How to Identify and Estimate the Largest Traffic Matrix Elements in a Dynamic Environment

Augustin Soule[1], Antonio Nucci[2], Rene Cruz[3], Emilio Leonardi[4], Nina Taft[2],*
[1] – LIP VI, Paris, France
[2] – Sprint Advanced Technology Laboratories, Burlingame, CA, USA
[3] – University of California San Diego, San Diego, CA, USA
[4] – Politecnico di Torino, Turin, Italy

ABSTRACT

In this paper we investigate a new idea for traffic matrix estimation that makes the basic problem less under-constrained, by deliberately changing the routing to obtain additional measurements. Because all these measurements are collected over disparate time intervals, we need to establish models for each Origin-Destination (OD) pair to capture the complex behaviours of internet traffic. We model each OD pair with two components: the diurnal pattern and the fluctuation process. We provide models that incorporate the two components above, to estimate both the first and second order moments of traffic matrices. We do this for both stationary and cyclo-stationary traffic scenarios. We formalize the problem of estimating the second order moment in a way that is completely independent from the first order moment. Moreover, we can estimate the second order moment without needing any routing changes (i.e., without explicit changes to IGP link weights). We prove for the first time, that such a result holds for any realistic topology under the assumption of *minimum cost routing* and *strictly positive link weights*. We highlight how the second order moment helps the identification of the top largest OD flows carrying the most significant fraction of network traffic. We then propose a refined methodology consisting of using our variance estimator (without routing changes) to identify the top largest flows, and estimate only these flows. The benefit of this method is that it dramatically reduces the number of routing changes needed. We validate the effectiveness of our methodology and the intuitions behind it by using real aggregated sampled netflow data collected from a commercial Tier-1 backbone.

Categories and Subject Descriptors

C.4 [Performance of Systems]: *Modeling techniques and Design studies.*

General Terms

Performance, Theory.

Keywords

Network Tomography, Traffic Matrix Estimation.

*This work was done when Nina Taft was at Sprint ATL; she is currently affiliated with Intel Research Berkeley, CA.

SIGMETRICS/Performance'04, June 12–16, 2004, New York, NY, USA.

1. INTRODUCTION

A traffic matrix is a representation of the volume of traffic that flows between origin-destination (OD) node pairs in a communication network. In the context of the Internet, the nodes can represent Points-of-Presence (PoPs), routers or links. In current IP backbone networks, obtaining accurate estimates of traffic matrices is problematic. There are a number of important traffic engineering tasks that could be greatly improved with the knowledge provided by traffic matrices. As a result, network operators have identified a need for the development of practical methods to obtain accurate estimates of traffic matrices. Example applications of traffic matrix estimation include logical topology design, capacity planning and forecasting, routing protocol configuration, provisioning for Service Level Agreement (SLAs), load balancing, and fault diagnosis.

The most straightforward approach is to directly measure traffic volumes between network endpoints. However, such approaches (e.g., Cisco's Netflow) still face challenging engineering obstacles related to the wide-spread deployment of a uniform measuring infrastructure, and the collection, storage, synchronization and processing of large amounts of information. Because such approaches will not be available to carriers in the near future, researchers have turned to statistical inference techniques.

The relationship between the traffic matrix, the routing and the link counts can be described by a system of linear equations $Y = AX$, where Y is the vector of link counts, X is the traffic matrix organized as a vector, and A denotes a routing matrix in which element 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. (We define our notation more thoroughly later on.) In networking environments today, Y and A are readily available; the link counts Y can be obtained through standard SNMP measurements and the routing matrix A can be obtained by examining IGP link weights together with the corresponding topological connectivity information. The problem at hand is to estimate the traffic matrix X. This is not straightforward because the matrix A does not have full rank, and hence the fundamental problem is that of a highly under-constrained, or ill-posed, system.

A first generation of techniques were proposed in [1, 2, 3]. Model parameters are estimated using either Moment Generating methods [1], Bayesian methods [2] or Maximum Likelihood estimation [3]. A common idea behind these approaches to tackle the highly under-constrained problem was to introduce additional constraints related to the second order moment of the OD pairs. Estimation is then carried out with two batches of constraints, one on the first order moment and one batch for the second order moment. However the combined set of constraints is not solvable without an assumption on the relationship between the mean and variance. For example, in [1, 2] the authors assume that the volume of traffic between a given

OD pair has a Poisson distribution. Since the mean and variance of a Poisson variable are identical, an estimate of the variance of the volume of traffic for some OD pair can be used to estimate the mean volume of traffic for the same OD pair. Cao et al. [3] assume instead that the traffic volume for OD pairs follows a Gaussian distribution, and that a power law relationship between the mean and variance exists. A comparative study of these methods [4] revealed that these methods were highly dependent upon the initial starting point, often called a *prior*, of the estimation procedure. Hence a second generation of techniques emerged [4, 5, 6] that proposed various methods for generating intelligent priors. Other second generation techniques focused on improving the speed of computation [7].

In practice, all of these statistical inference techniques for estimating traffic matrices suffer from limited accuracy. The error rates are distributed over a range, however, in short we can say that they typically fall between 10 and 25%. These methods can yield outliers such that some OD pairs can have errors above 100%. It has been difficult to drive the error rates down below these values. Some carriers have indicated that they would not use traffic matrices for traffic engineering unless the inference methods could drive the *average* errors below the 10% barrier.

The hardship in further reducing error rates is mainly due to fact that real traffic exhibits complex behaviors that substantially deviates from any simple traffic model. As a consequence, any apriori guess on OD flow behavior based on simple models may potentially have a significant impact on the degree of (in)accuracy in estimates. To develop realistic models, we start by studying one months' worth of sampled Netflow data from a commercial Tier-1 backbone traffic matrix. By examining properties of OD flows from direct measurement data, we observed that OD flows contain two critical components: diurnal patterns and a fluctuations process. We also test the "power-law" relationship between mean and variance of OD flows and we show how this assumption does not "fully" hold for real traffic. Thus our first goal is to establish models that do not require assumptions about the mean-variance relationship and that can incorporate the two components above.

In addition to using realistic models, we believe that further improvements in making the problem less under-constrained are needed to reduce error rates. One way to do this is to define a scheme that allows for the rank of the system to be increased. In [8], the authors, for the first time, proposed the idea of changing the link weights and then taking new SNMP measurements under this new routing image. The reason why this can potentially reduce the "underconstrainedness" of the system is as follows. If additional link counts are collected under different routing scenarios, then this yields additional equations into the linear system that would increase the rank of the system if they are linearly independent of the existing equations. In [8] the authors proposed a heuristic algorithm for computing link weight changes needed in order to obtain a full rank system. They incorporate some practical requirements and constraints that a carrier has to face in their solution. 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. To the best of our knowledge, this is the first paper to consider the cyclo-stationarity of traffic matrices.

In this work we use this "routing changes idea" and propose two new methods for traffic matrix estimation. The common idea of these two methods is to make use of the well-posed property of the new full rank system (after the sequence of weight changes has been applied). With this system, we can model the first and second order moments of OD flows separately. By coupling this approach with OD flow models that capture both the diurnal patterns and the fluctuation behaviors of real traffic, we will show that we can avoid any of the typical modeling assumptions (such as Poisson, Gaussian, or fixed mean-variance laws).

Our methods thus make use of multiple mechanisms each of which constitute one of many steps in an overall TM estimation procedure. The idea is to prepend basic inference techniques with some extra steps. In particular we add (1) a variance estimator, and (2) an algorithm for suggesting link weight changes that will increase the rank of the system. We derive a closed form solution to estimate the covariance of the TM based on our underlying OD models. To the best of our knowledge, this is the first time that an estimate for the covariance of a TM has been proposed. For our weight change algorithm, we rely on an algorithm such as that proposed in [8]. The last step in our method uses a basic inference step (e.g, a pseudo-inverse or Gauss Markov estimator).

In addition to providing an estimate for the covariance, we also prove that the variance will always be identifiable, i.e., that a unique solution exists. In particular, we show that for any general topology with minimum cost routing, and strictly positive link costs, it is always possible to estimate the covariance function without routing changes. Being able to estimate the covariance has significant consequences: (1) it can be used to improve the estimate of the mean; (2) we avoid having to make assumptions about the relationship between the mean and variance; and (3) we can use our variance estimate to define a method for identifying the top largest OD flows. Because 30% of the OD flows constitute 95% of the total traffic matrix load in the network we studied, it can be argued that estimating the largest flows alone would be sufficient for TM estimation. Others have also argued [5, 9] that carriers only care about estimating large flows because they carry almost all the traffic and because the largest errors occur only for small flows.

We are able to identify which flows are largest (i.e. uncover their identity) without estimating their mean rate. If we then modify the TM estimation problem to focus only on the large flows, we can then reduce the number of routing changes needed by methods that try to increase the rank. This is because we have reduced the number of variables to estimate. An important consequence of being able to identify the top flows is that it makes methods based on weight-change algorithms more practical since it helps limit the number of needed routing changes. In this paper we study the trade-off between accuracy of the estimates versus the number of routing changes needed to drive the linear system to full rank.

We will show that with our methods we can drive the average error rates down into a whole new range. Our method succeeds in driving the *average* error rates below the 10% target; and we are often able to reach 4 or 5% average error rates (depending upon the scenario). To the best of our knowledge, this is the first paper that consistently achieves errors below this 10% barrier.

