The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations

Antonio Nucci
Narus Inc.
Mountain View, CA
anucci@narus.com

Ashwin Sridharan
Sprint ATL
Burlingame, CA
ashwin@sprintlabs.com

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

ABSTRACT

There exist a wide variety of network design problems that require a traffic matrix as input in order to carry out performance evaluation. The research community has not had at its disposal any information about how to construct realistic traffic matrices. We introduce here the two basic problems that need to be addressed to construct such matrices. The first is that of synthetically generating traffic volume levels that obey spatial and temporal patterns as observed in realistic traffic matrices. The second is that of assigning a set of numbers (representing traffic levels) to particular node pairs in a given topology. This paper provides an in-depth discussion of the many issues that arise when addressing these problems. Our approach to the first problem is to extract statistical characteristics for such traffic from real data collected inside two large IP backbones. We dispel the myth that uniform distributions can be used to randomly generate numbers for populating a traffic matrix. Instead, we show that the lognormal distribution is better for this purpose as it describes well the mean rates of origin-destination flows. We provide estimates for the mean and variance properties of the traffic matrix flows from our datasets. We explain the second problem and discuss the notion of a traffic matrix being well-matched to a topology. We provide two initial solutions to this problem, one using an ILP formulation that incorporates simple and well formed constraints. Our second solution is a heuristic one that incorporates more challenging constraints coming from carrier practices used to design and evolve topologies.

Categories and Subject Descriptors:
C.2.5 [Computer Communications]: Local and Wide Area Networks - Internet C.4 [Performance of Systems]: Modeling Techniques H.1 [Information Systems]: Models and Principles.

General Terms:
Measurement. Design. Performance.

Keywords:
Traffic Characterization. Internet Traffic Matrices. Topology. Hypothesis Testing. Distribution Fitting.

1. INTRODUCTION

A traffic matrix is a network-wide description of the total traffic volume carried within a domain. Each element of such a matrix describes the volume of traffic that originates at one node and is destined for another, and corresponds to an origin-destination (OD) pair. The traffic matrix represents the demands and does not say anything about how the traffic is routed. Once a routing algorithm or load balancing scheme is applied to route the traffic matrix over a given topology, then the resulting link loads network-wide can be easily determined. The link traffic in turn impacts the queueing or scheduling performance at the downstream node. Because the distribution of link loads and behavior of router queues are the common metrics in performance studies, the traffic matrix provides a critical input to research efforts related to performance assessment.

There are a multitude of network design tasks whose performance evaluation requires a traffic matrix (TM) as input. Examples include the evaluation of: routing protocols, load balancing schemes, IGP link weight setting algorithms, reliability and failure analysis, multipath routing, capacity planning, bottleneck avoidance mechanisms, fairness schemes, the mapping of IP logical links to physical fibers, and so on. The performance assessment of any of these schemes is typically done either via simulation or computation, both of which need some mechanism to describe the anticipated network-wide load.

Why do we need a traffic matrix for all of these tasks? Consider the task of evaluating the performance of a particular routing scheme across an entire network. With a traffic matrix as input, one then applies the routing method under test, and the resulting loads on all links in the networks are determined. Common performance metrics for routing algorithms include average or maximum link load as well as the full distribution of link loads. One typically wants to assure that the maximally loaded link is below some threshold, or that the distribution of link loads is not excessively skewed. The same evaluation approach would be used for assessing a load balancing scheme, multipath routing, or link weight selection algorithms [9, 12] (for IGP protocols such as OSPF or IS-IS). This need is illustrated by two studies on link weight selection [23, 24] that already used traffic matrices for evaluating their algorithms. In one study [24] the authors had a proprietary traffic matrix available, while the other study [23] the authors focused on finding link weights that work well for a wide variety of traffic matrices.

Understanding network performance under failure conditions is another key example of when traffic matrix inputs are needed. Designing a network to handle failures can involve, for example, developing algorithms to compute backup paths, or designing the topology to include redundant equipment, or designing a routing protocol to find alternate paths quickly. In each of these cases, one needs to know where the traffic carried by the failed link will end up, and the resulting link loads after failure. One cannot do any of this without a network-wide description of the traffic demands. The same kind of performance assessment is done to evaluate the mapping of IP logical links to physical fibers [22]. When fiber cuts occur, traffic is moved around and the mapping determines the ensuing load on the remaining fibers. Having a traffic matrix would enable fairness studies to evaluate fairness at the OD pair level and network-wide. Finally, the capacity planning task for an evolving network is one that adds capacity into the network based upon current traffic loads (the TM) and predicted future loads.

Most research studies conducted on these topics make use of synthetically generated traffic matrices to evaluate the resulting performance of the scheme being designed. Because almost no data has been published on actual traffic matrices, researchers have had to resort to arbitrary assumptions, some of which can be quite unrealistic. For example, the most common assumption is that the origin-destination flows are uniformly distributed. Clearly researchers haven't had any choice before because neither actual traffic matrices, nor any of their statistical properties, have been available to them. The majority of research on how to estimate traffic matrices has come from carriers [2, 3, 5, 8, 10]. Because carriers view their traffic matrices as proprietary, such data is typically not published, and thus not available to the research community at large. The one source of traffic matrix data that is indirectly available comes from the Abilene network [4]. The Abilene community has collected flow-level statistics from its routers. They don't provide the traffic matrix itself, however it is computable from the Juniper flow sampling data. To the best of our knowledge the statistical properties needed for synthetic traffic matrix generation have not yet been extracted from Abilene's traffic matrix.

A few characteristics of OD flows are known. In [10, 14] it was shown that OD flows from a Tier-1 backbone exhibit strong diurnal patterns, thus indicating that highly aggregated OD flows are cyclo-stationary. These large 24-hour swings, illustrating some of the sources of variability in traffic matrices, imply that considering static traffic matrices for a performance evaluation task is likely to be insufficient. It was shown in [5] that gravity models are coarse but decent enough estimates of a traffic matrix so as to provide good initial conditions for optimization procedures that estimate traffic matrices from partial information. This implies that gravity models are also good candidates to consider for synthetic traffic matrix generation (although we do not consider them in this work).

Our goals in this paper are multiple. First, we discuss, and hope to raise awareness of, the many issues involved in generating synthetic traffic matrices. This task turns out to contain many subproblems depending upon the complexity and granularity of the desired traffic matrix, and how well the traffic matrix needs to be matched to the underlying topology that carries the traffic. Second, we provide some initial solutions for generating synthetic traffic matrices. Our solutions are based on two datasets, one collected from Sprint's European backbone and one collected from the Abilene network. Third, we will illustrate that there really is no justification for using uniform distributions to generate TMs. Fourth, we hope that our paper will encourage researchers to conduct studies in this area that we consider important for the general performance evaluation community. There exist few datasets of complete TMs, and they are typically proprietary. Thus if the end vision of this research area is to provide the community with a group of models for synthetic TM generation, it will require separate teams each working with their respective datasets.

There are two basic steps in synthetic TM generation, each of which can involve a set of problems. The first step is to generate traffic load levels for OD pairs. There are many possible outcomes from this step depending upon the type of traffic matrix desired. The simplest traffic matrix to generate is a *static traffic matrix*. A static TM represents one instance of a traffic matrix pertaining to a particular time interval. We need to randomly generate a set of traffic volumes (one for each OD pair) such that the set obeys a chosen probability distribution. This is where the uniform distribution has been applied before. This traffic volume typically represents an average volume level as it is defined over a time interval (5 mins, 1 hr, 1 day, etc.). We use our data sets to explore which distributions could be used to described hourly measured averages. Note that this is a spatial distribution as it describes mean values *across* OD flows. A more realistic TM to generate is a *dynamic traffic matrix*. A dynamic TM is essentially a three-dimensional object in which each matrix element (OD pair) is actually a time series. This thus requires using a temporal model for the dynamic evolution of an OD pair in time. A dynamic TM can evolve either in the stationary regime (if the evolution lasts less than 1 or 1.5 hours) or in the cyclo-stationary regime (if a multi-day TM is desired). Different models are needed for these two regimes.

The second basic step, or problem area, is that of assembling the traffic rates into a matrix such that the resulting TM is both feasible and well-matched to a given topology. After generating either a static or dynamic TM in the first step, all one has is a collection of random numbers. If a static TM is generated one has a set of numbers that represent the mean rates for the OD flows. If a dynamic TM was generated, one has a set of streams, i.e. a time series for each OD flow. For simplicity of exposition, we summarize this second problem for the static TM case. Each traffic volume (or mean rate) now needs to be assigned to a particular node pair. As we will discuss, this cannot be done at random or the constructed traffic matrix could be either infeasible to support or ill-matched to the underlying topology.

