Traffic Matrix Estimation: Existing Techniques and New Directions

A. Medina[a,b], N. Taft[a], K. Salamatian[c], S. Bhattacharyya[a], C. Diot[a]

[a]Sprint Advanced Technology Labs. Burlingame, CA.
[b]Department of Computer Science, Boston University. Boston MA.
[c]University of Paris VI. Paris, France.

ABSTRACT

Very few techniques have been proposed for estimating traffic matrices in the context of Internet traffic. Our work on POP-to-POP traffic matrices (TM) makes two contributions. The primary contribution is the outcome of a detailed comparative evaluation of the three existing techniques. We evaluate these methods with respect to the estimation errors yielded, sensitivity to prior information required and sensitivity to the statistical assumptions they make. We study the impact of characteristics such as path length and the amount of link sharing on the estimation errors. Using actual data from a Tier-1 backbone, we assess the validity of the typical assumptions needed by the TM estimation techniques. The secondary contribution of our work is the proposal of a new direction for TM estimation based on using *choice models* to model POP *fanouts*. These models allow us to overcome some of the problems of existing methods because they can incorporate additional data and information about POPs and they enable us to make a fundamentally different kind of modeling assumption. We validate this approach by illustrating that our modeling assumption matches actual Internet data well. Using two initial simple models we provide a proof of concept showing that the incorporation of knowledge of POP features (such as total incoming bytes, number of customers, etc.) can reduce estimation errors. Our proposed approach can be used in conjunction with existing or future methods in that it can be used to generate good priors that serve as inputs to statistical inference techniques.

1. INTRODUCTION

Traffic matrices (TM) reflect the volume of traffic that flows between all possible pairs of sources and destinations in a network. Estimation techniques based on partial information are used to populate traffic matrices because amassing sufficient data from direct measurements to populate a traffic matrix is typically prohibitively expensive. The term *Network Tomography* was introduced in [12] for the TM estimation problem when the partial data comes from repeated measurements of the traffic flowing along directed links of the network.

The knowledge represented by a traffic matrix is very valuable to a wide variety of *traffic engineering* tasks including load balancing, routing protocols configuration, dimensioning, provisioning and failover strategies. Information on the size and locality of flows is crucial for planning network growth and diagnosing problems. Because traffic matrices are not available to carriers today, they cannot quantify the cost of providing QoS as opposed to overprovisioning. Building such network-wide views is central to be able to efficiently engineer an IP backbone network.

Despite of the benefits that would be derived from having access to accurate traffic matrices for a network, very few techniques have been proposed for estimating them in the context of Internet. In this paper we focus on the traffic estimation problem in the context of backbone POP-to-POP topologies corresponding to Tier-1 carrier networks. In this case the nodes of the topology are POPs and the links correspond to the aggregated capacity of the links connecting any two given POPs. The data typically available for TM estimation are usually called *link counts* and in the context of the Internet, SNMP provides these data via incoming and outgoing byte counts computed per link every 5 minutes.

The primary contribution of this paper is to conduct a detailed comparative evaluation of the three main techniques that have been proposed in the literature to address the TM estimation problem. Our secondary contribution is to propose, based on the lessons learned from the comparative study, a new direction for TM estimation. We develop an initial model and perform a preliminary proof of concept evaluation based on real Internet traffic.

In the comparative study, the first approach studied is based on a straightforward application of Linear Programming (denoted as the *LP* approach hereafter) as proposed in [7]. The second uses Bayesian Inference techniques (called the *Bayesian* approach hereafter) as proposed in [11]. The third technique [3] is referred to as "Time-Varying Network Tomography" by the authors. We refer to this technique as the *EM* approach since the core of their approach is based on an *Expectation Maximization (EM)* algorithm to compute maximum likelihood estimates.

An important effort in the context of deriving traffic demands in an IP backbone is presented in [6]. This study addresses a slightly different problem in that their goal was to derive point-to-multipoint demands while we focus on point-to-point demands. The data used in [6] came from Netflow measurements, routing tables and routing configuration files. Their approach was an algorithmic one that uses this information to disambiguate the point-to-multipoint demands. The main focus of our paper is to evaluate the main statistical and optimization methods proposed to address the point-to-point traffic estimation problem.

In order to compare the three techniques that come from three

SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

different domains, we use a systematic approach to evaluate them within the same framework. We find that the existing techniques produce errors rates that are probably too high to be acceptable by ISPs. Furthermore, we show that decreasing such error rates is difficult in the face of very limited information and high sensitivity to noise in required prior information. We show that the marginal gain of being able to directly measure certain number of components of the traffic matrix is fairly small. This suggests that a new approach that allows the incorporation of additional knowledge about the network is needed. Indeed non-statistical knowledge about how networks are designed is available to network operators and should be combined with statistical data for achieving more accurate traffic matrix estimation.

Using data collected from a Tier-1 backbone, we demonstrate that the common statistical assumptions made with respect to the behavior of inter-POP flows generally do not hold for Internet traffic. New models are needed that can better capture the true nature of the volume of flow exchanged between a pair of POPs.

With this in mind, in the latter part of our work we propose a new direction for approaching the traffic estimation problem. We introduce the idea of using *choice models* to model POP *fanouts*. The selection of an egress POP for a given packet can be metaphorically viewed as a choice made by an ingress POP. Using the language of choice models, an ingress POP selects where to send a packet in order to maximize a utility. Such choices are made as a function of routing policies, where content and servers are located, etc. We attribute features to POPs, such as total capacity, number of customers/peers, etc. The combination of features at a particular egress POP can make it more or less attractive to a particular ingress POP. POPs that are more attractive will yield higher utility. We propose two inital choice models to provide a preliminary proof-of-concept of the applicability of this approach in the context of commercial network enviroments.

The paper is organized as follows. We describe the three estimation techniques in Section 2. In Section 3 we describe the comparison methodology we followed for the analysis. The results of the comparative analysis are presented in Section 4. In Section 5 we propose the use of choice models and describe the the theoretical aspects behind the derivation of a choice model. We define, calibrate and evaluate two initial choice models. Section 6 presents our concluding remarks.

2. TECHNIQUES EVALUATED

All three solutions are given using the same notation. Let c be the number of origin-destination (OD) pairs. If the network has n nodes, then $c = n*(n-1)$. Although conceptually traffic demands are represented in a matrix X, with the amount of data transmitted from node i to node j as element X_{ij}, it is more convenient to use a vector representation. Thus, we order the OD pairs and let X_j be the amount of data transmitted by OD pair j. Let $Y = (y_1, ...y_r)$ be the vector of link counts where y_l gives the link count for link l, and r denotes the number of links in the network. The vectors X and Y are related through an r by c routing matrix A. A is a $\{0, 1\}$ matrix with rows representing the links of the network and columns representing the OD pairs. Element $a_{ij} = 1$ if link i belongs to the path associated to OD pair j, and $a_{ij} = 0$ otherwise. The OD flows are thus related to the link counts according to the following linear relation:

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

The routing matrix in IP networks can be obtained by gathering the OSPF or IS-IS links weights and computing the shortest-paths between all OD pairs. The link counts Y are available from the SNMP data. The problem is thus to compute X, that is, to find a set of OD flows that would reproduce the link counts as closely as possible. The problem described by Equation (1) is highly underdetermined because in almost any network, the number of OD pairs is much higher than the number of links in the network, $r << c$. This means that there are an infinite number of feasible solutions for X.

In the case where there are several (K) measurement periods, we denote the link counts as Y_l^k, indicating the average load on link l in measurement period k, $k = 1, ..., K$. Similarly, we denote the traffic demands as X_j^k, indicating the traffic demand for OD pair j in measurement period k. The OD flows and link counts are related through A, as $Y^k = AX^k$.

The solutions proposed to date either fall into the category of optimization techniques [7] or statistical inference methods [12], [11] and [3]. The problem of determining Origin-Destination (OD) pairs based on statistical inference was pioneered in the context of networks by Vardi [12]. Vardi addressed the problem of identifiability and presented a general framework outlining a number of possible approaches along with their respective advantages and disadvantages.

In this paper we evaluate three methods. The optimization approach [7] poses the problem as a linear program (LP) and attempts to compute X directly. A reduction of the feasible solutions space is achieved by imposing linear constraints on X. The Bayesian [11] and EM [3] methods make assumptions about the distribution of X_j, i.e., modeling assumptions about OD flows. These methods use $E(X|Y)$ as their estimate for X. Thus the LP method uses the link counts Y as hard constraints, while the inference methods use Y to compute conditional distributions.

Both [12] and [11] assume that the OD pairs are generated from a collection of independent Poisson distributions; while [3] assumes that each OD pair follows a Gaussian distribution. Bayesian estimation is used in [11] to estimate the parameters of the Poisson distributions, while [3] relies on maximum likelihood estimation to estimate the Gaussian parameters. We consider that evaluating the *LP method* in the optimization category, and the Bayesian and EM methods on the statistical category allows us to evaluate three different algorithmic strategies (LP, Bayesian inference, EM algorithm), and three scenarios with respect to different assumptions made (none, Poisson, Normal). Following we give a brief summary of each of the three techniques we compared and due to lack of space we will refer the reader to the original papers ([7], [11],[3]) for more details.

2.1 Linear Programming

Because the traffic matrix estimation problem imposes a set of linear relationships described by the system $AX = Y$, the basic problem can be easily formulated using a Linear Programming model and standard techniques can be used to solve it. Knowing that the link count Y_l has to be the sum of all the traffic demands that use link l, the LP model is defined as the optimization of an objective function:

$$\max \sum_{j=1}^{c} w_j X_j \tag{2}$$

where w_j is a weight for OD pair j (examples of weights are discussed below). The objective function is subject to link constraints:

$$\sum_{j=1}^{c} A_{lj} X_j \leq Y_l \quad l = 1, ..., r$$

and flow conservation constraints:

$$\sum_{l=(i,j)} Y_l a_{lk} - \sum_{l=(j,i)} Y_l a_{lk} = \begin{cases} X_k & \text{if j = src of k} \\ -X_k & \text{if i = dst of k} \\ 0 & \text{otherwise} \end{cases}$$

and positivity constraints $X_j \geq 0 \quad \forall j$.