The rest of the paper is organized as follows. In Section 2 we give the formal problem statement for traffic matrix estimation. The essence of the idea of deliberately changing the routes to decrease the "underconstrainedness" of the problem is illustrated in Section 3. In Section 4 we discuss the dynamic nature of Internet traffic and we thus motivate the statistical traffic models we use. Section 5 presents the two methodologies we propose for estimating the traffic matrix, whereas the modeling aspects needed to estimate both the first and second order statistics of a traffic matrix in the context of both stationary and cyclo-stationary environments are presented in Section 6. In Section 7 we prove that the second order moment can be always estimated for any realistic topology under the assumptions of *minimum cost routing* and *stricly positive* link weights. In Section 8 we validate our methodologies using real and pseudo-real data while Section 9 concludes the paper.

2. PROBLEM STATEMENT

The network traffic demand estimation problem can be formulated as follows. Consider a network represented by a collection $\mathcal{V} = \{1, 2, \ldots, V\}$ of nodes. Each node represents a set of co-located routers (PoP). A set of L directed links $\mathcal{L} \subset \mathcal{V} \times \mathcal{V}$ is also given. Each link represents an aggregate of transmission resources between two PoPs. (We assume there are no self directed links, e.g. there are no links l of the form $l = (i, i)$.) We consider a finite time horizon consisting of K disjoint measurement intervals, indexed from 0 to $K-1$. We refer to each measurement interval as a **snapshot**. In each interval or snapshot we change the link weights, i.e. the paths followed for some OD pairs, and this gives a different image of OD pairs traversing the network. In all the following we assume the routing stays the same within the same snapshot, while two different snapshot are characterized by two different routing scenarios. Within each snapshot, we collect multiple consecutive readings of the link counts using the SNMP protocol at discrete time n, called **sample**, indexed from 0 to $N_s - 1$. Then, each snapshot lasts for $N_s \times 5$ minutes because SNMP reports link counts every 5 minutes. For simplicity we assume that the measurement intervals are of equal time durations, but this can be easily relaxed.

Each node is a source of traffic which is routed through other nodes, ultimately departing the network. Consider an origin destination (OD) pair $p = (v_1, v_2)$, and let $X_p(k, n)$ be the amount of traffic associated with the OD pair p during measurement interval k at discrete time n. In other words, $X_p(k, n)$ is the amount of traffic originating from node v_1 that departs the network at node v_2 during measurement interval k at time n. We assume that the measurement intervals are long enough so we can ignore any traffic stored in the network. Let $\mathcal{P}$ denote the set of all OD pairs; there are a total of $|\mathcal{P}| = P = V^2 - V$ OD pairs. We order the OD pairs p and form a column vector $X(k, n)$ whose components are $X_p(k, n)$ in some pre-defined order.

Let $Y_l(k, n)$ be the total volume of traffic which crosses link $l \in \mathcal{L}$ during measurement interval k at time n. Order the links $l \in \mathcal{L}$ in order to form the column vector $Y(k, n)$ whose components are $Y_l(k, n)$ for all $l \in \mathcal{L}$. Let $A_{l,p}(k)$ be the fraction of the traffic $X_p(k, n)$ from OD pair p that traverses link l during measurement interval k at time n. Thus, $Y_l(k, n) = \sum_{p \in \mathcal{P}} A_{l,p}(k) X_p(k, n)$. Forming the $L \times P$ matrix $A(k)$ with elements $\{A_{l,p}(k)\}$, we have in matrix notation

$$Y(k, n) = A(k)X(k, n) \;\; , \forall k \in [0, K-1] \text{ and } n \in [0, N_s - 1]. \tag{1}$$

In the literature, the $Y(k, n)$ vector is called the *link count vector*, while $A(k)$ is called the *routing matrix*. In IP networks, the routing matrix $A(k)$ during each measurement interval k can be obtained by gathering topological information, as well as OSPF or ISIS link weights. Link counts $Y(k, n)$ are obtained from SNMP data.

A general problem considered in the literature is to compute an estimate of the traffic matrix $X(k, n)$ for each k and n given the observed link count vectors $Y(k, n)$ for each k and n, assuming that the routing matrix $A(k)$ does not change in time, i.e. $A(k_1) = A(k_2) \forall k \in [0, K-1]$, and that $X(k, n)$ is a sample of a stationary (vector valued) random process. Furthermore it is generally assumed in prior work that the components of $X(k, n)$ are uncorrelated. The general problem is non-trivial since the rank of $A(k)$ is at most L and $L << P$, i.e. for each k the system of equations (1) is under-determined.

Our goal in this work is to estimate both the first and second order moments of the traffic matrix. To the best of our knowledge, this is the first attempt to try to estimate the second order moment. We will show how having estimates for the TM covariance enables new methodologies for estimating the first order moment used to populate a traffic matrix. In the next section we discuss traffic dynamics and further extend this goal to include such estimation when the input data is gathered over long time scales, i.e., outside the stationary regime.

3. ROUTE CHANGES CAN HELP

We now explain, via an example, the basic idea of exploiting routing changes to increase the rank of the system. We do so because this is a key component of our overall methodology, and in order to make this paper self-sufficient. For a complete explanation of the method and an algorithm for selecting such weight changes, see [8].

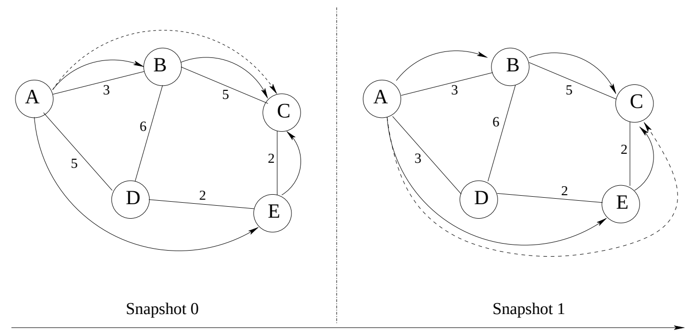

Figure 1: Example: Impact of Routing Changes

Consider the network shown in Fig. 1 composed of five nodes interconnected by six unidirectional links. Each link has an associated weight and the traffic from each OD pair is routed along the shortest cost path. For simplicity, we consider only five OD pairs (indicated by arrows). On the left of Fig. 1 we represent the network in its normal state, when no link weight changes have been effected. Snapshot 0 would generate the following system of linear equations.

$$\begin{bmatrix} Y_{AB} \\ Y_{AD} \\ Y_{BD} \\ Y_{BC} \\ Y_{EC} \\ Y_{DE} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{AB} \\ X_{AC} \\ X_{AE} \\ X_{BC} \\ X_{EC} \end{bmatrix}$$

The rank of routing matrix $A(0)$ is four. Two of the five OD pairs ((A, E) and (E, C)) can be estimated exactly because in this simple example they do not share their links with other OD pairs. On the right of Fig. 1 we show the effect of decreasing the weight of link $l = (A, D)$ from 5 to 3 (snapshot $k = 1$). This perturbation in the weights causes the re-routing of the OD pair (A, C) through the new path $\{(A, D), (D, E), (E, C)\}$. This snapshot generates a new system of linear equations, i.e. a new routing matrix $A(1)$, that can be appended to the previous set. One line of the new routing matrix would look like $Y_{AB} = [1 \;\; 0 \;\; 0 \;\; 0 \;\; 0] \;\; [X_{AB} \;\; ...]^T$. We can see that adding this to the original system of equations adds a new linearly independent equation into the system. As a consequence, the new system $[A(0)^T \;\; A(1)^T]^T$ is full rank and all five OD pairs can be estimated exactly.

The challenge in taking advantage of this idea, is to determine a minimal set of weight changes so that the impact of such changes on network traffic is as little as possible. Moreover, network administrators may simply not allow certain changes if their impact is too great (e.g., if a weight change increases the delay beyond allowable levels). In [8] the authors developed an algorithm for computing a minimal sequence of weight changes that takes into consideration a variety of practical administrative constraints.

In this paper, we assume we have a predetermined schedule that identifies a sequence of link weight changes to perform in a given order. This schedule would be the output of an algorithm such as the one given in [8]. The number of links whose weights get changed simultaneously in a single measurement interval k is small, i.e., typically 1, 2 or at most 3 links. When executing this method, two routing matrices $A(k_1)$ and $A(k_2)$ for $k_1 \neq k_2$ are usually sufficiently different so that the rank of the new routing matrix $A = [A^T(k_1), A^T(k_2)]^T$ is larger than the rank of either matrix alone. The schedule of link weight changes should be chosen such that after K measurement intervals, the rank of A is equal to the number of unknown coefficients needed to estimate the TM.

4. TRAFFIC DYNAMICS

In this section we discuss traffic dynamics and what they mean for the traffic matrix estimation problem. We begin by examining netflow data collected from a commercial Tier-1 backbone. Netflow was run for all the incoming links from gateway routers to backbone routers. The version of Netflow used is called *Aggregated Sampled Netflow* and deterministically samples 1 out of every 250 packets. Netflow samples fine granularity flows defined by the so called 5-tuple in IP packet headers. Using local BGP tables and topology information we were able to determine the exit link for each incoming flow. The resulting link-by-link traffic matrix is aggregated to form both a router-to-router and a POP-to-POP traffic matrix. We had roughly one month's worth of data at our disposal.

