Traffic Matrices: Balancing Measurements, Inference and Modeling

Augustin Soule[1], Anukool Lakhina[2],Nina Taft[3], Konstantina Papagiannaki[4],
Kave Salamatian[1], Antonio Nucci[5]*, Mark Crovella[2], Christophe Diot[4]
[1] Universite Pierre et Marie Curie LIP6 Paris, France. [2] Boston University, USA.
[3] Intel Research Berkeley, USA. [4] Intel Research Cambridge, UK. [5] Sprint Labs, USA.

ABSTRACT

Traffic matrix estimation is well-studied, but in general has been treated simply as a statistical inference problem. In practice, however, network operators seeking traffic matrix information have a range of options available to them. Operators can measure traffic flows directly; they can perform partial flow measurement, and infer missing data using models; or they can perform no flow measurement and infer traffic matrices directly from link counts. The advent of practical flow measurement makes the study of these tradeoffs more important. In particular, an important question is whether judicious modeling, combined with partial flow measurement, can provide traffic matrix estimates that are signficantly better than previous methods at relatively low cost. In this paper we make a number of contributions toward answering this question. First, we provide a taxonomy of the kinds of models that may make use of partial flow measurement, based on the nature of the measurements used and the spatial, temporal, or spatio-temporal correlation exploited. We then evaluate estimation methods which use each kind of model. In the process we propose and evaluate new methods, and extensions to methods previously proposed. We show that, using such methods, small amounts of traffic flow measurements can have significant impacts on the accuracy of traffic matrix estimation, yielding results much better than previous approaches. We also show that different methods differ in their bias and variance properties, suggesting that different methods may be suited to different applications.

Categories and Subject Descriptors:
C.2.3 [Computer Communications]: Network Operations C.4.3 [Performance of Systems]: Modeling Techniques

General Terms:
Measurement Performance

Keywords:
Traffic Characterization. Internet Traffic Matrix Estimation. Principal Components Analysis. Kalman Filtering. Statistical Inference.

1. INTRODUCTION

Network operators need to know how traffic flows through their network in order to make many of the design and management decisions they face. The traffic is typically described by a traffic matrix that captures the amount of traffic transmitted between every pair of ingress and egress points in a network. Each element of a traffic matrix is typically referred to as an Origin-Destination (OD) pair (or flow). Network operators have expressed a pressing need to obtain accurate traffic matrices since a multitude of their activities need such traffic matrices as inputs. These activities include failure management, provisioning, traffic engineering, routing policy design and load balancing.

In [8] the authors outline the reasons why today's monitoring technologies are unable to directly measure traffic matrices. For those reasons, the research community in recent years, has pursued statistical inference methods to estimate traffic matrices from readily available (but inadequate) link traffic measurements.

The traffic matrix estimation problem posed as an inference problem can be briefly stated as follows. The relationship between the traffic matrix, the routing and the link counts can be described by a system of linear equations $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{y}$ is the vector of link counts, $\mathbf{x}$ is the traffic matrix organized as a vector, and $\mathbf{A}$ denotes a routing matrix in which element $\mathbf{A}_{ij}$ is equal to 1 if OD pair j traverses link i or zero otherwise. The elements of the routing matrix can have fractional values if traffic splitting is supported. In networking environments today, $\mathbf{y}$ and $\mathbf{A}$ are readily available; the link counts $\mathbf{y}$ can be obtained through standard SNMP measurements and the routing matrix $\mathbf{A}$ can be obtained by computing shortest paths using IGP link weights together with the network topology information. The problem at hand is to estimate the traffic matrix $\mathbf{x}$. This is not straightforward because there are many more OD pairs (unknown quantities) than there are link measurements (known quantities). This is manifested by the matrix $\mathbf{A}$ having less than full rank. Hence the fundamental problem is that of a highly under-constrained, or ill-posed, system.

By looking back on the development of traffic matrix estimation techniques, we see that most techniques have been motivated by trying to tackle the fundamental issue of the problem being ill-posed. The idea of introducing additional side information is a common approach to solving ill-posed inverse problems. A first generation of techniques was proposed in [11, 2]. Their approach to handling the highly under-constrained problem was to introduce additional constraints related to the second order moment of the OD pairs. These methods used simple models for OD pairs (e.g., Poisson, Gaussian) that contained neither spatial nor temporal correlations *in the OD flow model*. A comparative study of these methods [6] revealed that these methods were highly dependent upon an initial prior estimate of the traffic matrix. This illustrated the need for improved OD flow models.

*A. Nucci has recently moved to Narus, Inc.

SIGMETRICS'05, June 6–10,2005, Banff,Alberta,Canada.
Copyright 2005 ACM 1-59593-022-1/05/0006 ...$5.00.

Shortly thereafter a second generation of techniques [13, 14, 7, 9] emerged that made use of side-information coming from additional sources of SNMP data. This extra measurement data was used to calibrate a model of OD flows. The calibrated model is then incorporated inside some type of estimation procedure. The OD flow model used in [13, 14] is that of a gravity model that captures the fraction of traffic destined to an egress node as a portion of the total arriving traffic at an ingress node. They assume that ingress and egress nodes are independent and calibrate the gravity model using SNMP data from access and peering links. This data is different from the SNMP data used inside the estimation which comes from inter-router links. The method in [9] tackles the issue of an ill-posed system via another approach. Their key idea is to explicitly change the link weights, thereby changing the routing and moving OD flows onto different paths. By doing this enough times, they increase the rank of the routing matrix. The inter-router SNMP data is then collected from each of the different routing configurations used. The side information here can be thought as this additional SNMP data coming from altered routing configurations. This route change method uses all of the collected data to calibrate a model for OD flows that is based on a Fourier expansion plus noise model. This is a dynamic model intended to capture the temporal evolution of OD flows. In fact, this method attacks the ill-posed nature of the problem two ways simultaneously, that of increasing the rank and of using an improved OD flow model. Their model is a purely temporal one for OD flows, while the approach in [14] uses a purely spatial model for the OD flows.

These second generation methods yielded improvements on first generation techniques and also provided useful insights. However these methods have not been able to push error rates sufficiently low for carriers. With recent advances in flow monitoring techniques, some of the issues involved in their use are dissipating. Moreover, it is becoming better understood [8, 12] what sorts of changes are needed to flow monitors in order to make obtaining the flow data needed for traffic matrices less cumbersome to obtain. Even assuming these changes will take place in the future, [8] shows that the communications overhead will remain large. Hence the authors propose the use of flow monitoring in a lightweight fashion - that of turning flow monitors on only when (in time) and where (which nodes) needed.

In this paper we further pursue this idea, generalize it to other methods, and explore a much broader range of questions that arise when one has the possibility of considering using partial flow measurements for traffic matrix estimation. Starting with the premise that we can do partial flow measurement, many important questions arise. Since such measurement data will contain information about spatial and temporal correlations, we can ask what are good OD flow models to exploit the temporal, spatial, or spatio-temporal correlations? How much measurement should be done? What are the improvements in error rate reduction that can be achieved for a given level of measurement overhead? How well do these methods handle the dynamic nature of traffic matrices? What are the strengths and weaknesses of different approaches with respect to other traffic engineering applications that rely upon traffic matrix data as an input? What are the tradeoffs in and limitations of inference methods coupled with partial flow measurement?

We address these questions by studying the behavior of five TM estimation methods. We include two second generation methods; that of [14] called the *tomogravity* method, and that of [9] which we refer to as the *route change* method. We label methods that use partial flow measurements as third generation methods, and consider three such methods. One method, called the *fanout* method hereafter, is the one proposed in [8]. We introduce a method that relies upon Principal Component Analysis (PCA), called the *PCA* method, that appears in this paper for the first time. The final method makes use of Kalman filtering techniques and is hence referred to as the *Kalman* method. This method is also introduced here for the first time.

If flow monitors are turned on network-wide for a period of time, then they yield an exact copy of the traffic matrix for that period of time. When *all* the flow monitors at a single PoP (router) are turned on, this yields an exact copy of one row of a PoP-to-PoP (router-to-router) traffic matrix. Because previous work [9, 8, 5] has found strong diurnal patterns in traffic matrices, the third generation methods we study assume that when flow monitors are turned on, they remain on for 24 hours. These OD flow measurements are then used to calibrate a model. With this kind of rich side-information, powerful models that capture spatial and temporal correlations via measurement (not via assumption) can be designed.

A traffic matrix is a dynamic entity that includes many sources of variability, including short-lived variations, long-lived variations (e.g., resulting from multi-hour equipment failures), and permanent sources of change (new customers/routers/link or removal of customers/routers/links). It is thus natural to expect that any underlying model of OD flows will need to adapt to these changes by being recalibrated. Hence each of the three flow-based methods studied herein include a mechanism to check for change. When changes are detected, flow monitors are turned on again for 24 hours, and the corresponding model is recalibrated.

Our contributions are multiple. First, we classify each method based on the type of correlation (spatial, temporal or spatio-temporal) they draw on to model OD flows. We clarify the amount, type and cost of supporting information they require. This taxonomy helps us better understand and explain the performance of each method. Second, we evaluate all of the methods in a unified fashion - on the same set of data, using the same error metrics. We find that the third generation methods (Fanout, Kalman, and PCA) outperform the second generation methods overall, and can yield accurate estimates for many of the smaller OD flows that are traditionally more difficult to handle. This is somewhat expected because the third generation methods rely on rich priors obtained from direct measurements. However, what is more surprising is that with only 10-20% additional measurements, we can reduce the errors by more than half. Thus, large gains in accuracy can come at a low measurement cost, which underscores the feasibility of the next generation of combined measure-and-infer techniques for computing traffic matrices.