There exists a rich problem space in that of matching a TM to a topology so that they are well matched. Being "well matched" means at a minimum, that we must ensure no link capacities are exceeded once the traffic matrix is routed on a particular topology. This can be viewed as a hard constraint as it defines feasibility. However, there are other soft characterisitics of being well matched that come from how ISPs design and evolve their topologies to match changing traffic loads. Examples of soft requirements for this problem include: link loads not being excessively skewed, limiting the maximum link load, and not assigning large flows to nodes that are intended to be used primarily as backup during failure episodes. The relationship between a traffic matrix and a topology is complex because in existing and evolving networks, the TM influences the topology and vice versa. We further discuss this relationship in Section 6. In the case when a feasible construction of the TM cannot be found for a given set of flows, we discuss how to scale down the TM while preserving the previously generated statistical characteristics.

In this work, we provide solutions for generating a static TM, a stationary dynamic TM, and for the placement problem. Our approach uses distribution fitting via hypothesis testing for generating static TMs, and calibration of temporal models for dynamic TMs. We provide two solutions for the placement problem, an ILP-based solution and an heuristic algorithm that incorporates some of the softer constraints mentioned above. We leave the case of dynamic TM evolution through the non-stationary regime for future work.

In Section 2 we describe the general problem and its many facets. Section 3 summarizes the data used and our overall approach. In Section 4 we study the problem of fitting the empirical spatial distributions of mean OD pair rates to known probability distributions. Handling traffic fluctuations is covered in Section 5. In Section 6 we describe the placement problem and present two solutions along with a simple validation. Section 7 concludes the paper.

2. PROBLEM DESCRIPTION

In general, a traffic matrix should be viewed as a dynamic entity. Let $X(i,j,t)$ denote the average bit-rate from source i to destination j at time t, where $1 \le i,j \le N$ in a network with N nodes, and t is a time index. The underlying unit of the temporal index can be anything; we often use a time unit of one hour. For a given OD pair (i,j), the description $X(i,j,t)\ \forall t$ gives a time series capturing the temporal evolution of the OD pair. It is known that traffic matrices exhibit strong diurnal patterns and are typically cyclo-stationary with a 24 hour cycle [10]. To capture the evolution of an OD flow over a multi-day period, this work proposed the following *dynamic cyclo-stationary model*,

$$X(i,j,t) = x(i,j,t) + W(i,j,t) \tag{1}$$

where $x(i,j,t)$ is a deterministic process that captures the diurnal cyclo-stationary behavior, and where $W(i,j,t)$ captures random fluctuations. The deterministic term can be modeled, for example, using either a Fourier expansion or from approximation signals of a wavelet transform. In [10] they also showed that typically a small number of basis functions are needed (e.g., around 5). The random term is modeled as a zero mean random process whose variance needs to be determined.

This dynamic cyclo-stationary description is the most general case of a traffic matrix. We have found that the Sprint TM is stationary within one hour periods. This agrees with the general understanding in the community that backbone IP traffic is typically stationary with a period of 1 to 1.5 hours. This general model can be easily simplified to describe the *dynamic stationary model*,

$$X(i,j,t) = \bar{x}(i,j,T) + W(i,j,t) \tag{2}$$

To be clear about the time indices, let t have 10 minutes as its time unit, while T has a time unit of 1 hour. Throughout a given hour T, the mean of the OD flow is given by the constant term $\bar{x}(i,j,T)$. During that hour, values for the (i,j)-th flow are generated every 10 minutes, i.e., $X(i,j,t)$ for $(T-1) \leq t \leq T$. The variations from one 10 minute time epoch until the next are determined by the term $W(i,j,t)$ (which recall has zero mean).

In the case of a static traffic matrix, defined for a particular time interval T, a TM should be populated with a single fixed value per OD pair. This thus provides a single instance of a TM. Our base model can by further simplified to capture the *static traffic matrix model* by setting

$$X(i,j,T) = \bar{x}(i,j,T) \tag{3}$$

Note we continue to use the bar notation ($\bar{x}$) because our intent in this case is to generate static TMs at an hourly time unit. Clearly this single value will represent an average over the 1 hour time interval. We use 10 minutes as our smallest time granularity because our data was available at that time unit; i.e., we have TM measurements every 10 minutes. We use 1 hour as our time unit for the static TMs and the dynamic stationary case, because we found our data to be stationary for that period oftime.

There are two basic steps or problems in synthetic TM generation are depicted in Figure1. The main problems for step 1 have to do with specifying the parameters of the three models above in Eq.'s (1-3). In our approach, we generate the more detailed TMs by building upon the model for the simpler TMs. In other words, our stationary dynamic TM generation incorporates parameters assigned to the static TMs and then generates additional parameters as needed. (Similarly, the cyclo-stationary TM builds on up that stationary TM.) The output of step 1 could be a TM at any one of these three granularity levels, depending upon what the researcher needs.

To specify the model parameters we need to do the following. For the basic static TM in a network with N nodes, we need to generate either N^2 or $N(N-1)$ fixed values representing the mean rates for the OD flows. Put alternatively, we need to generate the values for $\bar{x}(i,j,T)\ \ \forall i, \forall j$ for a given period T, as in Eq. (3). Call this step $1a$. Note that only $N(N-1)$ values are generated in the case that no self traffic is included (i.e., $X(i,i,T)$). This scenario may be more relevant, for example, if the task at hand is to solve the

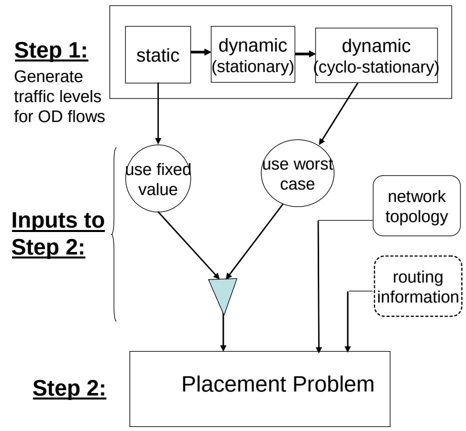

Figure 1: Problem Description

link weight selection problem. To generate a stationary dynamic TM, we use the mean rates generated in step 1a for the first term of the model in Eq. (2). We now need to generate values for the variance $\sigma(i,j)$ of the random fluctuations terms $W(i,j,t)$ for all i,j. We assume $E(W)=0$. Call this step $1b$. To generate a cyclo-stationary TM, a full model for $x(i,j,t)$ as in Eq. (1) needs to be specified (step $1c$). If a Fourier expansion is used this would imply specifying all the basis functions. Whichever model is used, it can incorporate those parameters already determined in steps 1a and 1b. In order to carry out step 1a, we need to understand something about the *spatial* properties among flows, while for steps 1b and 1c, an understanding of the *temporal* properties of flows is needed.

The second step in the overall methodology (as depicted in Figure 1) is that of assigning the OD values (or streams) to particular node pairs in a topology. We refer to this as the *synthetic traffic placement problem*. The output of step 1 is the input to step 2. As mentioned earlier, step 1 merely outputs a list of numbers that are disassociated from origin and destination nodes. If a static TM was generated in step 1, then the placement problem can use the fixed mean rates as input. If a dynamic non-stationary TM was produced in the first step, then either the mean or the maximum of each OD flow time series can be used as input. The task is to assign these flows to the a topology such that the TM is "well-matched" to the topology. The placement problem can take different forms depending upon the amount of information available to the researcher. We assume that the researcher always has, at a minimum, a topology that is given and specifies the connectivity and link capacities. If a set of link weights (i.e., to determine the routing) can be obtained for this topology, one can do a better job of matching the TM to the topology. The routing information in Figure 1 is depicted in a dashed box to indicate that this input may or may not be available. For some networks, these are available from the Rocketfuel project at the University of Washington [19].

The OD flows should not simply be "placed" on a topology, i.e., assigned to node pairs, in a purely arbitrary fashion. At a minimum, the OD flows once routed, cannot lead to a situation in which the link capacities are exceeded. We can state this capacity constraint more formally as follows. For a given assignment of rates to OD-pairs, let X' represent the traffic matrix in column vector format. For a specified routing, let A denotes the routing matrix, where element $a_{ke} \in A$ represents the fraction of traffic for OD flow k that is routed over link e. Then, the vector of link loads Y is determined by a simple linear transform of the OD flows, namely $Y = AX$. If C represents the capacity vector with $c_e \in C$ equal to the capacity of link e, we require that the traffic placement leads to an X' that yields link loads that satisfy $Y \leq C$.

In real ISPs, the topology and traffic matrix are related in that the topology is often designed to meet the needs of the TM. So as the TM evolves, the topology may be updated. For example, when a designer identifies a portion of the topology that is a bottleneck during failure, another node might be added in the portion. This node should probably not be assigned as the source or destination of a large OD flow since its primary purpose it not to connect customers but instead to provide resiliency. To the extent possible, solutions to the placement problem should try to incorporate such carrier design principles. In general, we believe the placement problem to be the most challenging issue of TM generation.