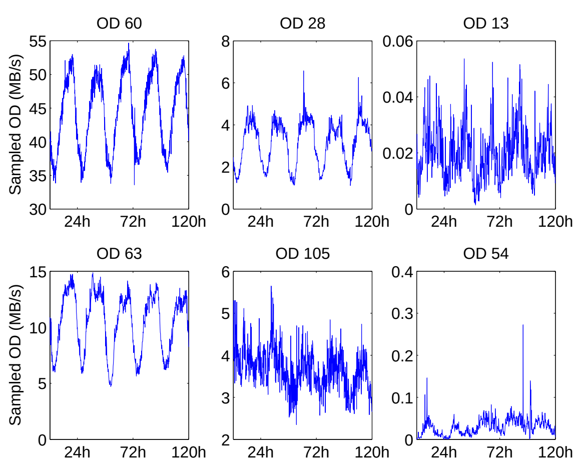

Figure 2: Netflow data for representative OD flows: *large* **on the left,** *medium* **on the middle and** *small* **on the right.**

Figure 2 shows the evolution of six OD pairs in time across a week (excluding weekends). Here we show three types of behavior and provide two OD pairs per type as illustrative examples. The two OD pairs on the left (top and bottom) are large and exhibit a regular cyclo-stationary behavior in time. There is a strong periodicity at intervals of 24 hours that fits the expected diurnal patterns. On the right we plot two OD pairs with extremely low mean rate and see that these flows are characterized by a very noisy shape. These OD pairs have lost any cyclo-stationary shape and tend to send most of their traffic in small time intervals. The two OD pairs in the middle column, are indeed in between these two behaviors: some may retain small diurnal patterns while others are

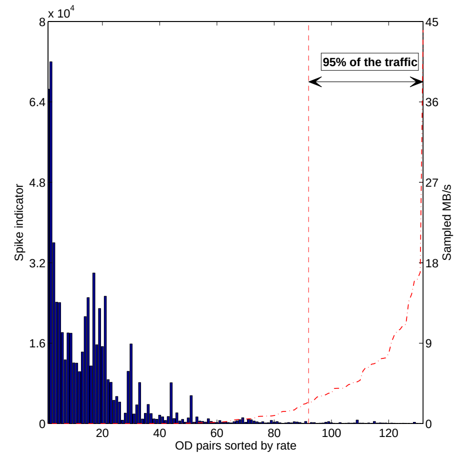

Figure 3: Spikiness and average traffic rate for each OD pair.

already becoming dominated by noisy behavior. We will use the terms "large", "medium" and "small" for these three types of OD flows; these terms are not defined precisely but are used just to ease the readability of the presentation.

Other traffic matrix studies have found high errors in small flows and have argued that this is not important because network administrators only care about the large flows [9, 5]. By using our data here, we can shed some light a why small flows pose problems for estimation, and further justify this suggestion.

Small flows are difficult to estimate accurately for multiple reasons. First, they are often very noisy and can have long periods of zero or close-to-zero transmission rates. Second, the small flows can often be two or three orders of magnitude less in volume than large flows. Hence the numbers are often so small that they cause numerical stability problems in estimation methods. Third, the small flows exhibit the largest relative spikiness of all flows. We define spikiness of a flow in terms of its time series. We difference the flow's basic time series by one measurement interval (10 minutes in our case) and divide by the mean to normalize. This helps illustrate how dramatically a flow can change from one measurement interval to the next. To have only one metric for the entire time series, we take the sum of the maximum and the absolute value of the minimum value observed along time. We plot our spikiness indicator in Figure 3. The OD pairs are sorted in order of increasing average rate. We see that this spikiness characteristic decreases as the average rate of a flow increases.

The good news is that these difficult small flows constitute an insignificant portion of the total network-wide traffic. In this same Figure 3, the right hand side shows the cumulative contribution of OD flows to the total network traffic. *We see that 30% of the OD pairs in the network carry 95% of the total network traffic. This justifies our approach in which we choose to focus only on medium and large flows as the target for accurate TM estimation.* Moreover, the flows in this batch do not suffer from relative spikiness problems. We define the cutoff threshold for flows in the medium/large category to be those above 3 MB/s, so that we include enough OD flows to capture 95% of the total network-wide load. Sometimes we use the term "top" OD flows to refer to those in the medium/large category defined by this threshold.

Figure 2 hints that OD flows contain (at least) two sources of variability, namely *diurnal patterns* and a noisy *fluctuations behavior*. These two types of variability could show up at different time scales. We believe these two types of variability are important to be captured explicitly in a model for the following reason.

Consider the implications of the idea of changing link weights to increase the rank of the system. In practice, network operators cannot change link weights in rapid succession. Once the weights have been changed, the routing protocol needs to compute new routes and then we need to collect multiple samples of SNMP link counts under the new routing. It was shown in [3] that it is typically advantageous to use around 10 consecutive SNMP samples for TM estimation. In this work on time-varying network tomography, it becomes clear that estimation methods need to take advantage of link correlations to make reasonable estimates. Therefore it is likely to be a few hours (at least) between weight change events.

This leaves carriers with two choices as to how to collect the data they need. One way is to collect all the data at the same hour every day, but over many days. The advantage of this method is that the traffic matrix will be stationary, but it will take many days to collect the required data. When the traffic matrix is in the stationary regime, a model is needed to capture the fluctuation behavior. Poisson or Gaussian assumptions have been used for this scenario [1, 3]. In our work, we consider a more general model for the fluctuations that does not require us to make assumptions about whether the process is is stationary or cyclo-stationary.

The second option is for network operators to collect all the SNMP samples from all the snapshots within a few hours; but then the traffic matrix is likely to be non-stationary. It is known that link fluctuations are not stationary over multi-hour periods. Our data indicates that the traffic matrix is at least cyclo-stationary due to the large diurnal swings.

We leave the decision as to when to collect all the needed measurements up to carriers as they will have other constraints regarding the management of the network that will impact this decision. In order to enable any choice they might make, and to develop a general model, we incorporate both cyclo-stationary behavior and noise fluctuations into our model and estimation procedure.

We can further motivate the need to explicitly model these two sources of variability with the following observation. In Figure 2 we observed these two behaviors across different flows. In fact, in the flows we care about (medium and large flows) both of these sources of variability often appear within each flow. To see this consider the two sample OD flows plotted in the top portion of Figure 4. The real OD pairs are plotted with continuous lines. We used five basis functions of a Fourier series to filter out the **diurnal patterns**, represented by dotted lines. This represents the first component of an OD flow. The signal that remains after the diurnal pattern is filtered out is shown in the bottom plots. We call this latter component of an OD flow the **fluctuations process**.

We make a few observations on these two components that strongly impact our choice of models. *An OD flow should contain two distinct components, one for the diurnal trend and one for the fluctuations process. The diurnal trend can be viewed as deterministic and cyclo-stationary, while the fluctuations process can be viewed as a stationary zero mean random process.* The stationarity of the fluctuations process is evidenced in the figure by a pretty consistent absence of any cycle trends over long time scales (hours and days). We said that one of our goals is to estimate the variance of the traffic matrix. In fact, what we will be estimating is the variance of this fluctuations process.

In the literature so far, it has been common to assume a fixed known relationship between the mean and standard deviation of an OD flow. The most common assumption is on the existence of a power law relationship between the two parameters. Such assumptions are needed to help the estimation problem to be more tractable. Figure 5 shows the relationship between the mean and standard deviation for our OD pair data sorted by their mean (from the smallest to the largest one). The points on the plot do not approximate a straight line as the assumption on the existence of a power law would entail. Note that this is a log-log plot hence the deviations from the straight line can be quite large. The best linear

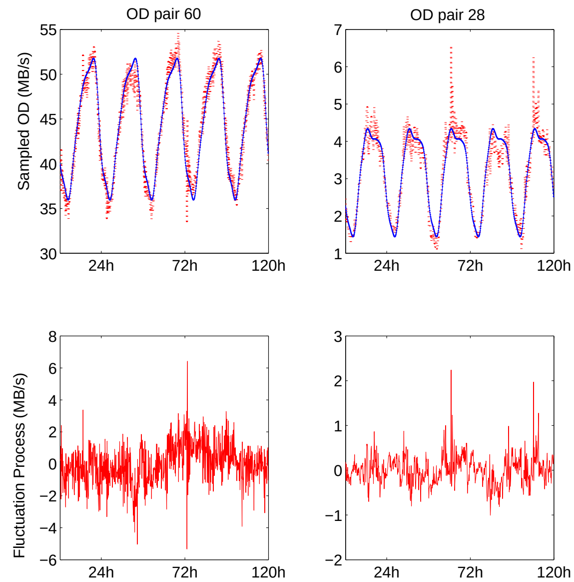

Figure 4: On the top is shown an example of two real large OD pairs (dotted lines) and their *diurnal patterns* (continuos lines). On the bottom is shown an example of *fluctuation process* for the two OD pairs obtained by removing the diurnal trends from the original signals.

fitting of the data we present in the figure corresponds to a power law with exponent approximately equal to 0.78, that implies a variance power-law coefficient equal to 1.56.When the power-law coefficient is computed for individual OD pairs, we find a large range spanning from 1 to 4. Existing methods require that a single value for this variance applies to all flows. Clearly this is not the case. Also, different researchers using different data sets have computed different values for this coefficient [3, 4]. Thus one cannot generally assume a particular value, nor a single value. It is unclear what the impact on errors is in methods that rely on this assumption.

