A Fast Lightweight Approach to Origin-Destination IP Traffic Estimation Using Partial Measurements

Gang Liang, Nina Taft and Bin Yu

***Abstract*— In this paper, we propose an approach to estimating traffic matrices that incorporates lightweight Origin-Destination (OD) flow measurements coupled with a computationally lightweight algorithm for producing the OD estimates. There are two key ingredients in our method, called PamTram, for PArtial Measurement of TRAffic Matrices. The first is to actively select a small number of informative OD flows to measure in each estimation time interval. To avoid the heavy computation of an optimal selection, we use a heuristic based on intuition from game theory. Randomized selection rules are developed based on the goals of reducing errors and adapting to traffic changes. We provide an algorithm for selecting a good flow to measure that is fast because it avoids the computations, such as integrating over past intervals, that are needed for optimal selection. The second key aspect of our method is an explanation and proof that an Iterative Proportional Fitting (IPF) algorithm can be used to approximate the traffic matrix estimate when the goal is a minimum mean squared error and the optimization starts from a maximum entropy initial estimate.**

In addition, we provide a one-step average error bound for PamTram when the randomized selection rule is uniform and no link counts are used. This bounds the average error for the worst case selection rule. Finally, we validate our method using data from Sprint's European Tier-1 IP backbone network. Results show that our method generates average errors below the 10% carrier target error rate. Interestingly, we show that it suffices to measure a single OD flow in each estimation interval, which renders our partial measurement method very lightweight in terms of measurement overhead.

***Index Terms*— iterative proportional fitting, minimax, origin-destination traffic matrix, partial measurement, statistical game**

I. Introduction

Origin-destination (OD) traffic matrices are network profiles that quantify the volume of traffic flow between all pairs of nodes in a given network. The traffic represented is the demand as it captures the traffic that originates at one node and is destined for the other node. Such matrices serve as important inputs for a variety of network traffic engineering tasks, including capacity planning, load balancing, and traffic provisioning; hence, the problem of estimating OD traffic matrices for backbone networks has recently attracted much interest from both service providers [1], [2], [3] and the research community [4], [5], [6], [7], [8].

In communication networks, packets are forwarded by internal routers or switches according to a routing scheme. The origin and destinations of these data packets are edge nodes that lie behind access routers. A general traffic matrix can be defined at any level of granularity: the traffic sources and destinations could be hosts, groups of hosts, routers or even PoPs (a large collection of co-located routers). The specification of a particular traffic matrix requires the selection of the level of aggregation. In a router-to-router traffic matrix, the traffic considered to be "sourced" at a given router includes all of the clients and peers attached to that router. Most research has focused on either router-to-router or PoP-to-PoP matrices, and we continue in the same vain, as these are the ones ISPs are primarily interested in. For a network with n_e edge (or access) nodes, the number of possible OD traffic flow pairs is n_e^2. The OD matrix also has a timescale associated with it - each entry gives an average volume level over some time interval (1 min, 1 hour, 1 day, etc.). Traffic matrices should be thought of as 3-dimensional matrices in which the third dimension is time. Each OD traffic flow is actually a time series, and thus the entire matrix evolves over time. It has been shown ([2], [9]) that traffic matrices are quite dynamic and exhibit strong diurnal patterns thus varying a great deal within a 24 hour period.

Current approaches for obtaining traffic matrices can be classified into two categories: direct and indirect. A direct approach is a pure measurement one in which the entire traffic matrix is repeatedly measured over time via monitoring technologies such as Netflow on Cisco routers. This software can either be resident on routers or located on separate monitoring equipment. In [2], the authors explicitly calculated the overheads of direct measurement using state-of-the-art flow monitors. They showed that today's solutions, which essentially mandate a centralized solution, are prohibitive in terms of communication and computation costs. They also illustrated that by moving towards a more distributed approach, the computation costs fall but the communications cost of full measurement (albeit smaller) still remains high.

The indirect approach relies on alternative data that is more readily available in networks, yet is incomplete. In particular, the Simple Network Management Protocol (SNMP), supplies statistics on links (such as total bytes seen in a given time window) and is widely deployed in today's ISP networks. SNMP supplies the link load levels every 5 minutes in most commercial networks. This is only partial information because typically the number of internal link constraints is much smaller than the number of OD pairs, thus creating an ill-posed problem. Vardi (1996) was the first to investigate the problem of estimating OD matrix through link traffic counts, and coined the term "network tomography" to illustrate its similarities with medical tomography. The challenge of the indirect approach lies in its ill-posed nature. For a general

Gang Liang is with the Department of Statistics, University of California at Irvine, Irvine, CA 92697, USA. (email: liang@uci.edu)

Nina Taft is with Intel Rearch Berkeley, Berkeley, CA 92710, USA. (email: nina.taft@intel.com)

Bin Yu is with the Department of Statistics, University of California at Berkeley, Berkeley, CA 92720, USA. (email: binyu@stat.Berkeley.EDU)

network, the number of links is usually proportional to the number of edge nodes n_e, which grows much more slowly than the number of OD pairs n_e^2. The problem becomes severely under-constrained even for a modest n_e. For instance, in a backbone network, n_e is is in the range of 20-40 at the PoP level, and is on the order of hundreds at the backbone router level.

Many approaches to tackle these problems try to find a simple model for OD flows, introduce constraints to ensure the identifiability of the model, and then employ some form of maximum likelihood estimation. Simple models, if they can work, are attractive in that they limit the computations from becoming excessive. Vardi [5] proposed a Poisson model assuming iid (independent identically distributed) Poisson distributions for the OD traffic byte counts. Based on LAN network data, Cao et. al. [4] revise the Poisson assumption to propose a Gaussian model coupled with an assumption of a power-law relationship between the mean and variance of an OD flow. Vaton and Gravey [6] propose an empirical Bayesian method and an iterative algorithm is used to learn the prior distribution. In [10] the authors proposed the use of gravity models for determining initial conditions for optimization methods (such as maximum likelihood estimation) to avoid local minima problems. In Zhang et. al. [1], a tomogravity model is proposed to regularized the gravity parameter estimate such that the final estimate is also faithful to the SNMP link counts. The computation of these methods is usually very high. Liang and Yu [11] propose a pseudo likelihood method to speed up the parameter estimation for general network tomography problems.

A key question regarding the indirect approaches is to what level of accuracy can the hidden OD traffic be recovered simply from aggregated link traffic counts? Most of the indirect methods achieve *average* errors in the range 20-30%. However carriers are hoping for error rates to fall below the 10% barrier. In order to achieve lower error rates, recent research seeks to obtain yet more data (referred to as *side information* in statistics) to bring into the problem. Nucci et. al. [9] propose to use routing changes to obtained more information about the underlying OD traffic. Zhang et. al. [10] use SNMP data not only from inter-router links (as in the traditional problem), but also from access and peering links in order to populate the gravity model.

In this paper we propose the approach of using partial OD flow measurements as a good type of side information to bring into the problem. The idea is to measure a small number of OD flows (e.g., one) directly using a flow monitor, in each measurement interval, and then to vary the flow(s) measured over the course of time. This idea was originally proposed in [12]; however in that short paper neither the theoretical foundation for this approach nor any validation using data was carried out. We do both of those herein. Three partial flow measurement approaches were proposed and evaluated in the comparative study done in [8]. The notion of partial flow measurement in those approaches is different because they all propose to turn flow monitors on at all routers, for a period of 24 hours to measure the traffic matrix throughout its diurnal cycles. All flow monitors are then turned off until sufficient change has been detected so as to require them to be activated again, for another period of 24 hours, in order to recalibrate the underlying models. While these approaches proved useful, the one we include in this study is far more lightweight. The measurement overhead in that study varied from 5-30% depending upon the particular scheme; using their same overhead metric, our approach yields a measurement overhead of 1-5%.

Our contributions in this paper are multiple. First we introduce a simple model to capture the 1-step temporal transitions of a traffic matrix. Although this model does not match the full OD flow behavior, and is not as rich as the models in [8], we illustrate that it is sufficient for the purposes of accurate TM estimation and enables the use of less intensive computations. Second, we propose a methodology that iterates over two steps. In one step we propose a mechanism to select the one OD flow that will be measured in each interval, and in the second step we compute an approximation to the minimum Kullback-Leibler divergence estimate to populate the traffic matrix. To select which flow(s) to measure we employ a game theoretic randomization scheme to choose informative OD pairs. Different OD flows will be measured in different time intervals and the choice of which flows to measure is based on previous estimates. Intuitively speaking, our choice of which pair to measure is based on the probability that an OD flow will generate large errors. The benefit of this approach is that it permits adaptation to dynamic changes in the traffic matrix. When changes in particular OD flows occur, those flows are likely to generate larger errors; as our method progresses in time, it eventually catches these changes. We contend that the original ill-posed problem can be substantially improved even if only a tiny fraction of OD pairs are measured in each time interval.

