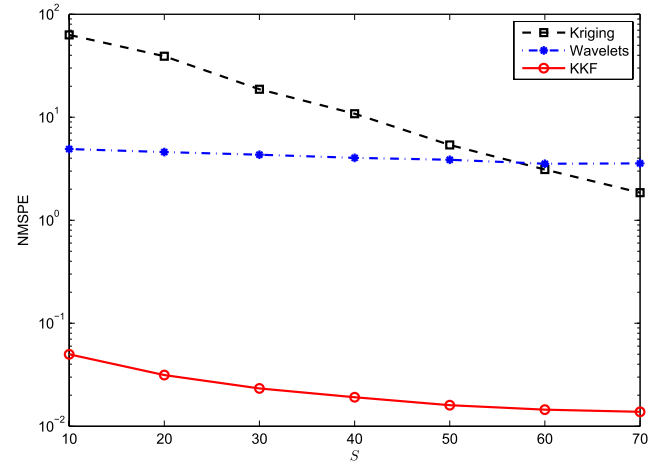


Fig. 3. NMSPE as a function of  $S$ , Internet2 network with random path selection.

kriging and compressive sensing approaches are very different from the true map. In contrast, the map obtained when using the KKF is close to the true map. In particular, observe that the delays of several paths change slightly around  $t = 80$  in Fig. 2(a). However, of the three maps, this change is only discernible in the KKF map in 2(d). The delay predictions provided by the KKF are thus sufficiently accurate for human inspection at control centers, even when monitoring a few paths.

It should be remarked that the maps in Fig. 2 are only for demonstration purposes, and not much can be inferred about the relative performance of different algorithms from these depictions alone. For a more detailed analysis of the different delay prediction approaches, consider the normalized mean-square prediction error (NMSPE), defined as

$$\text{NMSPE} := \frac{1}{(t_P - t_L)(P - S)} \sum_{t=t_L+1}^{t_P} \|\hat{\mathbf{y}}_S(t) - \mathbf{y}_S(t)\|_2^2. \quad (26)$$

The prediction performance of the three algorithms is first assessed by using delay measurements on randomly selected paths for each  $t$ . The (same) randomly selected paths are used for all three approaches. Fig. 3 depicts the NMSPE as a function of  $S$ , the number of paths on which delays are measured. Clearly, the KKF markedly outperforms the other two approaches across the entire range of  $S$ . As expected [6], the compressive sampling-based approach provides a more accurate prediction than network kriging.

Next, the performance of the three algorithms is analyzed for the case when paths for delay measurement are selected optimally. For the network kriging and the wavelet-based approaches, the optimal paths are obtained according to the selection procedures provided in [5] and [6], respectively. As pointed out in [6], performance of the wavelet-based approach can be improved by capitalizing on temporal correlations. This is done by solving (25) using measurements from  $\tau = 5$  consecutive time slots in a batch form. The temporal correlation is set to 0.5 and the optimal paths are obtained again using the selection strategy outlined in [6]. For the

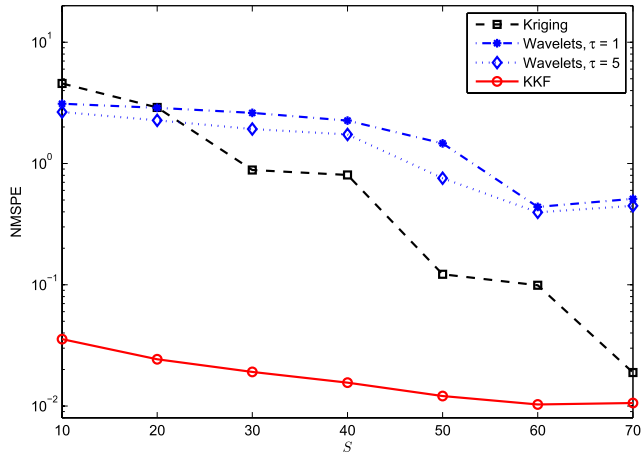


Fig. 4. NMSPE as a function of  $S$ , Internet2 network with optimal path selection.