In our work, we need not make any such assumption. Instead *we remove the power law assumption and choose to estimate the variance directly*, independently of the mean. What this plot does confirm is the hypothesis that OD flows with large variance are also the ones with large mean. Hence there is an implication that the order of magnitude of the standard deviation is closely related to the order of magnitude of the mean for an OD flow. We will make use of this observation to help identify the top flows.

5. METHODOLOGY

Before describing the details of our models and estimation procedures, we give here an overall summary of the methods proposed. We do so here because our methods involve combining a number of steps that are quite different from previous solutions.

In this paper we essentially propose two new methods. Method 1 has five critical components: (1) collect all the relevant SNMP data and estimate the variance of OD flows. To do this we provide a closed form solution for estimating the variance of OD flows; (2) use an algorithm for selecting a minimal set of weight changes (snapshots) to have a full rank linear system; (3) apply the route changes and collect all the relevant SNMP data; (4) we incorporate models into the estimation procedure to capture both the diurnal trends and the fluctuations process; (5) given this new model with a high rank system, basic inference techniques can be used (e.g., pseudo-inverse or Gauss-Markov estimators). The algorithm for finding appropriate snapshots is not new. The contributions here are

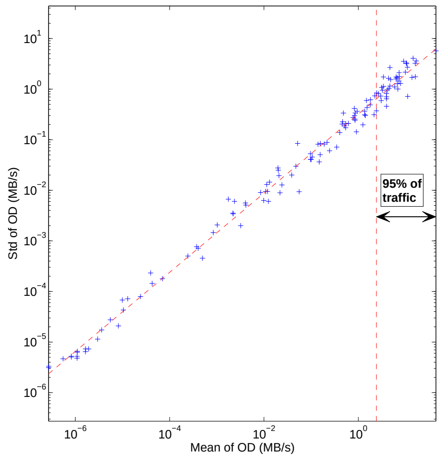

Figure 5: Standard deviation of traffic fluctuations vs the mean traffic volumes for all the OD pairs (in Mbytes/s).

in the modeling and estimation elements of steps (1) and (4), and in the validation of the composite method against real data (something that hasn't been done until now). We use Method 1 to estimate *all* of the OD pairs in the traffic matrix.

Note that step (1) can be omitted if a pseudo-inverse method is used, since it does not require any knowledge about traffic fluctuation statistics.

A key component in Method 2 is to identify the top largest OD flows (in advance of estimating their average volume). As we observed in the last section, a weak relation exists between the order or magnitude of the standard deviation and the order of magnitude of the means. By selecting the flows with the largest variance, we can be reasonably assured that we have identified the flows that are largest in mean. We then set the estimates for the small flows to zero and considered them known. We thus have fewer variables to estimate as only the large flows remain. Having a system with fewer unknowns, we can run our algorithm for finding the key link weight changes to increase the rank of the system. We expect to need fewer snapshots now since there are fewer variables to estimate.

Method 2 can be summarized by the following steps: (1) collect all the relevant SNMP data and estimate the variance of OD flows using the same model mentioned above; (2) use a threshold policy (defined later) and select all OD flows whose variance is above the threshold - call the set $\mathcal{X}_L$; (3) evaluate the route changes needed to make the subproblem (seeking only the OD pairs in $\mathcal{X}_L$) full rank; (4) apply the route changes and collect all the relevant SNMP data; (5) utilize the new models for diurnal and noise behavior; (6) estimate both the diurnal trends and the fluctuations process for OD pairs in $\mathcal{X}_L$ using basic inference methods. In essence, Method 2 uses one extra step to identify the subset $\mathcal{X}_L$. The last four steps of Method 2 differ from those in Method 1 only in that they are applied to a subset of the original system.

We evaluate these two methods in this paper. We point out that in our derived methodology (next section) we also provide one more component that is included in the mathematical development to establish a complete methodology; however it is not evaluated here due to lack of space. This last step is a close form solution for evaluating the goodness of the variance estimate.

By comparing these two methods we can evaluate an important tradeoff. On the one hand, in method 1 we hope to have very good accuracy since we have upgraded the system to full rank. On the other hand, network operators would prefer to do the smallest number of link weights changes necessary. In method 2 fewer snapshots are needed, but potentially at the expense of accuracy of the estimates. Because the small flows are set to zero, their bandwidth will be redistributed to other flows. A comparison of these two methods should illustrate how much additional error we might incur by greatly reducing the number of snapshots needed.

6. MODELS AND ESTIMATES

Before tackling the modeling aspects of the problem, we remind to the reader that we are interested in extracting two main components for each OD pair: the **diurnal pattern** and the **fluctuation process**. The diurnal pattern captures the evolution in time of the mean, while the fluctuation process captures the noisy behavior around the mean. In this context, estimating the mean corresponds to extract the main trend in time of the OD pair, while estimating the variance collapses to estimating the variance of the fluctuation process.

6.1 Stationary Traffi c Scenario

In this section we assume that $\{X(k,n) \forall k \in [0, K-1]$ and $n \in [0, N_s - 1]\}$ is a realization of a segment of a stationary discrete time random process. In particular, we model each OD pair as follows:

$$X(k,n) = x + W(k,n) \ \ , \forall k \in [0, K-1] \text{ and } n \in [0, N_s - 1], \tag{2}$$

where x is a deterministic column vector representing the mean of the OD pair, while $\{W(k,n)\}$ are zero mean column vectors, i.e. $E[W(k,n)] = \mathbf{0}$, representing the "traffic fluctuation" for the OD pairs at discrete time n in the measurement interval k. We define $W_p(k,n)$ as the p^{th} component of $W(k,n)$.

In practice, for each measurement interval k, there may be "missing" link measurements, due to collection errors, human errors, etc. In addition, depending on the network topology and the routing in measurement interval k, some of the equations in (1) may be redundant (linearly dependent). For simplicity of notation, we assume to keep all the equations in the system. Then we have:

$$Y(k,n) = A(k)X(k,n) \ \ , \forall k \in [0, K-1] \text{ and } n \in [0, N_s - 1]. \tag{3}$$

By definition, rank$(A(k)) = M_k$. Typically, $M_k \leq L < P$ so it is not possible to infer the value of $X(k,n)$ from $Y(k,n)$, even if there are no missing link measurements. As we mentioned before, we assume that the routing can change during different measurement intervals, so it is possible that $A(k_1) \neq A(k_2)$ if $k_1 \neq k_2$.

Define the following matrices, using block matrix notation:

$$Y = \begin{bmatrix} Y(0,0) \\ \vdots \\ Y(0, N_s - 1) \\ \vdots \\ Y(K-1, N_s - 1) \end{bmatrix}, \ A = \begin{bmatrix} A(0) \\ \vdots \\ A(0) \\ \vdots \\ A(K-1) \end{bmatrix}$$

$$W = \begin{bmatrix} W(0,0) \\ \vdots \\ W(0, N_s) \\ \vdots \\ W(K-1, N_s - 1) \end{bmatrix}, \ C = \begin{bmatrix} A(0) & 0 & \dots & 0 \\ 0 & A(0) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A(K-1) \end{bmatrix}$$

Note that in the new A matrix, each $A(k)$ matrix is repeated for N_s times, since the routing is assumed to stay the same within the same measurement interval k. In terms of dimensions, note that Y is a LN_sK-dimensional column vector, A is a $LN_sK \times P$ dimensional matrix, W is a KN_sP-dimensional column vector, and C is a $LN_sK \times KN_sP$ matrix. Putting (3) into matrix form, using (2), we obtain

$$Y = Ax + CW. \tag{4}$$

Our aim is to estimate x from the observations $Y(0,0), \ldots, Y(0, N_s - 1), \ldots, Y(K-1, N_s-1)$. We can use this estimate of x as an estimate of $X(k,n)$ for future time intervals. We shall denote our estimate of x by $\hat{x}$. Throughout the paper, we shall assume that A is of full rank.

One possible approach for synthesizing our estimate $\hat{x}$ is to minimize the Euclidean norm of $Y - Az$, i.e.

$$\hat{x} = \arg\min_z \{(Y - Az)^T (Y - Az)\}\,. \tag{5}$$

6.2 Estimating the Traffic Matrix Given Covariance of Traffic Fluctuations

In this section we show how simple and well-known inference techniques, i.e. *Pseudo-Inverse* and *Gauss-Markov* estimators, can be applied to solve the optimization problem shown in Equation (5) by using the full rank property of our approach.

To this end, let B be the covariance matrix of W, i.e.

$$B = E[WW^T]\,. \tag{6}$$

Let denote with t the discrete time across all the measurement intervals, from 0 to $T-1$ where $T = (KN_s)$, that identifies the total number of samples collected across all the experiment. Note that exists the bi-unique relationship between the discrete time t and the two temporal indexes k and n, i.e. $t = kN_s + n$, where $k = \lfloor t/N_s \rfloor$ and $n = t - \lfloor t/N_s \rfloor N_s$. Then the covariance matrix B can be written as:

$$B = \begin{bmatrix} R(0) & R(1) & \ldots & R(T-1) \\ R(1) & R(0) & \ldots & R(T-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(T-1) & R(T-2) & \ldots & R(0) \end{bmatrix}, \tag{7}$$

where $R(t)$ is the $P \times P$ matrix defined by:

$$R(t) = [r_p(t)] = \mathrm{diag}(E[W_p(\tau) W_p(\tau + t)]). \tag{8}$$

With this definition, the covariance vector of Y is equal to CBC^T. A basic issue is that B is unknown. For the time being, however, let us assume that B is known. With this assumption, we now consider the problem of estimating the traffic matrix x. The best linear estimate of x given Y, in the sense of minimizing $E[(Y - Az)^T (Y - Az)]$ with respect to z is known as the best linear Minimum Mean Square Error (MMSE) estimator. The best linear MMSE estimate of x can be obtained from the Gauss-Markov Theorem [10], and is stated below.