Third, we prove that the iterative proportional fitting (IPF) algorithm can be used for our two critical computational steps: (i) it approximates the minimum Kullback-Leibler divergence estimate (as used in [1], and (ii) can also be used to implement our game for selecting which OD flow to measure. Because IPF can be used inside these two steps of our methodology, our overall procedure yields an efficient and fast algorithm that is thus practical to implement. To handle the practical issue of disseminating a command from a centralized station to a router instructing it to take a measurement, we consider the possibility of delayed execution of our OD flow measurement choices. We demonstrate that even with 24 hours delay, our schemes are very effective. Fourth, we compute a bound on the 1-step error (the error of each successive estimate) and illustrate that the errors drop very quickly after a few iterations. Finally, we validate our methods on real data from a Tier-1 operational backbone.

This paper is organized as follows. In Section II, we briefly state the OD traffic estimation problem. In Section III, we explain our approach to partial measurement and introduce a few minimax randomization selection schemes for selecting those traffic matrix elements to measure. We prove that the IPF procedure can be used to approximate both a minimum mean squared estimate and to execute our game theoretic selection scheme. We also discuss the evolution of errors and derive a bound on the error from one interval to the next. The above methods are evaluated on a real network dataset in Section IV. We evaluate a number of performance metrics, including temporal and spatial errors, adaptability and overheads. In order to understand how well our minimax type schemes perform, we compare them to an oracle-based scheme in which full knowledge about which pairs are most informative is assumed. We conclude our paper in Section V and provide proofs of the theorems in the Appendix.

II. OD Traffic Estimation Problem

We denote the SNMP link counts as $Y = (Y_1, ..., Y_J)$ for a network with J links. Let $X = (X_1, \cdots, X_I)$ be the vectorized version of the traffic matrix where X_i denotes the i-th OD flow (for a total of I OD pairs). The OD traffic matrix X has been aligned into a vector for the convenience of mathematical manipulation. As in [5] and [4] there is a linear relationship between the unobserved X and observed Y:

$$Y = AX, \tag{1}$$

where A is an $J \times I$ routing matrix, determined by the network topology and the routing protocol. Mostly, elements of A take on the value of 0 or 1 where $A_{j,i} = 1$ if OD pair i traverses link j, and $A_{j,i} = 0$ otherwise. The elements of A could take on fractional numbers when traffic splitting is allowed. Such Markovian routing schemes are discussed in [5]. In this paper, we assume the network routing is fixed during each measurement period, and the routing matrix A is known.

Since at each time point, a traffic matrix is also naturally represented using two dimensional arrays, we will sometimes use the notation $X_{o,d}$ to denote the specific OD traffic from origin node o to destination node d. The total traffic originated at node o is represented by $X_{o,\cdot}$, and $X_{\cdot,d}$ captures the total traffic destined for node d. It should be clear from the context when the vector and matrix notations are used.

Since in the system $Y = AX$, A is known and Y is observable, the goal of the traffic matrix estimation problem is to recover X. Typically A is not full rank with $J \ll I$, so the estimation of the distribution of X is an ill-posed inverse problem. Constraints have to be introduced to ensure the identifiability of the model. Statistical modeling can be viewed as introducing constraints by taking characteristics of network traffic dynamics into account. There is a rich literature in statistics ([13], [14]) devoted to this topic from the point of view of regularization.

III. Methodology

A. Partial Measurement

On of the central ideas in our method is that of coupling the inference activity with the direct measurement of a small number (possibly just one) OD flow. To do this, it would be necessary for flow monitors to be universally deployed throughout a network. One might ask, if flow monitors are deployed everywhere, why not just measure the traffic matrix entirely? In [2] the authors outline the overheads involved for both centralized and distributed versions of full direct measurement. In both cases, the communications cost (information being shipped to a central Network Operations Center) remains very high. For this reasons, it is interesting to consider more lightweight uses of direct OD flow measurement. Our approach of measuring one, or a small number, of flows in each measurement interval is indeed very lightweight.

A recent discovery illustrated that seemingly high dimensional network OD traffic actually lives in a much lower dimensional space [15]. This provides compelling intuition for an active partial measurement approach, since it implies that there is potential to learn a great deal about all the flows by only measuring a few of them. In practice, it is challenging to get a low rank representation because the network traffic is volatile; hence, the representation changes over time. Our proposed partial measurement approach is to use only a few active measures to obtain some vital information to explore this low dimensional space dynamically. We contend that the original ill-posed problem will become more well-posed even if only a tiny fraction of OD flows are selected to be measured directly at each time point and the measured OD flows from previous time points also help due to the often present smoothness of the OD traffic.

B. Modeling

To model the OD flows, some previous efforts have chosen to assume that an OD flow is either Poisson or Gaussian. In this work we focus on the conditional random variable $X^{(t+1)}|X^{(t)}$, which is assumed to be a Gaussian distribution (We define η shortly.)

$$X^{(t+1)}|X^{(t)} \sim N(X^{(t)}, \eta^{(t)}\mathrm{diag}(|X^{(t)}|)). \tag{2}$$

Note that the same power mean-variance relationship Gaussian distribution was used in [4] and [11], and recently verified in [16]. Here, instead of on the marginal distribution of $X^{(t)}$, we have it on the conditional distribution of $X^{(t+1)}$ given $X^{(t)}$, and the unknown parameter $\eta^{(t)}$ characterizes the variability of the network traffic at time t. The $\eta^{(t)}$ varies over time, and we assume that they are bounded by a constant $\eta > 0$, that is, $\eta^{(t)} < \eta$. Empirical studies based our dataset in Section IV suggest small values of $\eta^{(t)}$ (cf. Fig. 4 (c)). Equation (2) accounts for the phenomenon that large flows have large variations, and this linear mean-variance relationship will be validated in the experimental section. The covariance matrix of this conditional distribution is diagonal, implying that all OD flows are independent of each other: it is an approximation to the real network traffic. The absolute value in the covariance matrix is introduced to ensure mathematical accuracies. In reality, the OD flow $X^{(t)}$ is always non-negative; as will be seen below in our approach the estimate of $X^{(t)}$ based on the model is always guaranteed to be non-negative.

The motivation for this conditional model is to introduce a time series structure between consecutive OD flows. In this paper, we are interested in monitoring network OD traffic continuously such that there is one traffic matrix (TM) estimate available at each time interval. This conditional model enables us to combine *past* traffic matrix estimates and the current link counts together to produce an estimate of a *current* traffic matrix. There are many ways to incorporate previous estimates, such as using it as an initial condition for an optimization procedure. To populate our traffic matrix we will use an estimate based on the expectation of the current random variable conditioned upon the link constraints and the additional measurements we obtain, given the immediate previous estimate.

In this approach, the transitions of the traffic matrix from one time interval to the next is controlled by the parameter η and small η's imply that these transitions are not excessive. Clearly the validity of non-excessive transitions depends upon the time scale the matrix intends to be used for. In our case, we make estimates of a traffic matrix every 10 minutes. Our model is intended to capture *local* behavior, that is, ”local” in a temporal sense (over a short window of time). We realize that our model would not be an accurate description of traffic over long timescales such as many hours or days. However, our intent is to capture the transitional behavior of a TM from one (short) interval to the next. Even though this model is coarse, it nevertheless works quite well for the purposes of TM estimation, and retains the advantage of being simple.

This modelling assumption has an alternate interpretation as a state-space model, which is used to describe internal unobservable states that evolve over time. The relationship between the observable and unobservable variables is usually specified as linear functionals typically with noise terms. In terms of state-space system notations, our model can be rewritten as follows:

$$X^{(t+1)} = X^{(t)} + \sqrt{|X^{(t)}|}\epsilon^{(t)} \tag{3a}$$

$$Y^{(t+1)} = AX^{(t+1)}, \tag{3b}$$

where the observable link traffic $Y^{(t)} \in \mathcal{R}^J$ is a linear function of the unobservable OD traffic $X^{(t)} \in \mathcal{R}^I$ at time t. The routing matrix A, relating the unobservable states and observations together, is a known sparse matrix (i.e., with many zero entries). The errors $\epsilon^{(t)}$ are identical independent distributed normal random variables:

$$\epsilon^{(t)} \sim N(0, \eta^{(t)}), \tag{4}$$

where, as discussed earlier, $\eta^{(t)}$ $(< \eta)$ is a unknown parameter quantifying the dynamics of the underlying OD traffic. We would like to comment that there is no need to estimate $\eta^{(t)}$'s in the OD flow estimation and selection because they only serve as scalers irrelevant to the final estimate. It will become clear by the equations in the following sections.