This is a deterministic model in which the link counts are viewed as hard constraints rather than as statistical data. This problem was first formulated as such in [7]. The choice of objective function influences the solution obtained. If we use a function that is the linear combination of all the demands, then we are trying to maximize the load carried on the network. In [7] the author argues that choices such as $w_i = 1 \quad \forall i$ will lead to solutions in which short OD pairs (i.e. neighboring nodes) will be assigned very large values of bandwidth while distant OD pairs (those with many hops between them) will often be assigned zero flow. Although such solutions are feasible, we know that these are not the solutions we are aiming to find. Therefore, [7] suggests that a good objective function is one in which the w_i reflect the path length. For small toy examples [7] demonstrated that this objective function does indeed yield the solutions one would want. We thus use that objective function in our evaluation of this method on backbone topologies.

2.2 Statistical Approaches

Figure 1 depicts a general diagram of the statistical approach.

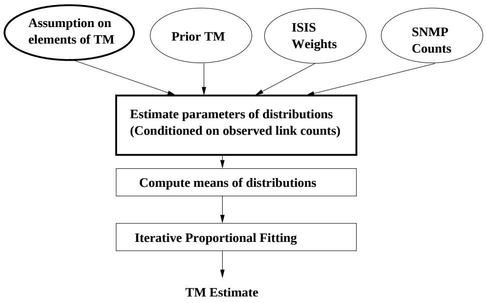
Assumption on elements of TM
Prior TM
ISIS Weights
SNMP Counts
Estimate parameters of distributions (Conditioned on observed link counts)
Compute means of distributions
Iterative Proportional Fitting
TM Estimate

Figure 1: General diagram for statistical approach

There are four general inputs to the statistical approaches. Although the assumptions made on the traffic demands are not actually an input, we may see them as influencing the specific statistical strategy to use. Statistical methods usually need a prior TM to get started. This important input may come from an outdated version of the traffic matrix, or an initial estimate obtained by some other mechanism. The ISIS weights are used to compute shortest paths which in turn generate the A matrix. The final input, SNMP data, gives the observed links counts Y. These inputs are used to impose constraints on the estimated traffic matrix.

Statistical approaches differ mainly on the statistical assumptions made on the components of the traffic matrix, and on the specific mechanisms used to estimate the corresponding parameters (the two darker boxes in Figure 1). Although, the EM method does not actually need a prior matrix in the sense that it may start with a random matrix as the initial point for the estimation process, a *bad* starting point may cause the method to get stuck in a local optimum.

Given the inputs, the first and main step of the estimation procedure is to estimate all the parameters of the distributions assumed for the traffic matrix components. This typically involves estimating Λ where $\mathbf{\Lambda} = \{\lambda_{\mathbf{1}}, ..., \lambda_{\mathbf{c}}\}$, denotes the vector of mean rates (i.e., each λ_j denotes the mean rate of OD pair X_j). (Other parameters can also be estimated if needed for a particular model.) Such estimates can be computed, for example, via an EM implementation of maximum likelihood estimation. The values assigned to the parameters will be conditioned on the SNMP counts observed on the links of the network. Once the parameters are obtained, the next step is to compute the conditional mean value for the distribution associated with each component of the traffic matrix. A final adjustment step is usually applied to the result from the previous step corresponds to an *iterative proportional fitting algorithm* (IPF). As described in [3], the IPF algorithm has been used extensively in the context of contingency tables. The idea is to express the linear constraints given by Equation (1) using a contingency table composed of the estimated traffic matrix and an extra value for each row and each column, corresponding to the row and column sums respectively. The IPF algorithm proceeds to adjust the values of the estimated traffic matrix such that the error with respect to the row and column sums is minimized. The convergence of the IPF algorithm can be proved [4].

2.2.1 Bayesian Approach

In the Bayesian approach, as proposed in [11], the goal is to compute the conditional probability distribution, $p(X|Y)$, of all OD demands X given the link counts Y. To achieve this goal we need to have a prior distribution for X, namely $p(X)$. In [11] it is assumed that $p(X_j)$ follows a Poisson distribution with mean λ_j, that is, $X_j \sim$ Poisson(λ_j), independently over all OD pairs. (Recall that $\mathbf{\Lambda} = \{\lambda_{\mathbf{1}}, ..., \lambda_{\mathbf{c}}\}$ denotes the vector of mean rates.) Since the set $\mathbf{\Lambda}$ is unknown and needs to be estimated, we need to define a prior for $\mathbf{\Lambda}$, which leads us to a joint model given by $P(X, \mathbf{\Lambda})$. The idea is then to observe the link counts and condition on them to obtain the conditional joint distribution given by $P(X, \mathbf{\Lambda}|\mathbf{Y})$.

Obtaining posterior distributions is computationally very hard. The approach adopted by [11] is to apply iterative simulation methods such as Markov Chain Monte Carlo (MCMC). Simulating a distribution means to draw a large number of samples to represent a complete histogram of the desired distribution.

Note that the ultimate goal is to compute $P(\mathbf{X}|\mathbf{Y})$. The aforementioned iterative simulation mechanism, relates the posterior we want to obtain to the joint posterior distribution that involves both X and $\mathbf{\Lambda}$ by the following equation:

$$p(\mathbf{X}|\mathbf{Y}) = \sum_{\mathbf{\Lambda}} \mathbf{P}(\mathbf{X}|\mathbf{\Lambda}, \mathbf{Y}) \qquad (3)$$

The problem at hand is thus to compute the posteriors $P(\mathbf{\Lambda}|\mathbf{X}, \mathbf{Y})$ and $p(\mathbf{X}|\mathbf{\Lambda}, \mathbf{Y})$. The simulation procedure starts with a prior matrix X^0 and the following iteration is performed:

1. Draw value of Λ^i from $p(\mathbf{\Lambda}|\mathbf{X^i}, \mathbf{Y})$

2. Using this Λ^i, draw a value of $\mathbf{X^{i+1}}$ from $p(X|\mathbf{\Lambda^i}, \mathbf{Y})$

3. Iterate until feasible solution is found. Feasibility equates a positivity constraint on the elements of X

[11] takes advantage of some useful structural properties of the problem formulation. The columns of A can be reordered with the form $A = [A_1 A_2]$ where A_1 is a nonsingular $r \times r$ matrix. Similarly reordering the elements of $\mathbf{X}$, we can rewrite $\mathbf{X} = [\mathbf{X_1 X_2}]$.

Then it can be proved that $\mathbf{X_1} = \mathbf{A_1^{-1}}(\mathbf{Y} - \mathbf{A_2}\mathbf{X_2})$ [11]. This implies that given Y and estimates for the $c-r$ OD pairs in vector $\mathbf{X_2}$, the remaining r OD pairs in $\mathbf{X_1}$ can be computed by straightforward algebra. The reordering of the columns of A and the related OD pairs of $\mathbf{X}$ can be done by standard QR decomposition methods [9]. This indicates that inferences only need to be made on a subspace of dimension $c - r$ rather than c. Rather than simulate $p(\mathbf{X}|\mathbf{\Lambda}, \mathbf{Y})$, it is thus enough to simulate $p(\mathbf{X_2}|\mathbf{\Lambda}, \mathbf{Y})$.

We implemented this technique using the code generously given to us from one of the authors. The technique as originally proposed does not make use of an IPF algorithm at the end. We have added this step as we found that it does give small improvements in the final estimates.

2.2.2 The EM Method

In this approach the OD pairs are modeled according to a Gaussian distribution, $X \sim \text{Normal}(\lambda, \Sigma)$, where the X_j are modeled as independent normal random variables. Because of the relation $AX = Y$, the Gaussian assumption on X implies that $\mathbf{Y} \sim \text{Normal}(\mathbf{A\Lambda}, \mathbf{A\Sigma A}^{'})$ where $\mathbf{\Lambda} = (\lambda_1, ..., \lambda_c)$ is again the vector of mean rates of the OD pairs, and the covariance matrix is given by $\mathbf{\Sigma} = \phi \quad \text{diag}(\lambda_1^b, ..., \lambda_c^b)$. Note that a specific structure is assumed regarding the relationship between the mean and the variance. In particular, it is assumed that they are related through $\Sigma_j = \phi\lambda_j^b$ where ϕ needs to be estimated along with the λ_j's. In [3] the authors showed that $b = 2$ fits the sample data from their LAN network well, and thus they use this value throughout their study.

The important extension that this method accomplishes over the previous two is that it incorporates multiple sets of link measurements. Let $\mathbf{y_1}, ..., \mathbf{y_K}$ denote a consecutive set of K SNMP measurements. This method assumes that these measurements correspond to iid random variables. Let $\theta = (\mathbf{\Lambda}, \phi)$ represent the set of parameters that we want to estimate. The maximum likelihood estimate is computed by finding the maximum of the following log-likelihood function:

$$l(\theta|\mathbf{y_1}, ..., \mathbf{y_K}) = \qquad (4)$$

$$-\frac{K}{2}\log|A\Sigma A^{'} - \frac{1}{2}\sum_{k=1}^{K}(y_k - A\Lambda)^{'}(A\Sigma A^{'})^{-1}(y_k - A\Lambda)$$

The method by Cao, *et. al.* [3] has the following steps. First, an Expectation Maximization (EM) algorithm is used to compute the estimate for θ, namely $\hat{\theta} = (\hat{\Lambda}, \hat{\phi})$. The EM algorithm is a broadly applicable algorithm that provides an iterative procedure for computing MLE's in situations where maximum likelihood estimation is not straightforward due to the absence some of data. EM algorithms require the existence of a prior to initiate the iterative procedure. Second, each OD pair j is estimated at time t by $\hat{X}_{j,t} = E[X_{j,t}|\hat{\theta}, Y]$.

In the third and final step, and iterative proportional fitting (IPF) algorithm [8] is applied. We implemented this method using MATLAB and its optimization toolbox.

3. COMPARISON METHODOLOGY

It is not possible to obtain an entire "real" traffic matrix via direct measurement. Therefore assessing the quality of TM estimations and validating TM models is difficult because one cannot compare an estimated TM to "the real thing". Indeed if it were possible to obtain real TMs, the inference problem would disappear.