Our third contribution is an examination of how each method responds to dynamic changes in the traffic matrix (TM). We observe that the tomogravity method, because of its memoryless model, accommodates large and sudden changes best. However, because the spatial correlations it uses in modelling are less accurate than those obtained from measurement, the tomogravity method suffers from biased estimates, producing less accurate estimates of the overall mean behavior of OD flows. Conversely, the third generation methods are able to capture the long term mean behavior of OD flows but track large and sudden changes in an OD flow less well. Our fourth contribution provides a closer look at estimation bias by relying on the key notions of estimation bias versus error variance. We find that, in stark contrast to the third generation methods, the tomogravity estimates are biased, but have low error variance. This suggests that tomogravity may be better suited for applications requiring responsive change detection, such as anomaly detection, whereas the hybrid measure-and-infer methods are well suited for capacity planning, that requires faithful estimates of the overall mean estimates.

The comparative part of our study differs from the comparative study in [4] because their study considers first and second generation methods, whereas ours is focused on second and third generation methods. Also they do not examine adaptivity and bias. To the best of our knowledge this is the first traffic matrix paper to examine the performance of estimation methods in terms of adaptivity and bias. These turn out to be important aspects because there is a tradeoff between the two, and because adaptivity and bias can be used to explain some of the results we see in the distribution of temporal and spatial errors. Moreover, our results indicate that partial flow measurements may be necessary to overcome bias.

The rest of this paper is organized as follows. Each of the five methods used herein is summarized in Section 2. Our performance analysis of errors, overheads, adaptivity and bias is presented in Section 3. We conclude our work in Section 4.

2. METHODS

In this section we briefly describe each of the five methods considered in this paper. We highlight three key aspects of each method: the type of OD flow model used, the type of data (or side information) brought in to calibrate the model, and the method of estimation. Focusing on these three aspects of each method is helpful in understanding the differences and similarities between various methods without getting lost in the details. We classify each model as being either spatial, temporal or spatio-temporal. A spatial model is one that captures dependencies among OD flows, but has no memory. In temporal models an OD flow model is dependent on its past behavior, but independent of other OD flows. Spatial models thus capture correlations across OD flows, while temporal models capture correlations in time. Clearly, spatio-temporal models are those that incorporate both types of correlation.

The three third-generation methods presented here use different underlying OD flow models. The common feature of these methods is that they rely on data from flow monitors to calibrate their models. All of these methods assume that flow monitors are initially turned on network-wide for a period of 24 hours for initial model calibration. The flow monitors can then be turned off until further notice. All of these methods include simple schemes for change detection, and when changes are detected, flow monitors are turned back on for another period of 24 hours.

Our validation data had an estimate of the traffic matrix at each 10 minute time interval. Hence all methods estimate the traffic matrix on a time scale of 10 minutes (the underlying time unit t).

2.1 Notation

To facilitate subsequent discussion, we introduce the relevant notational convention used in this paper. As mentioned in the introduction $\mathbf{x}$ represents the traffic matrix at a specific point in time, organized as a column vector with N elements. Likewise, $\mathbf{y}$ is the column vector of traffic on L links at a point in time, and $\mathbf{A}$ is the $L \times N$ routing matrix. We will also frequently refer to the traffic matrix over time and we use $\mathbf{X}$ to denote this structure; $\mathbf{X}$ is a $T \times N$ matrix where each column j corresponds to the timeseries of OD flow j. Similarly, $\mathbf{Y}$ is the T by L multivariate timeseries of link traffic. To represent the traffic matrix (or link traffic) at a particular time t, we use $\mathbf{x}_t$ (or $\mathbf{y}_t$). To identify the j-th OD pair at time t, we use $\mathbf{x}_t(j)$, and when needed, $x(i, j, t)$ represents the traffic at time t for the OD pair sourced at node i and destined to node j. We will reserve the terms $\hat{\mathbf{x}}$ and $\hat{\mathbf{X}}$ for the estimated traffic demands. In general, we will use boldfaced uppercase letters to denote matrices, boldfaced lowercase letters to denote vector quantities and italic lowercase letters to denote scalars. Finally, all vectors are column vectors unless otherwise stated.

2.2 Tomogravity Method

In [13] and [14], the authors developed a method for estimating traffic matrices that starts by building a prior TM using a gravity model approach. The basic principle of the gravity model is to assume proportionality relationships. For ease of exposition, we omit the time index in our notation. Let $x(i, *)$ denote the total traffic entering an ingress node i. If node i is a router, this corresponds to the incoming traffic on all the access and peering links. Let $x(*, j)$ denote the total traffic departing the network from node j. Again, if node j is a router, this includes all the traffic departing the AS in question on either access or peering links. The gravity model postulates that

$$x(i, j) = x(i, *)\frac{x(*, j)}{\sum_j x(*, j)} \tag{1}$$

This implies that the total amount of data node i sends to node j is proportional to the amount of traffic departing the network at j relative to the total amount of traffic departing the entire network. The authors of [13] call this the simple gravity model. This model essentially assumes complete independence between sources and destinations. However in typical IP backbones, there are two policies that lead to deviations from pure independence. Due to hot potato routing policies, traffic from a customer edge traveling towards a peer will be sent to the nearest exit point. The second policy is that there should be no traffic transiting the network from one peer to another.

Hence the authors define the generalized gravity model to capture these policies. This is interpreted as a conditional independence of source and destination, conditioned on additional side information identifying link types. Let $\mathcal{A}$ denote the set of access links, and $\mathcal{P}$ denote the set of peering links. The generalized gravity model is defined as follows.

$$\mathbf{x}'(i, j) = \begin{cases} 0, & \text{for } i \in \mathcal{P}, j \in \mathcal{P} \\ \frac{x(i,*)}{\sum_{i \in \mathcal{A}} x(i,*)} x(*, j) & \text{for } i \in \mathcal{A}, j \in \mathcal{P} \\ x(i, *)\frac{x(*,j)}{\sum_{j \in \mathcal{A}} x(*,j)} & \text{for } i \in \mathcal{P}, j \in \mathcal{A} \\ \rho \frac{x(i,*)}{\sum_{i \in \mathcal{A}} x(i,*)} x(*, j) & \text{for } i \in \mathcal{A}, j \in \mathcal{A} \end{cases} \tag{2}$$

where ρ is a normalization constant (see [13]).

Using an information theoretic perspective, the authors show that a gravity model is a prior capturing complete independence between sources and destinations and therefore is equivalent to a maximum entropy formulation. The gravity model is then used inside a convex optimization problem that combines a miniminum mean square error approach with a maximum entropy approach. This combination is achieved using a regularization strategy that is common approach for dealing with ill-posed problems. They formulate the optimization problem as

$$\min ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda^2 \;\; K(\mathbf{x}||\mathbf{x}') \;\; \text{subject to } \mathbf{x} > 0 \tag{3}$$

where $|| \cdot ||_2$ denotes the L_2 norm and $\lambda > 0$ denotes a regularization parameter. $K(\mathbf{x}||\mathbf{x}')$ is the Kullback-Leibler divergence which is a well known measure of distance between probability distributions. The K-L divergence is used here as a way to write the mutual information, as the goal is to minimize the mutual information. Since the optimization is written as a minimization problem, they minimize the mutual information (rather than maximize the entropy). The idea is thus that among all the traffic matrices that satisfy the link constraints, the method picks one closest to the gravity model. The overall method works as a two step process within each time interval t: first an initial estimate x' is calculated using (2), and then the optimization problem in (3) is solved.

The model used in this method is a spatial one that describes a relationship between OD flows. The gravity model essentially captures a node's fanout for each node. The *fanout* for a source node is simply the fraction of its total traffic that it sends to a given destination. For each source node, the sum of its fanout must equal one. So actually, although the gravity model assumes independence across *nodes*, it does not assume independence among *OD flows*. All the OD flows sharing a common source are interdependent because of this requirement that the fanouts sum to one. Similarly for OD flows sharing a destination node. The gravity model is not a temporal model because at any moment in time the calculation of the gravity model does not depend on history. The data used to calibrate the gravity model comes from SNMP data on access and peering links. The estimation part of this method uses SNMP link counts from inter-router links $\mathbf{Y}$, a routing matrix $\mathbf{A}$, and solves the minimization problem in (3) to produce an estimate for $\mathbf{X}$.

2.3 Route Change Method

We now summarize the method developed in [7, 9] where the latter paper uses the algorithm in the first as part of its overall methodology. In [7], the authors proposed the idea of changing the link weights and then taking new SNMP measurements under this new routing image. This strategy increases the number of constraints on the system because additional link counts, collected under different routing scenarios, yield new equations into the linear system, that will increase the rank of the system if they are linearly independent of the existing equations. In [7] the authors proposed a heuristic algorithm for computing link weight changes needed in order to obtain a full rank system. The advantage here is that with a full rank system, there is a huge potential to reduce errors. However, for such systems to be practical the number of times carriers have to change link weights needs to be kept small. Note that the problem is not yet solved by obtaining a full rank system because the additional measurements will be collected over time scales of multiple hours and thus the traffic matrix itself will change.

In order to capture this multi-hour evolution, the authors assume the diurnal pattern to be cyclo-stationary, while the fluctuation process, (i.e. random noise around the diurnal pattern), to be a zero mean stationary process with covariance matrix $\mathbf{Q}$. According to such assumptions, the authors proposed a Fourier expansion of the diurnal pattern and the OD flow model is formalized as follows.

$$x(i,j,t) = \sum_h \theta_h(i,j) b_h(t) + w(i,j,t) \tag{4}$$

where the first term is the Fourier expansion for the diurnal trends, and the second term captures the stationary fluctuations. Here $b_h(t)$ denotes the h-th basis function while $\theta_h(i,j)$ refers to the coefficients corresponding to OD pair sourced at node i and destined to node j.

