An Independent-Connection Model for Traffic Matrices

Vijay Erramilli, Mark Crovella, and Nina Taft *

Sept. 7 2006

BUCS-TR-2006-022

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.

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 timeseries of representative traffic matrices for a given (real or synthetic) 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. Based on these inputs, one would seek to model the entire set of origin-destination (OD) flows in the network.

Although complete TM models do not yet exist, some models for parts of the problem have been developed [23, 20, 14]. One of the most popular models in connection with traffic matrices is the 'gravity'

*V. Erramilli and M. Crovella are with the Department of Computer Science, Boston University; email: {evijay,crovella}@cs.bu.edu. N. Taft is with Intel Research, Berkeley, US; email: nina.taft@intel.com.

model, which estimates OD flow counts from ingress and egress counts. The gravity model has been used extensively in TM estimation (described below) and has been proposed as a tool for synthetic TM generation.

The key assumption underlying the gravity model is *independence* of ingress and egress. The assumption is that the traffic entering the network at any given node exits the network in proportion to the total traffic exiting at each node. 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, the 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 the underlying phenomena that shape traffic flow in a real network.

We seek to go beyond the gravity model towards one that reflects our understanding of underlying network phenomena. In doing so we seek a model that maintains the simplicity of the gravity model, while striving for model parameters that can be given a physical meaning.

To do so, we propose the *independent-connection* (IC) model. In this 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 think of each host (which can be an initiator or a responder) as being associated with a single network access point (although we discuss violations of this assumption later). [1] Then for any given connection, we model its initiator's access point as independent of its responder's access point.

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 traffic, 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" 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, as described in Section 3.

[1] For brevity, we will often refer to a connection as being initiated or responded to "at" an access point, when we mean that it was initiated or responded to by a host whose network ingress or egress is that access point. Likewise, we will often refer to access points as "nodes".

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. We focus on two applications: first, we characterize empirically observed model parameters for a number of real traffic matrices. This yields results that can be used to construct synthetic traffic matrices. Second, we use the model to construct inputs for the problem of TM estimation.

With respect to parameter characterization, we find that forward/reverse ratios and node preference show remarkable stability over time. Values for these parameters show strong similarity from week to week, over a span of up to seven weeks in our data. We find that the fraction of traffic flowing in a forward direction, as a fraction of total connection traffic, is generally in the range of 0.2 to 0.3 in our data; this is consistent with values that can be inferred from other published results. We find that network activity level shows familiar and predictable diurnal patterns, with noticeable changes on weekends. Finally, 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.

With respect to TM estimation, we show that the IC model forms a better starting point than does the gravity model. The TM estimation problem consists of estimating OD flow sizes given link counts, including ingress and egress counts. TM estimation is an under-constrained problem, and the usual approach is to bring in additional constraints; one way to do this is to start with a *prior*, that is, an initial guess at the TM that is a starting point for subsequent refinement.

We show how one can use the stability of node preference and/or forward/reverse ratios to construct priors for TM estimation. If one is able to measure an entire set of OD flows, we show how to extract activity values so that traffic matrices may be estimated from just ingress and egress counts in subsequent weeks. The resulting estimates are more accurate than those obtained using the gravity model. If one is only able to estimate the forward/reverse ratio for the network (a less demanding measurement task) then we show how to estimate both preference and activity values from ingress and egress counts. Remarkably, even these estimates form better priors for TM estimation than does the gravity model.

The paper is organized as follows: in Section 2 we cover related work. Then in Section 3 we discuss why the gravity model does not hold in the Internet, and we define the IC model. Section 4 describes the data we used in our results. In Section 5 we detail the characterization results of the data and discuss methods for generating synthetic traffic matrices using the IC model. Finally, Section 6 discusses the use of the IC model for TM estimation, and in particular deals with how one can infer the parameters involved using only node totals. We conclude in Section 8.

2 Related Work

The gravity model and its variants have been extensively used for the problem of IP network traffic matrix estimation [18, 11, 22]. The authors in [11] propose an approach for TM estimation based on choice models to model PoP fanouts, with a very similar structure to that of the gravity model. They also propose the use of priors as inputs to the estimation step. In [18, 22] the authors use the gravity model as a prior to TM estimation. Further, the work in [22] proposes a simple least-squares approach to estimate TMs called tomogravity. Studies of the effectiveness of different methods of TM estimation can be found in [19, 6].

In [23] the authors propose an information-theory based framework for TM estimation. In that paper, the gravity model is cast as capturing independence between source and destination at the packet level, and therefore is equivalent to a maximum entropy formulation. The gravity model is then used inside a convex optimization problem that combines a minimum mean square error approach with a maximum entropy approach. The idea is thus that among all the traffic matrices that satisfy the link constraints, the method

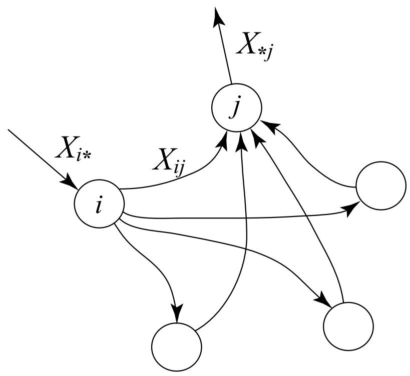


Figure 1: Flows in a Network

picks one closest to the gravity model (see [23] for more details). In fact, the central tenet of our proposed model is that independence between source and destination node at the packet level does *not* hold.

Our work also relates to synthetic traffic matrix generation. This problem has been discussed in detail in [13] and [17]. In fact [17] proposes that the gravity model itself can be used for this purpose, and suggests that the necessary inputs to the gravity model (node ingress and egress counts) can be modeled well using an exponential distribution. In contrast, we argue that the IC model is not only simpler than the gravity model for this purpose, but also more intuitive and flexible.

Our work in studying parameters of the IC model relates to prior traffic characterization results, notably [15] and [12]. In [15] the author proposes empirical models for applications which include FTP, SMTP, Telnet and NNTP based on extensive characterization studies of real TCP header traces. The results reported include the observation that the forward/reverse traffic ratio for Telnet traffic to be approximately 0.05. The authors of [12] propose a tool for analyzing bidirectional TCP connections called TStat. They use the tool to study traces collected on an access link from their university to the Internet. They define a parameter ξ which they call the 'asymmetry' parameter and which corresponds to f in our model. They report that majority of the connections exhibit asymmetry with ξ less than 0.5 most of the time. In addition they report the ξ value for HTTP traffic to be very low, around 0.06, while for P2P applications (Gnutella) ξ is around 0.35. These results are consistent with the f estimations we make in this paper.