The partial measurements can be incorporated into our model as follows. Let $M^{(t)}$ be the measurement matrix at time t, and $Z^{(t)}$ be the new vector of observations. We append $M^{(t)}$ below the routing matrix A according to,

$$C^{(t)} = \begin{pmatrix} A \\ M^{(t)} \end{pmatrix},$$

where $M^{(t)}$ is a $k \times I$ matrix in which each row is a unit vector: it contains e_i' if $X_i^{(t)}$ measured. The matrix $C^{(t)}$ now describes the relationship between the OD flows and link counts. Thus $Z^{(t)} = C^{(t)}X^{(t)}$ is the total observation available at time t. The first J entries in this vector contain the link counts while any additional entries contain the measured OD flows. In this paper, k, the rank of $M^{(t)}$ is preset, i.e., the number of OD pairs to be measured is determined. It is possible to treat it as a tuning parameter in different scenarios, however we find excellent performance when $k = 1$ and hence there is little motivation to explore other values (at least for the dataset we study).

Equation 3b is now replaced so that our new system equations, with the measurements incorporated, are given by

$$X^{(t+1)} = X^{(t)} + \sqrt{|X^{(t)}|}\epsilon^{(t)} \tag{5a}$$

$$Z^{(t+1)} = C^{(t)}X^{(t+1)}, \tag{5b}$$

Given this system we will populate our traffic matrix with the estimate

$$\hat{X}^{(t+1)} = E(X^{(t+1)}|X^{(t)} = \hat{X}^{(t)}, Z^{(t)}) \tag{6}$$

Algorithm 1 Summary of the PamTram approach

```
Initialization: Set $\hat{X}_0 = \mathbf{1}$
for each time interval $t$ do
    1. Measure OD pairs selected at step $t-1$;
    2. Estimate $X^{(t)}$ based on data $Z^{(1)}, \cdots, Z^{(t)}$ as in (6);
    3. Determine OD pairs to measure at $t+1$.
end for
```

Our proposed PamTram approach is summarized in Algorithm 1. In this method, the initial traffic matrix $\hat{X}^{(0)}$ is set to be component-wise vector 1. This initial choice of traffic matrix is not very important as the algorithm will quickly adjust itself to the right region. We could start with any constant vector, and this is equivalent to a maximum entropy estimation in the following sense: after normalization by the total OD traffic (which is naturally done during IPF), the OD traffic problem can be viewed as finding the best projection to the linear space of probability distributions specified by the link equations in terms of Csiszar's I-projection [17]. Any constant vector thus corresponds to a uniform distribution starting point to be projected. This is intuitively appealing because a maximum entropy estimate implies that we start knowing nothing and thus need no prior knowledge. Step 2 corresponds to the usual optimization problem for traffic matrix estimation. Many of the previous methods could be applied here. We will provide a fast implementation of an existing method. The challenge in step 3 is to determine which OD flows to measure.

In the subsequent sections, we will explain how we carry out steps 2 and 3 of this method. Before doing so we first introduce our error metric because minimizing this error is the objective of our optimization problem in step 2, and because these errors are also used in step 3 to assist in selection of which OD flow to measure. The intuition is to select flows that will reduce errors.

C. Error Metric

In this paper, we propose to use a variant of the mean square error (MSE) as the error metric to assess the performance of an estimator. Let $\hat{X}$ be an estimate of the unknown OD traffic X, then the MSE of $\hat{X}$ is defined as

$$\text{MSE}(\hat{X}, X) = ||\hat{X} - X||^2.$$

One drawback of the MSE metric is that it is very sensitive to large traffic flows' errors which are usually large in absolute scale. Model (3) postulates that the variance is proportional to the mean (conditionally on the previous traffic flow) and this relationship is used to devise the following scaled metric to mitigate the problem of MSE:

$$\text{sMSE}(\hat{X}, X) = \frac{||\hat{X} - X||^2}{||X||_1} = \frac{\sum_i (\hat{X}_i - X_i)^2}{\sum_i |X_i|}.$$

Other error metrics have been used in the past (e.g., [10]); a common one is the relative error defined as :

$$\text{Rel-Error}(\hat{X}_i, X_i) = \frac{|\hat{X}_i - X_i|}{|X_i|}.$$

It has been shown that in real networks, roughly 95% of the total load in the traffic matrix is carried by less than 1/2 or 1/3 of the flows [9]. Moreover, the volume of flow in these OD pairs can span several orders of magnitude. Hence there are typically very small traffic flows that generate extremely large relative errors but are essentially irrelevant. Our scaled MSE metric avoids this drawback, and works well as a performance metric for both large and small flows. We point out that in practice, the relative error is a useful measure to network operators as it is intuitively appealing; thus we also report on this metric in the results section.

Since we will use this sMSE error metric for assessing performance, we can also use it as our objective function inside the optimization of step 2 in our method. It is important to note that the scaling factor is a quantity that does not involve $\hat{X}$, hence in effect, minimizing the scaled MSE is the same as minimizing the MSE metric.

Based on the model we are using, we can derive a bound on the expected scaled MSE error. Suppose for a moment that $X^{(t)}$ were known and we take no measurements. Then using $X^{(t)}$ as a sensible estimate for $X^{(t+1)}$, the average error would be

$$\text{E}\left(\text{sMSE}(X^{(t)}, X^{(t+1)})\right) \approx \eta^{(t)} \leq \eta.$$

This indicates that the expected error will be close to $\eta^{(t)}$, which quantifies the variability of the traffic (according to our model) if we start from the previous true OD flow $X^{(t)}$. In practice, two factors are at play. On one hand, we do not know $X^{(t)}$ but only its estimate $\hat{X}^{(t)}$. On the other hand, we can add measurements to better the parameter estimation. The final expected error metric will be influenced by both factors.

D. IPF and Minimum Mean Square Error Estimation

We now state the algorithm, and its properties, for estimating OD traffic when both link traffic counts and some direct measurement information are gathered (Step 2 in Alg. 1). Since our goal is to estimate the traffic matrix on the order of once every 10 minutes, we seek a fast online solution. In this paper, we propose to use the iterative proportional fitting (IPF) algorithm to populate the traffic matrix. This IPF algorithm was used first by Cao et al. [4] as a post-processing step in their OD estimation algorithm based on a Gaussian OD traffic model. We describe IPF in the framework of I-projection in the space of probability distributions.

I-projection, first studied by Csisźar [17], gives a geometric view to minimum Kullback-Leibler (KL) divergence inference problems, where the KL divergence plays the role of squared Euclidean distance. Given a probability function q, minimizing KL divergence over the first argument

$$\hat{p} = \arg\min_{p \in \mathcal{L}} D(p||q). \tag{7}$$

can be viewed as a problem of projecting q into a convex set $\mathcal{L}$ of a probability distribution space. Linear constraints are special cases of convex sets. Algorithmically, this geometric view suggests that an alternating minimizing type of algorithm [17] is useful for solving the minimization problem (7) if the constraint set $\mathcal{L}$ is the intersection of a series of convex constraint sets $\{\mathcal{L}_l : l = 1, \cdots, L\}$. Several iterative algorithms [18], [19], [17] have been proposed to solve problem (7) with only linear constraints.

The OD flow estimate $\hat{X}^{(t)}$ is component-wise nonnegative, so it can be viewed as a probability function after scaling, and the IPF procedure, which applies when all $f_l(y)$ take only 0-1 values, is most relevant in OD traffic matrix inference. Below is the pseudo code of the IPF algorithm:

Algorithm 2 IPF Algorithm

```
Given $\mu = \hat{X}^{(t-1)}$ and $Y = AX^{(t)}$;
for $k = 1, \cdots, K$ or till converge do
  for $j = 1, \cdots, J$ do
    $\alpha = \sum_i 1_{\{A_{j,i}=1\}} \mu_i / Y_j$
    $\mu_i = \mu_i / \alpha$ for all $i$ with $A_{j,i} = 1$
  end for
end for
return $\mu$.
```

There are several appealing properties for IPF: 1) it is easy to implement; 2) it converges in exponential rates (cf. Liang et. al. [12]), and is thus very fast in practice. The IPF algorithm can be run satisfactorily in the order of $O(IJ)$ with a preset finite number of iterations. The starting point at time t is determined by $\hat{X}^{(t-1)}$, the estimate obtained from the previous step. It is reasonable to expect the starting value to be in a small neighborhood of the OD traffic $X^{(t)}$ to be estimated; this further speeds up the convergence rate. Further, Theorem 1 justifies the use of IPF for the OD flow estimation from a statistical viewpoint.

Theorem 1: For the network dynamic model (3), conditioning on $X^{(t-1)}$, if the mean vector μ (i.e., $X^{(t-1)}$) is assumed known, then the IPF estimate of $\hat{X}^{(t)}$ is approximately the minimum MSE estimate.

This implies that the iterative proportional fitting (the I-projection estimate) approximately gives the minimum MSE estimate when $\mu = X^{(t-1)}$ is known. In a real problem, $X^{(t-1)}$ is unknown hence replaced by the previous estimate

$\hat{X}^{(t-1)}$. Another advantage of the algorithm is that the resulting OD flow estimate is positive, which is not guaranteed by the minimum MSE error estimate.