By coupling route changes with temporal models for OD flows, the authors devise a new method for estimating the first and second order statistics of OD flows under both stationary and cyclo-stationary conditions. In order to do this, the authors develop an expanded system description of $\mathbf{y} = \mathbf{A}\mathbf{x}$ where the routing matrix, now a function of time, is modified to include many block matrices that incorporate the different routing images. The linear system is also modified to incorporate the models for $\mathbf{X}$; this has an impact on the pseudo-inverse solution used on the expanded full rank system to obtain a traffic matrix estimate. In this context, the estimation of the first order statistics collapses to the estimation of the diurnal pattern, i.e. the θ_h in Equation 4, while the estimation of the second order statistics becomes that of estimating the covariance matrix $\mathbf{Q}$ of the fluctuations process.

The OD flow model being used here is a temporal model in which each OD flow is dependent on its past. This is not a spatial model in that it is assumed that OD flows are independent of one another. This model operates on a different timescale than other methods. Let's say for the sake of example, that it takes 3 days to carry out each of the routing configurations. In this method, all the SNMP data from inter-router links is collected during this 3 day period, as well as each of the routing matrices that apply for each snapshot. The estimation step (using a pseudo-inverse method) is applied once for the three day period and estimates of the traffic matrix for each 10 minute slot of those 3 days are produced. In this method, the calibration of the OD flow model (i.e., determining the coefficients of the OD flow model) are done simultaneously with the estimation of the traffic matrix itself. Hence the side-information used for the OD flow models is the SNMP data from multiple snapshots.

To make this approach applicable for real operators, the authors coupled their ability to estimate the variance of each OD flow, with the observation (from empirical data) that flows with large variance are the flows with large average rate. By doing so, they were able to identify the top largest flows. This in turn enables them to reduce the number of routing configurations required for the overall process.

2.4 Fanout Method

The method in [8] is a purely data-driven method that relies on measurements alone to obtain the traffic matrix. No routing matrix is used, and no inference is performed. A node fanout is defined as the vector capturing the fraction of incoming traffic that each node forwards to each egress node inside the network, where a node can be a PoP, a router or a link. Let $f(i,j,t) = \frac{x(i,j,t)}{\sum_j x(i,j,t)}$ denote the fraction of traffic entering node i that will egress the network at node j at time t. A node's baseline fanout is then given by the vector $\mathbf{f}(i,*,t) = \{f(i,j,t)\ \forall j\}$ at time t. In both [8] and [4] the authors independently found that node fanouts exhibit strong diurnal patterns, and are remarkably predictable at the cycle of one day. This stability of fanouts means that the fanout of node i, at say 2:00pm on one day, can be used to predict row i of the traffic matrix at 2:00pm on subsequent days. This direct measurement technique exploits the finding of predictable fanouts.

Initially, the method assumes that flow monitors are turned on network-wide for a period of 24 hours. Using this data, the baseline fanouts can be computed for each node. Assuming the flow monitor measures the OD flows every 10 minutes, then in a 24 hour period, we have 144 measurement intervals in the baseline. Given the fanout for each 10 minute window within a day, we use the fanout from this first day at a particular time t to predict the traffic matrix on other day at the same time t. For example, the measured fanouts at 3:10pm on the day of netflow collection are used to predict the TM at 3:10pm on other days. The traffic matrix estimate is obtained by

$$\hat{x}(i,j,t) = \hat{f}(i,j,t) x(i,*,t) \tag{5}$$

where $x(i,*,t)$ denotes the total incoming traffic into node i, and $\hat{f}(i,j,t) = f(i,j,d+t\%144)$ where d denotes the number of days ago that the fanout was calibrated. $x(i,*,t)$ can be obtained by summing the SNMP link counts on all incoming access and peering links at the node. Hence the traffic matrix is obtained simply via multiplying the fanouts by SNMP link counts.

The observation on the stability of fanouts is critical, since the idea is for each node to measure its own fanout and to ship it to the Network Operations Center (NOC). The NOC has all the SNMP data readily available, and can thus produce the complete traffic matrix (given the fanouts from all nodes) using the above equation.

If the fanouts are stable for a period of a few days, then many days can go by before a node needs to ship a new fanout to the NOC.

Because the traffic matrix is a dynamic entity, the fanouts will change over time. This method thus includes a heuristic scheme for detecting changes in fanouts. Once a change is detected, then the fanouts need to be recalibrated, i.e., another 24 hours of netflow are needed. Each node randomly selects one 10 minute interval within the next 24 hours, and recomputes its fanout only for that 10 minute interval. The relative change between the newly measured fanout and the fanout vector corresponding to the same 10 minute interval in the baseline captures the diversion from the baseline. The diversion for node i at time t is defined as $\boldsymbol{\Delta}(i,*,t) = ||\hat{\mathbf{f}}(i,*,t) - \mathbf{f}(i,*,t)||$. If $\Delta(i,j,t) > \delta f_{i,j,t}$, for a randomly chosen time interval, then the entire row $\mathbf{f}(i,*)$ is re-measured for the following 24 hours. Otherwise another interval is randomly selected within the next 24 hours, and the process re-iterates.

We point out here that this procedure for checking diversion from the baseline is performed on a per node basis. If one node detects a fanout diversion, then a new collection of flow volume data is initiated *only on that node*. This means that a network-wide recollection of flow data is not needed. If some router fanouts change frequently, they will be updated often; whereas those exhibiting great stability may not get updated for weeks at a time. With this approach, only the portion the TM experiencing a dynamic change needs to undergo a recalibration.

The advantage of the fanout method over a full direct measurement method in which each node measures its own row of the traffic matrix in an ongoing basis, is the reduction in communications overhead. The saving comes by shipping fanouts only once every few days, rather than traffic matrix elements every 10 minutes. Moreover, flow monitors need not be on all the time, but are instead only turned on as needed.

This method is modeling OD flows via their fanout behavior. The model is a spatial model in the same way the gravity model was; namely that OD flow fanouts sharing the same source are correlated (due to the requirement that fanouts at a source node must sum to one). Similarly, correlation exists among OD flows with a common destination. However, it is clear from Equation (5) that this model is also a temporal model since an estimate depends upon the history of a fanout. The fanout method thus uses a spatio-temporal model for OD flow fanouts. The data used to compute the fanouts comes from flow measurements, as will be the case for all of our third generation methods. To summarize, this method uses no inference, but rather direct calculation to produce traffic matrix estimates. These are considered estimates because the fanouts are an approximation, when used outside the measurement period.

2.5 Principal Components Method

The Principal Components Method attacks the traffic matrix estimation problem by studying the intrinsic dimensionality of the set of OD flows. Recall that the central difficulty of the traffic matrix estimation problem stems from the fact that the apparent dimensionality of OD flows, $\mathbf{X}$, is much larger than the available link traffic measurements, $\mathbf{Y}$. In [5] the authors used Principal Component Analysis (PCA) to study the intrinsic dimensionality of the set of all OD flows. PCA is a dimension reduction technique that captures the maximum energy (or variability) in the data onto a minimum set of new axes called principal components. The authors of [5] found that the entire set of N OD flows, when examined over long time scales (days to weeks), can be accurately captured by low dimensional representations. In particular, the ensemble of OD flows are very well described by about 5 to 10 common temporal patterns (chiefly diurnal cycles) called eigenflows.

We can take advantage of this low dimensional representation of OD flows to develop a new approach for traffic matrix estimation. The key idea is that instead of estimating all the N OD flows, we need only estimate the k most important eigenflows. Because $k \ll N$, the problem of estimating the eigenflows from link traffic becomes well posed.

Formally, let $\mathbf{X}$ denote the T by N multivariate timeseries of OD flows, as per our convention. Then, we can use PCA to write $\mathbf{X}$ as:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \qquad (6)$$

where $\mathbf{U}$ is the T by N matrix of eigenflow timeseries and $\mathbf{V}$ is N by N matrix with principal components as its columns. Finally, $\mathbf{S}$ is a N by N diagonal matrix of singular values; $S(i,i)$ is a measure of energy captured by principal component i. The low effective dimensionality of $\mathbf{X}$ means that the top k principal components capture the overwhelming fraction of the energy of $\mathbf{X}$; the remaining principal components account for only a neglegible fraction and can be omitted. We can therefore reduce the dimensionality be selecting only the top k principal components. And, we can also approximate all the traffic demands at a time t as:

$$\mathbf{x}_t \approx \mathbf{V}'\mathbf{S}'\mathbf{u}'_t \quad t = 1, ..., T \qquad (7)$$

where $\mathbf{V}'$ is N by k with only the top k principal components, $\mathbf{S}'$ is the corresponding diagonal matrix of size k by k and vector $\mathbf{u}'_t$ has the values of the k most significant eigenflows at time t.

Traffic matrix estimation using PCA assumes that the transformation matrices ($\mathbf{V}'$ and $\mathbf{S}'$) are already known (*e.g.* from prior measurements of OD flows) and stable [5]. Thus the problem becomes one of estimating the top k eigenflows, $\mathbf{u}'_t$, from the set of link measurements $\mathbf{y}_t$, in:

$$\mathbf{y}_t = \mathbf{A}\mathbf{V}'\mathbf{S}'\mathbf{u}'_t \quad t = 1, ..., T \qquad (8)$$

Because the dimensionality of $\mathbf{u}'_t$ is much smaller, Equation 8 is a well-posed estimation problem (in fact, we set $k = 10$). To solve for $\mathbf{u}'_t$, we use the pseudo-inverse of $\mathbf{A}\mathbf{V}'\mathbf{S}'$ and obtain an estimate $\hat{\mathbf{u}}'_t$. Then, using Equation 7, we compute an estimate of the traffic demands, $\hat{\mathbf{x}}_t$. We set negative entries in $\hat{\mathbf{x}}_t$ to zero, and then rely on the iterative proportional fitting algorithm of [2] to refine our estimate, subject to the constraints imposed by the link traffic.