3 The Independent-Connection Model

Consider a network with n access points, as shown in Figure 1. The figure simply shows nodes in the network; the links that carry traffic (the network topology) are not shown. 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

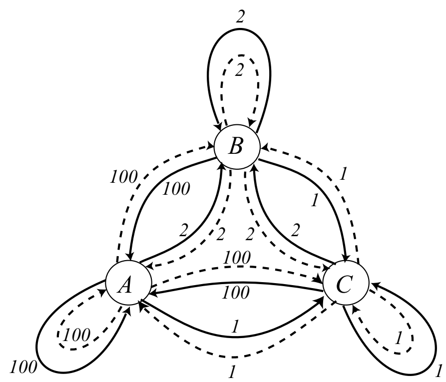


Figure 2: Example Traffic in an IC Setting.

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 $X_{i*}X_{*j}/X_{**}, \forall i, j$.

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. Since most traffic consists of connections, this means that for most packets, I is not well modeled as being independent of E, even if any given connection's initiator node is independent of its responder node.

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 (from initiator to responder) and the *reverse* direction (from responder to initiator). In actual practice forward and reverse volumes are not likely to be equal, but it simplifies the example.

Then consider what happens if some nodes initiate a larger traffic volume than other nodes. This is the situation in the three-node network shown in Figure 2. 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 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,

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

Under the gravity model, these probabilities should all be equal.

3.1 Model Definition

To formalize the notion of independence of connections, we proceed as follows. 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 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:

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

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**:

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

for $i, j = 1, ..., n$. In most of what follows we use versions of the simplified IC model, although we comment on situations in which the general IC model may be more appropriate.

The next aspect of the model to consider is stability of parameters in time. We consider three variants of the simplified IC model which incorporate increasingly restrictive assumptions about temporal stability of parameters. To clarify assumptions about temporal stability we cast model parameters as a function of $t = 1, ..., T$. In the most general case, all parameters may vary at each time step, yielding the **time-varying IC model**:

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

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

We also consider the case in which f shows stability in time. As discussed above (and evaluated in Section 5), at high levels of aggregation we may expect f to show stability in time. This assumption results in the **stable-f IC model**:

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

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

Finally, the most restrictive case we consider is that 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**:

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

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. The different variants of the IC model reflect different amounts of outside information that must be brought into the estimation process. In the case of the stable-fP model as used for TM estimation, one assumes that the stable values of f and P have previously been measured; the set $\{A_i(t)\}$ is then estimated from the data. In the case of the stable-f model, one only assumes that the stable value of f has previously been measured; both $\{A_i(t)\}$ and $\{P_i(t)\}$ are then estimated from the data.

4 Data Details

To explore the validity and utility of the IC model, we examined its applicability to real traffic matrix data. The results in Sections 5 and 6 are based on three data sets:

Géant Data: Géant [2] is a network connecting research institutions and universities across continental Europe. It has 22 PoPs, located in almost all major European capitals. Géant carries research and education traffic as well as commercial traffic. We use three weeks of sampled netflow data from this network, for the period of November 14, 2004 to December 8, 2004. 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. We call this data set **D1**.

Totem Data: The publicly available Totem [21] data set is also from the same Géant network. The data set consists of 4 months of TMs, obtained via netflow sampled flow data, sampled at the rate of 1/1000. We refer to this data set as **D2**. The two differences between **D1** and this dataset is that this data set consists of 23 PoPs; the PoP 'de' in **D1** is split into two PoPs ('de1,' 'de2') in **D2**. In addition, the time bin size is 15 minutes, leading to 672 sample points for each week's worth of data. More information on how the TMs were constructed, including description of measurement anomalies can be found in [21].

Full Packet Header traces from Abilene Backbone: This publicly available data set consists of a pair of two hour contiguous bidirectional packet header traces collected at the Indianapolis router node (IPLS), in the Abilene network[3]. The links instrumented are the ones eastbound and westbound, towards Cleveland (CLEV) and Kansas City (KSCY). The first 44 bytes of the IP header are anonymized. More information on the data can be found at [1]. We refer to this data set as **D3**.

In some cases we present results for a single week from **D1** or **D2**, while in other cases we use three weeks from **D1** or seven weeks from **D2.**

5 Characterizing IC Model Parameters

In this section we characterize empirically obtained estimates of model parameters for the IC model. Our purpose is twofold. First, if the IC model captures important characteristics of traffic matrices, then empirically observed parameter values are useful for generating synthetic TMs. Second, if we can observe stability of certain parameters over time, we can justify the use of the simpler stable-f or stable-fP models rather than the more complex time-varying model.

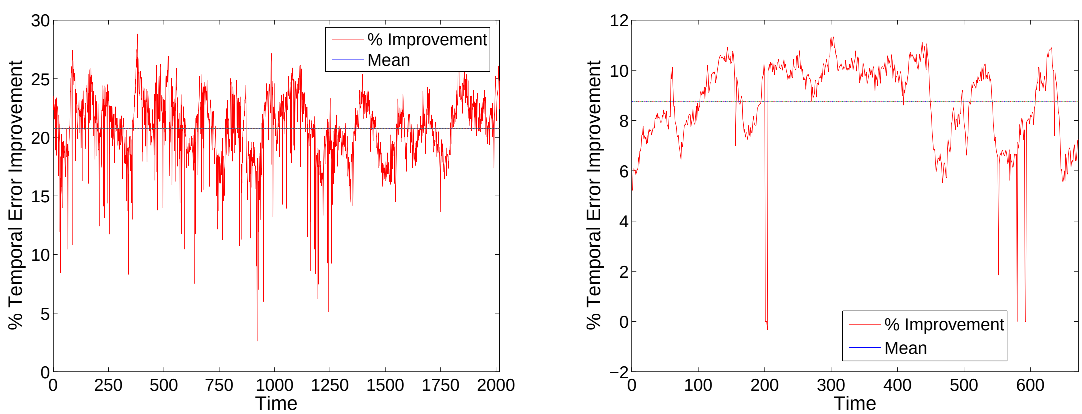
5.1 How Well Can the IC Model Fit Data?

