An Independent-Connection Model for Traffic Matrices

Vijay Erramilli
Dept. of Computer Science
Boston University
evijay@cs.bu.edu

Mark Crovella
Dept. of Computer Science
Boston University
crovella@cs.bu.edu

Nina Taft
Intel Research
Berkeley, CA
nina.taft@intel.com

Abstract

A common assumption made in traffic matrix (TM) modeling and estimation is independence of a packet's network ingress and egress. We argue that in real IP networks, this assumption should not and does not hold. The fact that most traffic consists of two-way exchanges of packets means that traffic streams flowing in opposite directions at any point in the network are *not* independent. In this paper we propose a model for traffic matrices based on independence of *connections* rather than packets. We argue that the independent-connection (IC) model is more intuitive, and has a more direct connection to underlying network phenomena than the gravity model. To validate the IC model, we show that it fits real data better than the gravity model and that it works well as a prior in the TM estimation problem. We study the model's parameters empirically and identify useful stability properties. This justifies the use of the simpler versions of the model for TM applications. To illustrate the utility of the model we focus on two such applications: synthetic TM generation and TM estimation. To the best of our knowledge this is the first traffic matrix model that incorporates properties of bidirectional traffic.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations; C.4.3 [**Performance of Systems**]: Modeling Techniques

General Terms

Measurement

Keywords

Traffic Matrix, Modeling, Independent-Connection Model, Gravity

1. INTRODUCTION

The flow of traffic through a network is a crucial aspect of the network's workload. The amount of traffic flowing from each ingress point (origin) to each egress point (destination) is called the traffic matrix (TM). Given the importance of the traffic matrix for many aspects of network operations, good models of traffic matrices are very useful.

Despite the importance of TM modeling, there has been little work to date focusing on *complete* models for TMs. By a complete model, we mean one that can be used to generate or characterize a time-series of representative traffic matrices for a given network topology. Such a model should ideally have a small number of physically meaningful inputs. Examples of such inputs would be the size of user populations served by each access point, or the nature of the application mix in the network.

Although complete TM models do not yet exist, some models for parts of the problem have been developed [16, 13]. One of the most popular models in connection with traffic matrices is the 'gravity' model, which estimates OD flow counts from ingress and egress counts. The gravity model has been used extensively in TM estimation and has been proposed as a tool for synthetic TM generation[11]. The key assumption underlying the gravity model is that the traffic entering the network at any given node exits the network at a particular egress node in proportion to the total traffic exiting at that egress. This can be thought of as a model in which the *ingress and egress points* for any given packet are *independent*.

This paper starts from a simple observation: *in the Internet, this independence assumption should not and does not hold*. The reason is that most network traffic consists of connections – two-way conversations usually carried over TCP. Each connection has an initiator (a host that requested the connection) and a responder (a host that accepted the connection request). The result is that the amount of traffic flowing from ingress i to egress j is not independent of the amount of traffic flowing from ingress j to egress i. Thus while the gravity model is appealing for its simplicity, it is divorced from key underlying phenomena that shape traffic flow in a real network.

To capture these phenomena, we propose the *independent-connection* (IC) model. Rather than assuming that ingress and egress of packets are independent, we assume that initiators and responders of connections are independent. The key is that we think of an aggregate flow (such as an OD flow) not as collections of packets, but as collections of connections. We focus on *connections* rather than *packets* because of the bidirectional nature of connections that contain packets flowing in two directions.

The IC model is based on three intuitive notions: first, each aggregated traffic stream from a single origin to a single destination consists of two kinds of traffic: forward traf-

IMC'06, October 25–27, 2006, Rio de Janeiro, Brazil.

fic, flowing from initiators to responders; and reverse traffic, flowing from responders to initiators. We assume that at a high enough level of traffic aggregation, the ratio of forward and reverse traffic may exhibit some stability in time and/or space (i.e., at different access points). The second notion is that each network access point has an *activity* level, meaning the rate at which bytes are flowing through the network due to connections initiated at that access point. Thus the network activity for a given access point consists of a portion of the traffic flowing both *to* and *from* the access point. Finally, the third notion is that each access point has an associated *preference.* This is the fraction of all connections whose responders are associated with that access point.