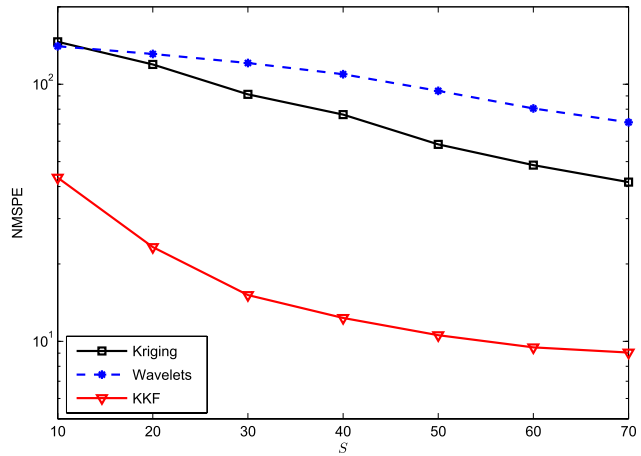


Fig. 5. NMSPE as a function of  $S$ , NZ-AMP network with random path selection.

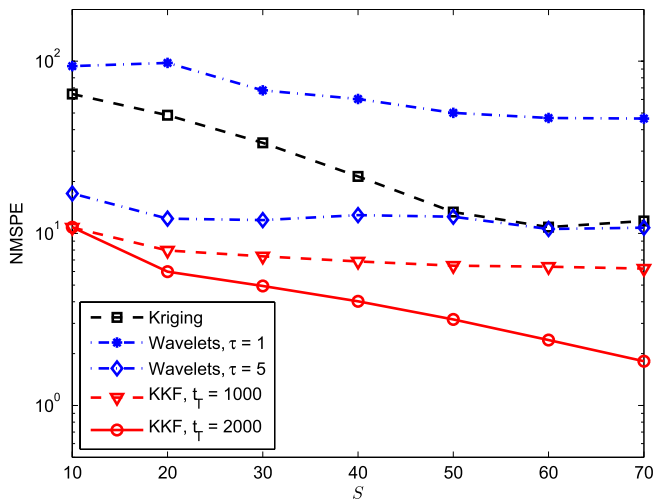
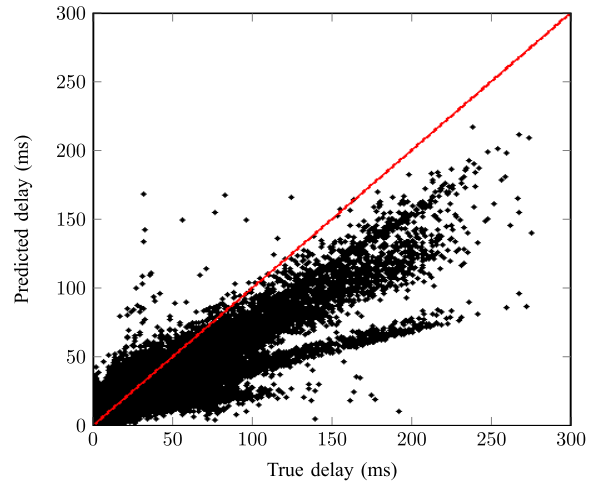
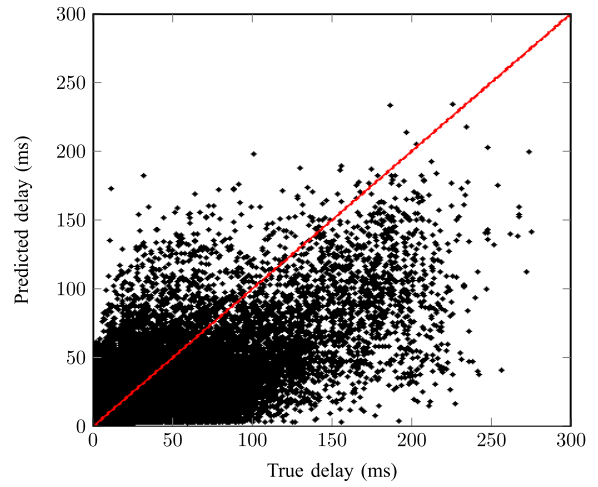


Fig. 6. NMSPE as a function of  $S$ , NZ-AMP network with optimal path selection.