In order to obtain the decomposition $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ we need the traffic matrix $\mathbf{X}$. This can be viewed as a prior traffic matrix and is obtained by using 24 hours of Netflow data. Given $\mathbf{S}$ and $\mathbf{V}$, we now simply collect the usual SNMP link counts and solve equation 8 to obtain the eigenflows.

Like the fanout method, the PCA method also has a recalibration step. Since the traffic matrix can change over time, the top k principal components that best describe the underlying dimension, could evolve as well. The recalibration step has two components: (i) deciding when to trigger re-measurements of the entire traffic matrix, and (ii) updating the PCA model (the $\mathbf{V}'$ and $\mathbf{S}'$ matrices).

Ideally, we should trigger measurements only when we are certain that the demand estimates obtained from the PCA model are erroneous. A sufficient condition to assess the accuracy of the demand estimates is to compare them against the link traffic counts. We can do this by producing link estimates, $\hat{\mathbf{y}}_t$, via our TM estimates, through $\mathbf{A}\hat{\mathbf{x}}_t\mathbf{y}_t$. When the demand estimates do not match the corresponding link traffic counts, it is likely that the traffic matrix has substantially changed and the PCA model is no longer accurate. One strategy to assess the accuracy of the PCA model is to examine the maximum relative error over all links:

$$\epsilon_t = \max |\frac{\mathbf{y}_t - \mathbf{A}\hat{\mathbf{x}}_t}{\mathbf{y}_t}|$$

where $\mathbf{y}_t$ is the traffic on all links at time t, $\hat{\mathbf{x}}_t$ is our estimate of the traffic demands at that timepoint, and ϵ_t is the resulting maximum relative error over all links (the vector division to compute the relative error is component-wise). To detect if a re-measurement is needed, we can check if ϵ_t exceeds a threshold, that is, if $\epsilon_t > \delta$, where δ is an error tolerance parameter.

Short-lived variations could potentially trigger unnecessary re-measurements in this scheme. To avoid expensive measurements due to transient changes, we monitor ϵ for a period of 24 hours. New measurements are only performed when there is a sustained change in ϵ. Sustained changes are quantified by counting the total number of entries in ϵ that exceed δ, and then checking if the result is more than some fraction κ of the monitoring period (24 hours). In general, if δ and κ are small, recalibrations will be triggered frequently, and when δ and κ are large, recalibrations will be rare. For the results in this paper, we set $\delta = 0.9$ and set $\kappa = 0.5$; we therefore trigger a recalibration if the relative error is more than 90%, and is sustained for more than half a day (because this combination of parameters produced a measurement overhead of 19% which is introduced in Section 3, allowing comparison with the other third generation methods). We emphasize that the recalibration strategy proposed here is one of many potential change detection methods.

Once a recalibration is triggered, new measurements of the entire traffic matrix are performed for a 24 hour period. The new set of OD flows $\mathbf{X}$ are then decomposed using Equation 6 to obtain the $\mathbf{V}'$ and $\mathbf{S}'$ matrices. In this manner, we update the PCA model of the traffic matrix, which is used to estimate all subsequent traffic demands until the next recalibration is triggered.

To summarize, the PCA model of the traffic matrix exploits the dependency among the ensemble of OD flows over long timescales. Therefore, like the tomogravity model, the PCA model also relies on spatial correlation in OD flow traffic. In the PCA model, each OD flow is decomposed into a weighted sum of a handful of common eigenflows, which then become the quantity to estimate. The estimation step treats each timepoint separately to determine the value of the dominant eigenflows. The weights that capture the dependency between the OD flows are specified by the $\mathbf{V}'$ and $\mathbf{S}'$ matrices (as detailed in [5]) and calibrated using direct flow measurements for a period of 24 hours. Because the original traffic matrix estimation problem is transformed into a well-posed problem (Equation 8), the estimation step in the PCA method is simply a pseudo-inverse solution.

2.6 Kalman Filtering Method

We now introduce a method that makes use of a novel approach for traffic matrix estimation. The idea is to use state space models from dynamic linear systems theory to capture the evolution of a system. We view the OD flows as the underlying *states* of a network traffic system. The states evolve in time because the OD flows evolve. However, the states are not directly observable. Instead the SNMP link counts Y can be viewed as an indirect observation of the item we are really seeking (the TM). Estimating the system state (traffic volumes in this case) from indirect observations, when these observations are linear combinations of the underlying state vector, is a common environment for the application of Kalman filtering. Kalman filtering is a powerful method because it can be used not only for estimation but also for prediction.

With $\mathbf{y}_t$ as the observation vector at discrete time t, we let $\mathbf{Y}_t = \{\mathbf{y}_t\}$ denote the set of all observations up to (and including) time t. The vector $\mathbf{x}_t = \{x(i,j,t)\forall i, \forall j\}$ denoting the entire set of OD flows at discrete time t, now also denotes the state of our system at time t. We model the evolution of the traffic state according to the following linear system,

$$\mathbf{x}_{t+1} = \mathbf{C}\mathbf{x}_t + \mathbf{w}_t \tag{9}$$

In this model $\mathbf{C}$ is the state transition matrix and $\mathbf{w}_t$ is the traffic system noise process. For a single OD flow, the diagonal elements of $\mathbf{C}$ capture the temporal correlations appearing in the flow's evolution. The non-diagonal elements of $\mathbf{C}$ describe the dependency of one OD flow on another, thus capturing any spatial correlations among the OD flows (if and when they exist). The noise process $\mathbf{w}_t$ captures the fluctuations naturally occurring in OD flows. This model follows that of typical linear time-invariant dynamical systems. Another way to interpret this model is to say that by $\mathbf{C}$ we capture the deterministic components of the state evolution while the term $\mathbf{w}_t$ captures the random or variable component. An attractive feature of this model is that it captures both temporal and spatial correlations in a single equation.

The traditional linear equation relating the link counts to the traffic matrix is now rewritten here as,

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{m}_t \tag{10}$$

where the term $\mathbf{m}_t$ represents additive measurement noise. It is well known that the SNMP measurement process has its own errors and thus we incorporate them to allow for minor differences between link observations and linear combinations of OD flows. Using Kalman filtering terminology we say that this equation captures the relationship between the observation vector and the state of the system.

Let $\hat{\mathbf{x}}_{t|t-1}$ refer to the *prediction* of $\mathbf{x}_t$ at time t based upon all information up to time $t-1$. We use $\hat{\mathbf{x}}_{t|t}$ to denote the *estimation* of $\mathbf{x}_t$ at time t. The estimation step takes the previous prediction and adds the most recent measurement (observation) to make the next estimate.

The task is to determine $\hat{\mathbf{x}}_{t+1|t+1}$ (written as $\hat{\mathbf{x}}_{t+1}$ for short) given a set of observations $\mathbf{y}_1, ..., \mathbf{y}_{t+1}$. The estimation error at time t is given by $\hat{\mathbf{x}}_{t|t} - \mathbf{x}_t$. We want an estimator that is optimal in the sense that it minimizes the variance of the error. The variance of the error is given by,

$$E[||\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1}||^2] = E[(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1})^T(\mathbf{x}_{t+1} - \hat{\mathbf{x}}_{t+1})] \tag{11}$$

Let $\mathbf{P}_{t|t} = E[(\hat{\mathbf{x}}_{t|t} - \mathbf{x}_t)(\hat{\mathbf{x}}_{t|t} - \mathbf{x}_t)^T]$ denote the covariance matrix of errors at time t. We assume both the state-noise $\mathbf{W}$ and the measurement-noise $\mathbf{M}$ to be zero-mean white-noise processes, uncorrelated and with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. Let the initial conditions of the state of the system be denoted by $\hat{\mathbf{x}}_{0|0} = E[\mathbf{x}_0]$ and $\mathbf{P}_{0|0}$. The Kalman filter estimates a process through a form of feedback control using two types of equations.

Prediction Step: This step predicts the state and variance at time $t+1$ dependent on information at time t.

$$\hat{\mathbf{x}}_{t+1|t} = \mathbf{C}\hat{\mathbf{x}}_{t|t} \tag{12}$$

$$\mathbf{P}_{t+1|t} = \mathbf{C}\mathbf{P}_{t|t}\mathbf{C}^T + \mathbf{Q} \tag{13}$$

Estimation Step: This updates the state and variance using a combination of the predicted state and the observation Y_{k+1}

$$\hat{\mathbf{x}}_{t+1|t+1} = \hat{\mathbf{x}}_{t+1|k} + \mathbf{G}_{k+1}[\mathbf{y}_{t+1} - \mathbf{A}\hat{\mathbf{x}}_{t+1|t}] \tag{14}$$

$$\mathbf{P}_{t+1|t+1} = (\mathbf{Id} - \mathbf{G}_{t+1}\mathbf{A})\mathbf{P}_{t+1|t}(\mathbf{Id} - \mathbf{G}_{t+1}\mathbf{A})^T + \mathbf{G}_{t+1}\mathbf{R}\mathbf{G}_{t+1}^T \tag{15}$$

where $\mathbf{Id}$ is the identity matrix and $\mathbf{G}_{t+1}$ is called Kalman gain matrix. (For lack of space, we do not include the derivation of $\mathbf{G}_{t+1}$.) The above equations (12-15), together with the initial conditions specific above, define the discrete-time sequential, recursive algorithm for determining the linear minimum variance estimate known as the **Kalman Filter**.

We make a few observations on this system. Equation (12) is a one-step ahead predictor whereas equation (14) serves as our estimate we can use to populate a traffic matrix. Note that equation (14) includes a self-correcting step. Our estimate is equal to our prediction (in the previous time slot) plus a gain factor, G_{t+1}, multiplied by the error in our *link estimate*, $\mathbf{A}\hat{\mathbf{x}}_{t+1|t}$. Using the new incoming measurement at $t+1$, we can compare the true link count versus our estimation for the link count. We adjust our estimate for the *traffic matrix* based upon our error in link estimation. The Kalman gain matrix is also a quantity that is updated and adjusted each measurement interval. It is adjusted so as to minimize the conditional mean squared estimation error.