Each of these assumptions has a natural physical interpretation. The ratio of forward to reverse traffic in a large set of connections reflects the properties of the underlying applications, and the application mix. For example, Web traffic will tend to have a much greater amount of traffic flowing in the reverse direction than in the forward direction, while P2P traffic may show less asymmetry. The activity level of a node corresponds to the number of users who access the network at that node, and their current level of network use. The preference of a node corresponds to the "desirability" or "popularity" of the services that are reached through that node — i.e., a level of interest expressed as a likelihood that any given user will seek to initiate a connection to any given service via that node.

In the IC model, these concepts are composed in a straightforward way (in Sec. 2). Depending on the assumptions one makes about the temporal or spatial stability of model parameters, one can obtain versions of the model that are useful for a number of network modeling tasks. Using real traffic matrices we study the behavior of the three key notions (parameters) of our model. We find, for example, that forward/reverse ratios and node preference show remarkable stability over time. Our findings indicate that the simpler versions of our models are sufficient for applications such as synthetic traffic matrix construction and TM estimation. We focus on these two applications and describe a method for traffic generation based on our model.

We also use the model to construct inputs for the problem of TM estimation. We show that although the IC model has fewer inputs that the gravity model, it does a better job at reproducing OD flow counts than the gravity model. Our purpose is not to propose another TM estimation method, but rather to validate our model by illustrating that it can work well in a known TM application, and do so with fewer parameters.

2. INDEPENDENT-CONNECTION MODEL

Consider a network with n access points. Traffic flows into and out of the network at each access point. During any fixed time interval, the amount of traffic in bytes that enters the network at node i and leaves the network at node j is denoted X_{ij}. This is called an Origin-Destination (OD) flow from origin i to destination j. All the traffic flowing into the network at node i is X_{i*} (which we refer to as the *ingress* traffic at i) and all the traffic flowing out of the network at node j is X_{*j} (the *egress* traffic at j). Finally, X_{**} denotes all the traffic in the network, i.e., the sum of X_{ij} for all $i = 1, ..., n$ and $j = 1, ..., n$.

The 'gravity' model treats the flow of traffic as a random process. For any packet, we let I be the random variable corresponding to the packet's ingress and E be the random variable corresponding to the packet's egress. The gravity model states that I and E are independent, that is, $P[E = j|I = i] = P[E = j]$ and likewise, $P[I = i|E = j] = P[I = i], \forall i, j$. Based on this 'independent-packet' assumption, the gravity model predicts that X_{ij} should be well approximated by $X_{i*}X_{*j}/X_{**}, \forall i, j$.

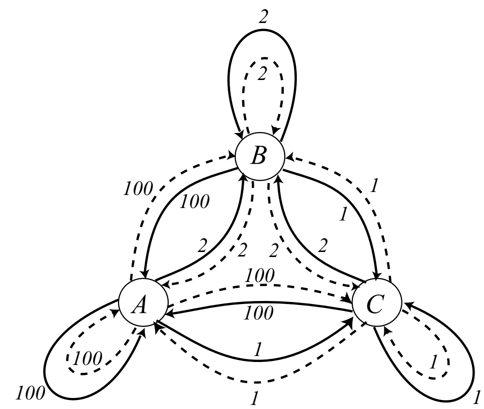

Figure 1: Example Traffic in an IC Setting.

We start with the simple observation that the independent-packet assumption is not accurate in the Internet. A typical packet in the Internet is part of a *connection* — a two-way exchange of packets, generally in the form of a single TCP connection. Packets flowing in opposite directions of a connection are not well modeled as being independent.

To see why this is the case, consider the following example. Let us assume that all connections consist of equal volumes of traffic flowing in the *forward* direction and the *reverse* direction. In actual practice forward and reverse volumes are not likely to be equal, but it simplifies the example.

Suppose some nodes initiate a larger traffic volume than other nodes, as in the three-node network example in Fig. 1. In the figure, directed arcs show the direction of traffic flow from origin to destination. Dotted arcs correspond to traffic flowing from initiator to responder (forward traffic) and solid arcs correspond to traffic flowing from responder to initiator (reverse traffic). Self-looping arcs correspond to connections between different hosts having the same access point. The total traffic flowing into the network at any node consists of all the arcs leaving that node, and traffic flowing out consists of all arcs pointing to that node.

Because we assume each connection contains the same number of bytes in both directions, pairs of forward and reverse arrows between an initiator and responder have the same value. In this case, node A initiates 3 connections, each of 100 packets in each direction; node B initiates 3 connections, each of 2 packets in each direction; and node C initiates 3 connections each of 1 packet in each direction. The key point is that the initiator and responder of any given connection are independent; that is, the probability that a connection's responder is any particular node is independent of the connection's initiator node.