Each test case is defined by a topology and a synthetic traffic matrix. We generate synthetic traffic matrices with entries $\mathbf{X_{ij}}$ denoting the average daily flow from POP i to POP j. We use the IS-IS weights to determine the shortest path routing for each pair of POPs. This defines the entries for the routing matrix A. The demands $\mathbf{X}$ are routed on the network according to A which determines the resulting load on each link. This link count data can be obtained simply by $Y = AX$. We give as input to each method the link counts, Y, and the routing matrix A. We can then compare the estimated TM, $\hat{\mathbf{X}}$, to the original TM $\mathbf{X}$ used to produce the link counts.

As mentioned earlier, the Bayesian approach only uses SNMP data from a single measurement interval, while the EM approach can use a window of such measurements. In order to conduct a fair comparison, we evaluate the EM technique using a window of size 1 for the majority of our tests. We also experiment with the EM technique using a larger window to assess the improvement that using multiple measurements can bring.

We compare these techniques with respect to the estimation errors yielded, sensitivity to prior information, sensitivity to modeling assumptions on OD pairs, and sensitivity to path length and link sharing on estimation errors. Using monitored backbone data, we assess the validity of the modeling assumptions made on OD pairs.

3.1 Topologies

We consider two topologies in our test cases. We first consider a small 4 node topology, depicted in Figure 2. We start with this simple scenario because it allows us to enumerate all the node pairs and link counts, which is useful for illustrating how the methods behave. The values on the links in this figure represent the link counts. We then evaluate the three methods on a large 14-node topology that closely corresponds to our Tier-1 backbone POP-to-POP topology. We focus on this topology since it represents a real commercial one. In this topology, each node represents a Point-of-Presence (POP) and each link represents the aggregated connectivity between the routers belonging to a pair of adjacent POPs (i.e., inter-POP links). For proprietary reasons, the numbers on the links represent hypothetical IS-IS weights.

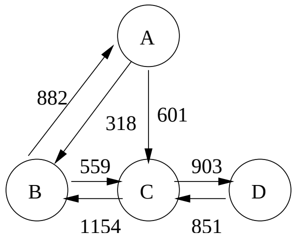

Figure 2: 4-Node Topology

3.2 Synthetic Traffic Matrices

One cannot avoid using synthetic traffic matrices since real ones aren't available. The best one can do it to generate synthetic TMs based on what one believes properties of real TMs to be. We also need to generate synthetic TMs that exhibit certain properties that expose the strengths and the weaknesses of the evaluated techniques. With this goal in mind, we generate five types of synthetic TMs that differ in the distribution used to generate their elements. Specifically, we consider *constant*, *Poisson*, *Gaussian*, *Uniform* and *Bimodal* TMs. It is required to try a variety of OD pair distributions because all statistical methods make some assumption about the distribution for the OD traffic demands. It is important to understand the sensitivity each method to the assumptions made.

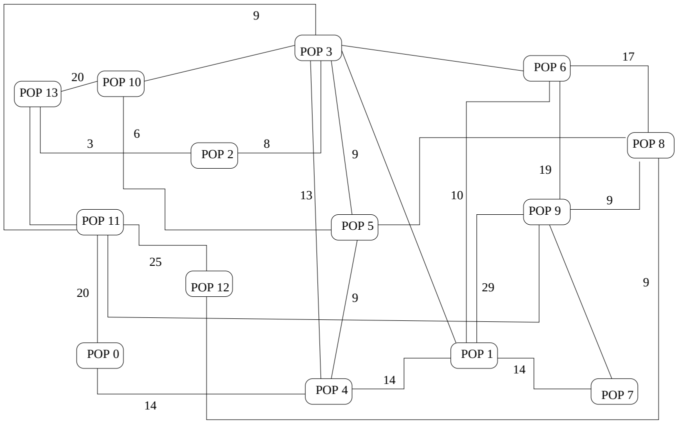

Figure 3: 14-Node Tier-1 POP Topology

Since the Bayesian technique relies on the Poisson assumption, in theory this method should perform best if the OD pairs follow a Poisson distribution. We can then ask how sensitive it is to this assumption by evaluating its performance using other distributions. Similarly, the EM approach assumes a Gaussian distribution and we want to examine its performance under both Gaussian and non-Gaussian scenarios. We consider the Uniform case because this is what is often used by researchers who need a traffic matrix in evaluating routing algorithms. We include the Constant case because this should be the easiest for these methods to estimate.

We suspect that these distributions do not properly reflect the distributions of OD traffic demands in the Internet backbone. Recent work [2] has shown that the Internet has "hot spot" behavior; i.e., a few POP pairs have very large flows, while the majority of POP pairs have substantially less flow between them. Thus, we include the bimodal distribution with the goal of having this style of relation between OD traffic demands.

Constant TMs are obtained by assigning a constant value, 300 Mbps, to all OD pair demands. *Uniform* TM were obtained by generating random values uniformly distributed in the interval $[100, 500]$. *Poisson* TMs were obtained by generating the parameters λ_i for each OD pair i uniformly from an interval $[100, 500]$. Next, random values are drawn for each $X_i \sim$ Poisson(λ_i). *Gaussian* TMs were obtained by generating μ_i uniformly distributed in the interval $[100, 500]$ and assigning the same variance for all OD pairs, $\sigma_i = 40$. Next, random values are drawn for each $X_i \sim N(\mu_i, 40)$. Finally, *Bimodal* TMs are generated by a mixture of two Gaussians, one with $(\mu_1 = 150, \sigma_1 = 20)$, and the second with $(\mu_2 = 400, \sigma_2 = 20)$. For each OD pair, a biased coin is flipped for which the probability of heads is 0.8. If the coin is heads, then a random number is generated according to the first Gaussian (μ_1, σ_1), and otherwise the second Gaussian is used.

For the smaller topology, we only use the Poisson and Gaussian traffic matrices. For the larger topology we study all 5 cases of traffic matrices.

4. RESULTS

Section 4.1 discusses the results for the small 4-node topology, and Section 4.2 discusses the results for the 14-node topology.

4.1 Small Topology

For the purpose of this illustrative example, we only use one synthetic TM per method. For the LP and Bayesian methods we used a Poisson traffic matrix, and for the EM method we used a Gaussian traffic matrix (TM). Weights were set to 1 on each link, and routes were computed according to shortest hop count.

The results for the three methods are presented in Table 1. The table shows the original traffic matrix, the estimated value for each OD pair and the relative error. The average error was 98% for the LP method, 13% for the Bayesian method, and 7.6% for the EM method. The LP method clearly does substantially worse than the other two. Notice that the LP method assigns zero to a number of OD pairs. It seems to try to match OD pairs to link counts. For example, it assigns 601 to the OD pair AC. Although this matches the link count for the link AC, we know that the link AC also carried traffic from OD pair AD. Similarly, we see that the LP method assigned a bandwidth value for 851 to the OD pair DB, which matches the link count on link DC. However the link DC carries traffic from OD pairs DB, DC and DA. Note that DA and DC are assigned zero in order to allow DB to match the link count. Because assigning zero to some node pairs implies an error of 100% for them, and since setting some node pairs to match link counts can generate high relative errors, the average error rate is very large.

The objective function chosen, as suggested in [7], sets the weights equal to the hop counts of the corresponding OD pairs. These weights may avoid the zero-assignment problem if a maximization objective is selected. Although this approach may have worked in the 3-node topology considered in [7], it does not seem to be effective in our 4-node topology. Because LPs are indeed sensitive to the objective function, we tried two other objective functions. One alternative is to minimize the errors (i.e., $\min Y - A\hat{X}$), and the other alternative is not to use any objective function at all (since satisfying the link counts may be achieved via the linear constraints constraints imposed on X). Both of these alternative objective functions produced exactly the same estimates.

For both the Bayesian and EM methods, half of the estimates are over-estimates (the estimation for a given OD pair is larger than the actual TM value), and about half are under-estimates. For this small topology the EM method performs better than the Bayesian method in terms of both the average error and the worst case error. It is interesting that the worst case errors for the two statistical methods do *not* correspond to the same OD pairs. The Bayesian method makes its two biggest errors for OD pairs CB and CD, while the EM method makes its two biggest errors for OD pairs AD and DB. The link CB carries the largest number of OD pair flows, namely four, among all the links in the small network. We see that the Bayesian method makes its worst error for the OD pair CB. This hints that estimation errors may be correlated to heavily shared links. We will explore this further for the larger topology.

4.2 Tier-1 POP-to-POP topology

Synthetic prior matrices, X', are generated by adding white noise to each element of the original TM. That is, $X_j^{'} = X_j + \epsilon$ where $\epsilon \sim N(0, 60)$. To test the sensitivity of the Bayesian and EM methods to the goodness of the prior matrix, we consider another prior matrix that adds larger errors to the TM, in particular $\epsilon \sim N(0, 100)$. To facilitate the discussion below, we refer to this prior as the *bad prior* and the previous one as the *good prior*.

The results in this section are presented in a series of tables with the following format. Each entry is expressed as a fraction; i.e., it should be multiplied by 100 to get a percentage value. In Tables 3 - 6 the '0.2' column denotes the fraction of OD pairs whose error is less than 20%; similarly for the 0.5 and 0.7 columns. The results for the LP method are given in Table 2. In this table the errors are so large that we instead indicate columns of 0.8, 1.0 and 1.5, since

Original TM	*LP*		*Bayesian*		Original TM	*EM*	
(*Poisson*)	Estimated TM	Error(%)	Estimated TM	Error(%)	(*Gaussian*)	Estimated TM	Error(%)
AB: 318	318	0	318	0	318.65	318.65	0
AC: 289	601	107	342	18	329.48	286.98	13
AD: 312	0	100	259	17	277.18	318.36	15
BA: 294	579	96	334	14	298.14	298.14	0
BC: 292	559	91	310	6	354.81	360.97	1.6
BD: 267	0	100	249	7	355.39	347.94	2
CA: 305	303	0.6	291	5	327.20	317.34	3
CB: 289	0	100	361	25	330.04	373.65	13
CD: 324	903	178	395	22	253.01	217.32	14
DA: 283	0	100	257	9	320.50	329.07	3
DB: 277	851	207	245	12	291.52	246.60	15
DC: 291	0	100	349	20	310.40	344.82	11

Table 1: Estimates for 4-node Topology

most of the errors are over 100%.