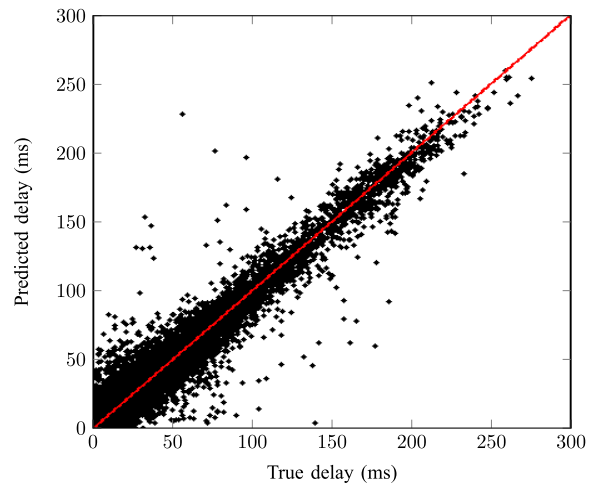
KKF, optimal paths are selected in an online fashion using Algorithm 1. Again, a significantly more accurate prediction of the path delays for the entire range of  $S$  is obtained via the KKF.



(a)



(b)



(c)

Fig. 7. Scatter plot for the NZ-AMP network,  $S = 30$  with optimal path selection. (a) Kriging. (b) Wavelets. (c) KKF.

*B. NZ-AMP Delay Data*

The KKF algorithm is tested here using delay data from NZ-AMP. The project continuously runs `ICMP` and `scamper` to determine the topology and delays between a set of nodes



in New Zealand. The data collected for this paper consist of end-to-end delays measured every ten minutes over the month of August 2011. The network has a total of 186 paths, whose delays range from almost constant to highly variable, at times reaching up to 250ms.

In Fig. 5, the NMSPE as a function of  $S$  is reported, for the case where paths that are to be measured are chosen randomly. Again, same paths are used for the three considered schemes. The KKF provides a markedly lower prediction error also for the NZ-AMP delay data. On the other hand, Fig. 6 shows the NMSPE on optimally selected paths for all three schemes. The KKF performs relatively better than the competing schemes for this data set as well. Observe though that the actual values of the NMSPE incurred for this dataset is at least an order of magnitude higher than those in the Internet2 dataset. Indeed, given the high variability in the data, it is possible to improve upon the prediction accuracy of KKF by training it better. This is showcased by the considerably lower prediction error curve for training interval  $t_L = 2,000$  shown in Fig. 6.

While the NMSPE is useful for characterizing the average performance, network operators are also interested in the prediction accuracy over the entire range of delay values. Towards this end, Fig. 7 shows the scatter plots of  $\hat{y}_s(t)$  versus  $y_s(t)$  for all  $t$  and  $S = 30$  optimally selected paths. The points cluster around the 45-degree line  $\hat{y}_s(t) = y_s(t)$ , and the thinner the “cloud” of points is, the more accurate the estimates are. Indeed, it can be seen that the points generated from the KKF estimates are crammed in a very close area around the 45-degree line, and accurate estimates are produced for

the entire range of experienced delays. Furthermore, the scatter plots corroborate the unbiasedness of the KKF predictor.

## VI. CONCLUSION

The present paper develops a spatio-temporal prediction approach to track and predict network-wide path delays using measurements on only a few paths. The proposed algorithm adapts a kriged Kalman filter that exploits both topological as well as historical data. The framework also allows for the use of submodular optimization in the selection of optimal delay measurement locations. The problem of path selection is formulated for different types of constraints on the set of selected paths, and solved in an online fashion to near-optimality. The resulting predictor is validated on two datasets with different delay profiles, and is shown to substantially outperform competing alternatives.