One of the most salient features of the Kalman filter estimation method is this self-correcting feature. It essentially provides an estimation method that continually adjusts the state prediction, each measurement interval, based on the error in the observations. This hints that such a method might be good at tracking changes, thus making it adaptable. We will examine this adaptivity in Section 3.

To apply Kalman filtering to traffic matrix estimation, one more step is needed. To produce our estimates in equations (12) and (14), we need to know the matrices $\mathbf{C}, \mathbf{Q}$ and $\mathbf{R}$ as well as the initial conditions $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$. We propose to use 24 hours of Netflow measurements to compute these variables. This step can be viewed as calibrating the system. Estimating the matrices $\mathbf{C}$, $\mathbf{Q}$ and $\mathbf{R}$ from Netflow data is a procedure that itself requires maximum likelihood estimation. For details on this estimation see [10].

The Kalman method also has a change detection and recalibration procedure to ensure the underlying state space model adapts to changes in the traffic matrix. Like the PCA method, Kalman computes estimates for the link counts $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$ using its traffic matrix estimates. Comparing these to the measured SNMP link counts yields an indirect error metric for the TM estimate. In Kalman filtering terminology, this error term is typically called the *innovation* process and is given by

$$\mathbf{i}_{t+1} = \mathbf{y}_{t+1} - A\hat{\mathbf{x}}_{t+1|t} \tag{16}$$

Recall that in the Kalman method, this error term is already used inside equation (14). The innovations process should be a zero mean process.

Our change detection method is as follows. In each 10 minute time slot we compute $\mathbf{i}_t$ and check if it is above a threshold. For the threshold we use twice the error variance, i.e., $2 \times \mathbf{A}\text{diag}(\mathbf{P}_{t|t})$. We do this throughout the day. At the end of each day (at midnight), we check to see what percentage of these errors were above this threshold. This checking of the innovations process is done on a per-link basis. If more than 10% of these errors exceeded the threshold, then we schedule 24 hours of flow measurements on a network-wide basis. Since our model incorporates spatial correlations, it is not possible to update only one link at a time. At the end of the 24 hour period, we recalibrate our $\mathbf{C}, \mathbf{Q}$ and $\mathbf{R}$ matrices using the measured traffic matrix.

Based on the same motivation as in the design of the PCA recalibration scheme, we wait until the end of the day before deciding that new measurements are needed, to avoid recalibration when transient changes occur. Because we examine a days worth of error metrics, we are essentially smoothing out the transient changes. The downside is that the recalibration could come many hours late, thus increasing our overall errors since we continue using the old models to generate estimates.

In summary, the Kalman method builds a spatio-temporal model for the OD flows. It is clear from Equation 9 that the behavior of OD flows at time $t+1$ is dependent upon the state of the flow at time t, and the spatial dependence is captured by the off-diagonal elements of the $\mathbf{C}$ matrix. Similarly, to the other third generation methods, this approach uses 24 hours of flow measurement data to calibrate its underlying model. The estimation procedure uses a Kalman filter which is the best linear minimum variance estimator in the case of zero mean Gaussian noise.

2.7 Discussion

For each method, Table 1 summarizes the type of OD flow model used, the source of data used to calibrate the model, the type of estimation used by each method, and the data used for estimation. Each method relies on insights that have been learned from measurements over the last few years. The Fourier model came from the realization of the presence of strong diurnal patterns in traffic matrix data. The PCA method is built on the observation that OD flows have low intrinsic dimensionality. The fanout method relies on the stability of fanouts. The generalized gravity model includes networking insights by incorporating routing policies such as hot-potato routing and no-transit traffic forwarding. Knowing that both spatial and temporal correlations exist among OD flows, and that the system is inherently linear system ($\mathbf{y} = \mathbf{A}\mathbf{x}$), the Kalman method builds a linear dynamic system that can capture both types of correlations in a single equation.

3. PERFORMANCE ANALYSIS

3.1 Measurement Data Used

The data we use for validation comes from Sprint's European IP backbone, a network comprised of 13 PoPs and 27 routers. During the summer of 2003 we obtained three weeks of traffic matrix data by enabling Netflow on all incoming peering links and all the links going from access routers to backbone routers. This latter set of links captures nearly all customer traffic. Netflow v8 [3] employs periodic sampling in which one sample is collected every 250th packet. Each router ships the collected Netflow statistics to a centralized collection station every 5 minutes. We aggregated the flow traffic counts into bins of size 10 minutes to avoid possible collection synchronization issues.

To obtain origin-destination flows from the raw flows collected, we have to identify the ingress and egress points of each flow. The ingress points can be identified based upon the ingress links where the flow monitors were turned on. For egress point resolution, we use BGP and ISIS routing tables according to the method detailed in [1]. Using this information we built a PoP-to-PoP traffic matrix, for each 10-minute interval, during our 3 week long experiment.

3.2 Spatial and Temporal Errors

To assess the overall performance level of each of the methods, we consider both temporal and spatial errors. Our main error metric is the relative L2 norm. By *spatial error*, we mean that we obtain an error metric per OD flow that summarizes its errors over its lifetime. For this we use,

$$RelL2_{SP}(n) = \frac{\sqrt{\sum_{t=1}^{T}(\mathbf{x}_t(n) - \hat{\mathbf{x}}_t(n))^2}}{\sqrt{\sum_{t=1}^{T}\mathbf{x}_t(n)^2}} \tag{17}$$

The ensemble of all spatial errors gives us a set of errors over all the OD flows. In Figure 1 we plot these spatial errors for each of our five methods. The x-axis represents the ordered flows, from largest to smallest, sorted according to their mean. In this plot we include all the OD flows constituting the top 95% of the load in the entire traffic matrix.

The temporal error gives an error metric for each time slot summarizing the errors over all OD flows at that instant. For this we

Method	Tomogravity	Route Change	Kalman	PCA	Fanout
Modelling					
* Data Used	SNMP (access, peer)	SNMP (inter-router) under multiple snapshots	Flow data 24 hours	Flow data 24 hours	Flow data 24 hours
* Model	Gravity	Cyclo-Stationary (Fourier+noise)	Linear Dynamic System	Eigenflows and Eigenvectors	Node Fanout
* Type	Spatial	Temporal	Spatio-Temporal	Spatial	Spatio-Temporal
Estimation					
* Data Used	SNMP (inter-router) 1 routing matrix A	SNMP (inter-router) multiple A matrices	SNMP (inter-router) A matrix	SNMP (inter-router) A matrix	SNMP (access,peer) no A matrix
* Method	regularized MMSE optimizaiton	pseudo-inverse	Kalman Filter	pseudo-inverse	none (direct formula)

Table 1: Summary of data types used for modelling and estimation

use,

$$RelL2_T(t) = \frac{\sqrt{\sum_{n=1}^{N}(\mathbf{x}_t(n) - \hat{\mathbf{x}}_t(n))^2}}{\sqrt{\sum_{n=1}^{N}\mathbf{x}_t(n)^2}} \qquad (18)$$

The temporal error is similar to what an ISP might see when trying to estimate the entire traffic matrix at a particular moment in time. This error metric is given in Figure 2 for all five methods. The x-axis here is in units of 10 minute slots. We include the estimation errors for 20 days, excluding the first day since it is used for calibration by the PCA, Kalman and fanout methods. (Recall that when flow monitors are turned on, we assume they are used to populate the traffic matrix and the modeling based inference approaches are ignored in those time slots.)

In the spatial error plots, we observe the well-known phenomenon that errors increase as the size of the flow decreases. However, this effect is far less pronounced in the third generation methods than the second generation methods. Since this holds across all the different third generation techniques, we conclude that we are seeing one of the benefits of using partial flow data coupled with improved OD flow models in that we can better estimate the smaller flows.

In the temporal error plots, notice that the PCA and Kalman methods appear to have exactly zero errors at certain periods. These periods are when recalibration occured and we thus exclude them from our error calculation. The moments of recalibration are not visible with the fanout method, because the fanout method does not recalibrate the entire TM model at the same time, but rather one or more PoPs at a time as needed.

Interestingly, we observe that the tomogravity method exhibits nearly constant temporal errors. We will examine this behavior further in later sections. The tomogravity method achieves roughly 25% temporal errors while all of the generation three methods achieve average temporal errors between 10-15%. The route change method experiences the largest errors, typically in the range of 30-50

An alternate way of viewing the spatial and temporal errors is to examine their CDF's, as given in Figures 3-4. We see that in terms of both spatial and temporal errors, the third generation methods always outperform the second generation methods. In particular, the route change method experiences the largest errors; the tomogravity method experiences the second largest errors; and the three methods using partial flow measurements all exhibit errors that are roughly similar with one another.

It is interesting to note that generally the distributions of temporal errors are fairly narrow. This would indicate that the range of errors made at a point in time, don't differ much at another point in time. The majority of the errors occurring in the tomogravity approach lie between 20-33%, while the majority of the errors for the third generation methods lie between 5-15%. We will be able to understand this behavior when we study the issue of bias in Section 3.6. We also see that the fanout and Kalman methods exhibit a bit of a tail in the last 5-th percentile of the distribution. We will be able to understand this when we examine the adaptivity of the methods, because as we will see, there are some moments in time when these methods do not adapt fast enough. We observe that the distributions of the spatial errors tend to be much less narrow than that of the temporal errors. The large dispersion of the spatial errors indicates that some flows may simply be harder to estimate than others. We know from the literature (and as confirmed by Figure 1), that in general smaller flows are harder to estimate. The flows included here (those constituting the top 95% of the load), span two orders of magnitude, roughly from $10^5 - 10^7$ Bps.