The true minimum MSE estimate of $X^{(t)}$ is the conditional expectation of $X^{(t)}$ given all observations:

$$\mathrm{E}^{(t)} = \mathrm{E}\left(X^{(t)} \,\middle|\, Z^{(1)}, \cdots, Z^{(t)}\right).$$

The computation of such a quantity is very high: it involves an integration over all past data points. In order to avoid such high computational cost, an approximation is use instead

$$\begin{aligned}\mathrm{E}^{(t)} &= \mathrm{E}\left(\mathrm{E}\left(X^{(t)}|X^{(t-1)}, Z^{(t)}\right) \,\middle|\, Z^{(1)}, \cdots Z^{(t)}\right) \\ &\approx \mathrm{E}(X^{(t)}|\hat{X}^{(t-1)}, Z^{(t)}).\end{aligned}$$

As stated in Theorem 1, the IPF algorithm can be used to compute this conditional expectation approximately starting from $\hat{X}^{(t-1)}$.

E. Measurement Selection Scheme

We now address the issue of how to select the OD flows to measure in each time interval (Step 3 in Alg. 1). The idea is to choose a scheme that will select the most informative of the unobservable flows. Clearly, the choice has to be made based solely on the observable variables. We focus on selecting a single OD flow because even just measuring one OD flow per interval provides excellent performance. Our ideas here could be generalized to selecting a few flows, although we don't believe there is much motivation to do so.

We realize that in practical systems, since the flow monitor is attached to a link when we turn it on, we will in fact capture all the flows sourced at that link. If the source node is a router and we turn a flow monitor on at all the links of this router, then again, we will obtain more OD flow measurements than one (in fact, we would obtain one row of the traffic matrix). However, in this paper we study the case of measuring only a single OD flow to understand the impact of this idea. In practice since we have more than one OD flows the errors will be lower than we calculate using only one OD flow. Our methodology for selecting which OD flow to measure remains the same whether or not just one OD flow is used or all elements of the corresponding OD row are used.

First, let us consider what an optimal solution would entail. Suppose X is a multivariate random variable (not necessary normal) with: $E(X) = \mu$, and $Var(X) = \Sigma$, where both μ and Σ are known (or can be estimated). Then the minimum MSE predictor for X is just μ with the MSE error

$$E||X - \mu||^2 = \mathrm{trace}(\Sigma).$$

Hence ideally, we would like to select an OD pair such that the resulting conditional covariance matrix given all observations

$$\Sigma^{(t)} = \mathrm{Var}\left(X^{(t)} \,\middle|\, Z^{(1)}, \cdots, Z^{(t)}\right) \tag{8}$$

has the smallest trace. In other words, to determine the observation matrix $M^{(t)}$ such that the trace of the conditional variance is minimized

$$M^{(t)} = \arg\min_{M^{(t)}} \mathrm{trace}\Sigma^{(t)}.$$

But this approach is not attractive because the computation of $\Sigma^{(t)}$, involving integration over all past observations, is too costly. Neither can we use

$$\mathrm{Var}\left(X^{(t)} \,\middle|\, X^{(t-1)} = \hat{X}^{(t-1)}, Z^{(t)}\right)$$

as an an approximation to the conditional covariance (8) because in general $\mathrm{Var}(D) = \mathrm{Var}(E(C|D)) + E(\mathrm{Var}(C|D))$, and it is difficult to approximate

$$\mathrm{Var}\left(E\left(X^{(t)} \middle| Z^{(t)}, X^{(t-1)} = \hat{X}^{(t-1)}\right) \middle| Z^{(1)}, \cdots, Z^{(t)}\right).$$

Since using $\Sigma^{(t)}$ to choose the optimal OD flow to measure is too computationally intensive, we develop instead heuristic randomization schemes motivated by game theory. Consider for a moment a uniform randomization scheme in which each OD flow is picked to measure at next step with equal probability $1/I$. The following theorem bounds the one-step error performance (the error made from one interval to the next) assuming uniform random sampling of flows.

Theorem 2: Let $\omega^{(t-1)}$ be the scaled MSE error at step $t-1$,

$$\omega^{(t-1)} = \mathrm{sMSE}\left(\hat{X}^{(t-1)}, X^{(t-1)}\right).$$

Assume no link measurements Y are made, and only one OD pair is selected for measurement by uniform random sampling, then the expected value of $\omega^{(t)}$, the error metric of $\hat{X}^{(t)}$, is approximately bounded by

$$E(\omega^{(t)}) \leq \frac{I-1}{I}\omega^{(t-1)} + \eta^{(t)} \leq \frac{I-1}{I}\omega^{(t-1)} + \eta,$$

where I is the number of total OD pairs. When t goes to infinity, $J\eta$ is an upper bound of the expected error.

This theorem is a comforting result in the sense that using a uniform randomization scheme is not going to lead to an error metric that can grow without bound. All of our alternative randomization schemes produce errors less than the uniform randomization scheme hence this bound on the average error appears to be a worst case bound. In the theorem, the expected value of the next step error metric is bounded by the sum of two parts: the first is the reduced previous error metric due to the randomized measurement, and the second part is η, that is due to the intrinsic uncertainty of the network dynamic system. When t grows, the expected error will be bounded regardless where we start. This is an appealing property of the randomized measurement approach.

In practice, link measurement $Y^{(t)}$ is obtained, then the residual of the parameter estimate at time t is

$$R^{(t)} = X^{(t)} - E\left(X^{(t)} \middle| X^{(t-1)} = \hat{X}^{(t-1)}, Y^{(t)}\right).$$

In order to reduce the scaled MSE (or equivalently the MSE), one should measure the OD pairs with the largest absolute residual(s). Note $\hat{X}^{(t-1)}$ is only an estimate. The picking the large residual can be viewed as a game between the nature (the network) and us, in which the nature sets the starting value $X^{(t-1)}$ against us. Suppose our goal is to maximize the probability of picking the largest residual, i.e.,

$$L(X^{(t-1)}, i) = 1\left(R_i^{(t)} = \max_j R_j^{(t)}\right).$$

The next theorem shows that the *uniform* is in fact the minimax rule, which guards ourselves against the worst scenarios.

Theorem 3: The uniform random sampling $(p(i) = 1/I)$ is the minimax decision rule of the pick-largest-random-number game with a 0-1 payoff (loss) function.

It is arguable that the network is an intelligent adversary; hence, choosing OD pairs uniformly will likely give relative poor results because the information in the previous traffic estimate is not exploited. Likely $\hat{X}^{(t-1)}$ will be close to the true state of the nature $X^{(t-1)}$ – that is, we can guess pretty well the move of the nature (network). If we assume $X^{(t-1)} = \hat{X}^{(t-1)}$, then $R^{(t)}$ is a mean zero normal random variable with variance (independent of $Y^{(t)}$)

$$\Lambda^{(t)} = \Sigma^{(t)} - \Sigma^{(t)} A'(A\Sigma^{(t)}A')^{-1} A\Sigma^{(t)}, \qquad (9)$$

where $\Sigma^{(t)} = \eta^{(t)}\mathrm{diag}(\hat{X}^{(t-1)})$, and the probability of $R_i^{(t)}$ being the largest residual in absolute value is

$$Q(i) = P\left(|R_i^{(t)}| = \max_j |R_j^{(t)}|\right). \qquad (10)$$

So a sensible randomization scheme is to pick ith OD flow with probability $P_{maxen}(i) = Q(i)$ in the above equation exactly. It is well know that it is the maximum entropy solution, which minimizes the negative log loss function if the distribution Q is assumed to be known:

$$P_{maxen} = \arg\min_P - \sum_i \log P(i) \log\left(\frac{Q(i)}{P(i)}\right) = Q.$$

Hence, we call this randomization scheme *maxen*.

The uniform and maxen randomization schemes approach the measurement selection from two different points of view. On one hand, the uniform scheme ignores the knowledge about the network from previous time intervals. On the other hand, the maxen randomization scheme is based on the rationale that the system will not change much and exploits the existing estimated OD traffic to its advantage. However, the network doesn't always behave the same way as before and the uniform scheme gives opportunities to depart from the existing model. Hence we combine these two schemes to produce a more efficient randomization scheme. Let $\alpha \in (0, 1)$. A weighted minimax randomization is defined as:

$$P_{wMaxen} = \alpha P_{uniform} + (1 - \alpha) P_{maxen}.$$

Here we assume that the parameter α is preset and can be tuned to adapt the network traffic pattern. We usually set it as a relative small number, such as 0.2, to favor more the existing estimated models. In the experiment section, we will discuss the performance difference of various randomization schemes.