## APPENDIX A

### ERROR COVARIANCE MATRIX

Towards deriving an expression for  $\mathbf{M}_s^y(t)$ , observe that the prediction error can be written as shown in (27a). The expressions in (27b) and (27c) follow from (6a), that allows the term  $\chi(t) - \hat{\chi}(t)$  to be written as

$$\begin{aligned} \chi(t) - \hat{\chi}(t) &= \chi(t) - \hat{\chi}(t-1) \\ &\quad - \mathbf{K}(t) [\mathbf{S}(t)(\chi(t) + \mathbf{v}(t)) + \epsilon_s(t) - \mathbf{S}(t)\hat{\chi}(t-1)] \\ &= \chi(t) - \hat{\chi}(t-1) + \mathbf{K}(t)\mathbf{S}(t)(\chi(t) - \hat{\chi}(t-1) + \mathbf{v}(t)) \\ &\quad + \mathbf{K}(t)\epsilon_s(t) \\ &= (\mathbf{I}_P - \mathbf{K}(t)\mathbf{S}(t))\tilde{\chi}(t) - \mathbf{K}(t)\mathbf{S}(t)\mathbf{v}(t) - \mathbf{K}(t)\epsilon_s(t) \end{aligned} \quad (28)$$

$$\begin{aligned} y_s(t) - \hat{y}_s(t) &= \bar{\mathbf{S}}(t)\chi(t) + \bar{\mathbf{S}}(t)\mathbf{v}(t) + \epsilon_s(t) - \bar{\mathbf{S}}(t)\hat{\chi}(t) - \bar{\mathbf{S}}(t)\mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1} [y_s(t) - \mathbf{S}(t)\hat{\chi}(t)] \\ &= \bar{\mathbf{S}}(t)(\chi(t) - \hat{\chi}(t) + \mathbf{v}(t)) + \epsilon_s(t) - \bar{\mathbf{S}}(t)\mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1} \\ &\quad \times [\mathbf{S}(t)(\chi(t) - \hat{\chi}(t) + \mathbf{v}(t)) + \epsilon_s(t)] \end{aligned} \quad (27a)$$

$$\begin{aligned} &= \bar{\mathbf{S}}(t)(\mathbf{I}_P - \mathbf{K}(t)\mathbf{S}(t))(\tilde{\chi}(t) + \mathbf{v}(t)) - \bar{\mathbf{S}}(t)\mathbf{K}(t)\epsilon_s(t) + \epsilon_s(t) - \bar{\mathbf{S}}(t)\mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1} \\ &\quad \times [\mathbf{S}(t)(\mathbf{I}_P - \mathbf{K}(t)\mathbf{S}(t))(\tilde{\chi}(t) + \mathbf{v}(t)) - \mathbf{S}(t)\mathbf{K}(t)\epsilon_s(t) + \epsilon_s(t)] \end{aligned} \quad (27b)$$

$$\begin{aligned} &= \bar{\mathbf{S}}(t)(\mathbf{I}_P - \mathbf{K}(t)\mathbf{S}(t))(\tilde{\chi}(t) + \mathbf{v}(t)) - \bar{\mathbf{S}}(t)\mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1}\mathbf{S}(t)(\mathbf{I}_P - \mathbf{K}(t)\mathbf{S}(t))(\tilde{\chi}(t) + \mathbf{v}(t)) \\ &\quad - \bar{\mathbf{S}}(t)\mathbf{K}(t)\epsilon_s(t) - \bar{\mathbf{S}}(t)\mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1} \\ &\quad \times (\mathbf{I}_S - \mathbf{S}(t)\mathbf{K}(t))\epsilon_s(t) + \epsilon_s(t) \end{aligned} \quad (27c)$$

$$\begin{aligned} \mathbf{Q}(t) &= (\mathbf{M}(t-1) + \mathbf{C}_\eta)\mathbf{S}^T(t)[\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S]^{-1} + \mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1} \\ &\quad - \mathbf{C}_v\mathbf{S}^T(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1}\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_\eta)\mathbf{S}^T(t)[\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S]^{-1} \end{aligned} \quad (31a)$$

$$= (\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)\mathbf{S}^T(t)[\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S]^{-1}. \quad (31b)$$

$$\mathbf{M}_s^y(t) = \mathbb{E}[(y_s(t) - \hat{y}_s(t))(y_s(t) - \hat{y}_s(t))^T] \quad (32a)$$

$$= \bar{\mathbf{S}}(t)(\mathbf{I}_P - \mathbf{Q}(t)\mathbf{S}(t))(\mathbf{M}(t-1) + \mathbf{C}_v + \mathbf{C}_\eta)(\mathbf{I}_P - \mathbf{S}^T(t)\mathbf{Q}^T(t))\bar{\mathbf{S}}^T(t) + \sigma^2\bar{\mathbf{S}}(t)\mathbf{Q}(t)\mathbf{Q}^T(t)\bar{\mathbf{S}}^T(t) + \sigma^2\mathbf{I}_{P-S} \quad (32b)$$