With the LP method the average errors are quite large, ranging from 170% to 200%. The problem of matching OD traffic demands to link counts is also observed in this case. It is interesting to note that the uniform TM presents the biggest difficulty for the LP method. This isn't surprising based on our observations in the small topology. Because the LP method assigns many OD pairs equal to zero, it overcompensates by assigning many other OD pairs very large values. This spread of very small and very large values (i.e., as large as the entire link count) clearly does not match a distribution in which the ensemble of OD pairs are evenly distributed throughout the bandwidth range. *Because the errors with the LP method as so high, it could not be used in practical networks* and thus we do not continue with with additional analysis of this method.

Dist	Avg	Max	.8	1.0	1.5
`Constant`	1.70	12.0	0.03	0.03	0.84
`Uniform`	2.08	24.0	0.05	0.07	0.85
`Poisson`	1.73	12.8	0.03	0.05	0.84
`Gaussian`	1.74	12.4	0.03	0.05	0.84
`Bimodal`	2.01	19.0	0.05	0.06	0.85

Table 2: LP Approach for 14-node Topology

The results of applying the Bayesian inference technique to estimate traffic demands on our POP-to-POP topology are given in Table 3. Since this method assumes the OD pairs are distributed according to a Poisson distribution, we would expect the method to perform best when this assumption holds, i.e. for synthetic Poisson TM. The method performs essentially the same regardless of the OD pair distribution, except for the bimodal case. Recall that a 0.90 in the '0.5' column means that 90% of the OD pair estimates had an error of less than 50%. With the exception of the bimodal distribution, the fact that the method is is not sensitive to the distribution can be viewed as an advantage as it makes the method more robust to various types of OD pair distributions. It is interesting to note that the bimodal distribution causes more difficulty. We believe this occurs because the Bayesian technique is trying to target average values in its estimates, and thus it will miss the two modes of the bimodal distribution.

Table 4 shows the results of the Bayesian method with the bad prior. With this prior, the average error is roughly 50% worse than with the good prior. This shows that the Bayesian inference method is indeed very sensitive to the prior matrix.

We ran the same set of tests for the EM method (using a window of size 1). The results with the good prior are given in Table 5, and the results with the bad prior are in Table 6.

Dist	Avg	Max	.2	.5	.7
`Constant`	0.20	1.16	0.60	0.92	0.97
`Uniform`	0.26	2.31	0.58	0.83	0.91
`Poisson`	0.23	1.99	0.57	0.90	0.94
`Gaussian`	0.23	1.78	0.59	0.88	0.94
`Bimodal`	0.41	5.00	0.41	0.76	0.87

Table 3: Bayesian. 14-node topology. Good prior.

Dist	Avg	Max	.2	.5	.7
`Constant`	0.41	2.13	0.41	0.75	.87
`Uniform`	0.43	5.55	0.43	0.71	0.84
`Poisson`	0.37	2.83	0.38	0.76	0.86
`Gaussian`	0.41	5.26	0.41	0.79	0.89
`Bimodal`	0.63	4.97	0.29	0.56	0.69

Table 4: Bayesian. 14-node topology. Bad prior.

Dist	Avg	Max	.2	.5	.7
`Constant`	0.12	0.54	0.81	0.99	1.00
`Uniform`	0.13	1.07	0.75	0.96	0.98
`Poisson`	0.11	0.42	0.81	1.00	1.00
`Gaussian`	0.14	0.57	0.71	0.98	1.00
`Bimodal`	0.22	0.90	0.82	0.89	0.96

Table 5: EM. 14-node topology. Good prior.

We see that the EM method shows a substantial improvement over the Bayesian method. With the good prior, the EM has a 14% average error while the Bayesian method has a 27% average error (averaged over the 5 cases examined). With a bad prior, the EM method achieves a typical average error of 26% while the Bayesian method achieves a typical average error of 45%. In comparing Tables 5 and 6, we see that the EM method is also quite sensitive to the prior. With the bad prior, the errors are a bit less than double the errors of the test cases with the good prior. Again, with the exception of the bimodal distribution, the EM method does not appear to be very sensitive to the assumed distribution for OD pairs.

One of the reasons why the Bayesian approach seems to perform worse than the EM approach is that the convergence of the MCMC algorithm is stochastic. Consequently, there is no guarantee to make actual improvements on the quality of the estimations from one iteration to the next. In contrast, the EM method (because it is analytic) has a deterministic convergence behavior, meaning that we are guaranteed to make some improvement after each iteration of the algorithm.

Dist	Avg	Max	.2	.5	.7
Constant	0.22	1.00	0.53	0.90	0.96
Uniform	0.24	1.01	0.57	0.85	0.92
Poisson	0.23	1.28	0.55	0.88	0.95
Gaussian	0.24	1.11	0.48	0.86	0.95
Bimodal	0.39	1.50	0.42	0.66	0.80

Table 6: EM. 14-node topology. Bad prior.

The cases we have analyzed so far consider only a snapshot of the network at a single measurement interval. We retested the EM method using a window size of 10, to take advantage of multiple measurement intervals. Table 7 shows the results. The constant case is not shown because it is not affected by incorporating multiple measurement intervals. As can be seen, in general there is only a slight improvement of about 2%. We further experimented by increasing the window size to 50 and the results were similar to those with a window size of 10. This indicates that after a certain point, more measurements do not provide additional gain. *These results indicate that the time-varying extension provides only a small improvement.*

Dist	Avg	Max	.2	.5	.7
Bimodal	0.37	1.93	0.38	0.69	0.84
Gaussian	0.22	0.95	0.52	0.92	0.98
Poisson	0.22	0.98	0.56	0.90	0.96
Uniform	0.28	1.70	0.49	0.83	0.91

Table 7: Time-Varying EM. 14-node topology. Bad prior.

4.3 Marginal Gains of Known Rows

All of the studied methods must provide a solution to the traffic estimation problem based on only partial information. This partial information ultimately limits how much can be inferred. It is thus interesting to ask the following question: what improvement be gained if we could measure some components of the traffic matrix directly? For example, suppose we could measure one row of the traffic matrix, then we no longer would have to estimate the OD pairs in that row, and the number of variables to be estimated would be reduced by n. We evaluated the benefit, or marginal gain, of having known rows by using these known values as additional constraints, and by seeing how the estimation improves when only a subset of all the OD pairs need to be estimated. This question is of interest for the following reason. During the planning and evolution of a network, a carrier may have the option at some point to deploy a certain amount of monitoring equipment. One could ask how many boxes are needed to substantially bring down the error rates. It is in this spirit that we explore the marginal gains described above.

We studied the case of the 14-node topology with the Poisson traffic matrix. We considered four cases: each of the priors (good and bad) for each of the methods (Bayesian and EM). We present the results from two of those cases (Bayesian with good prior, EM with bad prior) for compactness of presentation and because they illustrate all salient findings. The addition of rows is done in three different orders: (1) random; (2) row sum; and (3) error magnitude. By "row sum" we mean that we computed row sums on our synthetic traffic matrices. (In practice this could be computed by summing the SNMP counts from all inter-POP backbone links exiting a POP.) The row sum indicates the total volume of traffic output by a POP. The POPs are ordered from largest to smallest and the corresponding rows are added in that order. The error magnitude ordering was done as follows. For each row, we computed the maximum estimation error across all row elements (i.e., all OD pairs for a given source POP). We sorted the rows from largest to smallest error, and added them accordingly. Intuitively we would expect the overall errors to be brought down quickly at first by adding information about those OD pairs with large errors, and then more slowly as information about OD pairs with small errors is added.

The results from these experiments are given in Figure 4 and Figure 5. These figures plot the average error over all OD pairs versus the number of known rows. We observe three things. First, in both plots the error rate drops off roughly linearly with each additional row added. Second, the Bayesian technique does not seem to be sensitive to the order in which the rows are added. However, the EM technique does perform better when the rows are added according to largest-error-first ordering. Third, in this topology each row contains 13 OD pairs, about 7% of the 182 total. Therefore, providing a complete row of measured values corresponds to adding 7% of the components with exact values and we would hope that the error rate should be reduced by roughly 7%. The reduction in average error that is achieved by adding one additional row is approximately 2%. This result could be interpreted two ways. On the one hand, this implies that making the effort to measure a few specific rows of the traffic matrix may not yield commensurate benefits. On the other hand, if the long term goal is to achieve error rates of say between 5-10%, and current techniques can yield 20% errors, then each percentage improvement is meaningful.

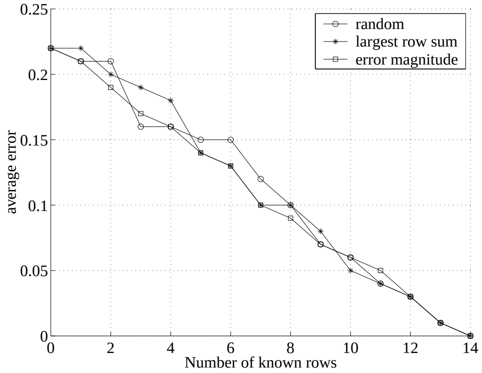

Figure 4: Marginal gains of known rows - Bayesian

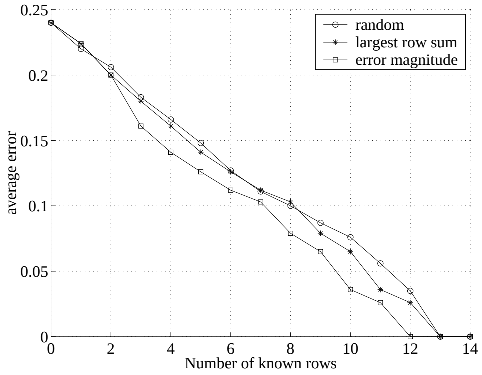

Figure 5: Marginal gains of known rows - EM

4.4 Which OD Pairs are Most Difficult to Estimate?

The results from both the small and large topologies reveal that all of these methods exhibit the following behavior. For a given traffic matrix, there are always some OD pairs that can be estimated very closely, while at the same time, other OD pairs are estimated very poorly. The errors of the poor estimates can be more than 10 times bigger than the errors of the good estimates. In all the test cases we saw that the minimum error is always zero while the maximum error is usually over 100%. It is thus natural to ask whether some OD pairs are more difficult to estimate than others. It this were true, it would be interesting to investigate the properties of paths associated with such "troublesome" OD pairs.