For these randomization schemes, the uniform is easy to realize, but the implementation of the maxen randomization rule is difficult because the probabilities defined in (10) are hard to obtain. Instead of computing these probability explicitly, the maxen scheme can be implemented if we can generate multivariate normal random numbers with the covariance matrix specified in (9). Let $\mu = \hat{X}^{(t-1)}$, and

$$X^{(t)} \sim N(\mu, \eta^{(t)}\mathrm{diag}(\mu)),$$

then

$$X^{(t)} - \Sigma A'(A\Sigma A')^{-1}(AX^{(t)} - A\mu) - \mu \qquad (11)$$

is a mean zero multivariate normal random variable with covariance matrix $\Lambda^{(t)}$. But (11) requires the inversion of the matrix $A\Sigma^{(t)}A'$, which is expensive computationally. Again, the result from Theorem 1 shows that the IPF algorithm can be used to approximate

$$X^{(t)} - \Sigma A'(A\Sigma A')^{-1}(AX^{(t)} - A\mu).$$

It can be solved approximately by applying IPF algorithm to starting from $X^{(t)}$ to fit the link constraint $Y = A\mu$. Thus a maxen randomization algorithm can be devised as follows:

Algorithm 3 Maxen Randomization Algorithm

```
Let $\mu = \hat{X}^{(t)}$ and $y = A\mu$;
1. Generate $\tilde{X} \sim N(\mu, \eta^{(t)}\mathrm{diag}(\mu))$;
2. Project $\tilde{X}$ onto $\{X | y = AX\}$ to get $\breve{X}$ using IPF;
3. Pick the $j$th OD flow if $j = \arg\max_i |\breve{X}_i - \mu_i|$.
```

Note that the parameter $\eta^{(t)}$ is actually irrelevant in the above formulation because it only serves as a scaler.

In summary, the total computational cost of PamTram is at most two IPF algorithm costs. The IPF computation is light and scalable to larger networks. Furthermore, there are other benefits of the proposed wMaxen selection rule. First, since it is a random scheme, wMaxen is not very sensitive to the estimation of the probabilities in Maxen and still works well even if our model estimation is a bit off. Second, the dynamics of the network traffic may exhibit many dramatic changes occasionally, as shown in Fig 2(b). Our dynamic model is not trying to predict sudden traffic changes, but rather to adapt to them as quickly as possible in conjunction with a small number of partial measurements. Since the wMaxen scheme is based on our model estimation, it inherents this adaptivity. Moreover, because the wMaxen rule keeps a balance between measuring of large and small traffic flows, it works effectively against sudden large traffic pattern changes.

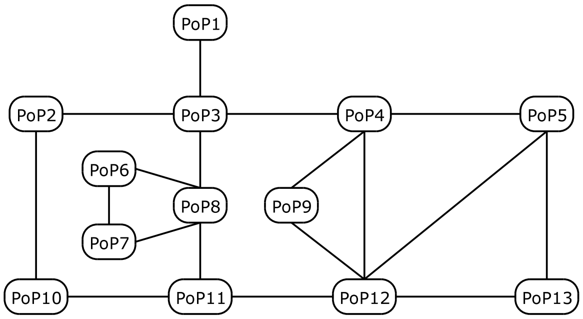
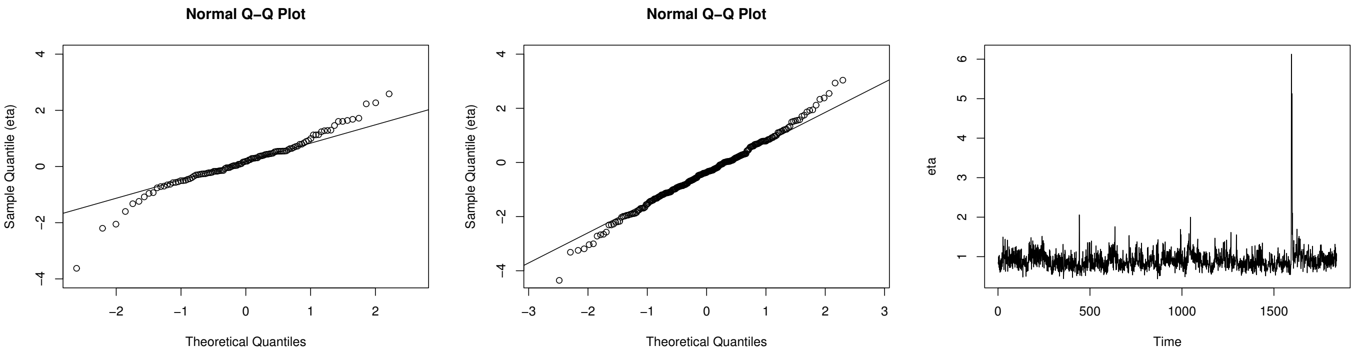


Fig. 1. The Sprint European PoP network topology.

IV. Experiments

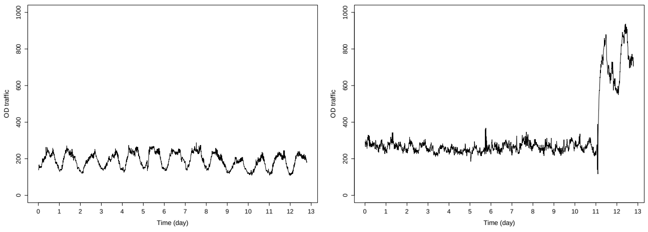
A. The Data

In this section, our proposed PamTram approach is validated using data from a real operational IP backbone. The data



(a) (b) (c)

Fig. 3. Exploratory data analysis. (a) and (b): Q-Q plots of $U_{t,i}$ within two chosen 5-point time windows: the first one is based on the time window 1-5 and the second time window 1601-1605; (c) Estimated $\eta^{(t)}$ over time.



(a) (b)

Fig. 2. Two sample PoP level OD traffic flows.

comes from Sprint's European backbone that constitutes 13 Points of Presence (PoPs) and 18 inter-PoP links. Fig. 1 shows the topology of the Sprint network at the PoP level. The network OD traffic information was collected by turning on Netflow (version 8) on all the Cisco routers. This version of Netflow uses a sampling scheme of monitoring 1 out of every 250 packets. The data was aggregated into PoP level flows at a time granularity of 10 minutes (i.e., average number of bytes sent between PoP pairs during each 10 minute window). The data collection interval of 10 minutes was chosen to mitigate possible measurement errors. To avoid inconsistencies between the link traffic and OD traffic, the link measurement data are derived from the flow level measurements X; this guarantees that the traffic matrix X, the routing matrix A and the link traffic counts Y are all in agreement with each other. This approach is well justified in [10].

We now show some behaviors of this OD traffic data that, although they have been pointed out before, are included here for completeness. Fig. 2 (a) and (b) show two time series plots of selected OD traffic flows. These two OD pairs are chosen because they represent common behaviors. The first one shows strong periodicity (very common among all OD pairs [15]). The strong periodicity of OD traffic also induces strong periodicity in almost all observed link traffic. The period of the traffic is exactly one day, while weekly period can also be seen over a longer time frame. The second one illustrates that sharp changes (day 11), different from the diurnal cycles and from the local noise, can occur. These can occur for reasons such as router failures, the addition of new customers, or the removal of previous customers.

The conditional linear mean-variance relationship is an important assumption in the dynamic network traffic model. Under the assumption, we have

$$U_{t,i} = \left(X_i^{(t)} - X_i^{(t-1)}\right) \Big/ \sqrt{X_i^{(t-1)}} \sim N(0, \eta^{(t)}).$$

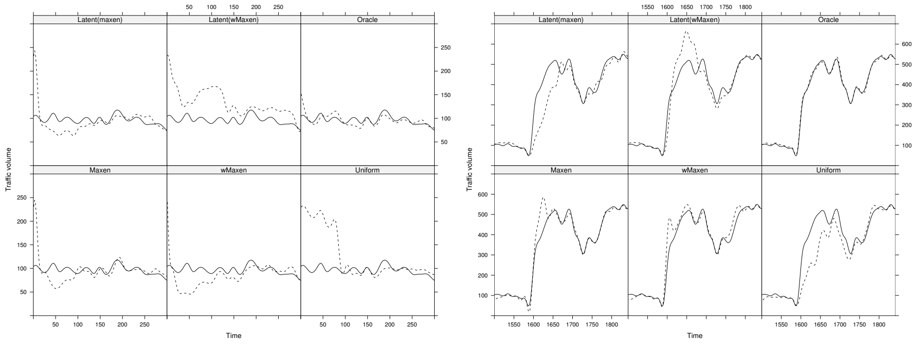
Even though $\eta^{(t)}$ varies over time, it is reasonable to assume that it is continuous, then we can estimate its value within each small moving window. Fig. 3 shows two QQ-plots at two different time points, and a time plot of the $\eta^{(t)}$ estimation. Because usually small OD traffic elements do not conform to the mean-variance relationship well, only the upper 90% of the traffic load are used to generate all three figures. On the other hand, the small traffic flows are less important. The Q-Q plots are produced based on all $U_{t,i}$ within a 50-minute window, i.e., 5 intervals. These two Q-Q plots are chosen because of their representativity; data in other time windows show similar features. Fig. 3(a) is drawn based on data points in the time window 1-5, and (b) within the time window 1601-1605. From both plots, we can see that the $U_{t,i}$ is very close to a normal distribution but with a longer tail. Fig. 3 shows the estimated $\eta^{(t)}$ over time. Because $U_{t,i}$'s have a longer tail than normal, a robust estimate of $\eta^{(t)}$ based on absolute moment is used:

$$\hat{\eta}^{(t)} = \left(\frac{\sum_i |U_{t,i}|}{0.799 \times I}\right)^2,$$

where $E(|V|) = 0.799$ for $V \sim N(0,1)$. From the plot, we can see that the values of $\eta^{(t)}$'s mostly oscillates around 1, which is very small given that a medium network traffic may take a value of several hundreds or thousands. There are occasional spikes in the figure – the most obvious one corresponds to the sudden traffic changes occurring around day 11. Other than this single point, the $\eta^{(t)}$ is well bounded. Overall, we have shown that the conditional linear mean-variance relationship is a good approximation to the raw data.