3.3 Over-Modeling

We see in both Figures 1-2 that the route change method performs the least well of all the methods. Before continuing we pause here to explain why this happens and to state why we no do not include this method in the remainder of our analysis. To see how this method behaves, we look at a sample OD flow and examine how the flow and the estimates evolve in time. Our eighth largest OD flow is depicted in Figure 5. In this plot, the original traffic is depicted via the black line and the estimate is given via the grey line. We see that the route change method yields a perfectly cyclical pattern that does not deviate throughout the 3 week period. This is because the route change method assumes that all OD flows follow the model in Equation 4. The implication of this model is very strong; it assumes each OD flow is cyclo-stationary, and any deviation from this model is not captured in the estimates. To see this even more closely, we plot a blow up of a short period of our fourth largest flow in Figure 6. We see here that modeling assumption is enforced too strongly.

This is in part due to the limited set of basis functions used (5). We have used 5 here because this was the number used in [9]. The performance could clearly be improved by using a larger number of basis functions for the Fourier expansion. Because the errors in this method are significantly larger than the other methods, we do not include this in the remainder of our study. We wish to point out that historically, this method was the first to introduce a temporal model for OD flows, thus paving the way for richer models based on properties learned from data. Part of the difficulty incurred by this method, may be due to the fact that it does not include any spatial correlations.

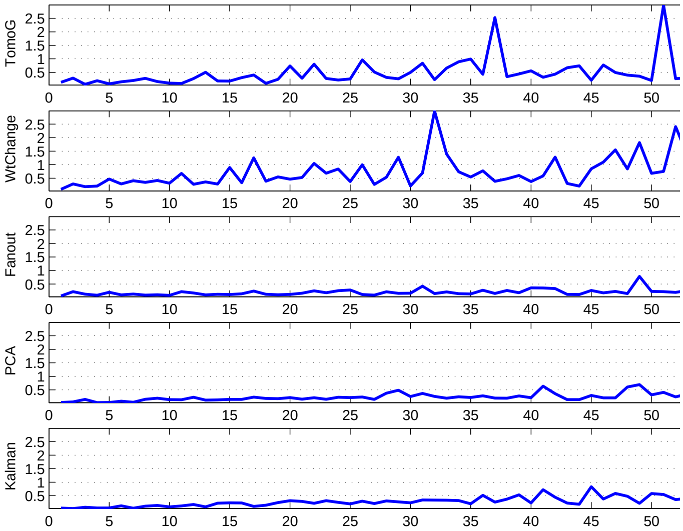

Figure 1: Spatial Relative L2 Norm Error (x-axis is flow id; flows ordered from largest to smallest in mean)

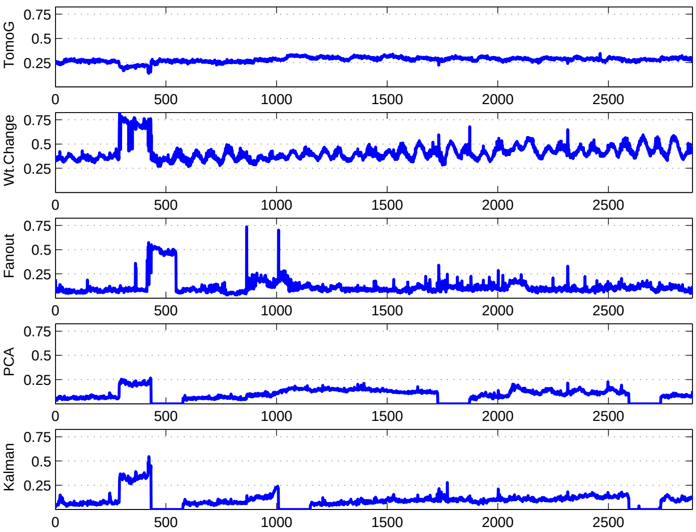

Figure 2: Temporal Relative L2 Error (x-axis in time units of 10 min)

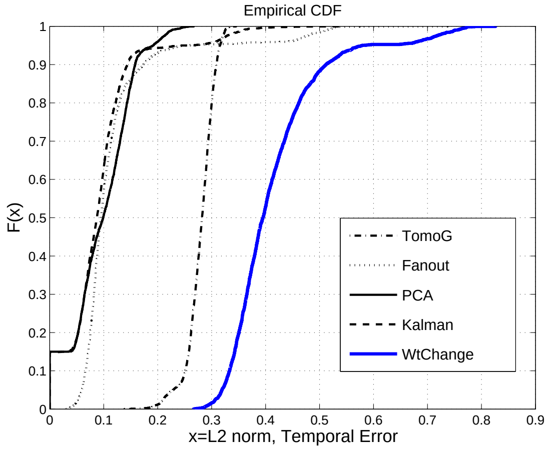

Figure 3: CDF of Temporal Errors

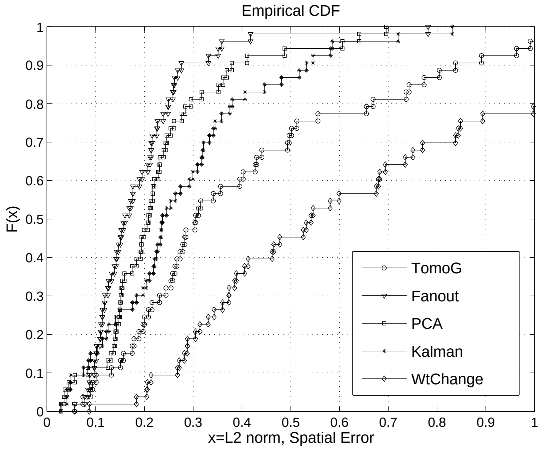

Figure 4: CDF of Spatial Errors

3.4 Measurement Overhead

We saw in Section 3.2 that the third generation methods outperform the second generation methods in terms of both spatial and temporal errors. This is not surprising considering that they have at their disposal a rich set of data to build useful models. It is thus important to now ask: at what cost do we obtain these improved error rates? To answer this, we define an overhead metric to capture the amount of measurement used by these new methods, and then look at the average error achieved for a given level of overhead.

Each time a flow monitor on one link is turned on, it is used for 24 hours. We thus define a metric whose units are *link-days* since it is intended to capture how many links were used (over a multi-week period) and for how long. We measure the overhead via a ratio intended to capture the number of link-days used divided by the total number of link-days that would be employed if flow monitors were left on all the time. Units such as *link-days* can be thought of as describing a spatio-temporal metric. More precisely, we define our overhead metric as follows. Let $D(l)$ be the number of 24-hour cycles that link l used netflow during our 3 week period. Hence $D(l)$ gives the total number of days that netflow was used on link l. The total overhead incurred by a single method is given by,

$$OH = \frac{\sum_{l=1}^{L} D(l)}{21 * L} \qquad (19)$$

where L is the number of links in the networks and 21 is the number of total days of our measured validation data. Both the numerator and denominator are capturing a network-wide property. If the flow monitors were turned on all the time and network-wide, then this overhead metric would be 1; this corresponds to the case when a traffic matrix is fully measured and no inference is used. If flow monitoring is never used, the metric is 0; this corresponds to the case of the tomogravity method. Our intent with this metric is to explore the error performance of methods whose overhead lies somewhere between the two extremes of 0 and 1.

We ran each of the three flow-based methods multiple times. In each case we changed the threshold that checks for diversion, so as to encourage more or less adaptation and recalibration, thus leading to the inclusion of different amounts of flow measurements. For each such case we computed both the spatial and temporal relative L2 norm error. The spatial errors are given in Fig. 7 and the tem-

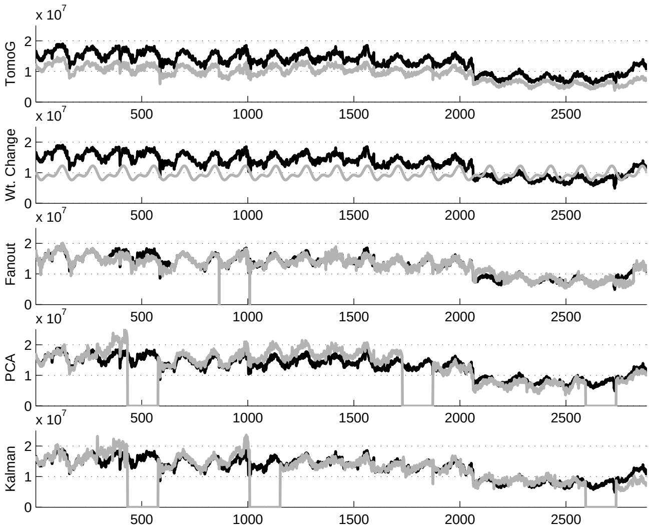

Figure 5: Eighth Largest OD Flow. Original in black; estimates in gray. Time unit is 10 min.

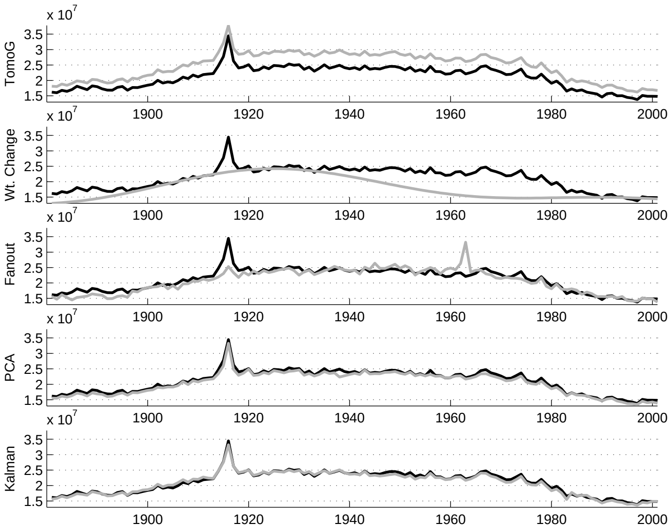

Figure 6: Fourth Largest OD Flow. Original in black; estimates in gray. Time Span from t=1860-2000.