However, we see that ingress-egress independence for *packets* is not a valid model. For example,

$$\begin{aligned} P[E = A|I = A] &= 200/403 \approx 0.50, \\ P[E = A|I = B] &= 102/109 \approx 0.93, \\ P[E = A|I = C] &= 101/106 \approx 0.95, \text{ and} \\ P[E = A] &= 403/618 \approx 0.65. \end{aligned}$$

These probabilities should be equal under gravity.

2.1 Model Definition

Rather than model connections individually, we note that what is important about a connection is that it consists of traffic in both the forward and reverse directions. Consider the collection of all connections with initiator node i and a responder node j. We use f_{ij} to denote the portion of the total bidirectional traffic due to these connections that is in the forward direction. That is, f_{ij} denotes bytes contained in forward traffic divided by bytes in forward plus reverse traffic, so $0 \leq f_{ij} \leq 1, \forall i, j$

Next we consider the total traffic due to connections *initiated* at node $i, i = 1, ..., n$, which we denote A_i (for 'activity'). This consists of some forward traffic flowing into the network at node i, and some reverse traffic flowing out of the network at node i.

Finally we consider how connection responders are chosen. Since we assume an independent connection model, the probability that a connection responder is at node i depends only on i. We denote this by $P_i, i = 1, ..., n$ (for 'preference'). We do not assume that the P_i values sum to one, but usually we will use them as probabilities and so will normalize by $\sum_{i=1,...,n} P_i$.

Then if X_{ij} is the amount of traffic between nodes i and j, the **general independent-connection model** has:

$$X_{ij} = \frac{f_{ij} A_i P_j}{\sum_{i=1}^{n} P_i} + \frac{(1 - f_{ji}) A_j P_i}{\sum_{i=1}^{n} P_i} \quad (1)$$

for $i, j = 1, ..., n$.

The first term captures the forward traffic flowing from i to j – traffic generated by the activity of users at node i; the second term captures the reverse traffic flowing from i to j – traffic generated by the activity of users at node j.

When considering only a single connection, f may vary considerably. However in a backbone network carrying highly aggregated traffic, we assume that f will show some degree of stability across different OD flows. In this case we simplify the general IC model and assume f is constant across the network. This leads to the **simplified IC model**:

$$X_{ij} = \frac{f A_i P_j}{\sum_{i=1}^{n} P_i} + \frac{(1 - f) A_j P_i}{\sum_{i=1}^{n} P_i} \quad (2)$$

for $i, j = 1, ..., n$.

The next aspect of the model to consider is stability of parameters in time. In the most general case, all parameters may vary at each time step. We term this version of the model the **time-varying IC model**:

$$X_{ij}(t) = \frac{f(t) A_i(t) P_j(t)}{\sum_{i=1}^{n} P_i(t)} + \frac{(1 - f(t)) A_j(t) P_i(t)}{\sum_{i=1}^{n} P_i(t)} \quad (3)$$

for $i, j = 1, ..., n$ and $t = 1, ..., T$.

We consider two other variants of the model which incorporate increasingly restrictive assumptions about the temporal stability of the parameters. At high levels of aggregation, one might expect f to show stability in time (shown in Sec. 4). This assumption results in the **stable-f IC model**:

$$X_{ij}(t) = \frac{f A_i(t) P_j(t)}{\sum_{i=1}^{n} P_i(t)} + \frac{(1 - f) A_j(t) P_i(t)}{\sum_{i=1}^{n} P_i(t)} \quad (4)$$

for $i, j = 1, ..., n$ and $t = 1, ..., T$.

Finally, we consider the case in which connection preferences are stable in time as well. This assumption may be justified if connection preferences reflect a relatively stable underlying 'popularity' of services available via each node. This assumption yields the **stable-fP IC model**:

$$X_{ij}(t) = \frac{f A_i(t) P_j}{\sum_{i=1}^{n} P_i} + \frac{(1 - f) A_j(t) P_i}{\sum_{i=1}^{n} P_i} \quad (5)$$

for $i, j = 1, ..., n$ and $t = 1, ..., T$.

There are a number of practical reasons for considering these different model variants. First, consider the problem of constructing synthetic TMs. Using the gravity model, one must somehow synthetically generate $2n$ values at each timestep t, namely $\{X_{i*}(t)\}$ and $\{X_{*i}(t), i = 1, ..., n\}$. The stable-$f$ IC model requires the same number of input parameters at each timestep: $\{A_i(t)\}$ and $\{P_i(t), i = 1, ..., n\}$ and hence presumably presents roughly the same level of modeling difficulty. However, the IC model can do significantly better: the stable-fP model requires only n inputs $\{A_i(t)\}$ at each timestep.