$$\begin{aligned} &= \bar{\mathbf{S}}(t)(\mathbf{M}(t-1) + \mathbf{C}_v + \mathbf{C}_\eta)\bar{\mathbf{S}}^T(t) - 2\bar{\mathbf{S}}(t)\mathbf{Q}(t)\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_v + \mathbf{C}_\eta)\bar{\mathbf{S}}^T(t) \\ &\quad + \bar{\mathbf{S}}(t)\mathbf{Q}(t)\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)\mathbf{S}^T(t)\mathbf{Q}^T(t)\bar{\mathbf{S}}^T(t) + \sigma^2\bar{\mathbf{S}}(t)\mathbf{Q}(t)\mathbf{Q}^T(t)\bar{\mathbf{S}}^T(t) + \sigma^2\mathbf{I}_{P-S} \end{aligned} \quad (32c)$$

$$= \bar{\mathbf{S}}(t)(\mathbf{M}(t-1) + \mathbf{C}_v + \mathbf{C}_\eta)\bar{\mathbf{S}}^T(t) - \bar{\mathbf{S}}(t)\mathbf{Q}(t)\mathbf{S}(t)(\mathbf{M}(t-1) + \mathbf{C}_v + \mathbf{C}_\eta)\bar{\mathbf{S}}^T(t) + \sigma^2\mathbf{I}_{P-S}. \quad (32d)$$

where  $\tilde{\chi}(t) := \chi(t) - \hat{\chi}(t-1)$ . After some manipulations, (27c) can be expressed as

$$\mathbf{y}_{\bar{s}}(t) - \hat{\mathbf{y}}_{\bar{s}}(t) = \bar{\mathbf{S}}(t)(\mathbf{I}_P - \mathbf{Q}(t)\mathbf{S}(t))(\tilde{\chi}(t) + \mathbf{v}(t)) + \mathbf{Q}(t)\boldsymbol{\epsilon}_s(t) + \boldsymbol{\epsilon}_{\bar{s}}(t) \quad (29)$$

where

$$\mathbf{Q}(t) := \mathbf{K}(t) + \mathbf{C}_v\mathbf{S}(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1} - \mathbf{C}_v\mathbf{S}(t)(\mathbf{S}(t)\mathbf{C}_v\mathbf{S}^T(t) + \sigma^2\mathbf{I}_S)^{-1}\mathbf{S}(t)\mathbf{K}(t). \quad (30)$$

Next, substituting for  $\mathbf{K}(t)$  from (7), the expression for  $\mathbf{Q}(t)$  simplifies to that shown in (31). Utilizing the fact that  $\tilde{\chi}(t)$ ,  $\mathbf{v}(t)$ ,  $\boldsymbol{\epsilon}_s(t)$ , and  $\boldsymbol{\epsilon}_{\bar{s}}(t)$  are mutually uncorrelated, with  $\mathbb{E}[\tilde{\chi}(t)\tilde{\chi}^T(t)] := \mathbf{M}(t-1) + \mathbf{C}_\eta$ , the error covariance matrix  $\mathbf{M}_{\bar{s}}^y(t)$  becomes as shown in (32).

Finally, substituting for  $\mathbf{Q}(t)$  [cf. (31b)] in (32d), and using the Woodbury matrix identity [32], the expression for  $\mathbf{M}_{\bar{s}}^y(t)$  becomes as shown in (9).

## APPENDIX B

### PROOF OF MONOTONICITY AND SUPERMODULARITY OF $f$

Let  $\Phi := \frac{1}{\sigma^2}(\mathbf{M}(t-1) + \mathbf{C}_\eta + \mathbf{C}_v)$ , and observe that  $f$  can be written as

$$f(S) = \log(\sigma^2) + \log \det [\mathbf{I}_{P-S} + \bar{\mathbf{S}}(\Phi^{-1} + \mathbf{S}^T\mathbf{S})^{-1}\bar{\mathbf{S}}^T] \\ = \log(\sigma^2) + \log \det [\mathbf{I}_P + \bar{\mathbf{S}}^T\bar{\mathbf{S}}(\Phi^{-1} + \mathbf{S}^T\mathbf{S})^{-1}] \quad (33a)$$