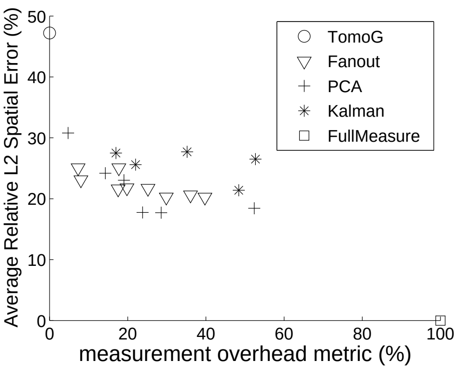

Figure 7: Spatial Errors vs. Overhead

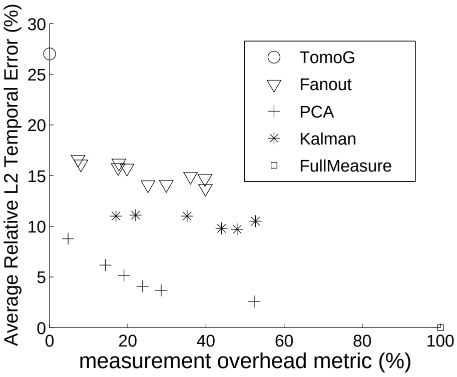

Figure 8: Temporal Errors vs. Overhead

poral errors are displayed in Fig. 8. In these two figures we also include the two extreme points mentioned above, displaying the error of the tomogravity method using zero flow overhead, and the point (100,0) for the case when a traffic matrix is obtained solely through continuous direct measurement. We assume the estimation error is zero (ignoring measurement errors) since flow monitors are on all the time. Although the error rate is zero, the measurement overhead is at its maximum. This overhead does not even include the large communications cost [8] of a full measurement approach. Through these plots we can see the balance that can be achieved using hybrid solutions that combine flow measurements with inference to improve inherent limitations in inference-based solutions.

Three interesting observations are evident from these plots. First, for a measurement overhead of anywhere between 10-20%, we can roughly cut the errors achieved by the tomogravity method in half. This large gain in error reduction for a small measurement overhead is true for both the spatial and temporal error metrics. Second, the third generation methods can produce highly accurate estimates with 90% fewer measurements than a full measurement approach. Third, beyond approximately a 20% overhead, there is little further gain by including additional measurements.

3.5 Handling Dynamic Changes in the Traffic Matrix Elements

The third generation methods have been designed to adapt to changes in the traffic matrix by recalibrating their models. The implicit assumption in this approach is that traffic matrices are very dynamic and thus an adaptive approach is required. This is indeed true; that dynamic nature can take a variety of forms and be due to a variety of reasons. For example, in our measurement data, we found some flows that exhibit rapid oscillations for extended periods of time. Other flows are seen to undergo a long-lived but not permanent change for a period of a day or so. This can happen if a set of ingress links fail and take a day to repair. The addition of new customers in a single PoP will lead to a permanent increase in the traffic demand emanating from that PoP. We ask the question as to how our various methods handle these types of dynamic behavior.

First we look at the case of a flow experiencing rapid oscillations (see Figure 9, that we have enlarged for a period of little less than a day). Here we see that all the methods can handle this type of behavior. The PCA and Kalman methods track the details extremely well. We point out that while the fluctuations are rapid, they are also contained within the same order of magnitude. We looked over the lifetime of this flow and saw that none of its oscillations

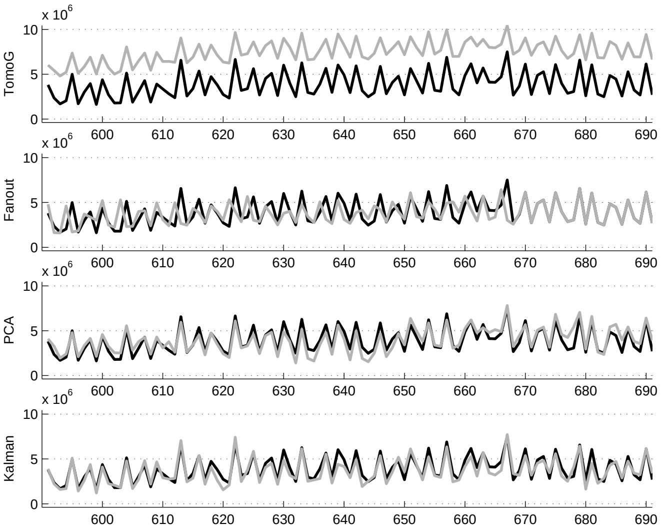

Figure 9: Example of Oscillating Flow. Original in black, estimates in gray. Variations within same order of magnitude.

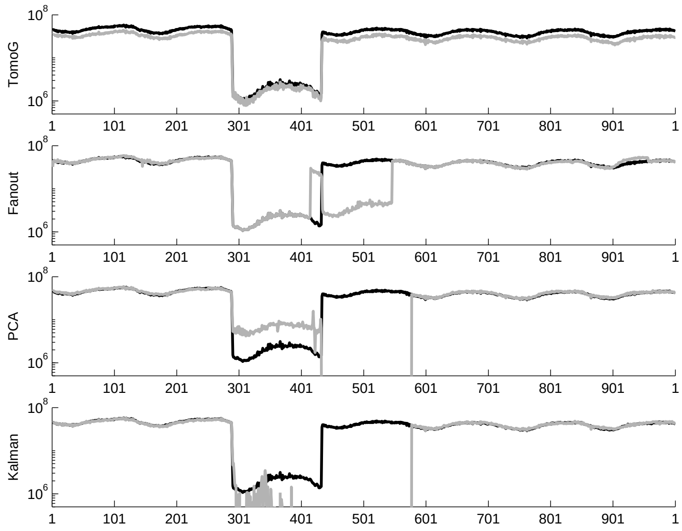

Figure 10: Dynamics in second largest flow; longer-lasting change over two orders of magnitude.

lead to model recalibration for the 3rd generation methods. This implies that there is sufficient flexibility to capture the variability for a flow that did not change an underlying trend.

In Figure 10 we see the example of our second largest OD flow that undergoes a dramatic change on day 3. This flow drops many orders of magnitude to a small, but nonzero level. (This was most likely due to the failure of some, but not all, of the links at the router sourcing this OD flow.) This is plotted on a log scale so that we may see what happens during the severe dip in flow volume.

The tomogravity method tracks this change best. The reason the tomogravity method tracks changes well is because it starts completely anew each time slot. In this case, the gravity model is recalculated every 10 minutes based on the newest SNMP data from access and peering links. In this case, the memoryless property of the tomogravity method serves it well.

We can see the adaptive behavior of the fanout method well here. For some other reason, the fanout method initiates a recalibration just a little before time unit 300 (before the dramatic change in the OD flow). During the day of problems with this flow, the fanout method is using the flow measurement data to populate the TM and hence we see no errors. However the fanout method has learned atypical behavior on day 3 (roughly), and so when it uses this on day 4 to generate estimates, it makes large errors. Since the fanout method only checks randomly once in a 24 hour period, it does not detect that the fanout model is out-dated until somewhere around 560, after which we see a complete recovery.

Looking back to Figure 2 we see that both the PCA and Kalman methods experienced larger errors during day 3 (starting at time slot 432 on that figure). We can now understand the source of this error; it was due to the large change in behavior of a dominant flow. The x-axis in Figure 10 starts at a different time than the one in Figure 2 because we have enlarged a portion of the plot for ease of viewing. We see (Fig. 10) that the PCA and Kalman do adapt to this change, but less well than the tomogravity method. Similarly to the fanout method, these methods react in a timely fashion to the first change (the dip in OD flow) but are slow to react to the recovery (return to normal of OD flow) due to their delayed approach to recalibrations, which is intentionally designed to avoid reacting to transient changes. Because the transient change is somewhat long lived in this case, these methods do not manage to avoid it. At the few timepoints when these methods do not adapt fast enough, larger errors will be generated. This explains why there is a bit of a tail in the CDF of the temporal errors in Figure 3.

We thus see a tradeoff between generation three methods and the tomogravity method. The tomogravity method adapts fast to changes, but experiences larger errors, than the other methods. Of course the rate of adaptivity has been defined by the method; tomogravity updates its model in every 10 minute time slot because the data it uses it typically available at that level of granularity. Because the data used by the generation three methods is more costly to obtain, they wait (defined at 24 hours) before adapting to ensure that a dynamic change is ongoing and that the method does not react to a change that will dissipate quickly.

3.6 Estimation Bias

Finally we seek to understand the issue of bias. We have seen in many figures the bias the tomogravity method exhibits (Figures 5, 6, 9, 10). For these flows the tomogravity method accurately tracks the shape of each flow, but exhibits a consistent over-estimate (e.g., 4th largest flow) or under-estimate (e.g., 8th largest flow). A consistent difference between estimated and true values is bias.

It is well known that biased estimators are not necessarily bad. An unbiased estimator is one whose expectation is equal to the true value of the parameter being estimated. However even if an estimator is unbiased, it may have high variance and thereby often fail to provide estimates that are close to the parameter's true value. On the other hand, sometimes biased estimators can have smaller variance and thereby typically yield estimates closer to the true value than those of unbiased estimators. Thus any consideration of an estimator's bias should also take into account its variance.

The sample bias of an estimator for OD flow n is computed as:

$$bias(n) = \frac{1}{T}\sum_{t=1}^{T}(\hat{\mathbf{x}}_t(n) - \mathbf{x}_t(n)) \tag{20}$$

We look at absolute bias rather than relative bias as this reflects the error a network operator would see employing the methods.