The second reason for considering different model variants relates to the TM estimation problem. These variants reflect different amounts of outside information that must be brought into the estimation process. When the stable-fP model as used for TM estimation, one assumes that the stable values of f and P have previously been measured; then only n inputs (namely $\{A_i(t)\}$) need to estimated from data. In the case of the stable-f model, we assume that f has been measured previously and can be reused; here both $\{A_i(t)\}$ and $\{P_i(t)\}$ are estimated from data.

3. MEASUREMENT DATA

To explore the validity and utility of the IC model, we used three data sets.

Géant Data: Géant [1] is a network of 22 PoPs connecting research institutions and universities across continental Europe. The sampling rate is 1 packet out of every 1000. The methodology used to construct OD flows from netflow data is detailed in [7]. We use a time bin size of 5 minutes to construct OD flows, giving us 2016 sample points for each week's worth of data.

Totem Data: The publicly available Totem [14] data set comes from the same Géant network. The data set consists of 4 months of TMs, constructed from sampled netflow data using the method described in [14]. Our results for the Totem TM are very similar to those obtained with the Géant data above, and are often omitted due to lack of space. We do however present this data when we want to illustrate multi-week long behavior.

Full Packet Header traces from Abilene Backbone: We used two hour contiguous bidirectional packet header traces collected at the Indianapolis router node (IPLS), in the Abilene network[2]. The links instrumented are the ones eastbound and westbound, towards Cleveland (CLEV) and Kansas City (KSCY).

4. MODEL EVALAUTION

We now examine how well our model can fit empirical datasets. We expect the time-varying model to fit the data best, followed by the stable-f model, with the poorest fit coming from the stable-fP model. This is based on the number of time-varying model parameters: $3n$ for the time-varying model, $2n$ for the stable-f model, and n for the stable-fP model. More precisely, if we are trying to fit a dataset of OD flows from a network with n nodes over t timesteps, the gravity model has $2nt - 1$ degrees of freedom (i.e., inputs), the time-varying IC model has $3nt$ degrees of

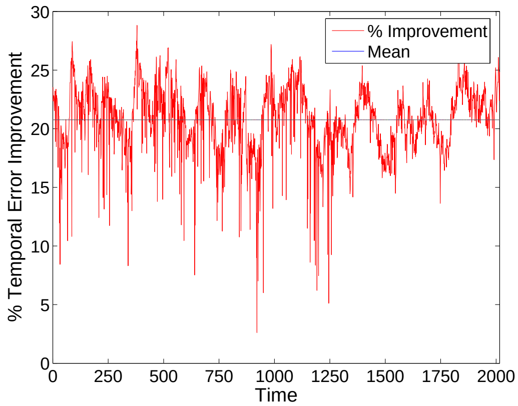

Figure 2: Temporal % Improvements of IC-Model over Gravity; Geant data

freedom, the stable-f model has $2nt+1$ degrees of freedom, and the stable-fP model has $nt+n+1$ degrees of freedom.

To be most conservative in our conclusions, we focus on our weakest model, the stable-fP model. The metric we use for measuring accuracy of model prediction here and throughout the paper is *relative* l_2 *temporal error* (as in [12]):

$$RelL2_T(t) = \frac{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(X_{ij}(t)-\hat{X}_{ij}(t))^2}}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}X_{ij}(t)^2}} \quad (6)$$

We estimate the values of f, P_i, and $A_i(t)$ via optimization, using the following nonlinear program:

$$\begin{aligned} \textit{minimize} \quad & \sum_{t=1}^{T} RelL2_T(t) \\ \textit{where } \hat{X}_{ij}(t) \quad &= \quad fA_i(t)P_j + (1-f)A_j(t)P_i \\ \textit{subject to:} \quad & \\ A_i(t) \quad &\geq \quad 0 \quad \forall i,t \\ P_i \quad &\geq \quad 0 \quad \forall i \\ \sum_i P_i \quad &= \quad 1 \end{aligned}$$

If we assume that errors have a Gaussian distribution, this is equivalent to a maximum-likelihood estimation of model parameters. We use the optimization toolbox provided by Matlab [3] to find the solution numerically.