$$= \log(\sigma^2) + \log \det [\Phi^{-1} + \mathbf{S}^T\mathbf{S} + \bar{\mathbf{S}}^T\bar{\mathbf{S}}] \\ + \log \det [(\Phi^{-1} + \mathbf{S}^T\mathbf{S})^{-1}] \quad (33b)$$

where (33a) follows from Sylvester's theorem for determinants [32]. Next, observing that  $\bar{\mathbf{S}}^T\bar{\mathbf{S}} + \mathbf{S}^T\mathbf{S} = \mathbf{I}_P$ , it is possible to write  $f(S)$  as

$$f(S) = \log(\sigma^2) + \log \det(\Phi^{-1} + \mathbf{I}_P) \\ - \log \det(\Phi^{-1} + \mathbf{S}^T\mathbf{S}). \quad (34)$$

Next, consider the decomposition  $\Phi = \mathbf{U}\mathbf{U}^T$ , and define the shifted function

$$h(S) := f(S) - \log(\sigma^2) - \log \det(\Phi + \mathbf{I}_P) \quad (35a)$$

$$= -\log \det(\mathbf{I}_P + \mathbf{S}^T\mathbf{S}\Phi) \quad (35b)$$

$$= -\log \det[\mathbf{I}_S + (\mathbf{S}\mathbf{U})(\mathbf{S}\mathbf{U})^T] \quad (35c)$$

where Sylvester's theorem has again been used in (35c). Finally, it is well known that a function of the form  $\log \det(\mathbf{I}_P + (\mathbf{S}\mathbf{U})^T(\mathbf{S}\mathbf{U}))$  is non-decreasing and submodular (see e.g., [25]), which allows one to deduce that  $f(S)$  is non-increasing and supermodular. Note further that the greedy approach from [27] can be used on  $h(S)$  by defining  $h(\emptyset) = 0$ .