We observe that the various methods perform very differently with respect to bias. Bias is plotted in Figure 11 for four methods. On the x axis are OD flows, sorted from largest mean to smallest.

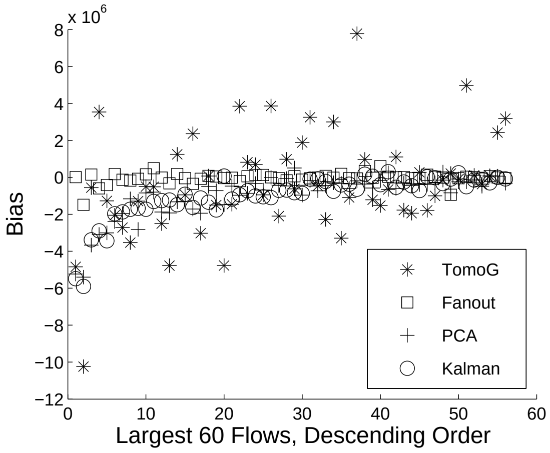

Figure 11: Bias Versus Mean Flow Size

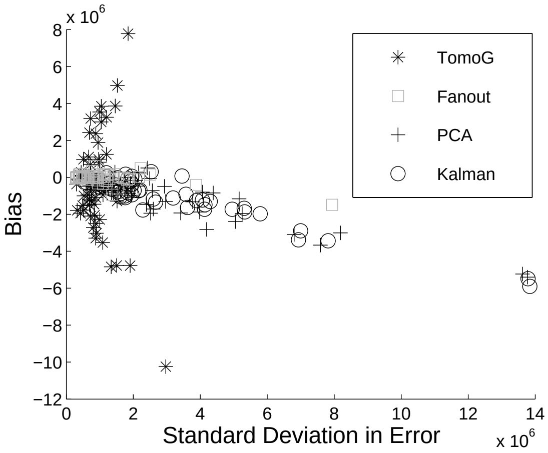

Figure 12: Estimation Bias Versus Standard Deviation

The figure shows that the most consistently unbiased method is the fanout method. Both the PCA and Kalman methods show a negative bias (*i.e.,* an underestimation) for the largest flows, with the amount of bias increasing with flow size. However the tomogravity method is quite different from the rest. It shows a much larger bias in general, with both positive and negative biases prominent. Furthermore, this bias is not consistently related to flow size, as in the PCA and Kalman methods. Rather the per-flow bias can be very large, even for quite small flows.

As mentioned above, the accuracy of an estimator is a function of both its bias and variance. To assess this we define the sample standard deviation of the estimator for flow n as:

$$ErrStd(n) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(err_t(n) - bias(n))^2} \qquad (21)$$

where $err_t(n) = \hat{\mathbf{x}}_t(n) - \mathbf{x}_t(n)$. Note that this metric is in the same units as sample bias, so we can directly compare the two when assessing their impact on accuracy.

Figure 12 plots the sample bias for each flow against its sample standard deviation, for the four methods. This plot confirms the striking difference between the tomogravity method and the other methods. It shows that while the tomogravity method has much lower variance than other methods, it exhibits much higher bias in general. In contrast, the Kalman and PCA methods show relatively high variance, with much lower bias in general, while the fanout method maintains relatively low bias and variance. Thus, the tomogravity method achieves a different tradeoff between bias and error variance than the third generation methods do. The existence of this bias explains why the CDF for the tomogravity method in Figure 3 is ”shifted” to the right of the others. Similarly, tomogravity’s low error variance explains the narrowness of this CDF.

Methods with low bias and high variance tend to estimate the flow mean well over long time intervals, while giving estimates for individual time points with less accuracy. Methods with high bias and low variance will tend to track changes between individual time points well (*i.e.,* accurately estimating short timescale variations) while long timescale averages (the mean) may be consistently wrong. These two different conditions are visible in Figure 6. The tomogravity method tracks the precise shape of the flow quite well, but its value is consistently offset. The Kalman and PCA methods tend to be closer to the true value on average, but their precise shape does not match that of the data as accurately.

This difference in the relative contribution of bias and variance can have implications for applications of these methods. Consider the use of traffic matrix estimation in anomaly detection. In this case, it is the variations from typical conditions that are important. Accurate estimation of the mean is much less important that correct tracking of the shape of the flow. Thus a method like tomogravity seems better suited in this case. On the other hand, consider an application such as traffic engineering (adjusting link weights to balance traffic) or network capacity planning. In this case, accurate estimation of flow mean is paramount. Further, accurate tracking of short timescale variations is less crucial since such decisions take place on relatively long timescales. For such applications, the PCA, Kalman, and/or fanout methods may offer a better choice.

4. CONCLUSIONS

None of the other traffic matrix papers have evaluated TM techniques in terms of 1) adaptivity or 2) bias. In this paper we do both. We find that there is an interesting tradeoff between these two factors. In order to reduce bias, direct measurements of the TM itself are needed every so often. However obtaining this data is expensive, so it cannot be done on small time scales (e.g., 10 minutes). The rate at which side-information is obtained determines the adaptivity of the TM estimation method. Tomogravity obtains its side-information (link data) for model calibration every 10 minutes and is therefore the most adaptive method. Third generation methods are less adaptive because they do not obtain their side-information (flow data) for model calibration more than once or twice a week typically. However, because flow data is so rich (thus enabling rich models) third generation methods can almost entirely eliminate estimation bias, unlike the tomogravity method. The third generation methods are able to overcome bias, and better capture the long term dynamic behavior of an OD flow, since they incorporate the *actual* correlations (either spatial or spatio-temporal) they have from measurements, as opposed to an *assumed* correlation structure as in the gravity model.

Although we say third generation techniques are ”less” adaptive than tomogravity, we note that these methods do adapt well enough. This is visible through their errors which are, in both space and time, generally far lower than tomogravity’s. Because of its bias, tomogravity exhibits errors consistently throughout the lifetime of an OD flow, whereas the third generation methods only exhibit larger errors during periods of adaptation. Furthermore, the next generation of hybrid measure-and-infer strategies can handle smaller flows much better, a problem that has so far plagued the known methods that rely on SNMP data alone.

An important and surprising finding of our work comes in comparing the results to the two extremes of using no flow measurements and that of a full brute force measurement approach to obtaining TMs. We find that with only 10-20% measurement overhead, we can achieve half the temporal and spatial errors reported by earlier methods that use no flow measurements. Relative to the brute force approach, we can achieve accurate estimates using 90% less measurements, thus implying there is little incentive to implement full direct measurement. The measurement, communications and computation overheads of the full measurement approach will be far greater than of the hybrid measure-and-infer strategies we have assessed here. The idea of using partial flow measurements to improve TM estimation could most likely be applied to any of the previous methods. Extensions to first and second generation methods could be envisioned not just via better model calibration, but also through the use of improved priors or by generating additional constraints.

We believe that pure inference methods have limitations as to how low they can push their errors; this is natural as they are ultimately constrained by limited input. This paper illustrates that some type of additional measurements, particularly of the OD flows themselves and not just extra link measurements, are needed to obtain more accuracy. As flow monitors become more accessible, network operators will have greater options to compute traffic matrices. We envision that techniques such as the Fanout, Kalman and PCA methods, will evolve into feasible and accurate alternatives that yield better performance than methods relying solely on inference, and at less cost than brute force measurement.

ACKNOWLEDGEMENTS

We would like to thank Sprint for generously providing us with their traffic matrix data. We are very grateful to Bjorn Carlsson, Jeff Loughridge, and Richard Gass of Sprint for their efforts in collecting the validation data used herein. We are also grateful to Prof. Eric Kolaczyk of Boston University for helpful discussions that led to the development of the PCA-based TM estimation method.

5. REFERENCES

[1] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft. Geographical and Temporal Characteristics of Inter-POP Flows: View from a Single POP. In *European Transactions on Telecommunications*, February 2002.

[2] J. Cao, D. Davis, S. VanderWeil, and B. Yu. Time-Varying Network Tomography: Router Link Data. *Journal of the the American Statistical Association*, 95(452), 2000.

[3] Cisco. NetFlow Services Solutions Guide, July 2001.

[4] A. Gunnar, M. Johansson, and T. Telkamp. Traffic Matrix Estimation on a Large IP Backbone - A Comparison on Real Data. In *ACM Internet Measurement Conference*, Taormina, Italy, October 2004.

[5] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft. Structural Analysis of Network Traffic Flows. In *ACM Sigmetrics*, New York, June 2004.

[6] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic Matrix Estimation: Existing Techniques and New Directions. In *ACM SIGCOMM*, Pittsburgh, USA, Aug. 2002.

[7] A. Nucci, R. Cruz, N. Taft, and C. Diot. Design of IGP Link Weight Changes for Estimation of Traffic Matrices. In *IEEE Infocom*, Hong Kong, March 2004.

[8] K. Papagiannaki, N. Taft, and A. Lakhina. A Distributed Approach to Measure Traffic Matrices. In *ACM Internet Measurement Conference*, Taormina, Italy, October 2004.

[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*, New York, June 2004.

[10] A. Soule, K. Salamatian, A. Nucci, and N. Taft. Traffic Matrix Tracking using Kalman Filtering. LIP6 Research Report RP-LIP6-2004-07-10, LIP6, 2004.

[11] Y. Vardi. Estimating Source-Destination Traffic Intensities from Link Data. *Journal of the the American Statistical Association*, March 1996.

[12] G. Varghese and C. Estan. The Measurement Manifesto. In *HotNets-II*, Nov. 2003.

[13] 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, CA, 2003.

[14] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An Information Theoretic Approach to Traffic Matrix Estimation. In *ACM SIGCOMM*, Karlsruhe, Germany, August 2003.