We fit both our model and the gravity model and compare their accuracy in Fig. 2. We plot the improvement of the IC model over the gravity using the $RelL2_T(t)$ metric. The improvement typically lies between 18 to 24%. The absolute improvements lie between 0.10 - 0.12. It is both surprising and encouraging that our model can fit the data well with a single constant f for the entire network. This implies that our simpler IC models may be sufficient for the TM applications we consider.

Characterizing f. Since the stable-fP model can indeed fit real traffic data quite well, we now examine which values for f are reasonable and whether such a parameter setting is stable in time. Ideally, to perform a thorough study of observed values of f, one would need unsampled netflow traces, or unsampled (to properly determine the initiator) packet header traces of all traffic in the network. To the best of our knowledge no such data sets exist. What we can do is to measure f for two large OD flows in the Abilene network since in our packet trace data the packets in the pairs *(KSCY,IPLS)* and *(IPLS,KSCY)* flow over the IPLS-KSCY link. Due to space limitations, we refer the reader to [5] for a complete description of our procedure for measuring f.

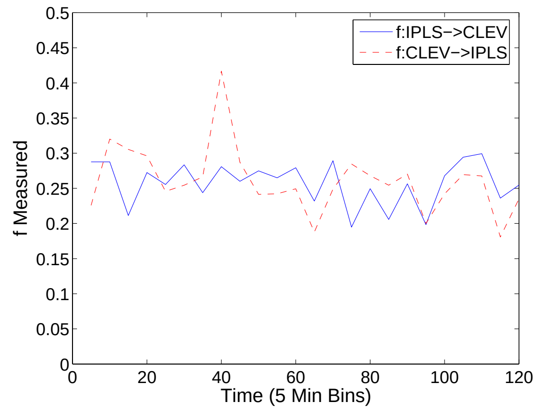

Figure 3: f values over time

In Fig. 3 we plot our measured values of f. We draw three conclusions: (1) f lies consistently in the range 0.2 to 0.3; (2) f is quite stable across time; and (3) the values of $f_{(CLEV,IPLS)}$ and $f_{(IPLS,CLEV)}$ are similar - which provides preliminary support for assuming spatial stability of f_{ij} over different (i,j) pairs.

We extracted the values of f computed from our fitting procedure in Sec. 4. We found that over a seven week period (Totem data) f did not vary outside the narrow range of [0.17,0.21]. The stability of f is encouraging as it means that if f could be measured once in a while, the obtained value for f could subsequently be used in modeling tasks for possibly multiple weeks. We believe that the particular values for f we observe are reflective of the application mix in highly aggregated traffic. While applications like Web browsing would have values of f lower than 0.2, other applications such as P2P could have higher f values.

Characterizing Preferences. Using our fitted data, we observed the values for $\{P_i\}$ over seven weeks in the Totem data. We make two observations from Fig. 4 (the values of $\{P_i\}$ for nodes i are in arbitrary order, and the $\{P_i\}$ sum to 1). First, values of P_i for any given node i are remarkably stable over time, even over seven weeks. Combining this observation with that of the stable f behavior, lends support for the use of the stable-fP model. Second, the $\{P_i\}$ values are highly variable across different node (with some nodes as much as ten times greater than the typical value). The high variability of $\{P_i\}$ values prompts us to examine their distributional tail. We found that a long-tailed lognormal distribution does a better job in fitting the tail.[5]. However the distributional fits should *not* be relied on too heavily; we have far too few data points (22 or 23) to reliably choose a distributional model for this data.

Characterizing Activity Levels. Finally we examine the nature of the timeseries of activity levels, $\{A_i(t)\}$, in our data. The variation in these values is the source of all time-variation in the stable-fP model. Intuitively, if $\{A_i(t)\}$ represents the rate at which traffic is being 'initiated' at node i, we expect to see familiar patterns of daily variation.

In [5] we provide the details of our findings on activity

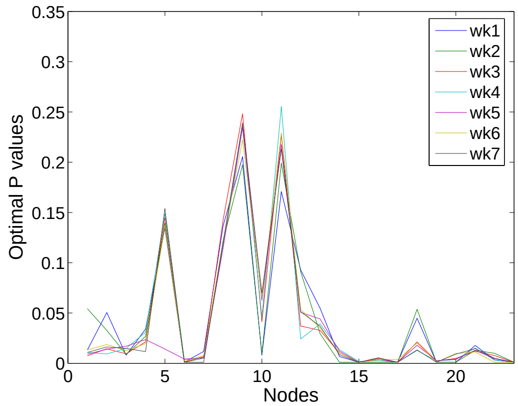

Figure 4: Optimal P Values over Time Totem

levels. In summary, we did find strong diurnal patterns in these timeseries, corresponding to daily variation as well as weekend variation.