With this in mind, we explore the relationship between the average errors and two topological properties: (1) the length of the path for an OD pair; (2) the number of OD pairs sharing a link. We call the second property the *maximum link sharing factor*, and it is obtained as follows. For each link, the number of OD pairs whose path traverses that link is computed. For each OD pair, we examine all the links in its path and record the link with the maximum amount of sharing. If OD pair j has a max-link-sharing factor of 23, then the most shared link in path j is used by 23 OD pairs. It is intuitive to hypothesize that it would be hard to estimate OD flows for OD pairs that traverse heavily shared links. If a link is shared by a large number of OD pairs, then it may be hard to disambiguate how much bandwidth belongs to each OD pair.

In order to generate enough data to obtain meaningful averages, we did the following. We considered both the Bayesian and the EM methods on the 14-node topology and generated 10 random traffic matrices according to the Poisson and Gaussian test cases. Since each TM has 182 OD pairs, this gives us error rates for 1820 OD pairs. The path lengths of all OD pairs in the topology range from 1 to 5. We grouped all the 1820 OD pairs into 5 bins, one for each possible path length and computed the average error for all OD pairs in each bin. This data is presented in Figure 6.

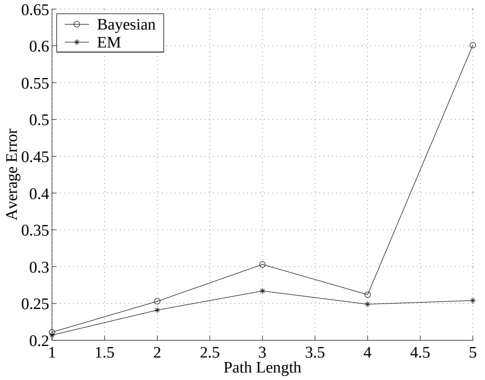

Figure 6: Average Errors and Path Lengths

Overall, we see that the Bayesian method is more sensitive to path length than the EM method; however the difference is small except for paths of length 5. This figure can be interpreted as follows. For example, with the Bayesian method paths of length 1 have a 20% error rate while paths of length 3 have a 28% error rate. We could thus say that paths of length 3 are 40% $(28-20)/20$ more difficult to estimate than paths of length 1. For the EM method paths of length 3 are roughly 26% more difficult to estimate than paths of length 1. The dip in the curve for paths of length 4 may be due to the fact that there were about twice as many OD pairs with path length 3 as OD pairs with path length 4. The sensitivity of the Bayesian method becomes huge for paths of length 5, whereas the sensitivity of the EM method does not continue to increase appreciably beyond paths of length 3.

The most "popular" link is shared by 33 OD pairs. The minimum amount of sharing is 1 OD pair. To examine the relation between average error and the maximum link sharing factor, we used the same 1820 OD pairs as above. We grouped these 1820 OD pairs into 6 equally sized bins over the range [1,33] (i.e., [1,5.5),[5.5-11), etc). The first bin contains all the OD pairs whose max-link-sharing factor is between 1 and 5.5, and so on. We compute the average error of all the OD pairs in each bin. The results are shown in Figure 7. *We observe that the average error is increasing as the amount of link sharing increases.* An OD pair whose maximally shared link is shared by 20 OD pairs can have an average error twice as big as another OD pair whose maximally shared link is shared by 5 OD pairs. The Bayesian method exhibits a slight drop in the average error rate as the sharing factor exceeds 20. This again may be a statistical error caused by the fact that only 90 out of 1820 OD pairs had sharing factors in the largest bin.

Note that if we compare the sensitivity of the EM method to path length and link sharing, we conclude that the EM method is more sensitive to link sharing than to path length. As the path length increased from 1 to 5, there was only an overall change in error of roughly 6%; while when the link sharing factor went from 1 to 33, the average error increased by roughly 15%.

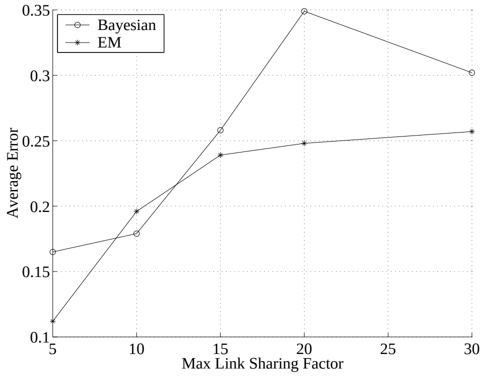

Figure 7: Average Errors and Heavily Shared Links

4.5 Validation of Statistical Assumptions

There are three key statistical assumptions made by the two inference methods. In this section, we verify whether or not these assumptions hold for actual Internet traffic. These three assumptions are (1) that OD pairs are Poisson; (2) that OD pairs are Gaussian; and (3) that the mean and variance are related according to the function $\Sigma = \phi\lambda^b$ where ϕ and b are constants. These methods also assume that the OD pairs are independent of each other. We did not have appropriate data to verify the independence assumption.

To validate these three assumptions, we obtained packet-level traces from a large scale monitoring project at a Tier-1 carrier network [2]. We used data collected at 3 different POPs on September 5, 2001. Our dataset included 5 inter-POP links from each of the 3 POPs. The traces from each of the 15 links were 12 hours in length. Using BGP tables collected from each of the 3 POPs we were able to construct 3 rows of the POP-to-POP traffic matrix. This was done by examining the destination address of each packet, doing a lookup in the BGP table, and mapping the next hop address to an egress POP. Because we only used traces and BGP tables within a single provider's domain, we could use the next hop address to identify the egress POP within the provider's domain. This approach may not be used to construct the entire traffic matrix as it is much too costly to monitor all backbone links in all POPs. With 3 rows, and 14 POPs per row, we now have the traffic flows for 39 OD pairs (we don't include traffic from a POP to itself). Our packet traces include GPS time stamps in each packet header allowing us to compute the rate of traffic flowing from one POP to another at fine time granularities. To assess the representativeness of the 5 measured inter-POP links per POP, we compared the total amount of data flowing through each POP (as given by SNMP statistics) to the total data flowing on the 5 measured links. We found that the measured data represents approximately 36% of the total POP traffic. Given the high amounts of traffic traversing through backbone high-speed links, we may argue that the characteristics of the fanouts for the three POPs as exhibited by the data collected are fairly representative of the overall behavior of each of the POPs.

Poisson assumption. To check the Poisson assumption we computed the traffic demands for each OD pair averaged over 1 second intervals. Although we have 12-hour long traces, we examined this assumption on 1-hour segments since it is known that flows do not exhibit stationarity over periods as long as 12 hours. Each 1-hour trace segment yields approximately 4000 samples of the traffic demands for a given POP pair. We used quantile-quantile plots to compare the quantiles of these data to the quantiles of a Poisson distribution with the same mean. We did this for the 39 measured OD POP pairs, and for many POP pairs we considered 3 or 4 different 1-hour segments. Due to space restrictions we include only a few graphs (Figures 8 - 10) that suffice to illustrate the main results we found.

Figures 8 and 9 show the quantile-quantile plots for the traffic from POP 8 to POP 10 for two different 1-hour periods. We see that in one hour (Figure 9) the fit appears reasonable while in the other hour (Figure 8) the fit appears quite poor. This illustrates our first observation that the goodness of the fit for such plots can depend upon the particular hour considered, even for the same pair of POPs. We found this same type of result for many other POP pairs, and thus conclude that the validity of this assumption is not consistent over different hours. We also give the same type of plot for the traffic traveling from POP 11 to POP 2. We see here a different shape for the match. In this figure, the match is ok for a portion of the distribution and does less well in the tail. This is not surprising. We asked ourselves at what point in the tail do the distributions start to diverge. Recall that the q-th quantile at a point (x, y) in the plot means that $Pr(\text{true-data} < x) = Pr(\text{observed-data} < y) = q$. In Figure 10 the data points diverge from the straight line for values of $q \geq 0.9$. We found that this divergence in the tail can happen anywhere from 90% to 98% depending upon the POP pair and the hour considered.

Gaussian assumption. We repeated the same procedure to compare the quantiles of the data for the POP pairs against the quantiles of a Gaussian distribution with similar mean and variance. In this case we make the plots easier to read by normalizing the data according to $x' = (x - \mu)/\sigma$. We can thus compare the observed data to a standard normal with a mean of 0 and a standard deviation of 1. The results from 3 representative POP pairs are provided in Figures 11 - 13. One of these plots, namely Figure 12 shows an excellent match. These other two would generally not be considered a good fit. We conclude that for some POP pairs the Gaussian assumption may be just fine, but for others it does not work well.

Note that the QQplot tests for Gaussianity in [3] yielded a closer fit for two reasons. First, their test was done on the Y data, and not the X data. More importantly, they studied a 7 node local-area network (LAN). We are studying a 14-node wide-area network (WAN) in which many of the POP pairs traverse the entire continental US. This indicates that the statistical assumptions that that may be made for local or metropolitan area networks (MAN) can be different to those that are legitimate for long-haul backbone networks.

QQ-plots are only one method to assess the match between two distributions. For Gaussianity, there are other well-known tests that can help us better understand the match. A classical measure of nongaussianity is kurtosis, or fourth-order cumulant. The kurtosis of a random variable X is defined by the average value of $(X - \mu)^4$ divided by σ^4. Since for the normal distribution, this ratio has the value 3, one usually defines the kurtosis as the value of this ratio minus 3. If kurtosis is positive the distribution has longer tails than a normal distribution with the same σ.