Before beginning to look at parameter characterization, it is useful to simply ask whether the IC model can provide useful outputs. That is, can the IC model succesfully reproduce observed data — better than, say, the gravity model?

In our case we expect the time-varying model to fit the data best, followed by the stable-f model; and we expect the poorest fit from the stable-fP model. This is just 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 freedom, the stable-f model has $2nt + 1$ degrees of freedom, and the stable-fP model has $nt + n + 1$ degrees of freedom.

Since the stable-fP model is expected to give the poorest fit to real data, to be most conservative in our conclusions we focus our attention on that version. The metric we use for measuring accuracy of model prediction here and throughout the paper is *relative l_2 temporal error* (as used for the same purpose in [19]):

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



(a) (b)

Figure 3: **Temporal % Improvements of Model over Gravity (a) Geant (b) Totem**

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 [4] to find the solution numerically. The resulting model fits the data very well. In particular, it fits the data considerably better than the gravity model, even though the gravity model has about twice as many degrees of freedom. We estimate the traffic matrix using both models and plot the improvement in $RelL2_T(t)$ of the IC model over the gravity model as a function of t.

This plot is shown in Figure 3, for one week taken from $D1$ and one week from $D2$. The plot shows that for the Géant dataset, the stable-fP model improves on the gravity model by about 20-25%, and for the Totem dataset, the IC model improves on the gravity model by about 6-8%.

5.2 Characterizing f

The results in the previous section show that, for some settings of f and $\{P_i\}$, the stable-fP model can fit real traffic data quite well. This motivates us to examine what values of these parameters are appropriate for modeling real traffic. Further, it suggests that it is worthwhile to investigate whether these parameter values in fact appear to be stable in time. In this section we ask these two questions as regards the model parameter f.

The parameter f_{ij} denotes the fraction of traffic in the set of connections initiated by i and responded to by j. Observing which end of a connection is the initiator is unreliable in sampled netflow. Ideally, to perform a thorough study of observed values of f for a given network, one would need unsampled netflow traces, or unsampled packet header traces of all traffic in the network. To the best of our knowledge no such data sets exist; the small number of traffic matrices that are available are based on sampled netflow records.

Although at the current time we cannot characterize f_{ij} over all (i, j) for any network, we can measure it for two particular (i, j) pairs in Abilene using dataset **D3**. In particular, we can measure it for the pairs *(KSCY,IPLS)* and *(IPLS,KSCY)* since all of the packets corresponding to these connections (presumably) flow over the IPLS-KSCY link — for which **D3** provides full packet traces.

To estimate f_{ij} using **D3** we proceed as follows:

- For a given pair of trace files, form connections by matching flows between the two links that have corresponding 5-tuples.

- Determine the amount of traffic on the link from i to j contained in connections that are initiated from node i and for which a response was found on the link from j to i. We identify the initiator of a connection as the sender of the TCP SYN packet. Call this traffic I_i.

- Determine the amount of traffic on the link from i to j contained in connections initiated at j. Call this traffic R_i.

- Proceed analagously to measure I_j and R_j.
- Classify the remaining traffic as unknown. In our experiments, the volume of unknown traffic was less than 20% of total traffic; but this is somewhat misleading, since connections that started before the beginning of the trace are classified as unknown.
- f_{ij} is computed as $\frac{I_i}{I_i+R_j}$.

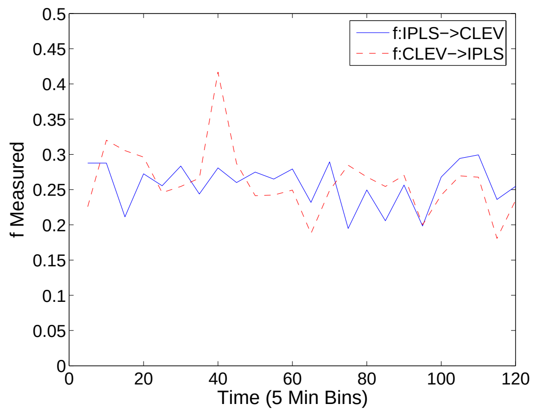


Figure 4: **f for IPLS-CLEV and CLEV-IPLS over Time**

We perform this procedure for each 5-minute timebin in the two-hour trace **D3**. The results are shown in Figure 4. We draw three conclusions. First, values of f in the range 0.2 to 0.3 seem reasonable: traffic that has a large fraction generated due to Web browsing should have an f value well below 0.5 (since HTTP response sizes are generally much greater than HTTP request sizes). Second, the value of $f_{(CLEV,IPLS)}$ is quite similar to the value of $f_{(IPLS,CLEV)}$. This provides some preliminary support for assuming spatial stability of f_{ij} over different pairs (i, j). Finally, in both cases, the values of f_{ij} are fairly stable in time, generally staying in the range of 0.2 to 0.3 at all times.

We can also ask the same questions about the estimated values of f obtained in successive weeks using the method of Section 5.1. We find very similar results: namely, f values close to 0.2 that are quite stable over time. In Figure 5 we demonstrate this for seven weeks from dataset **D2**. The results for **D1** (not shown) are quite similar.

We note that analysis yielding f values was performed in [12]. That study used a specially-developed tool to analyze properties of bidirectional TCP connections on a link connecting a university to the Internet; the tool was also able to distinguish connection types by application. That study found that traffic from initiator to responder was quite asymmetric with that from responder to initiator. For some applications it is highly asymmetric, such as for Web traffic and FTP traffic, for which the equivalent f value was close to 0.05. In the case of other applications, the asymmetry was not as great, such as for Gnutella and other P2P traffic, for which f was close to 0.35.

The significant differences of f across different applications suggest that observed f values are strongly affected by application mix. On the other hand, the temporal stability of f values in all our measurements suggest that the effect of application mix does not change rapidly on a week to week timescale.

5.3 Characterizing $\{P_i\}$

Next we turn to investigating the empirical values of $\{P_i\}$ and their stability over time. Again we look at successive weeks, where the $\{P_i\}$ values were obtained for each week using the method in Section 5.1.