5. TM GENERATION USING IC MODEL

Developing a comprehensive framework for synthetically generaing traffic matrices is a non-trivial task [10]. However, we claim that the IC model represents a relatively simple and accurate starting point for TM generation.

Based on the characterization results we have presented here, one can attempt to use the stable-fP IC model to generate synthetic traces by following these steps: (1) Choose an f value. Our results suggest that a in the range 0.2 to 0.3 is reasonable, (2) Use a long-tailed distribution to generate a set of preference values $\{P_i\}$. While we do not advocate a particular choice of distribution, our results suggest that a distribution like the lognormal is a reasonable choice, (3) Generate activity time series $\{A_i(t), i = 1, ..., n\}$. A model that explicitly incorporates daily variation, such as the cyclo-stationary model in [13], would make sense. and (4) Construct the timeseries of traffic matrices $X_{ij}(t)$ using equation (5). There are a number of advantages to this approach for TM generation. First, the parameters have an interpretation in terms of real network phenomena. For example, if an analyst wishes to incorporate knowledge of application mix, or explore the effects of changes in application mix, it is possible to do so by varying f. If an analyst wishes to model network 'hot spots' or 'flash crowds' this is possible by varying $\{P_i\}$. Finally if an analyst wishes to incorporate knowledge of user population levels, or explore the effects of varying such levels, it is possible to do so by varying $\{A_i(t)\}$. Second, as noted in Sec. 2, the stable-fP model requires relatively few inputs: $nt+n+1$ for a network of size n over t timesteps.

Finally, the last strength of this approach is made clear by contrast with the gravity model, which previously has been suggested as a starting point for synthetic TM generation [11]. In the gravity model, the set of inputs $\{X_{i*}(t)\}$ and $\{X_{*j}(t)\}$ are *causally* related. gravity models requires that the sum of the $X_{i*}(t)$s $\forall i$ equal the sum of the $X_{*j}(t)$s $\forall j$. This constraint is typically imposed per time slot t. Thus it is not a simple matter to synthetically construct inputs to the gravity model: the inputs at each timestep are causally constrained, in a complex way. On the other hand, the set of time-varying inputs $\{A_i(t)\}$ to the stable-fP model are not causally related. They may show strong correlations in time (which can be modeled stochastically), but they do not follow any constrained relationship that must be preserved. Hence synthetic TM generation is considerably simpler under the IC model than under using a method like the gravity model.

6. IC MODEL FOR TM ESTIMATION

In the last section we showed that the IC model estimates real TM data better than the gravity model. This motivates us to ask whether it forms a better starting point for TM estimation than the gravity model. The TM estimation problem has been well studied. The fundamental issue with this inference problem is that it is a highly under-constrained, or ill-posed, system [12, 16].

Although the specifics of particular solutions to TM estimation differ, many of the TM estimation solutions follow this blueprint: Step (1) Choose a starting point x^{init} as a prior to the estimation algorithm, Step (2) Run an estimation algorithm using Y, R and x^{init} to get x^{est} and Step (3) Run an iterative proportional fitting algorithm to make sure the estimated TM x^{est} adheres to link capacity constraints.

Most solutions to the TM estimation problem (such as [9, 4, 12, 16]) select different methods to carry out steps 1 and 2, while step 3 remains the same across many solutions. The studies in [8, 6] indicate that the quality of the initial prior is quite important and thus a better initial prior can lead to substantially better estimates.

A key question in using the IC model to generate the prior in Step 1 is what sorts of measurements are used in to determine the model parameters. When using the stable-fP model, we assume that f and $\{P_i\}$ have been previously measured. To use the stable-f model, we only need to assume that f is known or measurable. We will see that both cases yield improvements over the gravity model.

One possibility for obtaining measurements for our parameters is that of the hybrid scenarios proposed in [12] which combine direct TM measurement during a small set of weeks with inference used during the other weeks. (This was proposed to lighten the burden on flow monitors.) In this case old TMs can be used to calibrate our parameters. For our study, we vary the prior in Step 1 keeping Steps 2 and 3 the same. For Step 2 of the TM estimation procedure we use the least-squares estimation techniques proposed in [15] (also known as the tomo-gravity approach).