This problem differs from the traditional traffic matrix estimation problem because in that problem the input is a topology, a routing, link capacities and link loads. The goal is to infer the mean rates of the OD flows. The assignment of flow rates to node pairs is predetermined by the link loads that are known to come from specific links. In our case, the inputs are the mean rates, a topology, a routing, and link capacities. We do not have link loads as input, and the mean rates are given. Our goal is to assign these rates to particular node pairs such that the resulting TM is both feasible and reasonably matched to the topology. We do not consider the case when the topology itself can be altered to accommodate a potential TM. We do however consider both the cases of when the routing is and isn't available.

3. METHODOLOGY

3.1 Measurement Data Studied

The measurements used in this study come from two different backbone networks. Sprint-Europe (called Sprint hereafter) is the European backbone of a US Tier-1 ISP. It has 12 Points of Presence (PoPs) and carries commercial traffic for large customers (companies, local ISPs, etc). Abilene is the Internet2 backbone network and is made up of 11 large routers spanning the continental US. The traffic on Abilene is non-commercial, arising mainly from major universities in the US. We converted flow measurements into a PoP-to-PoP matrix for Sprint's backbone, and into a router-to-router matrix for Abilene's backbone. In both cases, we aggregated the measurements to 10 minute intervals. For the Sprint data, we have three weeks of data, for the Abilene network we have 1 week at our disposal.

3.2 Approach

Our approach to these problems can be summarized as follows. In order to generate N^2 (or $N(N-1)$) values for $\bar{x}(i,j,T)$ for a given T, we use our empirical data to see which probability distribution best describes the ensemble of values $\{\bar{x}(i,j,T)\}$. This is a spatial distribution defined across flows. Once such a distribution along with its parameters is identified, it can be used to generate random numbers to populate the static TM. We processed our empirical data for this purpose as follows.

We know that our data is stationary within one hour intervals. It was also shown in [11] that a TM in a particular one hour slot (e.g., 12-1pm) is quite similar to the TM in that same time slot on the next day. We show here, more specifically, that the time series of 12-1pm measurements from many days, concatenated together, yeilds a stationary time series. In Figure 2, the top plot illustrates the cyclostationary behavior of a sample OD flow. We have highlighted with vertical bars the 12-1pm time slot within each day.

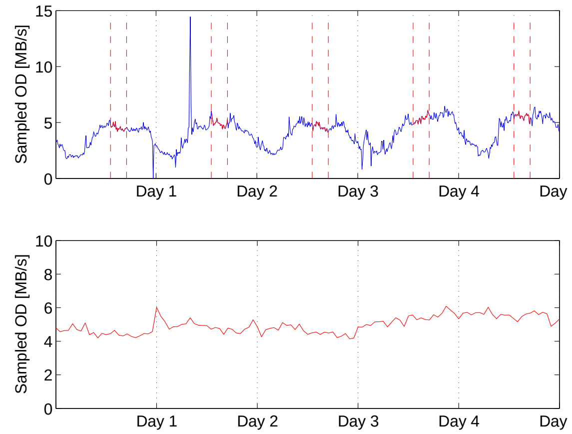

Figure 2: Example of cyclo-stationary and stationary behaviors

The lower plot is a zoomed in image, showing just the 12-1pm slot, concatenated over multiple days. This lower plots depicts a stationary process. (Similar plots can be shown for Abilene data.) We selected the 12-1pm data for the majority of our mean fitting tasks because this represents the peak hour traffic. This captures the worst case traffic matrix which is an appropriate one to consider for many traffic engineering tasks. Indeed capacity planning and reliability analysis, are currently carried out by many of today's ISPs using worst case loading scenarios. We fit the distribution of these sample mean values to a probability distribution using hypothesis testing and goodness-of-fit metrics. We wish to point out that although we present here the results for the peak hour time interval, we have carried out the same fitting procedures for other hours of the day. Our findings on the types of distributions that are good fits and bad fits are consistent over all hours of the day.

In order to populate a dynamic TM in the stationary regime, we need to understand the standard deviation behavior of each flow. We propose that to generate such a TM, one mean value for each OD pair should be generated as above. This mean should be used throughout the hour in question. At each 10 minute interval, a fluctuation or noise term should be added to the mean value $\bar{x}(i,j,T)$. This noise term can be generated using a Gaussian random variable with zero mean and a standard deviation that needs to be determined [10]. We find this standard deviation by examining the relationship between the mean and variance of each flow. It has been postulated in the past that the mean and standard deviation have a power law relationship in which $\bar{x}(i,j,T) = \psi\sigma(i,j,T)^{\gamma}$ [1]. According to our data, this rule holds reasonably. We use our datasets to calibrate the parameters (ψ, γ). Thus once the mean $\bar{x}(i,j,T)$ is determined as above, we can determine σ according to this power law relationship, and then use it to generate noise terms at each instance during the stationary regime.

Although we have described here the most general problem of generating cyclo-stationary TMs, we leave as future work the identification of specific model fittings. From [10], it is clear that one approach would be to use a Fourrier expansion with 5 basis functions to describe $x(i,j,t)$, and then the task is to determine the coefficients of the basis functions.

Our approach to the synthetic traffic placement problem is to devise simple solutions that are guided by standard metrics or well known properties of ISP topologies. For example, large PoPs (with either many links or a lot of capacity) are likely to be the source and/or destination of the largest OD flows. Hence, one of our techniques guides the mapping by preferentially assigning large traffic rates to OD-pairs with large fan-out/fan-in capacity. An alternate way to approach the problem is by observing that this relation is motivated by the need to reduce congestion. Towards this end, we also propose an Integer Linear Program to produce a mapping that minimizes the maximum congestion in the network. Our ILP solution is guaranteed to find a solution that satisfies the link capacity constraints if such a solution exists.

4. CHARACTERIZING MEAN RATES OF ORIGIN-DESTINATION FLOWS

Our goal in this section is to determine which probability distributions, and their parameters, are suitable to be used as random number generators to populate a static traffic matrix (i.e., the problem in step $1a$ of Section 2). We use the Sprint and Abilene measured traffic matrices, each of which represents a set of average OD flow values $\{\bar{x}(i,j)\},\ \forall i\ \forall j$. For each flow, we extracted each 10 minute sample from the peak hour (12-1pm) of each day in the trace (3 weeks for Sprint and 1 week for Abilene). All these samples of each flow are averaged to compute one rate value for each flow; the ensemble of the mean rates for all OD flows are used to create the dataset whose distribution we which to capture. We thus seek a spatial distribution as this is a description across flows.

4.1 Approach to Distribution Fitting

Our approach to distribution fitting and evaluating the goodness of fit for any specified distribution is based on hypothesis testing. In hypothesis testing, we formulate a null hypothesis H_0 such as "the distribution is lognormal with unknown parameters". An alternate hypothesis H_a is also given, such as "the distribution is not lognormal". We carry out distribution fitting and use maximum likelihood estimation methods to estimate the parameters of the distribution being tested. The next step is to determine whether to accept or reject the null hypothesis.

According to the Neyman-Pearson paradigm [20] a decision as to whether or not to reject H_0 in favor of H_a is made on the basis of a *test statistic* (typically a single number) computed on the empirical data and the fitted model. The set of values of the test statistic for which H_0 is accepted and rejected are called, respectively, the *acceptance region* and the *rejection region*. We reject the hypothesis if the test statistic (TS) is larger than a number called the *critical value* (CV). Hence the rejection region is defined by $TS > CV$, and the value CV separates the rejection region from the acceptance region. One of the mistakes we can make when deciding whether or not to accept H_0 is the following: we reject H_0 when it is true. Let α denote the probability that we make this kind of mistake; α is called the *significance level* of the test. The rejection region (and hence critical value) is defined for a given probability α under the null hypothesis.

We used two test statistics, the *Chi-Squared* (C-S), and the *Kolmogorov-Smirnov* (K-S) tests as goodness-of-fit measures to provide an indicator of the quality of the fitting. These statistics measure how well the distribution fits the input data and how confident we can be that the data could have been produced by the fitted distribution. It is well known that there is no hard rule to decide which test gives the best result. Each test has its strengths and weaknesses, and is focused on a different part of the distribution.

The *Chi-Squared test* (C-S) is the oldest and best known goodness-of-fit test. To calculate the chi-squared statistic, we must first break up the range of the empirical data (in this case the mean OD rate) into several "bins". The chi-squared statistic is then defined as:

$$\chi^2 = \sum_{b=1}^{B} (O_b - W_b)^2 / W_b \tag{4}$$

where B represents the number of bins, O_b the observed number of samples in bin b and W_b the expected number of samples in bin b of the distribution under test. A weakness of the chi-squared statistic is that there are no clear guidelines for selecting the number and location of the bins. A common approach to remove some arbitrariness of the bin selection is to use equi-probable bins.