For each of the three POP pairs in the figures we computed the kurtosis for 4 separate 1-hour segments. For the POP pair (8,10) the kurtosis values were (0.85, 8.53, 37.21, 43.78) for the four segments. Values close to zero are considered an excellent match. Typically values under 4 or 5 or even 6 are considered reasonable matches. Anything over 10 is considered very poor. Here we see that for a single POP pair, the kurtosis can be either very good or very poor depending upon which hour is considered. For the POP pair (11,2) we had kurtosis values of (56,73,31,116). For the POP pair (6,3) we had (0.75, 1.42, 0.51, 0.07). The latter POP pair is an excellent match while POP pair (11,2) is very poor at any hour.

We conclude so far that the Poisson or Gaussian assumption may hold well for certain POP pairs and for certain hours. However, it does not hold in any general sense. The fact that the validity of this assumption can vary according to both POP pairs and by the hour is both surprising and interesting.

Mean and Variance. In the EM-based method proposed in [3], it is assumed that the variance is related to the mean according to a power law relationship given by $\Sigma = \phi\lambda^b$. The parameter ϕ represents a constant scale factor. Using 16 points of LAN data, they show that b should be between 1 and 2, but closer to 2, to get a reasonable fit. We ask if such an assumption is satisfied by the means and the variances of the traffic flows for the OD pairs we are able to measure. Therefore, we ask the question with respect to WAN data and using hundreds of data points in the analysis.

For each measured POP pair j we have values of the bandwidth computed at 1-second intervals. For intervals of 100 seconds, we compute the mean and variance of the 100 bandwidth measurements, yielding a pair (m, v). Using 400 such intervals we obtain $[(m_1, v_1), ..., (m_{400}, v_{400})]$. Next we order these pairs according to the sorted mean and construct a log-log plot shown in Figure 14. There is one circle for each data point (m_i, v_i). We used a least squares approximation to fit this data to a line. This fitted line is also given in Figure 14.

With this fit, $\log \sigma^2$ will be approximately linear in $\log \lambda$ with slope b. In other words, the slope of this line gives the coefficient b that corresponds to the power law $\sigma^2 = \phi\lambda^b$. For this example $b = 1.97$. We analyzed this relationship for the 39 measured POP-pairs, and found that while the power-law relationship seems to hold, the exponent b does not remain constant over all pairs. We found that b varied fairly uniformly over the interval $[0.5, 4.0]$ for the 39 measured POP pairs.

5. NEW DIRECTIONS

We learned three main lessons from the comparative analysis. First, relying on SNMP link counts as the only concrete informa-

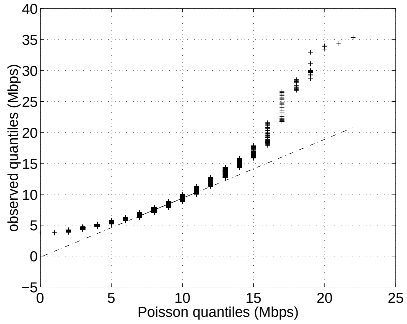

Figure 8: Poisson: Pop 8→10. Hour 2.

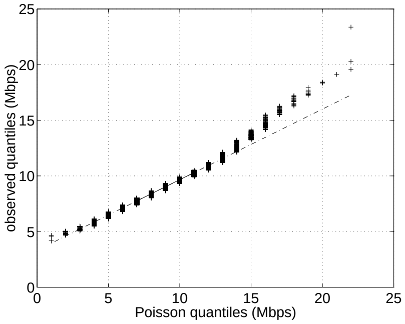

Figure 9: Poisson: Pop 8→10. Hour 3.

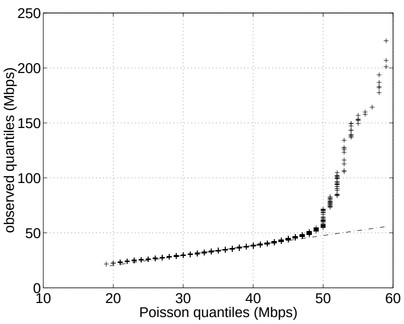

Figure 10: Poisson: Pop 11→2. Hour 2

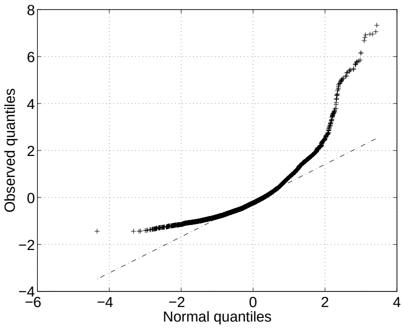

Figure 11: Gaussian: Pop 8→10. Hour 2

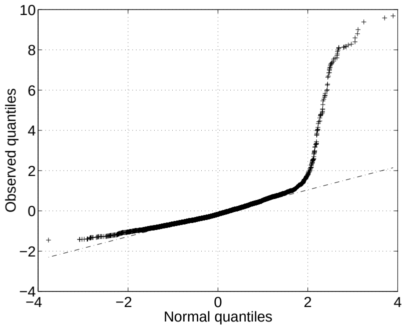

Figure 12: Gaussian: Pop 11→2. Hour 3

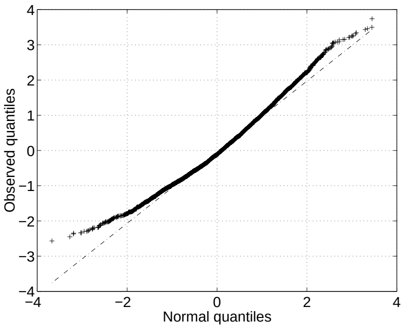

Figure 13: Gaussian: Pop 6→3. Hour 4

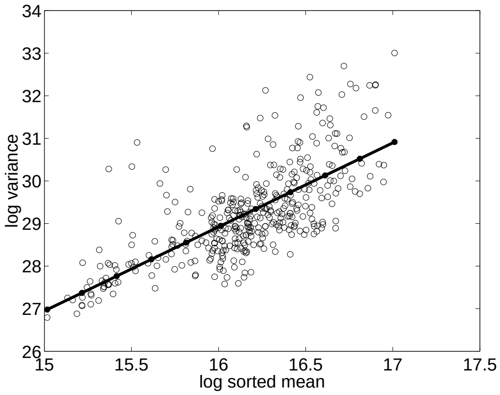

Figure 14: Log-Log Variance vs. Mean relationship

tion about the network yields a set of partial information that is too limiting. Network operators usually have significant amounts of knowledge and information about their network. An important step toward improving TM estimation methods is to devise methods that allow the incorporation of additional network-specific information. Such information may be obtained from either isolated packet-level or flow-level measurements, or be semi-static in nature, such as information on the size of POPs in terms of total capacity, number of customers/peers/data centers per POP, etc.

Second, these techniques are required to make modeling assumptions on the behavior of flows between OD pairs. If such assumptions do not reflect the true nature of their behavior, the corresponding estimates will have limited accuracy. Moreover, these methods actually populate the traffic matrix with estimates of the mean of the traffic demands between OD pairs. If the true nature of inter-POP flows is multimodal (i.e., in the style of the bimodal TM), then calculating the mean will push OD pairs toward some central value that misses the modes. Another important step in improving TM estimation is thus to build better models that make more accurate assumptions. By using direct measurement to obtain components of a traffic matrix, as done in [2], we can learn the true properties of some elements of the real TM, and using such properties, more realistic models can be built.

Third, the performance of the statistical inference techniques in terms of their errors is highly dependent upon the prior information required as input. Therefore a third important step to improve TM estimation is to devise methods to generate good prior TMs.

Motivated by these observations, we propose an approach that tries to make improvements along these three fronts. First, our model will allow us to incorporate additional information characterizing the network similar to the semi-static examples mentioned above. Second, we will make a fundamentally different kind of modeling assumption than the other methods. Rather than assume a probablistic model on a *single* matrix element, we will impose a deterministic model on a *group* of matrix elements. This model will be validated against measurement data. Third, what we propose here is a mechanism to generate a prior. This prior can be used with any statistical inference method to do the estimation steps (see Figure 1).

5.1 The POP Fanout Estimation Problem

Before describing our model we point out that there is an alternate way to state the POP-to-POP traffic matrix estimation problem. The total amount of bytes leaving a POP i corresponds to the sum of the SNMP counts for all outgoing links from POP i. Let O_i denote the total outflow from POP i. Let α_{ij} denote the fraction of the total outgoing bytes from POP i to some other POP j. The amount of traffic X_{ij} flowing from POP i to POP j can thus be described by:

$$X_{ij} = O_i \alpha_{ij} \quad (5)$$

The set of proportions, α_{ij}, $\forall j$ describes to the *fanout* of node i, i.e., how its traffic is distributed across the other POPs. Note that $\sum_j \alpha_{ij} = 1$. The O_i values are readily obtainable from the SNMP data. The traffic matrix estimation problem now becomes that of estimating the proportions α_{ij}. In order to estimate the α_{ij} we will develop a model for the fanout using *Discrete Choice Models (DCM)* [1]. Once we have specified the functional form of the fanouts, the estimation problem is reduced to estimating the parameters of that function.

5.2 Choice Models

The use of DCMs was motivated by asking which types of factors would influence the flow of a packet through the network, that is, which factors would cause a packet to go from a particular source POP i to a particular destination POP j. Clearly there are a large number of such factors, but we believe that they can be generally thought of as falling into two categories.

One category of factors is based on user behavior since most IP traffic is ultimately generated, or at least initiated, by users. The second category of factors comes from network design and configuration. Once a given unit of data is handed over to the network from a user, the path that it follows, and consequently its egress point, are determined by the way the ISP chooses to design its network. Network design choices that impact the path and egress point of a packet include the routing protocols used, routing policies applied (e.g peering, load balancing, etc.) and location issues such as the the location of content, customers, international trunks and servers.

Therefore, we may think of the geographic spread of Internet traffic as resulting from a *2-level decision process* that is taking place in the network at all times. At level 1 there are the users deciding what to send, seek or download, which consequently determines where traffic is generated from and impacts to where that traffic is destined. At the second level we have network design factors, ultimately determining how each packet flows through the network.