B. Partial Measurement Schemes

We tested PamTram using various partial measurement schemes to the Sprint PoP network data. The first three



(a) (b)

Fig. 4. Two selected time plots for estimated results of all PamTram schemes: (a) The inital portion; (b) The portion with the sudden traffic changes. In all panels, the solid line represents the true OD traffic.

are *uniform*, *maxen*, *wMaxen*. For the *wMaxen*, the weight parameter α is set as 0.2: we chose to make this number small to favor the maxen randomization scheme. Our experiments show that the scheme is not very sensitive to the choice of α. Furthermore, in order to better evaluate the performance of these randomization schemes, we also implemented an *oracle* scheme. In this scheme, the oracle has full knowledge of the true OD traffic and thus the largest residual can be precisely selected. In other words, we select the flow that results in the smallest scaled MSE error in the next parameter estimate. We can do this since we have the measured traffic matrix at our disposal. Although this cannot be done in practice, it provides us a means of assessing how far our schemes are from a sort of optimal (full knowledge) behavior. It is comforting that our schemes perform close to the oracle one.

In a real IP network, there is a practical problem with our scheme. The centralized node collecting the measurements and computing the estimates, needs to inform individual routers as to when to turn on a flow monitor to collect the chosen OD flow measurement. Such a command needs to be shipped across the network and the schedule loaded into the port of the relevant flow monitor. We assume that it would not be practical to do this every 10 minutes, however it is clear that selecting the measurement schedule a few hours in advance would provide the network ample time to disseminate and schedule the monitoring activities. We thus consider another version of our randomization scheme is which the OD flows to measure are selected 24 hours in advance. The idea is that a flow selected for measurement at 2:10pm on one day, is actually measured at 2:10pm the next day. The rational for such an approach comes from both the observation of strong daily periodicity (as in Fig. 2) which shows that traffic is generally similar from one day to the next at a particular time of the day, and from [2] in which the authors illustrate this notion more precisely using fanouts. In our implementation, the first day is special such that the OD chosen by the previous data point is used.

The latent scheme is a scheduling approach that needs to be combined with a randomized selection rule. In our experiment, two latent schemes are implemented: *Latent(maxen)* and *Latent(wMaxen)*. In total, six randomization rules are applied to this dataset: uniform, maxen, wMaxen, Latent(maxen), Latent(wMaxen), and oracle. In each case, we measure only one OD flow in each 10 minute measurement interval.

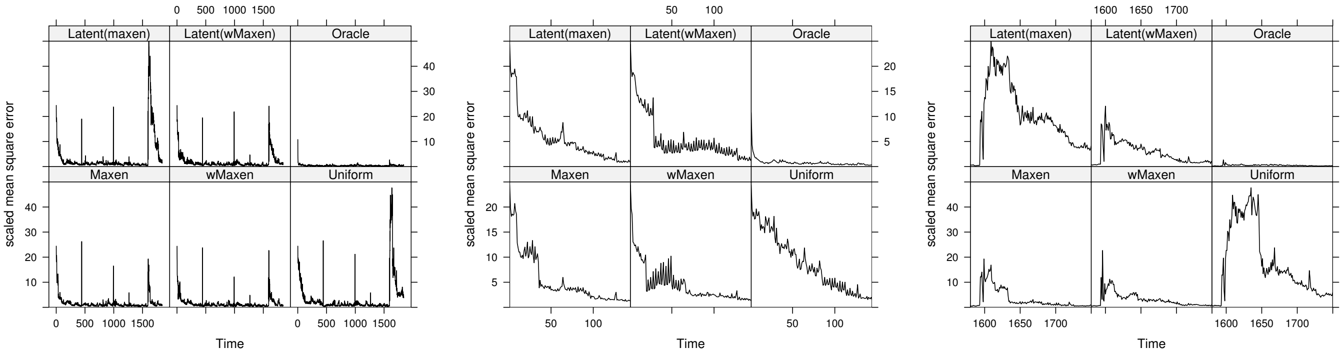
C. Experiment Results

Several performance measures are used to assess and compare all partial measurement methods. We look at both temporal errors (using the scaled MSE and relative error metric) and spatial errors. We will see how our error evolves over time and illustrate how these schemes adapt to unexpected traffic changes. We compare our results to some previous results. We discuss the tradeoff our method implies: the performance gain comes due to additional measurement overhead. We will show that the overhead incurred by PamTram is more lightweight than previously proposed partial measurement schemes.

Time plots. We start assessing the performance of PamTram estimators by viewing some raw time plots of OD flow estimates as shown in Fig. 4. Both figures are blown up in order to show more details. Because the network traffic is very volatile and hard to visulize, the smoothing spline method is used to remove unnecessary spikes of the true traffic while keeping the trend faithful. The same method is applied to estimated OD flows as well. Fig. 4(a) shows the initial trace of an OD flow (after smoothing) along with their estimates, and (b) shows a smoothed trace where a sudden change occurs. With only one additional measurement, all PamTram approaches adapt to the true OD traffic quickly. Below we will compare these schemes in various perspectives.

Scaled MSE errors. First we consider temporal errors. By temporal errors we mean that at each moment in time, we compute our error metric over all the flows giving a representative error for that timepoint. The ensemble of all such time points yields the temporal errors. Fig. 5(a) shows the scaled MSE plot along time for all six randomization methods. Overall, all partial measurement approaches drive the estimation error very low, even with only one OD flow is measured each time interval. We see that the majority of the errors are below 5%. This breaks new ground in terms of low error rates.

Evolution of Errors. In Fig. 5(b) we have blown up the initial portion of Fig. 5(a) by including only the first 150 estimation intervals. Overall, the scaled MSE decreases over



(a)

(b)

(c)

Fig. 5. The scaled MSE time plots: (a) the full scaled MSE plot; (b) the starting region; (c) the region with a large sudden traffic change.

time starting from the maximum entropy estimate for $X^{(0)}$ for all methods. As expected, the *uniform* performs worst and the *oracle* performs best. These plots illustrate that within 1 or 2 hours, we our errors drop below the 10%, and that within 3 or 4 hours, then can drop to below 5% (frequently). Although it takes a few hours to bring the errors down, this is not a lot considering that most operators will rarely start this procedure over from scratch; instead they will always have an old traffic matrix at their disposal that can be used as a starting point. This illustrates much less of a dependence upon the starting point than previous approaches [16]. For this first day the *maxen*, *wMaxen*, *Latent(maxen)* and *Latext(wMaxen)* all produce fairly similar behavior due to our implementation (of how we handle the latent schemes on the first day).

Adaptability. Recall that in Fig.2(b) we observed a sharp change on day eleven. We now examine how each of our methods adapted to this change (Fig. 5(c)). The fact that the *oracle* method essentially recovers immediately implies that our approach of selecting the OD flow to minimize errors is sound. The *uniform* scheme is strongly affected by this traffic change as the errors grow to around 40% and it takes many hours to return to lower error rates. The performance of *maxen* and *wMaxen* is comparable and good (the errors right to no more than 10-15% and recovery is quick). Interpreting the performance of *Latent(wMaxen)* and *Latent(maxen)* sheds some light as to the advantage and disadvantage of some of these schemes. The *Latent(maxen)* performs poorly because it is dependent upon yesterday's choices of which OD flows to measure. Since there has been a big change, these choices are not out of date and thus less relevant. In this case, adding in the uniform randomization selection permits PamTram to be less dependent upon out of date choices thus yielding lower errors. This validates the weighted randomization scheme *wMaxen* because it allows for practical issues to be met (requiring the use of a *Latent* approach) while simultaneously meeting the needs of a traffic matrix estimation technique to be able to adapt to major traffic changes.