The *Kolmogorov-Smirnov test* (K-S) is defined as:

$$D_n = sup[|F_m(x)| - \hat{F}(x)] \tag{5}$$

where m denotes the total number of sample points, $\hat{F}(x)$ the fitted cumulative distribution function, M_x the number of sample points less than x and $F_m(x) = M_x/m$. The K-S statistic does not require any binning. It is focused on the center of the distribution and therefore its weakness is that it does not detect tail discrepancies very well.

Each test statistic described above provides an indicator of the quality of the fitting of the specified distribution, and is thus called a *goodness-of-fit indicator*. This indicator reports a measure of the deviation of the fitted distribution from the input data: hence the **smaller** its value, the **better** the achieved fitting. To answer the question "how small a value is needed for a good fit?" we use the critical values. If the test statistic is less than the critical value, then we accept the hypothesis. The critical value is defined for a specific significance level α. The significance levels have a duality with confidence intervals [20]. If we accept the hypothesis for a given critical value associated with α, it means that we believe there is only an α percent chance that we are making a mistake using this decision criteria; put alternatively, we are $1 - \alpha$ confident in our decision. The final metric we use to assess a fit is the p-value which is an indicator of the probability of a good fit; the p-value can be defined as the $Pr(TS > c)$ for some value c. If $c > CV$, then $p < \alpha$. Put alternatively, the p-value is the smallest value of α for which the null hypothesis will be rejected.

4.2 Initial Observations on Our Data

We begin by looking at the empirical data for Sprint and Abilene in Figure 2. The two graphs on the left show the empirical density functions. We see that both of these datasets have a large proportion of their flows that are small. However the Sprint data set has a much larger fraction of its flows (nearly 55%) under the 1 MBps threshold. In fact, there are plenty of flows in the KBps range; this is more visible in the top right plot depicting the empirical cumulative density functions for both networks. The range between the smallest and largest OD pairs in Sprint's data spans roughly 7 orders of magnitude; while the Abilene data spans approximately 4 orders of magnitude. Finally in the lower right-hand plot, we examine what fraction of the OD pairs constitute 95% of the total network-wide traffic. The OD pairs are ordered from smallest to largest along the x-axis. The OD pairs to the right of the solid vertical line constitute 95% of Sprint's traffic. This includes roughly 1/3 of the OD flows. Similarly in the Abilene network, the flows to the right of the dashed vertical line contribute to the top 95% of traffic load; in Abilene's case this corresponds to roughly 1/2 of the OD pairs. These datasets appear to have the classic elephants and mice characteristic. We will see how these characteristics influence the task of finding a probability distribution that could have generated such a set of mean flow rates.

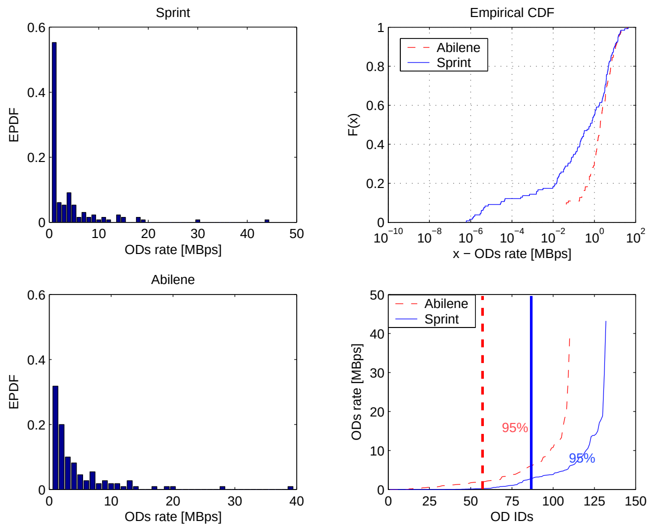

Figure 3: Empirical density functions (left) for Sprint (on the top) and Abilene (on the bottom). Empirical CDF for two datasets (top right). Traffic sent by OD pairs sorted by mean rate (bottom right)

4.3 Fitting the Empirical Means for Sprint

Using the fitting procedure described above, we tried to fit the Sprint data to about a dozen well known distributions. The first interesting result we found is that none of these distributions examined provided a "reasonable" fitting. They all failed the hypothesis test, and the p-values were below 0.02, indicating a very poor fit. Some distributions provided a good fit for the tail of the empirical density function, but produced a poor fit in the middle of the distribution; other distributions yielded the reverse.

We believe one reason this occurs is due to the numerical difference between the smallest and largest OD pairs that spans so many orders of magnitude. As we already saw in Figure 3, for the Sprint data there is a huge mass of very small flows, and seven orders of magnitude difference between the largest and smallest. Note that we could not find any decently fitting distribution, even though we tried some logarithmic distributions such as lognormal and loglogistic. We thus decided to separate the OD flows into two groups, one for the smaller flows, and one for the rest (including medium and large flows), and to carry out distribution fitting separately for these two empirical data groups. To separate the flows into two groups, we need to select a cutoff threshold. This threshold should partition the flows such that the two subsets each contain a reasonable number of sampled points and such that the two resulting empirical datasets can be efficiently fitted by well-known distributions. We tested many thresholds such that the larger group had flows spanning two, or three or four orders of magnitude. For all of these, the data in the resulting two groups can be fitted to a known distribution. This indicates that although we do need a separation threshold, the specific value of the threshold can be flexible. For the remainder of the results in this paper, we use a threshold of 1 Mbps. This was selected because the resulting group of medium and large OD flows constitute 95% of all the traffic load. In this way we capture most of the traffic via a single distribution.

A summary of our findings, for the medium and large OD flows in the Sprint-Europe network, is given in Table 1. We found three distributions that consistently passed the hypothesis test. These were the *lognormal*, *loglogistic*, and *inverse gaussian*. A random variable X has a lognormal distribution if the random variable $\ln X$ has a normal distribution. The density of a lognormal distribution is given by,

$$f_{LogNormal}(x;\mu,\sigma) = \frac{1}{x\sqrt{2\pi}\sigma} exp\{\frac{-(\ln x-\mu)^2}{2\sigma^2}\} \quad (6)$$

where μ is the scale parameter and σ is the shape parameter.

Similarly, a random variable X has a loglogistic distribution is the random variable $\ln X$ has a logistic distribution. For ease of exposition, we state the logistic distribution with location parameter μ and scale parameter σ, that is given by,

$$f_{Logistic}(x;\mu,\sigma) = \frac{e^{\frac{x-\mu}{\sigma}}}{\sigma\{1+e^{\frac{x-\mu}{\sigma}}\}^2} \quad (7)$$

The density of the inverse gaussian distribution is specified by,

$$f_{InvGauss}(x;\mu,\lambda) = \sqrt{\frac{\lambda}{2\pi x^3}} exp\{-\frac{\lambda}{2\mu^2 x}(x-\mu)^2\} \quad (8)$$

By "consistently" passing the test, we mean that these three distributions repeatedly passed the hypothesis test, even when we moved the threshold separating small from medium/large flows around, or when we normalized the data different ways, or when we varied the binning when computing the CS test statistic.

All of our distribution fitting was carried out for a significance level $\alpha = 0.05$. For this significance the critical value for the C-S tests in Table 1 was 14.07. The C-S test statistics for the Lognormal, Loglogistic and Inverse Gaussian distributions were all below this critical value. Thus we have no reason to reject a hypothesis that suggests any one of these three distributions is a fit for the data, and hence all three of these are acceptable. Although there are no hard and fast rules as to what value of 'p-value' constitutes a "good" fit, often anything larger than 0.1 is considered acceptable [20]. While all three of these distributions have p-values above 0.1, the values in the table indicate that the loglogistic is the least good fit with respect to the Chi-Squared test. All three of these distributions also perform well with respect to the K-S test, yielding very low values of the K-S test statistic and high corresponding p-values. Different test statistics examine different parts of the distribution, therefore it is not surprising that the CS test and the KS test yield different results for the ranking of these distributions (e.g., the CS test says the lognormal distribution is best, while the KS test suggests the loglogicstic fit is best). However, the important point is that all of these distributions do well with respect to both tests.

We also include the performance of the uniform distribution since this has been used in many research studies. It is clear that the uniform fails the hypothesis test, and performs extremely poorly. Our calculations generated p-values on the order of $10^{-}9$ or smaller, and thus we simply represent such values in the table as zero. At this point, we do not wish to promote any one of the three successful distributions over the other, nor any particular value of their parameters. Instead we wish to illustrate that these types of distributions are a far better choice than the uniform distribution for generating sample mean values of OD flows.

In Figure 4 we provide another perspective on the quality of the fittings. In this plot, we can see visually the quality of the fit of any of these three distributions. The curves are difficult to distinguish because they are very close to one another, and are all close to the data points. The grey line with steps in it corresponds to the original data. We also include the 95% confidence bounds for the lognormal distribution. We can see that all of the empirical data points lie within these bounds.