For the purpose of modeling we allow ourselves to think of the proportion of traffic going from POP i to POP j as being determined by *choices* made by the source POP i. Clearly POPs are not intelligent agents that make choices in the ordinary sense of the term "choice". However a POP can be thought of as representing the choices made by users and by network designers. This is useful because in a POP-to-POP traffic matrix the POP is the traffic source. A POP represents user choices in that it aggregates all the traffic from many users located "behind" the POP. A POP represents network design choices because the choice of egress node made by the forwarding table is a composite decision based on the many types of policies and location issues outlined above.

Decision behavior is usually modeled by *choice models* (CM) [10]. There is a solid body of mathematical foundations on which CMs have been developed and used in other areas such as transportation modeling and marketing research [5]. Choice models typically have 4 key elements: *decision makers*, a set of *alternatives*, *attributes of the decision makers as well as of the alternatives*, and *decision rules* that govern the decision process. In the context of POP-to-POP traffic matrix estimation, these four elements are defined as follows. As explained above, we allow POPs to be viewed as decision makers since they are the aggregation level of interest here and can be thought of as aggregating user and network design decisions. Each decision-maker (POP) has a set of attributes that characterize it. The set of alternatives corresponds to the set of egress POPs. Similarly, each alternative egress POP has a set of attributes associated with it, influencing its attractiveness. For example, attributes of a POP, that might determine whether a packet should be destined for that POP, include its capacity, the number of attached customers, the number of peering links, etc. We model the decision process as based on a *utility maximization criteria*, in which the *utility* of making a given choice is expressed as a function of, both, the attributes of the decision-maker as well as of the alternatives. As discussed in the next section, we incorporate some level of uncertainty into the decision process by using *random utility models*. Note also that we apply DCMs instead of *pure* CMs because in this context the set of alternatives is discrete.

5.3 Random Utility Model

In general, we can express the random utility of POP i choosing to send a packet to POP j as the sum of a deterministic component, V_j^i, and a random component, ϵ_j^i, as follows:

$$U_j^i = V_j^i + \epsilon_j^i \quad (6)$$

The deterministic component captures observable characteristics of POPs and we use this factor to represent the impact of level-2 engineering factors (described in the previous section) on the choices made by a POP. The random component is intended to capture some level of uncertainty in the decision process. This uncertainty is assumed to arise from our inability to measure all the factors influencing the decision process, such as transient link failures or changes in SLAs.

Expressed this way the utility is a random variable. The probability that POP i selects choice j from a choice set C, denoted $P_C^i(j)$, is stated by equation 7. This is the probability that the random variable U_j^i has the largest value among the utilities of all alternatives. In other words, in a given realization of all the utility random variables, what is the probability that U_j^i will be get the largest value.

$$P_C^i(j) = P[U_j^i = \max_{k \in C}\{U_k^i\}] \quad (7)$$

In order to compute this probability distribution we need to select a specific random utility model which entails defining how we model V_j^i and our assumptions about the random components ϵ_j^i. First we discuss the deterministic component.

We can think of V_j^i as quantifying the *attractiveness* of choice j for POP i because as V_j^i increases, the utility U_j^i also grows. This attractivity may be modeled as a function of the attributes of the POP to be chosen as well as the attributes of the POP making the decision. The question now is what should be the functional form of V_j^i. In most cases of interest, DCMs are designed to use functions that are linear in the attributes.

It is convenient to define a vector of attributes, $\{w_j^i\}$, that includes both the attributes of the alternative j and those of the decision-maker i. Therefore, if M attributes are considered, the function for the deterministic part of the utility function may be specified as follows:

$$V_j^i(X_j^i) = \sum_{m=1}^{M} \mu_m w_j^i(m) + \gamma_j \quad (8)$$

where μ_m defines the relative importance of attribute m with respect to the others, and γ_j is a scaling term representing the amount of attractivity of POP j not captured by the attributes.

There are many factors that could be included as attributes of a given POP that might influence another POP to choose the given one as the egress POP for a packet. Consider the following simple intuitive examples. If a POP in Chicago has many large address blocks "behind" it, whereas a POP in Omaha has small address blocks behind it, then Chicago may be a generally more popular destination. If the ingress POP has many peering links attached to it, than one egress POP with many peering links may be more attractive for that ingress POP than another egress POP with few peering links. If one POP has many customers attached while another has many peering links, then one needs to quantify the balance between these two factors. The weighted sum plays this role.

Modeling the attractiveness of a POP this way allows us to incorporate additional knowledge that network operators typically have about their network. POP features are incorporated via the attributes $w_j^i(m)$ in the attractivity factor V_j^i. Many different choice models could be defined based upon how many and which combination of attributes are included in this deterministic component. Determining which attributes to include for POP-to-POP traffic matrix estimation is an important and involved modeling step. We will consider some initial sample models as a proof of concept for this approach.

We now discuss our assumptions about the random component in our utility model. Let us consider a binary scenario to illustrate the derivation of our model. In this scenario a POP has to choose one alternative from a two-component choice set $C = 1, 2$, meaning that it has to choose between sending a unit of data downstream to either POP 1 or POP 2. The probability that alternative 1 is chosen by POP i is given by:

$$\begin{aligned} P_{1,2}^i(1) &= P[U_1^i \geq U_2^i] \\ &= P[V_1^i + \epsilon_1 \geq V_2^i + \epsilon_2] \\ &= P[V_1^i - V_2^i \geq \epsilon_2 - \epsilon_1] \end{aligned} \tag{9}$$

Without loss of generality, the error term $\epsilon_2 - \epsilon_1$ can be assumed to be centered. Equation (9) relates the choice probability to the cumulative distribution function of the error terms which needs to be defined.

As we explained before, the error term models the uncertainty originated by *unobservable* factors influencing the decision process. Assuming that there are a large number of unobserved attributes, and that they are independent, we can use the Central Limit Theorem to justify modeling the random error as a Gaussian random variable. The solution to equation 9 assuming Gaussian errors would lead to a model for the choice function called *Normal Probability Unit* or *Probit model*. The difficulty with the Probit model is that it has no closed form and thus its practical applicability is limited.

An alternative model, which approximates the Probit model and has a closed form for the density function is the *Logistic Probability Unit* or *Logit model*. In the Logit model, the error terms are assumed to be independent and identically distributed according to a *Gumbel* distribution. The Gumbel distribution is similar in shape to the normal distribution. Even though it is not symmetric with respect to the mean, it matches the normal distribution well over an significant range of the domain of the pdf. In the binary scenario, with a hypothesis of a Gumbel distribution for the errors terms, the probability function for the decision process is given by:

$$P_{1,2}^i(1) = \frac{e^{V_1^i}}{e^{V_1^i} + e^{V_2^i}} \tag{10}$$

This model for the choice probabilities is appealing for several reasons besides its analytical tractability. The logit model enables us to capture the fundamental assumption about the source of uncertainty in the decision process. Furthermore, the logit function captures behavior in which a few elements are large and dominate the overall behavior, and in which there can be great differences between small and large elements. This is attractive due to the well known elephant and mice behavior of Internet flows. Finally, we should mention that the logit model has also proved to be very useful and practical in both the information theory and transportation domains.

This model can be easily extended to multiple alternatives leading to a *multinomial logit model*. In the multinomial case the choice set C representing the set of alternatives can be of any size (certainly larger than two). The probability of choosing a given alternative j from the set C is given by the following equation:

$$P_C^i(j) = \frac{e^{V_j^i}}{\sum_{k \in C} e^{V_k^i}} \tag{11}$$

We use the probability given by Equation 11 to model the POP fanouts, α_{ij} that we were originally seeking to estimate. We thus model the traffic between a pair of POPs using the following equation:

$$X_{ij} = O_i \frac{e^{V_j^i}}{\sum_{k \in C} e^{V_k^i}} \tag{12}$$

where O_i represents the total outgoing bytes sent into the network by POP i.

In summary our model is described by Equations 12 and 8. Equation 12 states that we believe that POP fanouts behave according to an mlogit function, while Equation 8 states that POP attributes influence the fanout via the exponent in the mlogit function. In subsection 5.5 we will assess the validity of this model using direct measurement data.

5.4 Analogy to Gravity Models

Before defining specific models and evaluating them, we pause to point out that our choice model as defined by Equation 12 has the same form as a *Gravity Model*. This analogy is helpful because gravity models define concepts that are intuitively appealing for the interpretation of this equation.

Gravity models are trip distribution models that have been widely used in transportation applications for estimating vehicular traffic demands between urban areas. A gravity model adapts the gravitational concept, as advanced by Newton, to the problem of estimating traffic distribution throughout an urban area. Basically, a gravity model says that the amount of vehicle traffic between zones in an urban area is directly proportional to a *repulsion* factor of the source zone, an *attraction* factor of the destination zone, and inversely proportional to a friction factor between the two zones.

A general formulation of a gravity model may be given by the following equation:

$$X_{ij} = \frac{f(R_i, A_j)}{g(i,j)} \tag{13}$$

where X_{ij} is the traffic volume from i to j; R_i is a parameter representing *repulsive* factors which are associated with "leaving" i; A_j is a parameter representing *attractive* factors related to "going" to j; $f(.,.)$ is a function relating the repulsion and attractivity factors of i and j respectively; and $g(i,j)$ is a *friction* factor between i and j.

Gravity models facilitate an intuitive interpretation of our choice model as given in Equation 12. The notion of a repulsion factor can be interpreted as the amount of traffic that a POP emits. This corresponds to the O_i factor in our model. We combined the attractivity and friction factors together by merging $A_j/g(i,j)$ into the fanouts, α_{ij}, which are modeled by the mlogit function. This clarifies why it is intuitively appealing to view the deterministic factor in our utility model as representing the attractivity of an egress POP.

5.5 Results with Initial Choice Models

Using different combinations of POP attributes, we can build different multi-logit models. Any particular model can include attributes of the egress POPs (i.e., the choices) and/or attributes of the decision maker (i.e., the ingress POP). As an initial proof of concept, we build two basic models. Model I includes a single attribute, namely the total amount of bytes incoming into an egress POP. Hence $V_j^i = \mu_1 W_j(1) + \gamma_j$ where $W_j(1)$ is an attribute of the choice and can be obtained by summing the SNMP link counts for all incoming links into an egress POP. Model II adds one additional attribute into this model, namely the total amount of bytes leaving the ingress POP. Now we have $V_j^i = \mu_1 W_j(1) + \mu_2 W^i(2) + \gamma_j$ where the second attribute is of the decision maker. By incoming and outgoing links of a POP, we are refering to inter-POP links that connect backbone routers in different POPs. Since we are interested in POP-to-POP traffic matrices, we do not include customer access links and peering links.