PROPOSITION 1. *The best linear MMSE estimator $\hat{x}$ of x given Y is*

$$\hat{x} = \hat{x}(Y, B) = \left(A^T (CBC^T)^{-1} A\right)^{-1} A^T (CBC^T)^{-1} Y\,. \tag{9}$$

Note that the estimate $\hat{x}$ in (9) reduces to the pseudo-inverse estimate when B is the identity matrix. If W has a Gaussian distribution, it can be verified that the estimate in (9) is in fact the maximum likelihood estimate of x.

COROLLARY 1. *Regardless of whether or not W is Gaussian, the estimate in Proposition 1 is unbiased, i.e. $E[\hat{x}] = x$, and furthermore we have*

$$\hat{x}(Y, B) = x + \left(A^T (CBC^T)^{-1} A\right)^{-1} A^T (CBC^T)^{-1} CW \tag{10}$$

and

$$E[(\hat{x} - x)(\hat{x} - x)^T] = \left(A^T (CBC^T)^{-1} A\right)^{-1}\,. \tag{11}$$

Note that (11) allows us to estimate the accuracy of our estimate $\hat{x}$ of x given Y and B. In particular, the p^{th} element of the diagonal of the matrix $E[(\hat{x} - x)(\hat{x} - x)^T]$ is the mean square error of our estimate of the p^{th} element of the traffic matrix.

An advantage of using the pseudo-inverse estimate above is that it does not require knowledge about the statistics of W. On the other hand, knowledge of the statistics of W can be used to obtain an improved estimate of x. In particular, from (4), it is evident that the components of Y may have different variances. For example, if the variance of a component of Y is small, the square of the difference between that component and the corresponding component of $A\hat{x}$ should be heavily weighted in the determination of an estimate of x.

6.3 Estimating the Covariance of Traffic Fluctuations

Next, we are interested in estimating the variability of traffic between OD pairs over time, so that a better estimation of x can be obtained using Gauss-Markov method (Equation 9) and *confidence intervals* for our estimate can be defined (Equation 11). Moreover, we have shown previously in Section 4 how flows with large variance are also the ones with large mean. Having a model that allows us to estimate the variance of each OD pair can help the identification of the top largest flows carrying the most significant fraction of network traffic. To this end, we will consider the problem of estimating the covariance function of the fluctuation process $\{W\}$. In doing that, we assume that different OD pairs fluctuation can be modeled as independent random processes.

We shall present a method for obtaining such an estimate. We highlight two nice characteristics of our model. First, it does not require any knowledge of the first order moment statistics. Previous approaches assumed to know exactly the mean to recover the covariance and vice-versa. As a consequence, the new model does not suffer anymore of potential error propagation problems introduced by the first order moment estimation. Second, the model does not require any routing configuration change. It uses only a large number of measurements of the link counts under the same routing configuration. We prove in Section 7 that such a result generally holds for any topology under realistic assumptions of *minimum cost routing* and *strictly positive link costs.*

Let us consider a generic discrete time $t \in [0, T-1]$ presented in Section 6.2. We can define the *link correlation matrix* as $C_Y(t) = E[Y(\tau) Y^T(\tau + t)]$. For two links l and m, each entry of this matrix can be defined as follows:

$$E\left[\sum_{i=1}^{P} \sum_{j=1}^{P} A_{l,i}[X_i + W_i(\tau)] A_{m,j}[X_j + W_j(\tau + t)]\right]$$

$$= \sum_{i=1}^{P} A_{l,i} A_{m,i} r_i(t) + \sum_{i=1, j=1, j \neq i}^{P} A_{l,i} A_{m,j} E[X_i] E[X_j].$$

Note that in the previous statement we assumed that each OD pair is independent. By using a matrix notation, we can write:

$$C_Y(t) = AR(t)A^T + E[AX]E[AX]^T = AR(t)A^T + E[Y]E[Y]^T. \tag{12}$$

Then, we can estimate the $R(t)$ as follows:

$$\hat{R}(t) = \arg\min_Z ||C_Y(t) - AZA^T - E[Y]E[Y]^T||_2^2.$$

Finally we notice that we can re-write equation (12) as $\gamma_y(t) = \Gamma r_w(t)$, where $\gamma_y(t)$ is $C_Y(t) - E[Y]E[Y]^T$ ordered as a L^2 vector, Γ is a $L^2 \times P$ matrix whose rows are the component-wise products of each possible pair of rows from A, and $r_w(t)$ is a P vector whose elements are $r_p(t)$.

As a consequence an estimation of $r_w(t)$ can be obtained using the pseudo-inverse matrix approach:

$$\hat{r}_w(t) = (\Gamma^T\Gamma)^{-1}\Gamma^T\gamma_y(t). \tag{13}$$

We conclude this section with some considerations on the accuracy of the estimation of $\hat{r}_w(t)$. After some computations it results:

$$E\left[(\hat{r}_w(t) - r_w(t))^T\,(\hat{r}_w(t) - r_w(t))\right] =$$

$$\frac{1}{T^2}||\Phi||\sum_{l=1}^{L}\sum_{m=1}^{L}\sum_{i=0}^{T-1}\sum_{j=0}^{T-1}E[y_l(i)y_m(i+t)y_l(j)y_m(j+t)] -$$

$$\frac{1}{T^2}||\Phi||\sum_{l=1}^{L}\sum_{m=1}^{L}\sum_{i=0}^{T-1}\sum_{j=0}^{T-1}E[y_l(i)y_m(i+t)]E[y_l(j)y_m(j+t)]$$

being $\Phi = [(\Gamma^T\Gamma)^{-1}\Gamma^T]^T[(\Gamma^T\Gamma)^{-1}\Gamma^T]$.

PROOF. Let us consider the case in which $t = 0$. This simplifies the notation (the other cases are similar). Let $Y_2(\tau)$ be an L^2 vector whose components are $y_l(\tau)y_m(\tau) - E[y_l(\tau)]E[y_m(\tau)]$, after some easy computation we can write:

$$\hat{r}_w(0) = \frac{1}{T}\sum_{\tau=0}^{T-1}(\Gamma^T\Gamma)^{-1}\Gamma^T Y_2(\tau) \tag{14}$$

but since $r_w(0) = \frac{1}{T}[\sum_{\tau=0}^{T-1}(\Gamma^T\Gamma)^{-1}\Gamma^T]\gamma_y(0)$ it results:

$$r_w(0) - \hat{r}_w(0) = \frac{1}{T}\sum_{\tau=0}^{T-1}(\Gamma^T\Gamma)^{-1}\Gamma^T[\gamma_y(0) - Y_2(\tau)] \tag{15}$$

from which the assert easily follows just remembering that given any pair of real valued vectors V and W and a real valued matrix D; $V^TD^TDW \le ||D^TD||V^TW$, being $||D^TD||$ the maximum (in module) eigenvalue of the symmetric matrix D^TD. This property can be easily checked since D^TD is a symmetric matrix and thus self-adjoint; which implies that a complete system of orthogonal normalized eigenvectors of D^TD exists. $\square$

We notice that Equation (15) can be used to relate the confidence of the estimation of $R(t)$ to the forth order moments of link-count statistics which can be evaluated through standard statistical techniques.

6.4 Cyclo-Stationary Traffic Scenario

Next, we consider a cyclo-stationary model for traffic matrices and the associated estimation problem. As before, let $A_{l,p}(k)$ be the fraction of the traffic $X_p(k,n)$ from OD pair p that traverses link l during measurement interval k at discrete time n. For simplicity of notation we hide the snapshot information by using the discrete time t defined in section 6.2. Thus, (1) and (3) hold as before. Rather than assume that $\{X(t)\}$ is a random process with constant mean, we consider a model where $\{X(t)\}$ is cyclo-stationary. Specifically, instead of (2), we assume that

$$X(t) = x(t) + W(t) \ \ ,\forall t \in [0, T-1]. \tag{16}$$

We assume that $\{X(t)\}$ is cyclo-stationary with period N, in the sense that $X(t)$ and $X(t+N)$ have the same marginal distribution. More specifically, we assume that $W(t)$ is zero mean and that $\{x(t)\}$ is a deterministic (vector valued) sequence, periodic with period N. In Section 4 we have shown that the "fluctuation" vectors $\{W(t)\}$ can be assumed to be stationary at 5 minutes time scale. Since this claim does not hold at smaller time scales, next we introduce a more general model based on the assumption that the "fluctuation" vectors $\{W(t)\}$ is cyclo-stationary to second order, i.e. $E[W(t)W^T(t+m)] = B^{(t,m)}$ where the covariance matrices $B^{(t,m)}$ are such that $B^{(t,m)} = B^{(t+N,m)}$ for all k. Note that if the "fluctuation" vectors $\{W(t)\}$ is stationary to second order, the model is still valid but the covariance matrices $B^{(t,m)}$ are such that $B^{(\tau,n)} = B^{(t,n)}$ for all t, τ and n.