We now look at the fitting of the small OD flows (see Table 2). For these cases the critical value for the C-S statistic was 11.07.

Distributions Fitted	Parameters	C-S		K-S	
		Test Value	*p-value*	*Test Value*	*p-value*
LogNormal	$\mu = 15.45$,$\sigma = 0.885$	4.7	0.702	0.097	0.591
LogLogistic	$\mu = 15.43$,$\sigma = 0.509$	6.9	0.22	0.079	0.819
InvGaussian	$\mu = 7.73$x10^6,$\lambda = 6.91$x10^6	6.2	0.511	0.089	0.688
Uniform	$a = 1.06$x10^6,$b = 4.43$x10^7	142	0	0.57	0

Table 1: Sprint data. Results of fitting the Medium-Large OD pair data. Critical value for C-S tests is 14.07.

Distributions Fitted	Parameters	C-S		K-S	
		Test Value	*p-value*	*Test Value*	*p-value*
LogNormal	$\mu = 11.25$,$\sigma = 2.02$	10.1	0.071	0.124	0.336
Gamma	$a = 0.56$,$b = 4.18$x10^5	2.7	0.73	0.069	0.95
Weibull	$a = 1.87$x10^5,$b = 0.69$	9.1	0.1	0.073	0.92
Uniform	$a = 168$,$b = 8.8$x10^5	72	0	0.42	0

Table 2: Sprint data. Results of fitting the Small OD pair data. Critical value for C-S tests is 11.07.

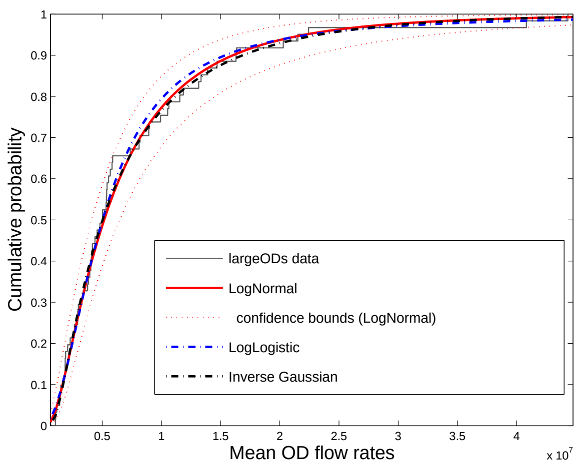

Figure 4: Empirical and fitted CDFs for Medium/Large OD Flows in Sprint data.

The lognormal distribution is borderline in that technically is passes the hypothesis test (CS test statistic of 10.1 is less than the critical value), however the corresponding p-value is quite small, plus the performance in terms of the KS test is borderline. The loglogistic and inverse gaussian distributions were also borderline but were just on the inside of the rejection region (thus results excluded due to space constraints). The two best fits we obtained were with the Gamma and Weibull distributions. The Weibull distribution does well for the KS test but is also borderline for the CS test. Only the gamma distribution does well with respect to both test statistics. Again we include the results for a uniform fit to illustrate the complete failure of this fitting. These small OD flows are harder to model because some are so small that they essentially constitute noise. We leave as future work the investigation as to why the Gamma distribution yields the best fit.

4.4 Fitting the Empirical Means for Abilene

We applied the same methodology to fit the means of the Abilene TM to numerous distributions. Since the mean OD flow rates in the Abilene data only span 4 orders of magnitude, rather than 7 as in the Sprint data, we tested our hypotheses about the different distributions on the entire set of OD flows from Abilene. As before, a significance level of 0.05 was used for all these tests. We see in Table 3, that the lognormal distribution is best with respect to the CS statistic, while the loglogistic distribution is best for the KS statistic. (The critical values are given in the table captions.) We observed the same behavior for the Sprint data. However, the lognormal distribution is the only one that does well for both statistics, as the loglogistic is borderline for the CS test.

Unlike for the Sprint data, we find here that the inverse gaussian no longer performs very well. To explore this we carried out the same approach of separating out the smaller flows. Using a cutoff threshold of 1 Mbps, we include enough flows so as to constitute 90% of the total network-wide load. Table 4 shows that by doing this we obtain a better fit for the three distributions. We remind the reader that although we primarily report the statistics for four distributions, each time we tested them, we consider all twelve. The details of the other fittings are not included because they all consistently produced very poor fittings.

We point out that Tables 3 and 4 together indicate that the lognormal distribution is more robust in the sense that it can fit the empirical data in a wider variety of scenarios (with or without the cutoff threshold). We also wish to point out that the results reported in these four tables were consistent (in terms of general ranking and passing the hypothesis tests) for a variety of binning scenarios.

A visual perspective on the fittings is given in Figure 5. In this figure we put the x-axis values on a log scale. We can see the good fit of the lognormal and loglogistic distributions. We can also see how the inverse gaussian has difficulty in the middle of the distribution.

4.5 On Using Uniform Distributions

It is clear visually from Figure 3, that the mean OD flow values, for both Sprint and Abilene, could not have been generated from a uniform distribution. We have also seen in Tables 1-4 that the uniform distribution consistently fails the hypothesis test, and performs extremely poorly for the KS test. The hypothesis that a uniform distribution fits the data will reject for a significance level less than 0.01. This indicates that by rejecting the uniform distribution, the likelihood that we are making a mistake is less than 1%. We can also illustrate visually how poorly a uniform distribution fits these empirical datasets. Figure 6 shows this for the Sprint-Europe data while Figure 7 shows this for the Abilene data. This is not surprising, as it is clear from Figure 3 (as we mentioned earlier) that elephants and mice exist in this traffic, implying that some kind of heavy-tail distribution is more likely to be a better de-

Distributions Fitted	Parameters	C-S		K-S	
		Test Value	*p-value*	*Test Value*	*p-value*
LogNormal	$\mu = 16.6, \sigma = 1.04$	9.2	0.75	0.0546	0.8534
LogLogistic	$\mu = 16.6, \sigma = 0.56$	14.6	0.33	0.0356	0.9975
InvGaussian	$\mu = 2.56\text{x}10^7, \lambda = 1.44\text{x}10^7$	15.73	0.26	0.12	0.05
Uniform	$a = 6.1\text{x}10^5, b = 1.5\text{x}10^8$	590	0	0.583	0

Table 3: Abilene Data. Results for fitting the mean rate of *all* OD flows. OD flows span 4 orders of magnitude. Critical value for C-S tests is 22.36.

Distributions Fitted	Parameters	C-S		K-S	
		Test Value	*p-value*	*Test Value*	*p-value*
LogNormal	$\mu = 16.6, \sigma = 1.04$	5.3	0.62	0.0546	0.8534
LogLogistic	$\mu = 16.7, \sigma = 0.48$	5.5	0.59	0.046	0.966
InvGaussian	$\mu = 2.7\text{x}10^7, \lambda = 2.6\text{x}10^7$	2.3	0.80	0.03	0.99
Uniform	$a = 3.5\text{x}10^6, b = 1.5\text{x}10^8$	325	0	0.583	0

Table 4: Abilene Data. Results for fitting the mean rate of Medium/Large OD flows. OD flows span 3 orders of magnitude. Critical value for C-S tests is 14.07.

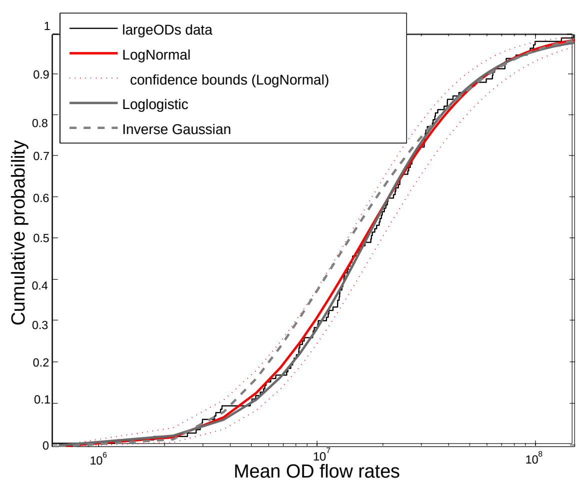

Figure 5: Empirical and fitted CDFs for all OD Flows in Abilene data. X-axis is log scale.

scription of the spatial distribution. We would thus like to suggest that researchers who wish their data to mimic realistic medium to large ISPs, should not use uniform distributions to generate sample traffic matrices.

4.6 Summary on Spatial Distributions

In order to populate a static TM, we thus suggest that one use a lognormal distribution. Due to space limitations we have only presented the results here for peak hour traffic. However, we remind the reader that all of these test were conducted on other 1-hour time periods for both the Sprint and Abilene data. Our findings remain the same, the lognormal distribution consistently produces a good fit, while other distributions (such as uniform) consistently produce poor fits.