Spatial Errors. The spatial errors give a different view of the errors in OD flow estimation. By spatial errors we mean that one error is computed per flow since the summing operation is done over time. This gives a summary error per flow over its lifetime. The ensemble of these errors illustrates the different errors experiences by different flows. In order to have a comparison with several other methods, we use the weighted L2 norm proposed in [8] to measure the spatial error:

$$d_{\text{spatial}}(i) = \sqrt{\sum_{t=1}^{T}\left(\hat{X}_i^{(t)} - X_i^{(t)}\right)^2 / \sum_{t}(X_i^{(t)})^2}.$$

Fig. 6 shows the ensemble of spatial errors across large OD flows. The OD flows in the plot are sorted in decreasing order according to their total traffic volume. The OD flows included in the plot constitute 99% of the total load. The two verticle bars represent the 90% and 95% cut-off points respectively (i.e., all the flows to the left of the bar constitute 90% or 95% of the load). We see in this plot that OD flow with smaller average size tend to have larger errors. This is a well known phenomenon and is consistent with results in almost all previous traffic matrix estimation papers. More importantly, we observe very small spatial errors for the majority of the traffic. All of these partial measurement schemes perform reasonably closely to the oracle one. For all schemes, except the *Latent(maxen)* and *Latent(uniform)*, the weighted L2 norm spatial errors are mostly below 40%. For comparison, we compute the average spatial errors for 95% of the total load. Latent(maxen) has the worst average spatial error: 18.4%, uniform 16.8%, Latent(wMaxen) 14%, and both maxen and wMaxen around 13%. For the same dataset, the methods in [8] have average spatial error ranging from $17\% - 45\%$ depending on the scheme. Specifically, the tomogravity method [1] has a average spatial error of 47%. For other methods, the PamTram approaches also generates better results with less measurement overhead, which we will also discuss later. Since we include 99% of the traffic in the plot, there are numerous small flows still included in the set of flows presented here. These are often disregarded in traffic matrix estimation because they are less important and hard to estimate. This thus shows that our methods can handle some of the small flows as well.

Relative Errors We now look at the instantaneous relative errors, summing neither over time nor over space, but instead just assembling the errors achieved at each interval for each flow. Fig. 7 shows the cumulative distribution function (CDF) of the absolute values of relative errors. With such an error

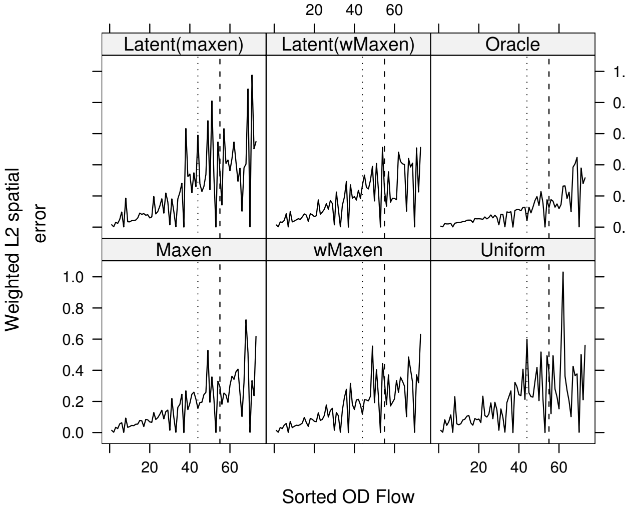


Fig. 6. Spatial error metric plots. The two verticle lines are respectively 90% and 95% quantitle.

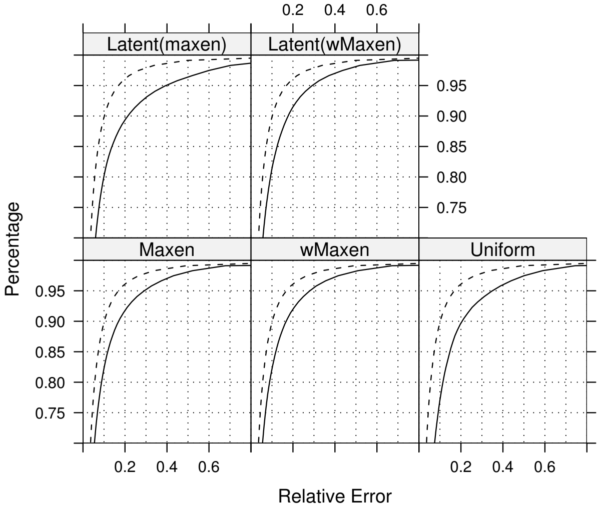


Fig. 7. Cumulative distribution plots of relative errors. The dash line in each panel is that of the oracle scheme.

metric, the very small traffic matrix elements can generate extremely large relative errors. To discount the effect of such small traffic matrix elements from misrepresenting overall performance, each relative error is assigned a weight proportional to the true OD traffic count. This weights the errors by their relative importance. Large flows are considered more important as they are the ones that count for capacity planning and link weight selection algorithms. Thus our resulting CDF plot focuses on the larger OD traffic pairs. Again, we see that our *Maxen*, *wMaxen* and *Latent(wMaxen* schemes all perform well and aren not far off from the *oracle*. Note that the y-axis on these plots begins at 70%. With these methods, more than 80% of OD flows experience an error less than 10%.

TABLE I

AVERAGE RELATIVE ERROR, SCALED MSE AND RUNNING TIME OF DIFFERENT MEASUREMENT SCHEMES.

	Avg(relError)	Avg(sMSE)	Time (sec)
Oracle	4.4%	0.42	N/A
Maxen	7.5%	1.57	0.31
wMaxen	7.5%	1.45	0.28
Latent(maxen)	9.2%	3.35	0.27
Latent(wMaxen)	7.9%	1.79	0.245
Uniform	9.4%	3.50	0.145

Summary. Table I reports the average relative errors, and average scaled MSE for all partial approaches. We only report the average relative errors for large OD traffic flows. The percentage of total traffic we choose is 90%, i.e., the reported average relative errors are based on OD traffic flows which constitute the top 90% of the total traffic load. The average scaled MSE is the simple average of all the scaled MSEs at each time interval. These results show that in general all the PamTram approaches perform well in estimating the true network OD traffic. The performance of *maxen* and *wMaxen* are comparable, while the *wMaxen* is more robust. The *Latent(wMaxen)* method retains similar performance and thus yields a practical way of implementing PamTram.

Overheads. Table I also reports the run-time or computation time for our proposed randomization methods. The computation of the PamTram is very light; it is a very appealing property of the proposed approach, especially important for the implementation of such online algorithms. The Sprint dataset is processed on a 3.2GHZ computer using the R package [20]. It takes *maxen* approximately 0.31 seconds to generate one traffic matrix estimate per 10 minute window. This includes two iterations of the IPF algorithm (one for selecting which flow to measure and one to approximate the minimal MSE estimate). In total, *maxen* takes 572.58 seconds to process the total 1842 time intervals (more than 12 days worth of data) in the Sprint PoP dataset. The *uniform* scheme approximately further cuts the running time by half because only one round of the IPF procedure is needed in each time interval because the flow selection is simple for the uniform scheme. This is fast because the complexity of an IPF algorithm is $O(IJ)$. It avoids matrix inversion as is needed by many maximum likelihood estimation or regularization approaches.

The PamTram is lightweight not only computation-wise, but also in terms of measurement overhead. The strong performance of PamTram's partial measurement approach does not come for free; we are using flow monitors to achieve such performance levels. Thus we should assess the measurement overhead in order to understand the tradeoff between performance and measurement. In [8] the authors introduced a measurement overhead metric. Their metric was defined as $\sum_{i=1}^{I} D(i)/(\text{NumDays} * \text{NumLinks})$, where $D(i)$ was the number of days that link i was turned on for flow measurement. This metric, with units of *link-days*, made sense in their context because each time a flow monitor is turned on it remains on for 24 hours. The idea was to count the amount of time a flow monitor is on over many links and days, and to create a ratio so as to compare it to the case of full measurement when all flow monitors are on all the time. The measurement overheads in [8] ranged from 5-30% depending upon the scheme. Our scheme is equivalent to the case when one flow monitor is on all the time because at any moment we have one flow being monitored. Hence in terms of this metric, the overhead of PamTram is $1/I$ or roughly 5% since we have a network with $I = 18$ links. This is the overhead when a flow monitor is turned on and collects everything on the link to which it is attached. If the flow monitor could be configured to monitor only a single OD flow, then the measurement overhead would drop to $1/169$ (one over the number of OD pairs) which is less than 1% measurement overhead. This is so lightweight that the tradeoff of measurement versus performance gain is immaterial.

V. Discussions and Future Work

In this paper, we propose a partial measurement approach for OD traffic matrix estimation. There are two key ideas in this approach. The first is to use partial flow measurement in a lightweight fashion by only measuring one flow per estimation time interval. We couple this with a dynamic traffic model that allows us to incorporate past information into the current estimate. Such an approach is successful in achieving excellent performance with minimal measurement cost. There is no magic in this approach. One OD flow at one time is only a little extra information, but all the measurements accumulate over time via the dynamic network model. Our second key contribution is the illustration that an IPF algorithm can be used both for approximating an MSE error and for selecting which OD flow to measure. Because the IPF algorithm is fast and we only measure one OD flow per time interval, PamTram is lightweight both in computation time and in measurement overhead. We thus believe that PamTram has potential to be considered for deployment in operational networks.