Our aim is to estimate $x(t)$ for all t (the "traffic matrix") given the observations $Y(t)$, $0 \le t < T = KN_s$.

We shall assume that $x(t)$ can be represented as the weighted sum of $2N_b + 1$ given basis functions, i.e.

$$x(t) = \sum_{h=0}^{2N_b}\theta_h b_h(t) \tag{17}$$

where for each h, θ_h is a $P \times 1$ vector (of "coefficients"), and $b_h(\cdot)$ is a scalar "basis" function that is periodic with period N. In particular, we will consider a Fourier expansion where

$$b_h(t) = \begin{cases} \cos(2\pi th/N) & \text{, if } 0 \le h \le N_b \\ \sin(2\pi t(h-N_b)/N) & \text{, if } N_b + 1 \le h \le 2N_b \end{cases} \tag{18}$$

Substitution of (17) into (16), and then (3), we obtain

$$\begin{aligned} Y(t) &= A(k)(\sum_{h=0}^{2N_b}\theta_h b_h(t)) + A(k)W(t) \\ &= A'(t)\theta + A(k)W(t), \end{aligned}$$

where we define the $(2N_b+1)P \times 1$ vector θ according to

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{2N_b} \end{bmatrix}, \tag{19}$$

and $A'(t)$ is the $L \times (2N_b+1)P$ matrix defined as

$$A'(t) = \begin{bmatrix} A(k)b_0(t) & A(k)b_1(t) & \cdots & A(k)b_{2N_b}(t) \end{bmatrix}. \tag{20}$$

Next we re-define the matrix A to be of dimension $LT \times (2N_b + 1)P$ as follows:

$$A = \begin{bmatrix} A'(0) \\ A'(1) \\ \vdots \\ A'(T-1) \end{bmatrix}. \tag{21}$$

The matrices C and W are defined as before. With this notation we have an equation similar to equation (4), namely

$$Y = A\theta + CW. \tag{22}$$

Thus, we can use essentially the same method as before to estimate θ, and hence estimate $x(t)$ for each time instant t (see Equation (17)). The same approaches proposed in Section 6.3 can be applied to the problem described by Equation (22) to estimate the covariance matrices $B^{(t,m)}$.

7. IDENTIFIABILITY OF SECOND ORDER MOMENT

We want to prove that is always possible to estimate the covariance function without requiring any routing configuration changes for any topology.

THEOREM 2. *For a general connected topology the rank of* Γ *is* P *under any minimum cost routing in which link costs are strictly positive.*

PROOF. We will prove the theorem by contradiction. Suppose the rank of Γ to be smaller then P then there exists a non null vector $V_o \in R^P$ that is mapped through Γ in the null vector (i.e, $\Gamma V_0 = 0$). Let $T \leq P$ be the number of non null components of V_0. Let $\mathcal{V}$ be the vectorial space of cardinality T comprising all the vectors which have null components in correspondence to null components of V_0.

We consider the correspondence $F : \mathcal{V} \rightarrow R^T$ which maps any vector $V \in \mathcal{V}$ into a vector $W \in R^T$ by discarding null components of V. Let $\hat{\Gamma}$ be the matrix obtained by Γ by discarding the columns that in the multiplication ΓV with $V \in \mathcal{V}$ would not give contribute (since multiplied by null elements of V).

By construction $\Gamma V = \hat{\Gamma} W$ for any $V \in \mathcal{V}$. Finally let W_0 be the vector which corresponds to V_0 though F. We notice that every component of W_0 is not null.

We will show that necessarily $\hat{\Gamma} W_0 \neq 0$, thus leading to a contradiction with the previous assumptions, since $\hat{\Gamma}$ has a row which necessarily contains one and only one non null element.

To prove that a row of $\hat{\Gamma}$ contains one (and only one) non null element let us consider the set Z_{od} of OD pairs which correspond to the elements of W_0 and compute path-cost (sum of the link weights) for any OD pair oin Z_{od}. Let z_{od} be an OD pair in Z_{od} which corresponds to a maximum path cost. Consider the first and the last link (l and m respectively) spanned by the z_{od} path. We claim that the set of OD pairs that pass through both links necessarily comprises only z_{od}. As a consequence the row of $\hat{\Gamma}$ which corresponds to the considered links must contains only one element different from 0.

To complete the proof we only have to show that links l and m are both crossed only by z_{od}: i) by construction z_{od} crosses both l and m. ii) no other OD pairs can cross l and m. Indeed assume $z'_{od} \neq z_{od}$ crosses both links l and m then there are only two possibilities: 1) l is the first and m the last link of the $z`_{od}$ path; in this case however both the origin the destination of z'_{od} would result coincident with the origin and the destination of z_{od} then contradicting the fact that $z'_{od} \neq z_{od}$; 2) either l is not the first link or m is not the last link of the path spanned by z'_{od}; in this case however the path cost of z'_{od} result larger than the path cost of z_{od} (since the sub-path of z'_{od} from l to m necessarily have the same cost of the whole z_{od} path) thus contradicting the fact that the path cost of z_{od} is maximum. $\square$

8. RESULTS

We now evaluate our two methods using the Netflow data from our commercial Tier-1 backbone. We focus on the scenario in which operators want to estimate the TM within hours (as opposed to using measurements from the same hour each day over many days); in other words, this corresponds to the cyclo-stationary environment. In Section 8.1 we validate our model for the mean estimation, i.e. the diurnal pattern of a PoP-to-PoP traffic matrix. We show that Method 1 can push error rates down into a whole new range. In Section 8.2 we show how the second order moment model in Method 2 helps the identification of the top largest flows carrying the largest amount of network traffic. As a consequence of being able to do this, we can dramatically reduce the number of routing changes needed to obtain a full rank system. This in turn is the key to driving error rates down consistently. We illustrate the tradeoff in terms of error rates versus number of routing changes used. For both methods, we used a pseudo-inverse estimator as our basic inference method in the last step. We used this, rather than the Gauss-Markov estimator because (as will be explained later), we did not have enough months of netflow data to do a proper comparison using Gauss-Markov estimators.

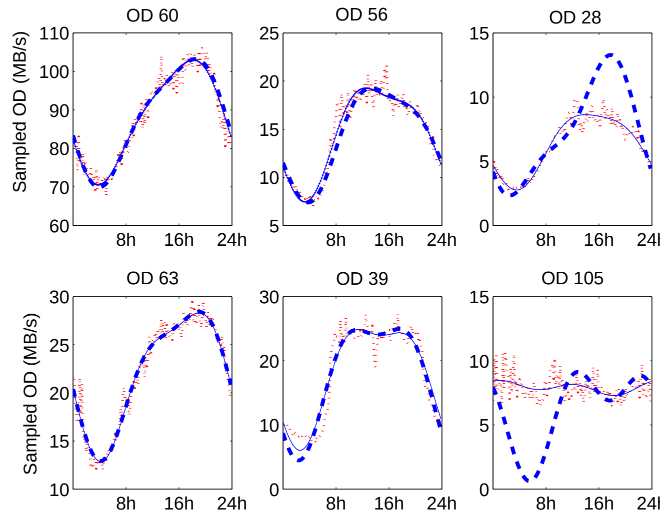

Figure 6: Method 1. Mean Estimation. Real OD flow (dotted lines), de-noised OD flow (continuous lines), and estimated OD flow (dashed lines).

8.1 Method 1: mean estimation

In this section we assess the accuracy of the mean estimation when the traffic matrix is computed using Method 1. We remind the reader that in this section, our goal is to estimate *all* of the OD pairs. For component (2) of this method, we used the heuristic presented in [8] to generate a sequence of snapshots (a schedule of link weight changes) in order to obtain a full rank A matrix. For our network scenario considered, the algorithm determined that $K = 24$ snapshots were needed to identify all the OD pairs. Of these 24 snapshots, 22 of them involve only one link weight change at a time, (i.e. *single weight change*), while the last two involve two simultaneous link weight changes, (i.e. *double weight changes*).

Before providing summary statistics for all the OD pairs, we give results - for illustrative purposes - from six particular OD pairs. The six pairs in Figure 6 all come from the large and medium flow categories. For each flow, these graphs shows the temporal shapes of the real, de-noised, and estimated OD pair. The de-noised OD pair refers to an OD pair with everything filtered out except the first 5 basis functions of Fourier series; put alternatively, this illustrates how well a simple Fourier model captures the changing mean behavior. We see that our model fits the large and medium flows extremely well. It is interesting that the quality of the estimation obtained decreases as the average rate drops. The two on the right do suffer from larger errors. Note that these two OD flows (#28 and #105) were the worst case performing OD flow estimates from within the medium and large category. Our method exhibits the same type of performance behavior as other methods in that it estimates large flows well and has difficulty as the flows get smaller and smaller.