The values of the location and scale parameters can be found in the ranges indicated in Tables 1-4. However, in general, these parameters will need to be scaled so that the resulting flows and subsequent link load levels are reasonably matched to the topology (e.g., (i) we cannot exceed link capacities, or (ii) it may not be interesting to consider TMs that result in extremely low loads). We discuss this issue of scaling in Section VI. Finally, we point out that for many applications, it may be sufficient to use just the model for the medium and large flows, and ignore the small flows, when generating the TM. ISPs don't care much about the very small flows; instead they focus on the large flows when doing capacity planning, load balancing, failure planning, etc. Hence this may be enough for the researcher depending upon the application being studied.

5. GENERATING FLUCTUATIONS FOR ORIGIN-DESTINATION FLOWS

Recall that our model for the dynamic behavior of an OD flow is given by $X(i,j,t) = \bar{x}(i,j,T) + W(i,j,t)$ for stationary period T, as in Eq. (2). Having specified $\bar{x}(i,j,T)$ for all i and j in the previous section (step $1a$), we now need to characterize $W(i,j,t)$ to complete the specification of the above model (step $1b$) for a dynamic TM in the stationary regime. Our model says that $W(i,j,t)$ is a zero mean random process intended to capture the fluctuations of the flow.

With our static TM we have one value that represented the average behavior for the peak hour T. We now want to generate values for the TM at finer time scales t, say every 10 minutes. We adopt the mean value from the static TM, but at each 10 minute interval add a different amount of fluctuation or noise. This can be done by randomly generating zero-mean Gaussian noise with the appropriate variance parameter. Thus the task at hand is to estimate this variance $\sigma(i,j)$ for each OD pair. (We sometimes write $\sigma(i,j)$ rather than $\sigma(i,j,T)$ for simplicity since we are focused on a particular time interval T here (the peak hour).

To do this we propose coupling our OD flow model with the model for the mean-variance relationship proposed in [1]. As mentioned earlier, this model describes the relationship between the mean and standard deviation as one given by the power law $\bar{x}(i,j) = \psi\sigma(i,j)^\gamma$. We use our two datasets to calibrate (ψ, γ). Figure 8 shows the relationship between the mean and standard deviation for our OD pair data sorted by their mean (from the smallest to the largest one). The top graph captures this data for the Sprint network while the bottom one represents Abilene's data. The points on the plot approximate a straight line reasonably, as the assumption on the existence of a power law would entail. This assumption has been adopted before [1]. The best linear fitting for the Sprint data corresponds to $(\gamma = 0.8, \log\psi = -0.33)$. For the Abilene data the best linear fit for the log-log plot is $(\gamma = 0.93, \log\psi = -0.31)$.

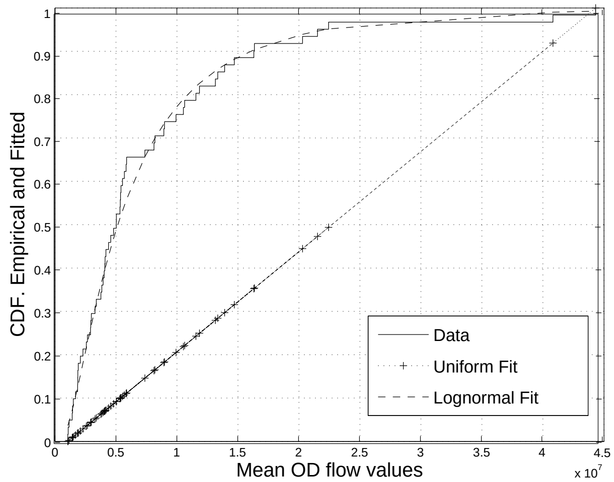

Figure 6: Sprint Data. Uniform and Lognormal Fitting. Data from peak hours across 21 days from 132 OD flows.

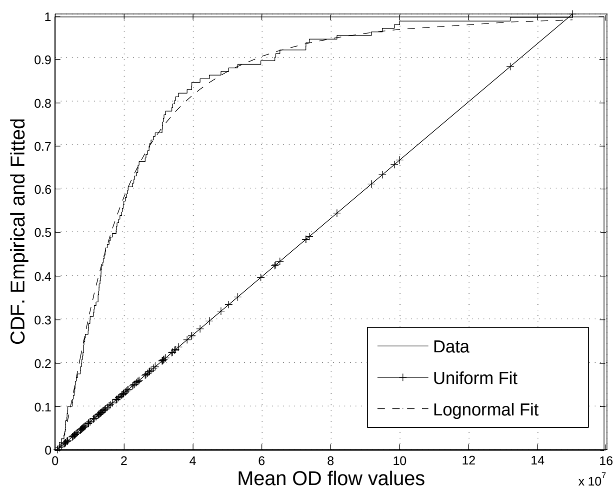

Figure 7: Abilene Data. Uniform and Lognormal Fitting. Data from peak hours across 7 days from 121 OD flows.

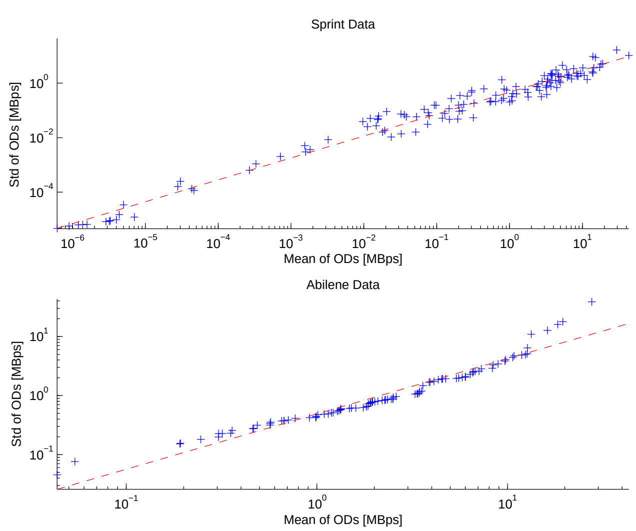

Figure 8: Standard deviation of traffic fluctuations vs the mean traffic volumes for all the OD pairs (in Mbytes/s).

Coupling these parameters with our estimate for the mean $\bar{x}(i,j)$ we select the standard deviation for OD flow (i,j) as follows

$$\log \sigma(i,j) = \frac{\log \bar{x}(i,j) - \log \psi}{\gamma}$$

6. ASSIGNING ORIGIN-DESTINATION FLOWS TO A TOPOLOGY

In this section we discuss the issue of mapping an unordered set of values $\{\bar{x}(i,j)\}$ to particular pairs of nodes in the given topology. We first discuss the interplay between a traffic matrix, a topology and routing, to illustrate the notion that a traffic matrix can be well-matched or ill-matched to a topology.

The traffic matrix can influence both the topology and routing, albeit at different time scales. On a scale of months to years, network operators add capacity in the form of extra links or modify the logical topology to accommodate increased traffic volumes and long-term shifts. At the time scale of days to weeks, network operators modify the routing to accommodate sharper shifts in the traffic matrix. An extensive amount of work on optimizing both the topology and routing matrix for a given traffic matrix is available in the literature.

The problem we wish to address however is the inverse problem. Specifically, we wish to assemble a traffic matrix by identifying an assignment of the synthetically generated rates to OD-pairs in a topology. We shall henceforth address this problem as the "synthetic traffic placement problem". Both the topology and routing influence can affect the traffic matrix. Consider the case of peering with another network. The choice of the peering point would, among other things, depend on the available capacity and backup paths. Also, shortest path routing protocols like OSPF/IS-IS can influence the matrix by changing the egress point via BGP.

We point out that this particular problem needs to be distinguished from that of the traffic matrix estimation problem. In the traffic matrix estimation problem, the link loads are given as input data (typically through SNMP), and the task at hand is to infer the values of the average OD flow rates. Inference is used since this problem is usually under-constrained even if both the topology and routing are known. Several techniques have been proposed in the literature to identify the most satisfactory inference of the traffic matrix ([1], [18], [3], [5], [8]). The synthetic traffic matrix mapping problem differs from the inference problem in that the OD flow rates are known, and possibly generated using the distribution(s) identified in Section 4. The task is to determine which rates should be assigned to which node pairs, i.e., how to place them on the topology.

Similarly to the TM estimation problem with known link loads, there is no "universally" best metric which can quantify the goodness of a solution. Instead, to reflect realistic scenarios, one must construct metrics and constraints that reflect desirable network properties to measure the performance of a mapping. For example, some properties which any traffic matrix should satisfy are :

1. The traffic matrix must be feasible. In other words, there should exist a routing under which the link loads do not exceed link capacities.