Our intent was to use the dynamic network model (5a) to capture short-term network dynamics. The model itself is flexible and can be easily extended to accommodate additional information. For instance, a natural extension of the model is to install an auto-regressive term for each OD

$$X_i^{(t+1)} = B_i X_i^{(t)} + \sqrt{B_i X^{(t)}}\epsilon^{(t)}, \tag{12a}$$

where B_i is a polynomial function of the backward operator. The periodicity and more fine structure of the network traffic can be incorporated through the autoregression formula, resulting better OD traffic estimates. Such an extension does bring in an additional layer of complexity. The major challenge is how to determine the order the autoregression terms and how to update them in a dynamic way. We defer all these questions as future research.

Other interesting directions for future research include using more advance traffic monitoring and estimation capabilities to develop dynamic traffic profiles. This can be useful for security applications and to provide enhanced performance for subsets of the total traffic belonging to specific applications (such as VoIP) that may have its own performance and robustness requirements.

Acknowledgments

We would like to thank Sprint Advanced Laboratory for allowing us the access to the PoP network dataset. We would also like to thank Antonio Nucci, Dina Papagianaki, and Anukool Lakhina for preparing the dataset and many helpful discussions. Partial support to Bin Yu and Gang Liang is gratefully acknowledged from the National Science Foundation (CCR-0106656 and FD01-12731) and the Army Rearch Office grant (DAAD19-01-1-0643).

VI. Appendix

A. Proof of Theorem 1

It is easy to show that the conditional expectation

$$E(X|Y) = \mu - \Sigma A'(A\Sigma A')^{-1}(A\mu - Y).$$

Because Σ is a diagonal matrix proportional to η, the coefficient η cancels out and does not appear in the above equation. This conditional expectation is also the solution to the weighted least-square estimate with square root weights in Zhang et al. [1]

$$\min \sum\nolimits_i (X_i - \mu_i)^2/\mu_i \ \text{ subject to } AX = Y.$$

Then similarly, we may borrow the argument pointed out by Zhang et. al.[1] that

$$\begin{aligned} D(X/N||\mu/N) &\approx \sum\nolimits_i X_i/N\,(1 - X_i/\mu_i) \\ &\approx 1/N \sum\nolimits_i (X_i - \mu_i)^2/\mu_i, \end{aligned}$$

where N is the total traffic. The first approximation is a linear expansion of logarithm function, and the second approximation due to our assumption that $\sum_i \mu_i \approx N$.

B. Proof of Theorem 2

Let K denote the index of the OD pair to be measured; hence, we have $P(K = k) = 1/I$. Under the assumption that no any link measurement is obtained, we have

$$\hat{X}_k^{(t)} = \begin{cases} X_k^{(t)} & \text{if } K = k \\ \hat{X}_k^{(t-1)} & \text{otherwise.} \end{cases}$$

Similarly, we define $\tilde{X}^{(t+1)}$ as

$$\tilde{X}_k^{(t)} = \begin{cases} X_k^{(t)} & \text{if } K = k \\ X_k^{(t-1)} & \text{otherwise,} \end{cases}$$

which is the parameter estimate if we start from the true value.

Fix $X^{(t-1)}$ and $\hat{X}^{(t-1)}$ at first, then the expected value of the scaled MSE is

$$\begin{aligned} & E\left(\frac{||\hat{X}^{(t)} - \tilde{X}^{(t)}||^2}{\sum_i \tilde{X}_i^{(t)}}\right) \approx \frac{E||\hat{X}^{(t)} - X^{(t)}||^2}{\sum_i X_i^{(t-1)}} \\ = \quad & \frac{E||\hat{X}^{(t)} - \tilde{X}^{(t)}||^2 + E||\tilde{X}^{(t)} - X^{(t)}||^2}{\sum_i X_i^{(t-1)}} \end{aligned} \tag{13}$$

The first approximation is obtained by the delta method, and the second equality holds because

$$\begin{aligned} & E\left(||(\hat{X}^{(t)} - \tilde{X}^{(t)}) + (\tilde{X}^{(t)} - X^{(t)})||^2\right) \\ = & E\left(E\left(||(\hat{X}^{(t)} - \tilde{X}^{(t)}) + (\tilde{X}^{(t)} - X^{(t)})||^2 \middle| K{=}k, X_k^{(t)}\right)\right). \end{aligned}$$

Note given $K = k$ and $X_k^{(t)}$, $\hat{X}^{(t)} - \tilde{X}^{(t)}$ are determined, and $\tilde{X}^{(t)} - X^{(t)}$ is a mean 0 multivariate normal random variable. The cross terms disappear after expanding the square term.

For each term in (13), we have

$$\begin{aligned} & \frac{E||\hat{X}^{(t)} - \tilde{X}^{(t)}||^2}{\sum_i X_i^{(t-1)}} \\ = & \frac{||\hat{X}^{(t)} - \tilde{X}^{(t)}||^2 - \sum_k P(k)(\hat{X}_k^{(t-1)} - X_k^{(t-1)})^2}{\sum_i X_i^{(t-1)}} \\ \leq & \frac{I-1}{I}\omega^{(t-1)}, \end{aligned}$$

and

$$\frac{E||\tilde{X}^{(t)} - X^{(t)}||^2}{\sum_i X_i^{(t-1)}} \leq \eta.$$

Further, note that the above bound actually does not depend on the value of $X^{(t)}$, implying it holds generally.

Taking an expectation over both side of the inequality, we have

$$E\left(\omega^{(t)}\right) \leq \frac{I-1}{I} E\left(\omega^{(t-1)}\right) + \eta. \qquad (14)$$

If $E(\omega^{(t-1)}) > I\eta$, then the expected error metric will decrease after one step, i.e.,

$$E\left(\omega^{(t)}\right) < E\left(\omega^{(t-1)}\right).$$

Easy to show that any value larger than $I\eta$ will not be the stabilization point of equation (14). It implies that $I\eta$ will be the upper bound of the expected value of the error metric in the long run.

C. Proof of Theorem 3

If only we can show that the uniform selection rule $P(i) = 1/I$ is an equalizer decision rule. First note that

$$E_P\left(L(R^{(t)}, i)\right) = 1/I,$$

independent of the distribution of $R^{(t)}$ as long as the i is chosen independent of $R^{(t)}$. It implies that the such a decision rule is actually an equalizer for the game:

$$E_p\left(\max_{X^{(t-1)}} L(R^{(t)}, i)\right) = 1/I.$$

So the uniform rule is minimax.

References

[1] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "An information-theoretic approach to traffic matrix estimation," in *ACM SIGCOMM*, 2003.

[2] K. Papagiannaki, N. Taft, and A. Lakhina, "A distributed approach to measure ip traffic matrices"," in *ACM Internet Measurement Conference*, October 2004.

[3] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large ip backbone - a study on real data," in *Proc. ACM IMC*, October 2004.

[4] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography: router link data," *Journal of American Statistics Association*, vol. 95, pp. 1063–1075, 2000.

[5] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, vol. 91, pp. 365–377, 1996.

[6] S. Vaton and A. Gravey, "Network tomography : an iterative bayesian analysis," in *Proc. ITC 18*, August 2003.

[7] G. Liang and B. Yu, "Pseudo likelihood estimation in network tomography," in *IEEE Infocom 2003*, San Francisco, April 2003.

[8] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffice matrices: Balancing measurement, modeling and inference," in *ACM Sigmetrics*, June 2005.

[9] A. Soule, A. Nucci, E. Leonardi, R. Cruz, and N. Taft, "How to identify and estimate the largest traffic matrix elements in a dynamic environment," in *ACM Sigmetrics*, June 2004.

[10] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *ACM Sigmetrics*, San Diego, USA, June 2003.

[11] G. Liang and B. Yu, "Maximum pseudo-likelihood estimation in network tomography," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2043–2053, August 2003.

[12] ——, "Maximum entorpy models: convergence rates and application in dynamic system monitoring." in *International Symposium on Information Theory*, 2004.

[13] M. Hanke and P. Hansen, "Regularization methods for large-scale problems," *Surveys on Mathematics for industry*, vol. 3, pp. 253–315, 1993.

[14] G. Wahba, "Spline models for observational data," in *SIAM*, Philadelphia, 1990.

[15] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *ACM Sigmetrics*, June 2004.

[16] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *ACM Sigcomm*, Pittsburg, USA, August 2002.

[17] I. Csiszár, "*I*-divergence geometry of probability distributions and minimization problems." *The Annals of Probability*, vol. 3(1), pp. 146–158, 1975.

[18] J. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The Annals of Mathematical Statistics*, vol. 43, no. 5, pp. 1470–1480, 1972.

[19] S. D. Pietra, V. D. Pietra, and J. Lafferty, "Induce features of random fields," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[20] R. Ihaka and R. Gentleman, "R: A language for data analysis and graphics," *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, pp. 299–314, 1996.