The gain of our method comes in terms of the actual error rates achieved for these top flows. To examine estimation errors in general, we use the following two metrics. First, we examine the difference between the *first component* of the Fourier series, i.e. the

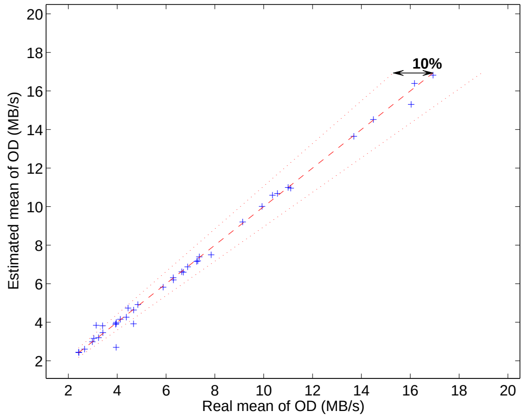

Figure 7: Method 1. Relative error of 1^{st} component of Fourier series for top flows with $N_s = 100$.

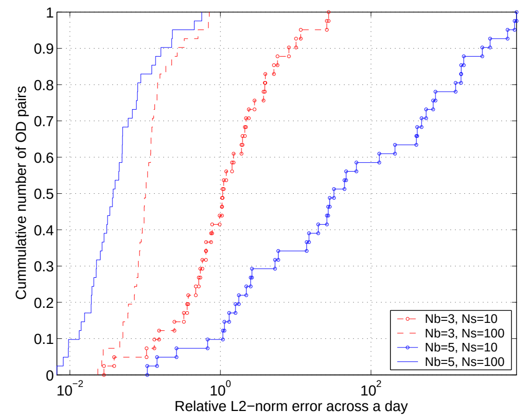

Figure 8: Method 2. Cumulative distribution of the relative *L2-norm* error for top flows as a function of N_b and N_s.

continuous component, of the netflow data and the estimation provided by our models. Since this part of our estimate would be used to populate a traffic matrix, we compute the relative error of this estimate. Second, since we have a temporal process, we need to examine the errors in estimation over time. We thus use the difference in energy between the estimate $\hat{X}_p(t)$ and the real data $X_p(t)$. In particular, we use the relative L2-norm to estimate the goodness of the model fitting, $||X_p(t) - \hat{X}_p(t)||_2^2 / ||X_p(t)||_2^2$.

Figure 7 shows our first metric on the difference of the first component of the Fourier series for the real data (x-axis) and the estimated OD pairs (y-axis) for medium and large flows when $N_s = 100$ samples per snapshot are used. The average error is 3.8%. This is a large improvement upon other methods whose *average* errors typically lie somewhere between 11%-23%. [1] Some carriers have indicated that they would not use traffic matrices for traffic engineering unless the inference methods could drive the average errors below the 10% barrier. We believe that this is the first study to achieve this.

Among these flows, the worst case relative error across all OD pairs is 28%. If we look carefully at the figure, we can see that all the flows have less than 10% relative error, except for four outliers. These outliers correspond to OD pairs whose average rate is less than 5 MB/s, i.e. among the smallest OD flows within the "top" set represented here. For 90% of these top flows (i.e., excluding these four outliers), the average error drops below 2% and the worst case relative error drops to 4.6%. (See the first column in Table 1 for 24 snapshots.) We point out that small OD pairs (those whose rate is under 3 MB/s) do not appear in the figure but are considered in the system and are thus a part of the overall computation.

Figure 8 shows the *L2-norm* relative error cumulatively. In most of our calculations the number of basis functions we used was $N_b = 5$ and the number of samples per snapshot used was $N_s = 100$. This figure shows that the worst case L2-norm relative error was 56% corresponding to OD pair #28 (the second worst case is 44%, corresponding to OD pair #105). We see that 85% of the flows had an L2-norm error of less than 10% (see OD pairs #56 and #39 as an example). Figure 6 helps us to understand the L2-norm error. The flows with low errors are typically going to be the large and medium OD pairs since our model succeeds in capturing the diurnal trends. The errors are typically going to appear in the estimates for smaller flows; Figure 6 illustrates that even though we have trouble tracking the dynamics, we do still track the mean behavior of these flows. We are including the examples of the worst case behavior for completeness of analysis and to describe how our method can perform in estimating small OD flows.

Others have reported relative errors on their mean estimates where the means are computed over some specific interval (usually somewhat long). Our L2-norm relative error is an error rate on our estimation (or fitting) of the dynamic OD flow varying in time; put alternatively it summarizes "instantaneous" errors in OD flow modeling. We cannot compare the value of this latter metric to other studies because other studies have not tried to build a model capturing the temporal behavior.

The performance of our method is influenced by the number of samples per snapshot and the number of basis functions used. In this same figure, we can examine the influence of these two parameters. Intuitively, a larger number of basis function N_b will lead to a better quality estimate, although at the cost of a larger number of samples. Note that the number of samples N_s plays an important role independently of the number of basis functions implemented. The more samples collected the more is learned about the temporal evolution of each OD pair, and a better estimation can be provided. For a fixed number of basis functions, going from $N_s = 10$ to $N_s = 100$ yields a substantial improvement. Our experimentation showed that using more than 5 basis functions yielded insignificant gains and thus we decided to set $N_b = 5$ for the remainder of our evaluation.

8.2 Method 2: identification and estimation of the top OD pairs

In this section we study the performance of Method 2. We start by estimating the variance of the OD flows. This information is used to isolate the "top" flows that we want to estimate. We do this by setting the small flows to zero and hence removing any need to estimate the associated variables. As we will see, Method 2 makes an interesting tradeoff surface, namely that of reducing the number of snapshots required at the expense of some accuracy in estimation.

In Section 4 we saw that 95% of the network traffic is carried by only 30% of the OD pairs, each of whose rate is greater than 3 MB/s. The first two steps of Method 2 are used in order to try to identify as many of these large flows as possible. First we compute the variance of the OD flows using Equation (14). Second we order the OD flows by size of variance. By relying upon our observation that flows that have large variance are also typically large in mean,

[1] It is hard to compare numbers exactly because different studies use different amounts of total load. However we capture more total traffic than most other studies that typically include 75% or 80% of the network-wide load.

the task now is to set a threshold and select the "top" flows above this threshold. There are two issues arising from these steps.

As discussed in Section 6, the nice feature about our variance estimator is that it does not require any routing changes. However a large number of SNMP samples are required to force the relative error of the variance estimate to be under 5%. We selected the 5% target arbitrarily. After some experimentation we found that roughly 25,000 samples were needed to achieve this target level of accuracy. In a real network this implies one needs about 3 months worth of SNMP data. In commercial networks obtaining this much historical data is not a problem as all ISPs store their SNMP data for multi-year periods. Although we had plenty of months (years) of *SNMP* data, we did not have 3 months of *Netflow* data available to us that would have been needed to do a complete validation of this method.

We therefore decided to use pseudo-real data for this evaluation. We call this pseudo-real data because it is generated based on a model fitted to actual data (but only one month's worth). To create sample OD flows, we filter out the noise from each of our sampled OD pairs and keep only the first five components of the matched Fourier series. We generate sample OD flows, over longer periods of time, using this Fourier series model to which we add a zero mean gaussian noise with a power-law variance whose coefficient is set to 1.56 (in accordance with the empirical data observed in Figure 5). By comparing Figures 6 (Netflow data) and 11 (pseudo-real data), we can see how well our pseudo-real data generator matches the actual sampled data. We route this traffic (according to the original A matrix, i.e. snapshot $k = 0$) and generate what would be the resulting SNMP link counts for the 3 month period under study. This last step is the same as the methodology presented in [6].

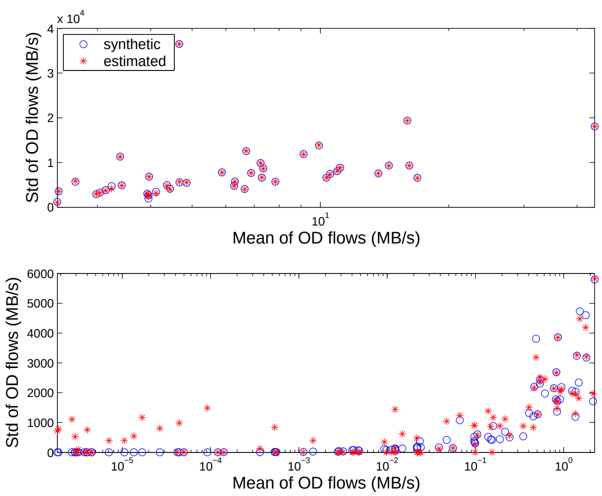

Figure 9: Real and estimated standard deviation for all OD pairs, large and medium (on the top) and small (on the bottom).

We compare our estimate of the standard deviation (std) to the standard deviation of the pseudo-real data in Figure 9. The top plot is for the medium and large flows, while the bottom plot includes the comparison for small flows. The variance estimate for the medium and large OD flows is quite good in that it achieves an average estimation error of less than 5%. As expected, it is harder to estimate the variance of the smaller OD pairs and we see the errors can span a large range. This challenge cannot be met by merely increasing the number of samples because it is due to the difference in order of magnitude of large and small OD pairs. As a consequence, a small error in the std-estimate of large OD pairs will be spread across multiple OD pairs causing large errors in the std-estimate of small OD pairs.