2. The matrix should not be "skewed", that is, it should not load any particular link excessively.

We propose two solutions based on different metrics to determine the mapping for a synthetic traffic matrix. The first method, termed the Load Minimization Solution uses an Integer Linear Program (ILP) to determine the mapping based on optimizing a single metric. The second method, termed the Ranking Metrics Heuristic computes a mapping by ranking the OD-pairs based on 3 metrics. We assume the researcher has available a topology along with link capacities either generated from tools like [15] and [17] or discovered through tools like [19]. Furthermore, we consider both the cases when a routing is a priori known, for example, in the form of OSPF/IS-IS link weights obtained through some tool like Rocketfuel, and also when the routing solution is itself variable. The researcher is not assumed to possess any link load information in the form of SNMP, Netflow or packet trace data.

6.1 Load Minimization Solution

A unique feature of the placement problem for a synthetic traffic matrix is that link loads are unknown. This implies that the choice of how to organize the flow rates into a traffic matrix is impacted by both feasibility and congestion concerns. A particular placement is considered feasible if none of the link capacities are exceeded. In other words, by an appropriate choice of a traffic matrix and routing, one can minimize congestion (note that the matrix may not be feasible. We address this issue in a subsequent section.) Minimizing congestion is a desirable and widely used metric for traffic engineering. Hence, our first method, the Load Minimization Solution, seeks to determine a mapping that tries to achieve this goal. Our ILP solution will find a solution that meets the capacity constraints if such a solution exists; this partly comes as a consequence of minimizing congestion. For ease of exposition, we define some terminology below, before presenting the ILP.

The input to the ILP is :

1. The set of uni-directional links E with c_{uv}, the capacity of link $(u, v) \in E$.

2. The set of OD-pairs $\mathcal{K}$ with $K = \|\mathcal{K}\|$. Let $s(k)$ represent the origin node for OD-pair k.

3. The set of synthetically generated rates $\{R_l\}, l = 1, \ldots, K$

The output (variables) obtained by solving the ILP are :

1. The mapping indicator variable I_{kl} which takes a value of 1 if OD-pair k is assigned the rate R_l and 0 otherwise.

2. r_k, the rate of OD-pair k, which is decided by the mapping I_{kl}.

3. x^k_{uv} is the amount of flow from OD k that traverses link (u, v). If the k-th flow traverses link (u, v), then $x^k_{uv} = r_k$ when fractional routing is *not* supported, or x^k_{uv} is a portion of r_k if fractional routing occurs. If flow k does not traverse link (u, v) then $x^k_{uv} = 0$.

The ILP may now be formulated as :

$$\min z$$

where

$$\sum_{l=1}^{K} I_{kl} = 1 \tag{9}$$

$$\sum_{k\in\mathcal{K}} I_{kl} = 1 \tag{10}$$

$$r_k = \sum_{l=1}^{K} I_{kl} R_l \tag{11}$$

$$\sum_{v:(v,u)\in E} x^k_{uv} - \sum_{v:(u,v)\in E} x^k_{uv} = \begin{cases} r_k & \text{if } u = s(k) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$z \geq \sum_{k\in K} x^k_{uv}/c_{uv} \qquad (u, v) \in E \tag{13}$$

$$x^r_{u,v} \geq 0 \quad (u, v) \in E,\ k \in \mathcal{K} \tag{14}$$

From Eq. (13), z is the maximum link load. Constraints Eq. (9) and (10) ensure that each traffic rate R_l is mapped to exactly one OD-pair rate r_k and vice versa. Eq. (12) are the flow conservation equations. The optimal solution of the above integer linear program gives a rate-to-OD-pair mapping that minimizes congestion. If $z < 1$, the solution yields a feasible mapping, i.e., one that does not violate any link capacity. If a feasible placement exists our solution will find it since we use minimizing congestion as an objective. It should be clear that for some sets of OD flow rates no feasible solution exists. If the results yields $z \geq 1$, then one must scale down the traffic matrix to achieve feasibility. Details of scaling are presented in Section 6.4.

In the case, when a routing is given a priori, the above formulation can be modified by replacing the flow x^k_{uv} by $r_k \cdot f^k_{uv}$, where f^k_{uv} is the *fraction* of flow k on link (u, v) fixed by the routing.

6.2 Ranking Metrics Heuristic

The Load Minimization heuristic possesses the nice property of quantifiable performance since it seeks a mapping that minimizes a common metric. However, the technique possesses several drawbacks. One, the problem is a Generalized Assignment Problem and NP-complete. Two, the heuristic tunes the mapping to a specific routing. Three and most importantly, the heuristic focuses only on a single metric. In practice, the interplay between traffic matrix, topology and routing produces a mapping that must satisfy multiple "soft" metrics like low congestion, failure insensitivity etc.

To overcome these drawbacks, we propose a simple heuristic, the Ranking Metric Heuristic, which takes into account multiple metrics. The heuristic works as follows. We take the K samples of mean OD traffic loads and order them in a list from largest to smallest. Call this *list1*. Next we want to create a list of node pairs that are ranked according to a metric that gives hints as to how likely a node pair is to carry a large OD flow. Our *list2* orders the node pairs from most likely to carry a large flow to least likely. Given two such lists, we simply match them one to one: the ith traffic volume in *list1* is assigned to the ith OD pair in *list2*. The last step in our algorithm is to ensure the capacity constraints are met.

We now define three metrics used to rank the node pairs to create *list2*. These metrics should be viewed as guidelines for ordering node pairs that generate soft constraints rather than hard constraints on node pair ordering. Our intent is to make use of know-how on how carriers evolve their backbones. We remind the reader that in both of our datasets 95% of the traffic load was carried by either half or one third of the OD flows. Thus it is more important that the order among the top half of these two lists be accurate than the bottom half. In fact the ordering among the small traffic loads and small OD pairs might be quite unimportant since it matters little where tiny flows are placed.

It is reasonable to expect, that nodes with large fanout capacities can support a large amount of traffic and hence act as big sinks/sources of traffic, while nodes with limited incoming/outgoing capacities would contribute only a small amount of traffic. Let $F_{out}(src)$ denote the fan-out capacity of a source node, and $F_{in}(dst)$ denote the fan-in capacity of a destination node. We define our first ordering metric m_1 on a pair of nodes $N1$ and $N2$ as

$$m_1(N1, N2) = min(F_{out}(N1), F_{in}(N2))$$

where $N1$ refers to the source and $N2$ refers to the destination node. We can order all nodes pairs this way. Clearly this is going to produce a list in which many node pairs have the same value for the metric.

To disambiguate between node pairs having the same value for $m1$, we use a second metric $m2$ to create a finer level of ordering. A second piece of information obtainable from the topology is the *connectivity*, in terms of number of links, of the node. A node that has large connectivity is more likely to be a large source/sink than a node that is sparsely connected. This classification is motivated by the practice carriers follow in the placement of PoPs and the inter-connectivity amongst them to maintain *robustness*. If a PoP is a large source of traffic, it is likely to have large connectivity (to other PoPs) so as to prevent disruption of traffic due to link failures. Let $NL(N)$ denote the number of links at node N. Then the value for a node pair is determined by

$$m2(N1, N2) = min(NL(N1), NL(N2))$$

We point out that we always need to consider the minimum of each of the metrics at the source and destination node, because it is the smaller node that determines the likelihood of the pair to carry a large OD flows. Also observe that the Load Minimization heuristic too is influenced by these two metrics, albeit indirectly, since it seeks to minimize load and hence will typically map large rates to OD-pairs with large fan-out/fan-in capacity.

We note that the actual numbers produced are not going to lie in between the numbers produced from metric 1. The numbers here should be used to produce a finer level of ordering while still maintaining the order of the previous metric. The actual values used for this can be adjusted as we move forward in this process.

To further subdivide among node pairs that are still experiencing ties due to the above two metrics, we use a third metric to create an even finer level of ordering. This metric is based on routing information. Using the routing information (i.e., the matrix A defined in Section 3.2), we can count the number of OD pairs traversing a node. The intent here is to capture some transit traffic information and not the traffic sourced at or destined for this node. We determine the maximum number of OD flows the node could carry under *all cases* of single link failures. Let $NFUR(N)$ denote the Number of Flows Under Failure for node N, i.e., this gives the *worst-case* number of origin-destination pairs that could traverse a node. This helps us to disambiguate a node that has a large fanout capacity which could either belong to a large origin-destination pair or merely act as a support for transit traffic. Our third ordering metric is thus given by

$$m3(N1, N2) = \frac{1}{max(NFUR(N1), NFUR(N2))}$$

This metric uses an inverse because we want one node pair whose NFUR metric is higher than a second node pair to be ranked lower because we prefer to leave this node pair unattached to large OD flows in order to reserve it for failures.

6.3 Heuristic Algorithm

Our heuristic can now be concisely stated.