## REFERENCES

- [1] H. Singhal and G. Michailidis, "Structural models for dual modality data with application to network tomography," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5054–5071, Aug. 2011.
- [2] M. G. Rabbat, M. A. T. Figueiredo, and R. D. Nowak, "Network inference from co-occurrences," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 4053–4068, Sep. 2008.
- [3] B. D. Ripley, *Spatial Statistics*. New York, NY, USA: Wiley, 1981.
- [4] N. Cressie, "The origins of kriging," *Math. Geol.*, vol. 22, no. 3, pp. 239–252, 1990.
- [5] D. B. Chua, E. D. Kolaczyk, and M. Crovella, "Network kriging," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 2263–2272, Dec. 2006.
- [6] M. Coates, Y. Pointurier, and M. Rabbat, "Compressed network monitoring for IP and all-optical networks," in *Proc. ACM Internet Meas. Conf.*, San Diego, CA, USA, Oct. 2007, pp. 1–3.
- [7] W. Xu, E. Mallada, and A. Tang, "Compressive sensing over graphs," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2087–2095.
- [8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," in *Proc. ACM SIGCOMM*, Portland, OR, USA, 2004, pp. 15–26.
- [9] Y. Liao, P. Geurts, and G. Leduc, "Network distance prediction based on decentralized matrix factorization," in *Proc. IFIP Netw.*, Chennai, India, May 2010, pp. 15–26.
- [10] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. ACM SIGMETRICS*, New York, NY, USA, 2004, pp. 61–72.
- [11] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. van Mieghem, "Analysis of end-to-end delay measurements in Internet," in *Proc. Passive Active Meas. Workshop*, Fort Collins, CO, Apr. 2002, pp. 1–33.
- [12] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso, "The kriged Kalman filter," *Test*, vol. 7, no. 2, pp. 217–285, Dec. 1998.
- [13] C. K. Wikle and N. Cressie, "A dimension-reduced approach to space-time Kalman filtering," *Biometrika*, vol. 86, no. 4, pp. 815–829, 1999.
- [14] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Feb. 2011.
- [15] J. Cortés, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Trans. Autom. Controlled*, vol. 54, no. 12, pp. 2816–2827, Dec. 2009.
- [16] B. D. O. Anderson and J. B. Moore, *Optimal Filtering* (Information and System Sciences Series). Englewood Cliffs, NJ, USA: Prentice-Hall, 1979.
- [17] P. Casas, S. Vaton, L. Fillatre, and T. Chonavel, "Efficient methods for traffic matrix modeling and on-line estimation in large-scale ip networks," in *Proc. 21st Int. Teletraffic Congr.*, Paris, France, 2009, pp. 45–54.
- [18] E. Dall'Anese, S.-J. Kim, and G. Giannakis, "Channel gain map tracking via distributed kriging," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1205–1211, Mar. 2011.
- [19] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.
- [20] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. Autom. Control*, vol. 15, no. 2, pp. 175–184, Apr. 1970.
- [21] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Trans. Autom. Control*, vol. 21, no. 4, pp. 520–523, Aug. 1976.
- [22] M. K. Tsatsanis and G. B. Giannakis, "Modeling and equalization of rapidly fading channels," *Intl. J. Adapt. Control Signal Process.*, vol. 10, nos. 2–3, pp. 159–176, 1996.
- [23] M. K. Tsatsanis, G. B. Giannakis, and G. Zhou, "Modeling and equalization of rapidly fading channels," *Signal Process.*, vol. 53, nos. 2–3, pp. 211–229, 1996.
- [24] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
- [25] F. Bach, "Learning with submodular functions: A convex optimization perspective," *Found. Trends Mach. Learn.*, vol. 6, nos. 2–3, pp. 145–373, 2012.
- [26] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proc. ACM Symp. Theory Comput.*, Victoria, BC, Canada, May 2008, pp. 45–54.
- [27] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, no. 1, pp. 265–294, Dec. 1978.
- [28] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques* (Lecture Notes in Control and Information Sciences), vol. 7, J. Stoer, Ed. Berlin, Germany: Springer-Verlag, 1978, pp. 234–243.
- [29] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, "An analysis of approximations for maximizing submodular set functions—II," *Math. Program. Study*, vol. 14, no. 1, pp. 73–87, 1978.
- [30] R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comput. Harmonic Anal.*, vol. 21, no. 1, pp. 53–94, 2006.

- [31] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *Ann. Math. Statist.*, vol. 35, no. 2, pp. 876–879, 1964.
- [32] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.

**Ketan Rajawat** (M'13) received his B. Tech and M. Tech degrees in Electrical Engineering from Indian Institute of Technology Kanpur, in 2007; and his Ph.D. degree in Electrical and Computer Engineering from University of Minnesota, in 2012. Currently, he is an assistant professor in the Department of Electrical Engineering, IIT Kanpur.

His research interests lie in the areas of Signal processing and Communication Networks. His current research focuses on cross-layer network optimization and dynamic network monitoring.

**Emiliano Dall'Anese** (M'11) received the Laurea Triennale (B.Sc Degree) and the Laurea Specialistica (M.Sc Degree) in Telecommunications Engineering from the University of Padova, Italy, in 2005 and 2007, respectively, and the Ph.D in Information Engineering from the Department of Information Engineering (DEI), University of Padova, Italy, in 2011. From January 2009 to September 2010 he was a visiting scholar at the Department of Electrical and Computer Engineering, University of Minnesota, USA. Since January 2011, he has been a post-doctoral associate at the Department of Electrical and Computer Engineering and Digital Technology Center, University of Minnesota, USA.

His research interests lie in the areas of signal processing, communications, and smart power systems. Current research focuses on optimization and control of power distribution systems with renewable sources of energy; robust, distributed, and sparsity-leveraging statistical inference for grid analytics; and, monitoring and optimization of communications networks.

**Georgios B. Giannakis** (F'97) received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engineering, 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. Since 1999 he has been a professor with the Univ. of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing - subjects on which he has published more than 330 journal papers, 550 conference papers, 20 book chapters, two edited books and two research monographs (h-index 98).

Current research focuses on compressive sensing, cognitive radios, cross-layer designs, wireless sensors, social and power grid networks. He is the (co-) inventor of 21 patents issued, and the (co-) recipient of 8 best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a distinguished Lecturer for the IEEE-SP Society.