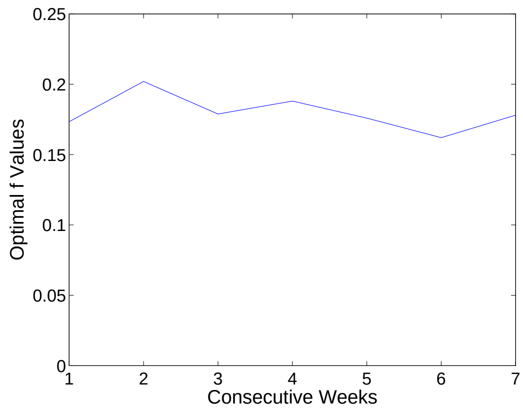
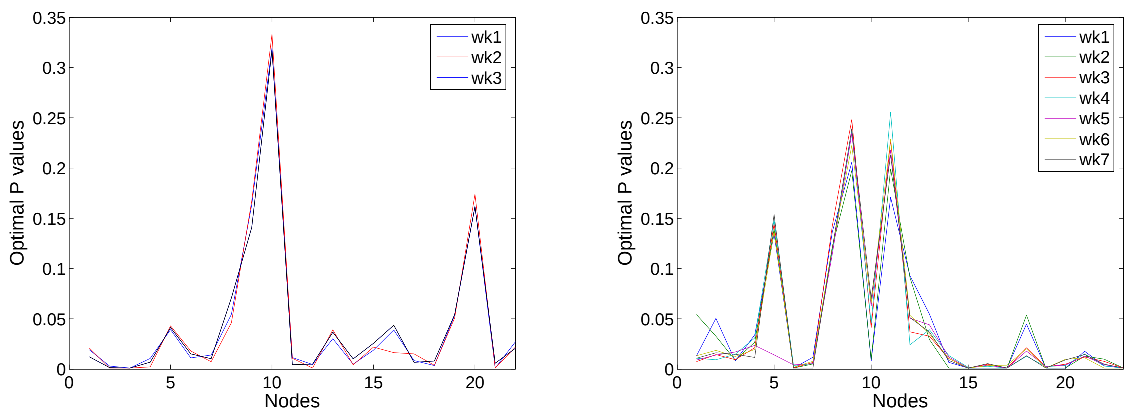


Figure 5: f **over Time, Totem**



(a) (b)

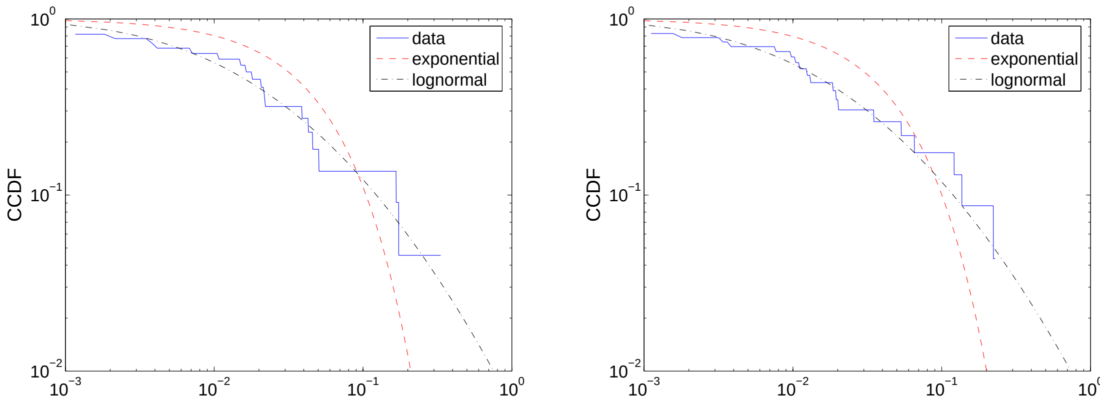
Figure 6: **Optimal** P **Values over Time (a) Géant (b) Totem**

We show results for datasets **D1** and **D2** in Figure 6. Each figure plots the values of $\{P_i\}$ for nodes i in arbitrary order; the $\{P_i\}$ sum to 1. We make two observations from the figure. First, values of P_i for any given node i are remarkably stable over time. Even over seven weeks in the case of **D2** there is very little variation over time. In combination with results in the previous section, this lends support to the use of the stable-fP model.

Second, the $\{P_i\}$ values are highly variable; most are small, but a few are quite large (as much as ten times greater than the typical value).

The high variability of $\{P_i\}$ values prompts us to examine their distributional tail. In Figure 7 we show the log-log complementary distribution function of $\{P_i\}$ values for one week each from **D1** and **D2**. For comparison purposes, we also show analytical curves for the best fit obtainable using an exponential and a lognormal model.

The figure confirms our intuition that the distribution of $\{P_i\}$ values shows a long tail. 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. However the long-tailed lognormal distribution clearly does a better job at approximating the tail shape of this dataset than does the exponential. For both datasets, the maximum likelihood estimates of the lognormal parameters were $\mu \approx -4.3$ and $\sigma \approx 1.7$.



(a) (b)

Figure 7: **CCDF Optimal P Values (a) Géant (b) Totem**

To gain further insight into how P_i values vary across the network, it is helpful to gauge P_i against the traffic volume X_{*i} at node i. This comparison is shown in Figure 8. This shows that the amount of traffic exiting the network at a particular node is not a very good indication of the node's preference level. While nodes with small amounts of traffic must necessarily have low preference levels, among the nodes with greater than a median level of traffic there seems to be little correlation between traffic volume and preference level.