We point out that we are using the simple gravity model herein, and the authors of [16] have also proposed a generalized gravity model, which takes into account side information about link types (e.g., access or peering). It is known that this extension to the gravity model improves upon the simple gravity model. We did not compare our prior to the generalize gravity model prior because such additional information was not available to us.

6.1 TM Prior with Stable-fP Model

In these experiments, we use two weeks of data. Week 1 data is used to compute f and $\{P_i\}$. The goal is to produce TM estimates for week 2. But before producing a TM prior for week 2, we need to produce estimates for $\{A_i(t)\}$ during week 2. We next explain how to estimate $\{A_i(t)\}$ (called $\{\tilde{A}_i(t)\}$) using only the ingress and egress node counts along with the f and $\{P_i\}$ values.

In what follows, we organize the values of $\{A_i(t)\}$ into an $n \times t$ matrix $\mathcal{A}$, in which $\mathcal{A}_{it}$ corresponds to $A_i(t)$. We also reorganize $X_{ij}(t)$ into the $n^2 \times t$ matrix $\mathcal{X}$ where we have one (i, j) pair per row, and each row is a time series.

In the stable-fP model, $X_{ij}(t)$ is a linear function of $\{A_i(t)\}$ as given by Equation (5). Thus we can use f and $\{P_i\}$ to construct a matrix ϕ such that:

$$\mathcal{X} = \phi\mathcal{A} \quad (7)$$

We cannot use a pseudo-inverse solution on this equation to estimate A from $\mathcal{X}$ because $\mathcal{X}$ is unavailable (indeed, this is the traffic matrix itself). However, what is available are the ingress and egress nodes counts, specifically $\{X_{i*}\}$ and $\{X_{*j}\}$. We thus need to convert $\mathcal{X}$ in (7) to the ingress and egress counts.To do so, we define a matrix H whose elements H_{ij} are 1 if TM flow j contributes to the total ingress count for node i, and 0 otherwise. In other words, if TM flow j originates at node i, then its traffic will be counted in that nodes ingress count. Because there are n nodes and n^2 OD flows in the network, the dimensions of H are $n \times n^2$. With this definition of H we can now write, for a node i, $X_{i*} = H(i,*)\mathcal{X}$ where $H(i,*)$ is the i-th row of H. Let $\mathcal{X}_{ingress}$ denote the column vector of ingress counts for all nodes; thus $\mathcal{X}_{ingress} = H\mathcal{X}$, and the dimensions of $\mathcal{X}_{ingress}$ are $n \times t$ (i.e., we have the time series for each node). Similarly we define the 0-1 matrix G such that $\mathcal{X}_{egress} = G\mathcal{X}$.

Next we define the block matrix Q that is composed by stacking matrix H on top of G:

$$Q = \begin{bmatrix} H \\ G \end{bmatrix}$$

whose size is thus $2n \times n^2$. Then

$$Q\mathcal{X} = \begin{bmatrix} \mathcal{X}_{ingress} \\ \mathcal{X}_{egress} \end{bmatrix}$$

which is the data that *is* available to us.

After some simple manipulation, we get our IC-model prior, $\tilde{X}$, is now given by:

$$\tilde{X} = \phi((Q\phi)^T Q\phi)^{-1}(Q\phi)^T Q\mathcal{X} \quad (8)$$

6.2 TM Prior with Stable-f Model

In our final scenario, we assume that the only IC model parameter that can be obtained directly from measurement is f. Again, we assume we only have ingress and egress node counts available to estimate $\{A_i\}$ and $\{P_i\}$.

Simple algebraic manipulation can give us

$$\tilde{A}_i = \frac{fX_{i*} - (1-f)X_{*i}}{2f-1} \quad (9)$$

We perform a similar exercise to obtain estimates for $\{P_i\}$, namely,

$$\frac{\tilde{P}_i}{\sum_{j=1}^{T} \tilde{P}_j} = \frac{fX_{*i} - (1-f)X_{i*}}{(2f-1)\sum_{j=1}^{T} A_j} \quad (10)$$

With these relationships, we construct a new prior for TM estimation. For each time bin, the most recent ingress and egress counts are used to estimate $\{A_i\}$ and $\{P_i\}$, and these estimates are combined with f according to the stable-f IC model (4) to produce a TM prior.