In order to validate our modeling approach we need to see whether the mlogit function properly describes fanouts and whether weighted sums of attributes incorporated in exponents are useful in improving fanout estimation. To do this we make use of three rows of a real POP-to-POP traffic matrix that we were able to measure direclty. We use this data to calibrate our model. The problem of calibrating the model is one of estimating the model parameters (μ_1, μ_2, γ_j).

The data and method to generate the three rows of an actual POP-to-POP traffic matrix were described in Section 4.5. Each row describes one fanout, i.e., $\alpha_{ij} \quad \forall j$. Since the fanout describes the fraction of traffic going from POP i to POP j, and since $\sum_j \alpha_{ij} = 1$, each α_{ij} can be viewed as the probability that a packet from POP i will be selected to egress at POP j. This can also be viewed as representing the *decision* made by the ingress POP. We make this point to tie together the concepts of using decision models, POP fanouts and how our data represents these concepts.

Using the actual fanouts as our data set, we seek to find model parameters for our two basic models to see if either of the two mlogit functions can be made to fit the data well. We use maximum likelihood estimation to calibrate the model parameters. To provide a preliminary assessment of the correspondence of the form of the logit function and the observed fanouts, we use a goodness-of-fit test. An analogy that we may use to illustrate the goal of using goodness-of-fit tests is the following. Consider the case in which we have a set of data points and we are asked to fit them specifically with an exponential function. In this case we need to find the parameter of an exponential function that best fits the observed data points. If we are able to find an exponential parameter that fits well the data points we would have a first indication showing that the observed data has actually an exponential shape. If the observed data points are, say, random, there is no exponential function that would fit the data points well.

Figure 15 compares the actual fanout observed from data to the fanout predicted by Model I for a POP in New York City. The x-axis indicates the names of the other POPs included in our topology. We see that the fanouts are well predicted for a number of the egress POPs, but there are two or three egress POPs whose fanouts are poorly predicted. This indicates that additional attributes may be needed to improve fanout estimation. Figure 16 compares the actual fanout of the same ingress POP to the estimated fanouts using Model II (that includes one attribute more than Model I). Model II significantly improves the errors for those egress POPs that exhibited high errors using Model I. Specifically the fanouts for the last six POPs are all improved.

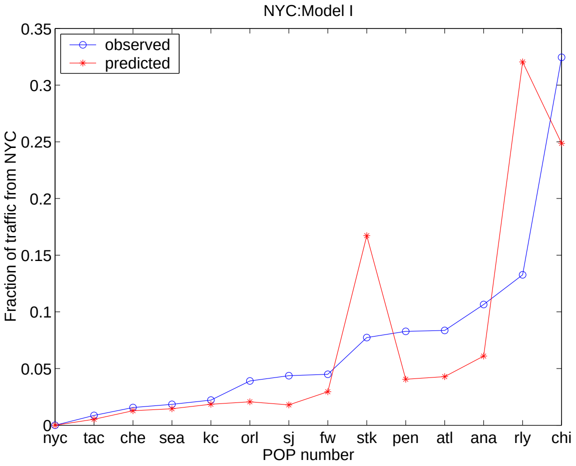

Figure 15: Choice Probabilities (NYC, Model I)

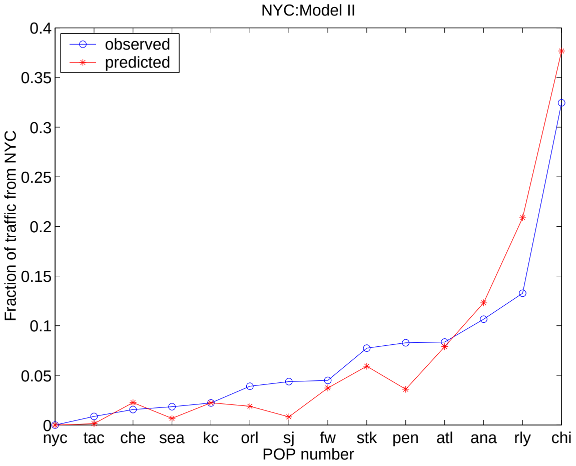

Figure 16: Choice Probabilities (NYC, Model II)

For model II we used a *root mean square error metric* as a additional measure of goodness of fit. Our model yielded RMSE's of 0.0299 for POP *NYC*, 0.0729 for POP *PEN* and 0.0377 for POP *SJ*. The closer the RMSE is to zero the better fit is provided by the logit model. The RMSE values are not too distant from zero and although there is still room for improvement, we consider these results very promising.

These results illustrate a few points. First our mlogit function can be used to properly describe fanouts. Second, adding additional attributes into the attractivity factor appears promising for improving fanout estimation. Overall, we see that a choice model that includes the total SNMP byte counts entering and exiting a POP can give a decent prior estimate for a POP-to-POP traffic matrix.

Before concluding this section we make one final point about comparing the performance of these choice models to the three existing techniques we studied in the first part of this paper. We could consider making a comparison as follows. We could use our choice model to generate a prior and see if that can improve either the Bayesian or EM methods. However it no longer makes sense to evaluate these two methods using the synthetic TMs we presented earlier. These TMs were generated using methods that violate the mlogit function behavior. If our assumption is that TMs can be describe by mlogit functions, then we need to generate a synthetic TM based on some mlogit function as a test case. Hence the overall evaluation of our model needs two steps: first, it needs to be demonstrated that the mlogit model is valid, and second, the performance of a statistical method using our prior should be assessed on a variety of synthetic TMs generated according to different mlogit models. In this paper we have carried out the first step; step 2 remains for future work.

6. CONCLUSIONS

Our systematic comparison of the three existing techniques shed light on the pragmatic implications of applying these approaches in real environments. In our 14-node POP-to-POP topology, the LP method had an average error of 170%. We described the problems with the LP method and conclude that it cannot be considered a viable candidate for use in carrier networks. The Bayesian approach had errors ranging from $20-45\%$, while the EM approach had errors ranging from $10-25\%$. The errors span this range largely due to the sensitivity of both methods to the goodness of the prior. It is clear than the EM method is the best of the existing techniques.

Both methods showed some sensitivity to path lengths and link sharing in that it is harder to estimate OD pairs that have longer paths or traverse links that are highly shared. The Bayesian method is more sensitive to these factors than the EM method. We found that trying to improve these methods either by adding a few known rows of the traffic matrix, or by expanding the window to include additional measurement intervals, does not result in a major reduction in estimation errors. These observations lead us to conclude that, rather than add more data of the same type already being used, it is worth pursuing methods that allow the incorporation of different kinds of data.

Using real traffic traces from a Tier-1 backbone network we evaluated the validity of the basic assumptions made about the traffic by the two statistical approaches. We found that the Poisson and Gaussian assumptions hold for some POP pairs and not for others; similarly they hold in some one hour periods but not in others. Therefore these assumptions are not true in any general sense. We observe in our traces that the relationship between the mean and the variance for OD-pair traffic flows, namely $\sigma^2 = \phi\lambda^b$ is acceptable; however, the value of the exponent b is not constant for different OD pairs.

We have introduced a new approach to TM estimation based on choice models that model POP fanouts according to an mlogit function. The three rows of a real POP-to-POP matrix that we obtained illustrates that this model is very reasonable. Through the exponents in this model, we can incorporate POP features such as the number of customer links, number of routers, the size of an address block behind a POP, etc., that influence the attractiveness of sending packets to a given POP. For two simple models, that include one or two POP attributes, we calibrate our model by estimating the model parameters. We compared our model to actual data via a goodness of fit test. The small RMSEs serve as a proof of concept of this approach. Our Model II indicates that by using the total amount of bytes incoming and outgoing from a POP, we can generate a decent TM prior with RMSE's under 0.8.

In the future, we will focus on developing more elaborate choice models that include many more POP features. This method will allow us to understand which POP features impact POP-to-POP flows and which don't. Given these elaborate models, we will compare choice models to the existing methods. Since our method can be used simply to generate a good prior, we will also evaluate to what extent our method can improve the existing techniques.

ACKNOWLEDGEMENTS. We would like to thank Dr. Claudia Tebaldi for useful discussions and for generously providing her code implementing the Bayesian method; Prof. Bin Yu for helpful discussions on the EM method; Profesors Ibrahim Matta and Mark Crovella for useful discussions during the development of this work; Dr. Bryan Lyles for initialing pointing us in this direction; and finally Dr. Jennifer Rexford for helping us prepare the paper.

7. REFERENCES

[1] M. Ben-Akiva and S. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, 1985.

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

[3] J. Cao, D. Davis, S. Vander Weil, and B. Yu. Time-Varying Network Tomography. *J. of the American Statistical Association.*, 2000.

[4] W.E. Deming and F.F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, pages 427–444, 1940.

[5] S. Erlander and N.F. Stewart. *The Gravity Model in Transportation Analysis – Theory and Applications*. 1990.

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

[7] O. Goldschmidt. ISP Backbone Traffic Inference Methods to Support Traffic Engineering . In *Internet Statistics and Metrics Analysis (ISMA) Workshop*, San Diego, CA, December 2000.

[8] L. Rschendorf. Convergence of the iterative proportional fitting procedure. *Annals of Statistics*, pages 1160–1174, 1995.

[9] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 1993.

[10] J. Swait. Probabilistic Choice Set Information in Transportation Demand Models. Technical Report Ph.D. Thesis, MIT, Department of Civil and Environmental Engineering, 1959.

[11] C. Tebaldi and M. West. Bayesian Inference of Network Traffic Using Link Count Data. *J. of the American Statistical Association.*, pages 557–573, June 1998.

[12] Y. Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. *J. of the American Statistical Association.*, pages 365–377, 1996.