5.4 Characterizing $\{A_i(t)\}$

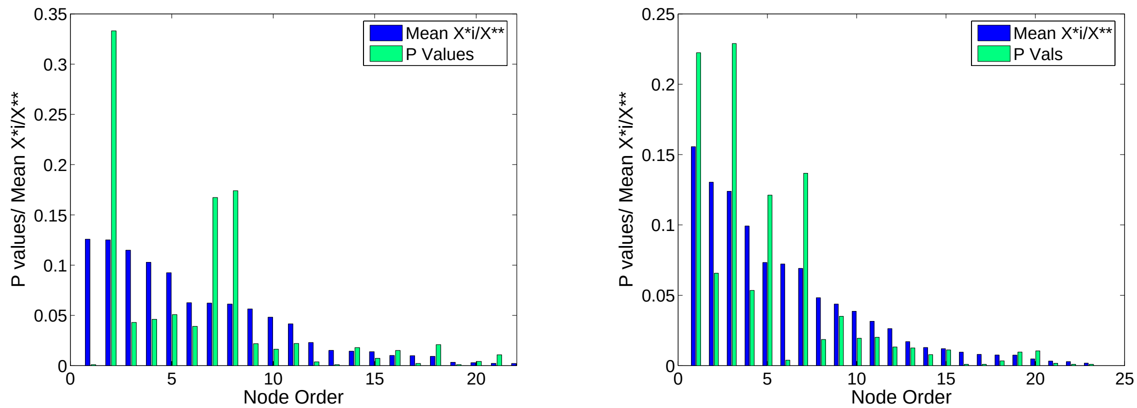
Finally we examine the nature of the ensemble of timeseries $\{A_i(t)\}$ estimated from our data, measured in bytes. The variation in these values is the source of all time-variation in the stable-fP model. Intuitively, if $\{A_i(t)\}$ represents the rate at which traffic is being 'initiated' at node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

In Figure 9 we show the time series for $A_i(t)$ values corresponding to 3 nodes in each network: the node with the largest mean $A_i(t)$ value over the week, a node with an intermediate mean $A_i(t)$ value, and a node with one of the smallest mean $A_i(t)$ values.

We make a number of observations from this figure. First, there are strong periodic patterns in these timeseries, corresponding to daily variation as well as to reduced activity on the weekend. Second, the timeseries corresponding to higher activity levels show a more distinct and pronounced pattern, which is consistent with the aggregation of a higher number of users responsible for the activity level.

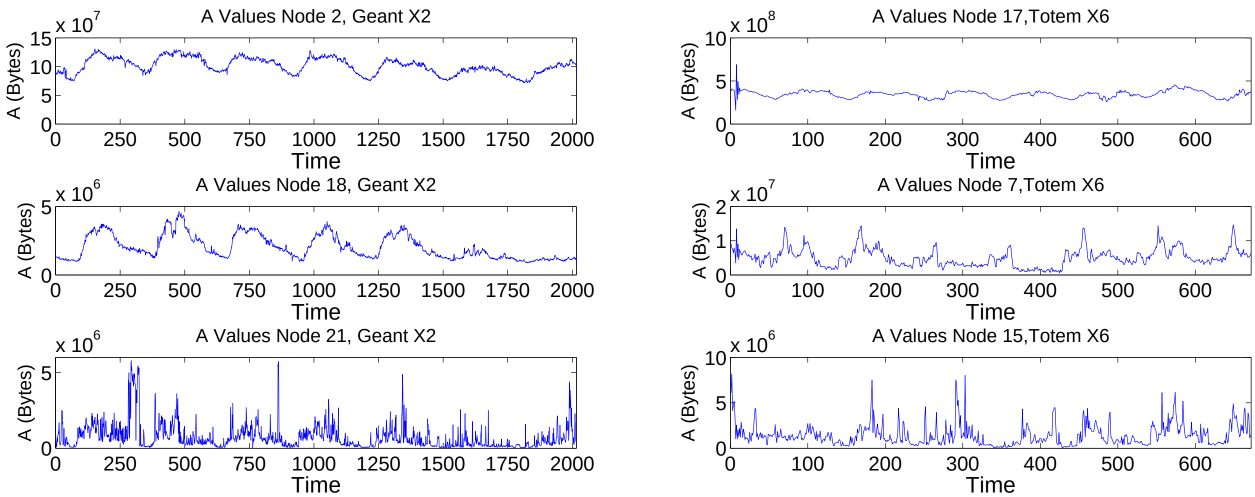
To gain further insight, we examined whether preference levels $\{P_i\}$ showed correlation with mean activity levels $\{A_i(t)\}$. Our analysis showed no evidence of correlation. This adds some confidence that preference levels and activity levels are measuring different things.

Considerable previous work has examined the properties of traffic on week-long timescales, e.g., [20, 8, 9]. In particular [20] has proposed a cyclo-stationary model for traffic variation on this timescale. The cyclo-stationary approach models traffic as the superposition of a limited number of periodic waveforms. We note that the cyclo-stationary model may be suitable for describing the timeseries of $A_i(t)$ for each i as well, although we leave this for future work.



(a) (b)

Figure 8: **Comparison of Optimal P values with Normalized Egress Counts (a) Geant (b) Totem**



(a) (b)

Figure 9: A **Time Series, Largest, Medium-Sized and Smallest Node (a)Géant (b) Totem**

5.5 Using the IC Model for TM Generation

Developing a comprehensive framework for synthetically generaing traffic matrices is a non-trivial task [13]. 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:

- Choose an f value. Our results suggest that a reasonable value may be in the range 0.2 to 0.3.
- 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.
- Generate activity time series $\{A_i(t), i = 1, ..., n\}$. A model that explicitly incorporates daily varia-

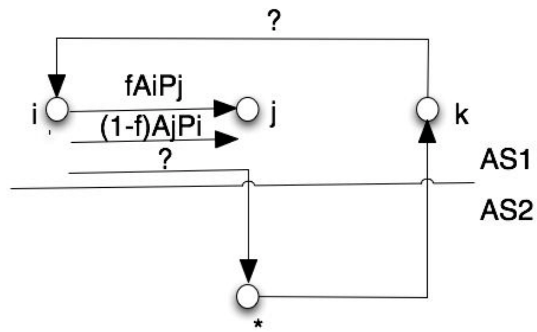


Figure 10: Problem with Asymmetry

tion, such as the cyclo-stationary model [20], may be reasonable.

- Construct the timeseries of traffic matrices $X_{ij}(t)$ using equation (5).

There are a number of advantages to this approach to TM generation. First, the parameters have an interpretation in terms of real network phenomena. Thus, if an analyst wishes to incorporate knowledge of traffic mix, or explore the effects of changes in traffic 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 Section 3, 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 [17]. In the gravity model, the set of inputs $\{X_{i*}(t)\}$ and $\{X_{*j}(t)\}$ are *causally* related. All traffic entering the network must leave the network at some point, so the sum of the $X_{i*}(t)$s must equal the sum of the $X_{*j}(t)$s. However, the relationship between $X_{i*}(t)$ and $X_{*j}(t)$ for any given i and j is not easily captured. 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.