We find that the IC model yields substantial improvements over the gravity model. Whether using measured ϕ from the previous week or two weeks back [5], improvements are in the range of 10-20%. We thus improve on the gravity model, using only ingress/egress counts to estimate $\mathcal{A}$, and a previous week to estimate f and $\{P_i\}$.Similarly we see improvements of around 8% with the stable-f IC prior as a prior. Even when very little side information is available to the analyst (just an estimate of f), the IC model outperforms the gravity model as a prior for TM estimation. Refer to [5] for detailed results.

7. CONCLUSIONS

In this paper we introduced a new model for traffic matrix data that takes into consideration the bidirectional nature of most Internet connections. We compared our model to another popular traffic matrix model, namely the gravity model. We found that our IC model better fits actual traffic matrix data. This was our first validation of our model. We considered the TM application of traffic matrix estimation, and showed that when TM estimation uses the IC model as a prior, it can do a better job, than with a gravity model prior. This comparison serves as a second validation of our model. In studying the application of our model to our data we learned interesting things about our model parameters; (1) the network-wide ratio of forward to reverse traffic is quite stable over multiple days and weeks; (2) the preferences or popularity of nodes is also stable over multiple weeks. These two findings justify the use of the simplest version of our model for the application of synthetic traffic generation.

This work was supported in part by NSF grants ANI-9986397 and CCR-0325701, and by Intel.

8. REFERENCES

[1] Geant network http://www.geant.net.

[2] Internet Abilene Network http://www.internet2.org.

[3] Optimization Toolbox Matlab http://www.matlab.com/products/optimization/.

[4] Cao, J., Weil, S. V., and Yu, B. Time-Varying Network Tomography. *Journal of the American Statistical Assoc. 2000* (2000).

[5] Erramilli, V., Crovella, M., and Taft, N. An Independent Connection model for Traffic Matrices. Tech. Rep. 2006-022, Computer Science Dept., Boston University, Boston, MA, USA, 2006.

[6] Gunnar, A., Johansson, M., and Telkamp, T. Traffic Matrix Estimation on a Large IP Backbone - a Comparison on Real Data. *Proceedings of ACM/Usenix IMC 2004* (2004).

[7] Lakhina, A., Crovella, M., and Diot, C. Diagnosing Network-wide Traffic anomalies. In *In Proceedings of ACM SIGCOMM* (2004).

[8] Medina, A., Salamatian, K., Taft, N., Matta, I., , and Diot, C. A Two Step Statistical Approach for Inferring Network Traffic Demands. Tech. Rep. 2004-011, Boston University, Computer Science Department, March 2004.

[9] Medina, A., Taft, N., Salamatian, K., Bhattacharyya, S., and Diot, C. Traffic Matrix estimation: existing techniques and new directions. In *Proceedings of ACM Sigcomm 2002* (New York, NY, USA, 2002).

[10] Nucci, A., Sridharan, A., and Taft, N. The problem of synthetically generating IP Traffic Matrices: initial recommendations. *ACM Computer Communication Review 35*, 3 (2005), 19–32.

[11] Roughan, M. Simplifying the synthesis of Internet Traffic Matrices. *SIGCOMM Comput. Commun. Rev. 35*, 5 (2005), 93–96.

[12] Soule, A., Lakhina, A., Taft, N., Papagiannaki, K., Salamatian, K., Nucci, A., Crovella, M., and Diot, C. Traffic Matrices: Balancing Measurements, Inference and Modeling. In *Proceedings of the ACM SIGMETRICS* (2005).

[13] Soule, A., Nucci, A., Cruz, R., Leonardi, E., and Taft, N. How to identify and estimate the largest Traffic Matrix elements in a dynamic environment. *Proceedings of ACM SIGMETRICS* (2004).

[14] Uhlig, S., Quoitin, B., Lepropre, J., and Balon, S. Providing public intradomain traffic matrices to the research community. *SIGCOMM Comput. Commun. Rev. 36*, 1 (2006), 83–86.

[15] Zhang, Y., Roughan, M., Duffield, N., and Greenberg, A. Fast accurate computation of large-scale IP Traffic Matrices from Link Loads. In *Proceedings of the 2003 ACM SIGMETRICS* (New York, NY, USA, 2003), ACM Press, pp. 206–217.

[16] Zhang, Y., Roughan, M., Lund, C., and Donoho, D. L. Estimating point-to-point and point-to-multipoint Traffic Matrices: an Information-theoretic Approach. *IEEE/ACM Trans. Netw. 13*, 5 (2005), 947–960.