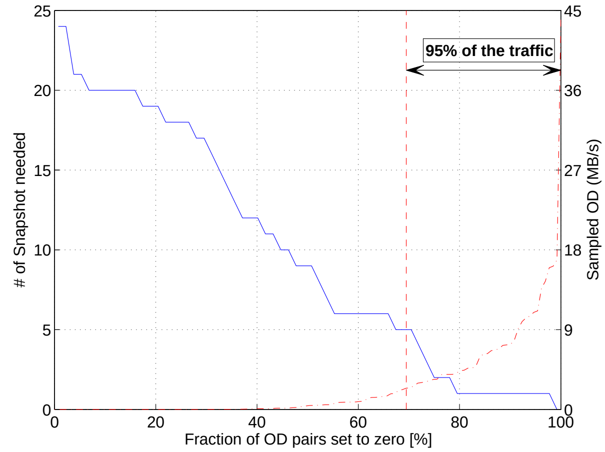

Figure 10: Number of snapshots needed as a function of OD pairs to be estimated.

We were not able to use Equation (15) as a metric for confidence of our variance estimator because this involves a detailed study for which space is lacking. Without a method for extracting exactly the top 30% largest OD pairs from the rest, we thus rely on a simple threshold scheme for now. We keep all OD pairs whose rates are above a threshold set to be two orders of magnitude lower than the largest OD pair. The OD pairs below this threshold are set to zero.

We are now ready to examine the impact of removing the small OD pairs on the number of snapshots required. Having isolated the top OD flows, our methodology next uses a heuristic algorithm to select the sequence of snapshots needed - however this time we seek identifiability on a much reduced set of OD pairs. Figure 10 depicts the number of snapshots required as a function of the number of OD pairs set to 0. Recall that as more OD pairs get set to zero, there are fewer variables to estimate and so we expect the number of snapshots to be decreasing. We start with 24 snapshots, when all OD pairs are considered by the heuristic, and end with 0 when no OD pair has to be identified. Note that only 5 snapshots are needed to disaggregate 30% of the largest OD pairs carrying 95% of network traffic. Recall that when we estimated the entire traffic matrix using Method 1, our algorithm requires 24 weight changes, or snapshots. Here we see that if we contend ourselves with estimating 95% of the load of the traffic matrix, then we can drop the number of needed snapshots to 5, i.e., an 80% reduction! This is indeed a dramatic decrease in the number of needed snapshots. With the number of snapshots so small, ISPs are more likely to be able to execute such a method. One of our key findings here is thus that by focusing only on large flows and being able to identify them, we can enable a method (the weight change method) that appears to be able to significantly improve the error rates as compared to previous methods.

Finally we evaluate the impact of removing the small OD pairs on the accuracy of estimation for the remaining top OD flows. When flows are "removed" (i.e., set to zero) for the purpose of our method, the traffic they carry still appears inside the SNMP link counts. Thus if we assume some OD flow has zero rate, then its actual rate will be transfered to some other OD pair retained in the estimation process. This will increase the inaccuracy in the flows being estimated.

Figure 11 shows an example of mean estimation for 6 top OD pairs as a function of the number of snapshots implemented. Setting small OD pairs to zero does not have a significant impact on the large flows. It can have an impact on the smallest of our medium flows (again OD #28 and #105 are the worst case flows in our top

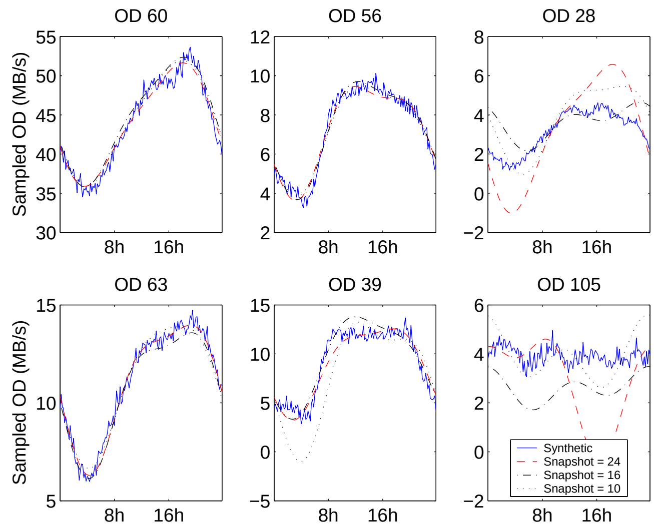

Figure 11: Method 2. Example of mean estimation for six OD pairs within top flows, as a function of number of snapshots carried out.

OD [%]	Snapshot=24		Snapshot=16		Snapshot=10	
	ME	*WC*	*ME*	*WC*	*ME*	*WC*
100%	3.79	28.04	8.96	57.31	10.88	58.40
95%	2.34	15.48	5.79	31.23	7.99	31.20
90%	1.72	9.30	4.54	22.51	6.78	24.85

Table 1: Average (ME) and Worst-Case (WC) relative errors of 1^{st} component of Fourier series for 100%, 95% and 90% of top OD pairs.

set).

We provide summary statistics for our errors as a function of the number of snapshots in Table 1. Clearly both the average relative error and the worst case error decline as the number of snapshots is reduced. For example, 90% of our top flows have an average error of under 2% when 24 snapshots are used, an average error under 5% when 16 snapshots are used. The average error climbs to 7% when only 10 snapshots are used. We point out that all of the average errors reported in this table are under the 10% barrier for target error rates.

9. CONCLUSIONS

In this paper we propose a new approach for dealing with the ill-posed nature of traffic matrix estimation. In previous approaches, additional constraints on the variance of OD flows were added into the estimation procedure to provide more information than simply the constraints on first order moments. To estimate the mean, both batches of constraints are solved together along with needed assumptions relating the mean to the variance. In our work we derive an estimate for the variance that is independent of the mean, and thus doesn't require assumptions about the mean and variance relationship. Our variance estimator is based on a new model for OD flows that we establish after studying a set of directly measured OD flows. This model explicitly incorporates both the diurnal patterns and the fluctuations behavior.

Armed with a variance estimator, we can address the ill-posed nature of the problem by using our variance estimates to identify the largest flows in the traffic matrix. This is useful because our data shows it is usually less than half of the traffic matrix elements that are responsible for carrying the majority of the traffic load. The complexity of the traffic matrix estimation problem is thus reduced because the number of variables to estimate has decreased. We use an algorithm to identify a sequence of IGP weight changes that should be applied to backbone links. By collecting additional SNMP measurements under these different routing snapshots, we can add new linearly independent equations into the basic linear system thereby increasing its rank. Increasing the rank of the system is the key to making the problem less under-constrained.

We thus make use of all of these mechanisms to propose a new methodology that incorporates the variance estimator, the algorithm for finding appropriate routing snapshots, and an inference technique as the final step. Both the variance estimator and the inference scheme are based on our OD flow model that incorporates two types of flow variability behaviors, namely a cyclo-stationary diurnal pattern and a stationary fluctuations process. We show that our OD flow models work well in that they accurately capture the temporal dynamics of Internet backbone traffic.

One of our key findings here is thus that by focusing only on large flows and being able to identify them, we can enable a method (the link weight change method) that is able to significantly improve the error rates as compared to previous methods. In all our test cases, the *average* errors stayed consistently below the 10% target carriers have suggested, and sometimes even reached as low as 1 or 2% (depending upon the scenario evaluated). We believe that this is the first proposed method to achieve average error rates in this range.

We show the tradeoff in accuracy of the estimates versus the number of snapshots used in increasing the rank. On the one hand, as more and more small flows are ignored, the errors in the estimates of the large flows increase. On the other hand, dropping more and more small flows means that fewer snapshots are needed to obtain a full rank system for the remaining flows of interest. The advantage of this is that it renders the weight-change approach more practical since only a few such changes may really be needed in an operational environment. For the backbone network considered here, this tradeoff is not too dramatic in that we can still achieve average error rates under the 10% barrier even when a limited number of link weight changes are carried out.

10. REFERENCES

[1] Y.Vardi, "Estimating Source-Destination Traffic Intensities from Link Data", *Journal of the the American Statistical Association*, 91(433), March 1996.

[2] C.Tebaldi and M.West, "Bayesian Inference of Network Traffic Using Link Count Data", *Journal of the the American Statistical Association*, 93(442), June 1998.

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

[4] A.Medina, N.Taft, K.Salamatian, S.Bhattacharyya and C.Diot, "Traffic Matrix Estimation: Existing Techniques Compared and New Directions", *ACM Sigcomm*, Pitsburgh, PA, August 2002.

[5] Y. Zhang, M. Roughan, N. Duffield and A. Greenberg, "Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads", *Proceedings of ACM Sigmetrics* San Diego, CA, June 2003.

[6] Y. Zhang, M. Roughan, C. Lund and D. Donoho, "An Information Theoretic Approach to Traffic Matrix Estimation", *Proceedings of ACM Sigcomm*, Karlsruhe, Germany, August 2003.

[7] Gang Liang, Bin Yu, "Pseudo Likelihood Estimation in Nework Tomography", *IEEE Infocom*, San Francisco, CA, March 2003.

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

[9] A. Feldmann, A. Greenberg, C. Lunc, N. Reingold, J. Rexford and F. True, "Deriving Traffic Demands for Operational IP Networks: Methodology and Experience", *IEEE/ACM Transactions on Networking* June 2001.

[10] H. Stark and J.W. Woods, "Probability, Random Processes, and Estimation Theory for Engineers", *Prentice-Hall*, Englewood Cliffs, New Jersey 07632.