5.6 Issues with the IC Model

Finally, we consider some ways in which the IC model is not representative of real network phenomena.

The simplified IC model, in which f_{ij} is constant for all node pairs (i, j) in the network, is likely to be inaccurate in the presence of routing asymmetry. Routing symmetry has been widely studied and can arises due to a number of reasons [16]. In particular, 'hot potato' routing among peer ASes that connect at multiple points may result in the forward traffic for a connection leaving the network at a node different from where the connection's reverse traffic enters.

The problem is illustrated in Figure 10. The figure shows two ASes (AS1 and AS2) that connect at two points, j and k. Assume we want to model the traffic flowing from i to j, i.e., X_{ij}. The simplified IC model will treat this as shown in the figure: $X_{ij} = fA_iP_j + (1 - f)A_jP_i$. However, some connections initiated by node i are responded to by a host in AS2 denoted $*$. Traffic flowing from $*$ to i does not return via node j, but rather via node k. Thus f_{ij} is somewhat higher than f_{ji}.

In practice, the results in this section show the the simplified IC model yields a good fit to our data. Thus it appears that routing asymmetry in the Géant network is not so severe as to invalidate the simplified IC model. We leave it to future work to determine whether routing asymmetry requires use of the general IC model in other networks.

6 Using the IC Model in 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 extremely well studied. Posed as an inference problem, it can be briefly stated as follows. The relationship between the traffic matrix of a network, the routing and the link counts can be described by a system of linear equations $Y = Rx$, where Y is the vector of link counts, x is the traffic matrix organized as a vector, and R denotes a routing matrix in which element R_{rs} is equal to 1 if OD pair s traverses link r or zero otherwise. The elements of the routing matrix can have fractional values if traffic splitting is supported. In networking environments today, Y and R are readily available; the link counts Y can be obtained through standard SNMP measurements and the routing matrix R can be obtained by computing shortest paths using IGP link weights together with the network topology information. The problem at hand is to estimate the traffic matrix x. This is not straightforward because there are many more OD pairs (unknown quantities) than there are link measurements (known quantities). This is manifested by the matrix R having less than full rank. Hence the fundamental problem is that of a highly under-constrained, or ill-posed, system [19, 23].

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}
- 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 [11, 5, 19, 23]) select different methods to carry out steps 1 and 2, while step 3 remains the same across many solutions. A common solution for Step 1 is to construct an initial estimate – a 'prior' of the traffic matrix – using the gravity model and available measurements.

The sensitivity of different estimation techniques to the quality of initial priors has been studied [10, 6]. These results indicate that the quality of the initial prior is quite important and thus a better initial prior can lead to substantially better estimates. Given that the IC model appears to be more accurate than the gravity model, we hypothesize that by using the IC model to improve upon the solution for Step 1, any procedure for Step 2 can yield better TM estimates.

A key question in using the IC model concerns what sorts of network measurements are used in estimating model parameters. We consider three scenarios. First, we consider the case in which all parameters are continually estimated online. In this scenario, we imagine that the operator has the capability of measuring

$\{A_i(t)\}$, $\{P_i\}$, and f. This is a less demanding task than full netflow collection, and so represents a reasonable solution. From an accuracy standpoint, this is the best scenario and allows us to measure the maximum gain the IC model can provide over the gravity model.

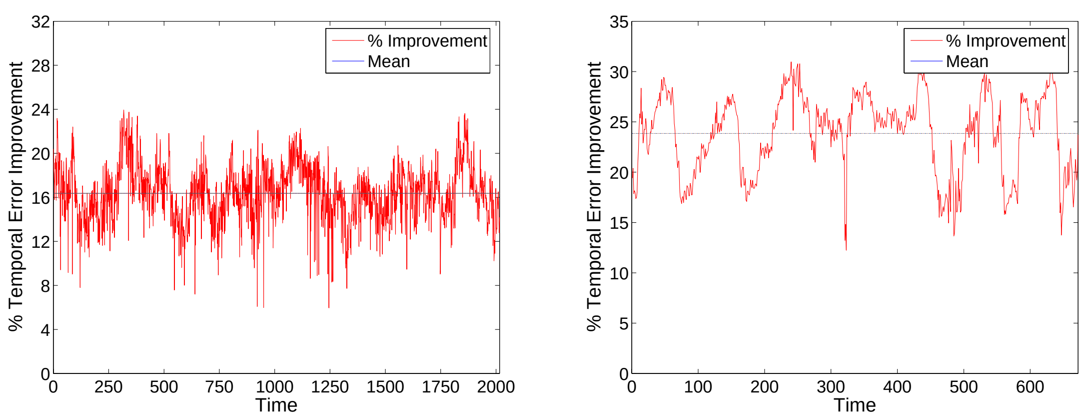
However, continuous online measurement may not be desirable, and so we consider alternatives that do not suppose continuous collection of any data other than the ingress and egress counts used in the gravity model. In the next scenario, we assume that f and $\{P_i\}$ have been measured at a previous point in time. In this case we explore whether we can exploit the observed stability of f and $\{P_i\}$. We show that even in this case, the stable-fP IC model yields good improvements over the gravity model as a prior for TM estimation.

Finally, we consider the case in which only f is known or measurable. In this case we show that the stable-f model still yields improvements over the gravity model.

6.1 IC Model with Measured Parameters

To approximate the case in which IC model parameters are continuously measured, we make use of the parameter settings computed via the optimization procedure in Section 5.1. To do this we use one week's worth of traffic matrix data. We use this same data to produce an IC-model prior and a gravity model prior. For Step 2 of the TM estimation procedure we use the least-squares estimation techniques proposed in [22] (also known as the tomo-gravity approach). We run the estimation procedure twice, keeping steps 2 and 3 the same in both cases, while varying the prior used in Step 1.

To compare the TM estimates produced, we compute the relative L2 temporal errors according to Equation (6) between the estimates and our data, for each test run. These errors are plotted in Fig. 11 for two different datasets. In the Géant dataset the time bins are 5 minutes (i.e., we produce an estimate every 5 minutes), and thus in Fig. 11(a) the x-axis reaches 2000 time units (each time unit is 5 minutes). The Totem data (Fig. 11(b)) uses bins of 15 minutes, and thus with an underlying time unit of 15 minutes, less than 700 estimates are produced in one week. For the Géant data, the improvement typically falls between 10-20% whereas in the TOTEM data, the improvement was mostly in the range of 20-30%. For estimation purposes, this is a large improvement.



(a) (b)

Figure 11: **Temporal Improvements of TM Estimation over Gravity - All Optimal (a) Géant (b) Totem**

We point out that we are using the simple gravity model herein, and the authors of [23] 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 generalized gravity model prior because such additional information was not available to us. In the Totem data, this information is lost because the per link flow records have been transformed directly into OD flow data. In the Géant data, the data is presented at the router level (not the link level which is needed for the generalized gravity model); moreover the notion of peering traffic here is not exactly the same as that in the generalized gravity model because peers here include other university networks and the business relationship is not the same as in ISP networks.

This experiment was conducted as a sort of thought experiment for us to understand the bounds of the gain the IC model can achieve over the gravity model. If f, $\{P_i\}$ and $\{A_i\}$ are obtained from flow records, then this scenario is not desirable, because the TM itself could be derived from these records. However it is not hard to imagine a method that measures our parameters directly from each access point, without using flow records. In this case, the above scenario becomes useful.

6.2 TM Prior with Stable-fP Model

Another possibility for obtaining measurements for our parameters occurs in the types of hybrid scenarios proposed in [19] 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. This would allow use of the stable-fP IC model.

In this case we use one week's worth of data to estimate values for IC model parameters f and $\{P_i\}$. As we have shown in Section 5, these seem to be stable and hence we hypothesize that these parametes estimates can be used for estimation in subsequent weeks. To facilitate discussion, consider a scenario spanning two weeks. 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:

$$\mathcal{X} = \phi \mathcal{A} \tag{7}$$

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

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

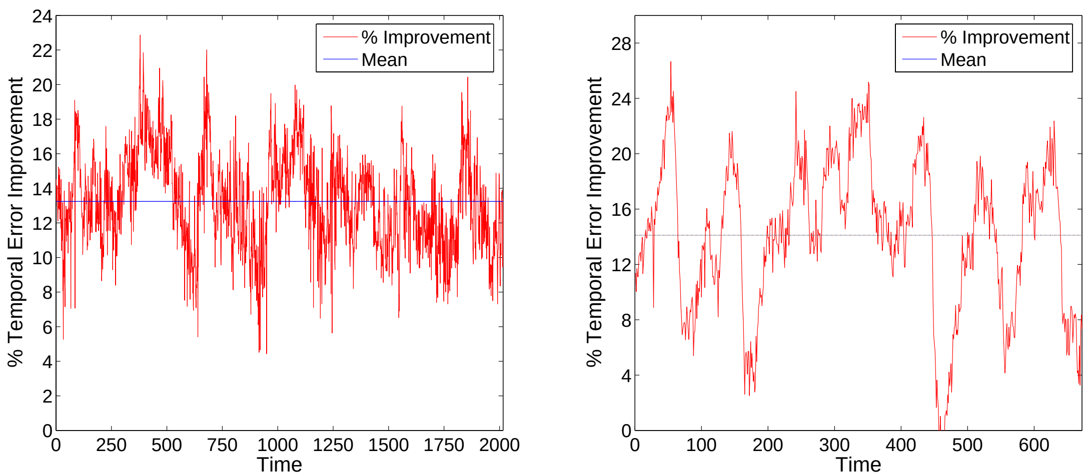
which is the data that *is* available to us.

Then to estimate $\mathcal{A}$, we premultiply both sides of (7) by Q and then use the pseudo-inverse of $Q\phi$. That is, we construct the following estimate:

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

Intuitively this gives us an estimate for $\mathcal{A}$ that yields a $Q\phi\mathcal{A}$ closest to the ingress and egress counts $Q\mathcal{X}$ in a least-squares sense. Our IC-model prior, $\tilde{X}$, is now given by:

$$\tilde{X} = \phi((Q\phi)^T Q\phi)^{-1}(Q\phi)^T Q\mathcal{X} \tag{9}$$



(a) (b)

Figure 12: **Temporal Improvements TM Estimation over Gravity, f and P available (a) Géant (b) Totem**

To evaluate this approach we run the TM estimation procedure using this prior, and compute the percent improvement of this procedure over the one that uses a gravity model prior. In the case of Géant, we used the previous week's data to compute f and $\{P_i\}$. To explore the effect of using even older measurements, for the Totem data, we computed f and $\{P_i\}$ using data from two weeks prior to the week being estimated.

The results are depicted in Fig. 12. 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, improvements are in the range of 10-20%. Thus we can substantially improve on the gravity model, using only ingress/egress counts to estimate $\mathcal{A}$, and a previous week to estimate f and $\{P_i\}$.

6.3 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 $\{P_i\}$.

Using the stable-f IC model in (4) (and omitting the time index on X, A and P) we sum over all j to get

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

$$X_{i*} = f A_i + \frac{(1-f)\sum_{j=1}^{T} A_j P_i}{\sum_{i=1}^{T} P_i} \tag{10}$$

Similarly, summing over i, we get

$$X_{*j} = \frac{f \sum_{i=1}^{T} A_i P_j}{\sum_{i=1}^{T} P_i} + (1-f) A_j$$

Using Equations (10,6.3) together (and interchanging indices i and j in (6.3)) we can eliminate the P_i terms, yielding the following estimates for $\{A_i\}$:

$$\tilde{A}_i = \frac{f X_{i*} - (1-f) X_{*i}}{2f - 1} \tag{11}$$

We perform the same exercise to obtain estimates for $\{P_i\}$, namely,

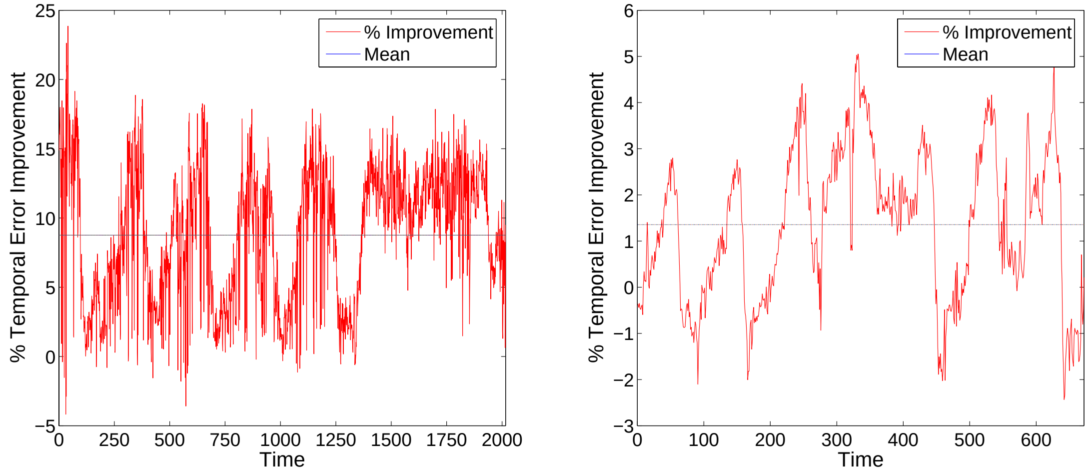
$$\frac{\tilde{P}_i}{\sum_{j=1}^{T} \tilde{P}_j} = \frac{f X_{*i} - (1-f) X_{i*}}{(2f-1)\sum_{j=1}^{T} A_j} \tag{12}$$

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.

The results are shown in Figure 13. This figure shows the percent improvement of TM estimation with the stable-f IC prior over TM estimation with a gravity prior. For the Géant data, we still see considerable gains of around 8%. However, the percentage gain for the Totem data was less significant, somewhere between 1-2%. We conclude that even when very little side information is available to the analyst (just an estimate of f), the IC model is nonetheless preferable to the gravity model as a prior for TM estimation.

7 Conclusions

Good TM models are useful for studying many problems related to network operations. The most widely used model today for TMs is the gravity model, which is based on the assumption of independence of source and destination at the packet level. We have shown that this assumption does not hold in the Internet, as most packets are part of a two-way connection. Thus the the amount of traffic which a source sends to the destination has some dependence on the amount of traffic which the destination sends to the source.



(a) (b)

Figure 13: **Temporal Improvements TM Estimation over Gravity, using Stable-f IC model.**

In the paper we present a new model for traffic matrices, called the independent-connection model, that is based on the independence of connections rather than of packets. We construct this model based on three key notions of underlying network behavior. The first notion is that an OD flow consists of two types of traffic: forward traffic initiated by the origin and reverse traffic initiated by the destination. The second notion is that every node has an *activity* level associated with it which is the rate that traffic is being generated due to activity of users at that node. The final notion is that each node has an associated *preference* level which corresponds to the likelihood of connecting to that node, and does not depend on activity levels. With these concepts we compose a family of models that differ in the assumptions they make about parameter stability in space and time. The model has the appealing property that its parameters have a natural interpretation that aids "what-if" investigations.

Using real TM data, we first demonstrate that the IC model, in spite of being simpler than the gravity model, does a better job in fitting the data. We go on to characterize empirically estimated parameter values and show that two model parameters (forward/reverse ratio and node preference) show remarkable stability over time, while the third parameter (activity level) shows strong periodic behavior. We discuss how these results can lead to improved methods for generating synthetic traffic matrices.

We then show how the IC model can be used as a prior for the TM estimation problem. We show that the temporal stability of model parameters means that measurements of parameter values made in a previous week can be used in the current week to yield significant improvements in TM estimation accuracy as compared to the gravity model.

8 Acknowledgments

We would like to thank Anukool Lakhina for help with the data and for helpful discussions on a preliminary version of this work. We are also grateful to R.A. Martin and Michael Menth for very detailed feedback on a draft version, as well as to the anonymous reviewers for IMC 2006 whose comments made this work better. This work was supported in part by NSF grants ANI-9986397 and CCR-0325701, and by Intel.

References

[1] Abilene-1 OC48c Backbone Traces http://pma.nlanr.net/traces/long/ipls1.html.

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

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

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

[5] CAO, J., WEIL, S. V., AND YU, B. Time-Varying Network Tomography. *Journal of the American Statistical Assoc. 2000* (2000).

[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] LAKHINA, A., PAPAGIANNAKI, K., CROVELLA, M., DIOT, C., KOLACZYK, E. D., AND TAFT, N. Structural Analysis of Network Traffic Flows. In *Proceedings of ACM SIGMETRICS / Performance 2004* (June 2004), pp. 61–72.

[9] LELAND, W. E., TAQQU, M. S., WILLINGER, W., AND WILSON, D. V. On the self-similar nature of Ethernet Traffic (extended version). *IEEE/ACM Trans. Netw. 2*, 1 (1994), 1–15.

[10] 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.

[11] MEDINA, A., TAFT, N., SALAMATIAN, K., BHATTACHARYYA, S., AND DIOT, C. Traffic matrix estimation: existing techniques and new directions. In *Proceedings of Sigcomm 2002* (New York, NY, USA, 2002).

[12] MELLIA, M., CIGNO, R. L., AND NERI, F. Measuring IP and TCP behavior on edge nodes with Tstat. *Comput. Networks 47*, 1 (2005), 1–21.

[13] 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.

[14] PAPAGIANNAKI, K., TAFT, N., AND LAKHINA, A. A Distributed Approach to Measure IP Traffic Matrices. In *Proceedings of ACM IMC/Usenix* (2004).

[15] PAXSON, V. Empirically derived analytic models of wide-area tcp connections. *IEEE/ACM Trans. Netw. 2*, 4 (1994), 316–336.

[16] PAXSON, V. End-to-End Routing Behavior in the Internet. *IEEE/ACM Trans. Netw. 5*, 5 (1997), 601–615.

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

[18] ROUGHAN, M., GREENBERG, A., KALMANEK, C., RUMSEWICZ, M., YATES, J., AND ZHANG, Y. Experience in measuring backbone traffic variability: models, metrics, measurements and meaning. In *Proceedings of IMW ’02* (New York, NY, USA, 2002), ACM Press, pp. 91–92.

[19] 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).

[20] 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).

[21] 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.

[22] 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.

[23] 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.