1. Order $\bar(x)(i, j)$ from largest to smallest to create *list1*.
2. Order node pairs according to ranking metric $m1$ from largest to smallest.
3. For all node pairs experiencing ties for $m1$, apply ranking metric $m2$ to break the ties.
4. For all node pairs still tied in their ranking, apply metric $m3$ to further subdivide the ordering among these pairs.
5. Assign item $list1(i)$ to node pair $list2(i)$.
6. Route the flows according to the routing matrix A, and check if any link capacities are exceeded. If yes scale down the TM according to the method outlined below in Section 6.4 .

6.4 Scaling

It is possible that the mapping solution obtained by either of the two techniques may be infeasible, that is, it violates some link capacities. This can happen due to two factors. The first is that the technique itself produces an infeasible mapping (this can only happen for the Ranking Metrics heuristic since the ILP always finds a feasible mapping if one exists). The second factor could be that the parameters (eg. μ, σ) used in the probability distribution for generating the synthetic rates are incompatible with the topology itself. We now address both these issues.

The first issue can be dealt with in a straightforward fashion by scaling down the traffic matrix, which also has the added benefit of preserving the original spatial traffic rate distribution. The scaling factor is simply given by the excess congestion on the most loaded link. Alternatively, if one does not wish to scale down the traffic rates, we propose a modified form of the ILP formulation that perturbs the solution obtained by the Ranking Metrics heuristic to determine if a feasible solution exists. The reader is referred to [21] for details of the formulation.

Yet another approach would be to apply metric $m2$ first and metric $m1$ second in the above algorithm, and then checking if this yields an assignment that meets the capacity constraints. We tried many orderings of these three metrics. In our validation of this algorithm with the Sprint TM and topology, the ordering of three metrics as stated above yielded the closest solution to the real TM.

The second potential source of infeasibility are the synthetically generated rates themselves. They may be ill-matched to the topology if the parameter set (eg. μ and σ for lognormal distribution) is chosen independently of the topology available to the researcher. The question of what mean and variance (which are usually the independent variables defining the distribution parameters) to choose for the traffic rate distribution given a topology is an open question. It strongly depends on the desired congestion, the routing mechanism etc. and hence has no simple answers.

As a rule of thumb the mean traffic rate could be chosen to be equal to the average link capacity (or scaled capacity depending on targeted load) divided by the average number of OD-pairs per link (over all failure scenarios). The variance could be computed in a similar fashion but using the largest link capacity instead. However, we do not claim these to be definitive methods. Instead, the researcher could choose a mean and variance using this method and then scale the distribution based on the mapping solution. For

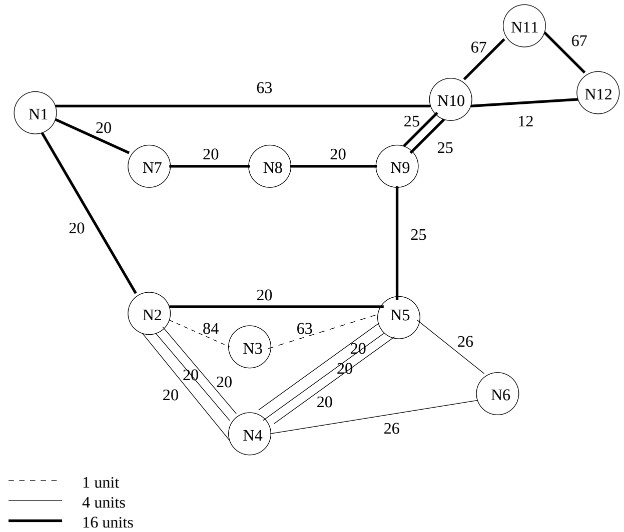

Figure 9: Illustration of an ISP Backbone

e.g. if the congestion level of the mapping solution from the ILP or Ranking Metrics Heuristic is low, the matrix can be scaled up.

Given a target mean and variance for the OD flows, the parameters for the lognormal distribution (μ, σ) can then be easily from the system of equations below, which relate the Mean and Variance of the lognormal random variable with the scale and shift parameters:

$$\begin{aligned} \ln \text{Mean} &= \mu + \frac{\sigma^2}{2} \\ \ln \text{Variance} &\approx (\sigma^2 + 2\mu) + \sigma^2 \end{aligned} \tag{15}$$

6.5 Validation

In order to validate our heuristics, we took the OD flows from Sprint's actual TM and scrambled them (i.e., we keep the rates but disassociate them from particular node pairs). We recomputed their placement using our two solutions, and compared our placements to their actual placement in the original Sprint TM. We used the PoP level topology for Sprint's European network as shown in Figure 9. It has 3 classes of links (based on the link speed), which we have scaled for proprietary reasons. We do show the *actual* link weights used for IS-IS routing. Each link actually constitutes a pair of uni-directional links in either direction but were assumed to have the same weight setting.

We solved the ILP for the Load Minimization Solution using CPLEX 7.0 on a Dell 1.2 GHz PowerEdge server. No a priori routing was assumed. It took 9 hours to produce the optimal placement in terms of minimizing congestion, indicating that this approach may be intractable for larger topologies. The placement computed by the ILP correctly identified 40 of the top 61 OD-pairs that account for 95% of the traffic, although the actual placement of rates differed from the original placement among the 40 OD-pairs. Since the goal is to provide a reasonable placement, we believe that identifying the location of large OD-pairs is more important than the exact rate assigned (as long as it is "large"). There were no capacity violations in this solution.

In Table 5 we list the value of our metrics for the Ranking Metrics heuristic on a per-node basis. We do not give them for the node pairs, for ease of exposition, because this would yield a table with 132 rows. Our intent here is merely to illustrate how these ranking metrics end up affecting our ordering. The IS-IS weight setting was used to compute the metric $m3$.

Node	Fan-Out Capacity	Connectivity	Worst Case # Transit OD pairs
1	48	3	60
2	44	6	56
3	2	2	0
4	28	7	20
5	49	6	63
6	8	2	0
7	32	2	24
8	32	2	30
9	64	4	72
10	90	5	38
11	32	2	20
12	32	2	20

Table 5: Three Ranking Metrics *per Node*

We now look at some examples of how these metrics work. According to our first metric, we see that nodes $N1, N2, N5, N9$ and $N10$, are likely to be large sinks and/or sources of traffic by virtue of their large fanout capacity.

All of these large nodes, when paired with node 2 will yield the same value for $m1$; namely $m1(N1, N2) = m1(N5, N2) = m1(N9, N2) = m1(N10, N2) = 44$. The metric $m2$ will break all these ties since it yields $m2(N1, N2) = 3,\ m2(N5, N2) = 6,\ m2(N9, N2) = 4,\ m2(N10, N2) = 5$ leading the pair $(N1, N2)$ to be the highest rank among these four pairs.

We notice that two nodes, $N3$ and $N6$, have zero as their worst case transit number of OD pairs. If they carry no transit OD pairs, this indicates that the fanout capacity is reserved exclusively for traffic originating from this node. If we look at the topology in Fig. 9, we can see that this metric is working properly as nodes n3 and n6 are not well connected (in terms of capacity) to the rest of the topology and hence little traffic would be routed through them during failure. We also notice from the Table that node 9 has the largest number of transit OD pairs. This is useful for breaking ties achieved with $m1$ and $m2$ to reduce the load on n9. For example, it is easy to see that the node pairs $(N7, N8),\ (N7, N9),\ (N7, N11)$, $(N7, N12)$ will all tie with respect to metrics $m1$ and $m2$. Metric $m3$ will break these ties and put the flow including node 9 (namely $(N7, N9)$) at the bottom of these four.

The Ranking Metric heuristic correctly identified 36 of the top 61 flows (95% of the total bytes) and did not violate any capacity constraints. This indicates that simple metrics may work well enough to produce a TM assignment to a topology that is approximate, but reasonable, and is likely to avoid very skewed assignments (as random assignment could produce).

7. CONCLUSIONS

In this paper, we introduced and described the problem of synthetically generating traffic matrices. We discussed many issues involved in TM generation, and presented a first methodology for handling this problem. A simple OD flow model that covers both static and dynamic TMs was included. We described the placement problem of assigning end-to-end traffic volumes to specific node pairs, and highlighted why the inter-dependence of topology and traffic matrices can render this challenging.

To the best of our knowledge, this is the first paper to provide some key properties of actual traffic matrix data from large domains, that can be used in synthetic TM generation. We found that the mean rates of OD flows do *not* obey a uniform distribution, and we encourage researchers to avoid this approach. Instead, we discovered that either the lognormal, loglogistic or inverse Gaussian distributions provide a good distribution to use if one wants to populate a static TM that can be said to represent at least some realistic traffic matrices. Among these, the lognormal distribution is perhaps the best, although only marginally so, since it appears more robust to variations in the fitting procedure. The lognormal distribution was a good fit for the majority, but not all, of the OD flows when the measured TM had flows diverse enough to span seven orders of magnitude. A distinction between large and small flows was not needed for the Abilene TM, whose OD flows could *all* be described by a lognormal distribution.

Our methods provide a more justifiable approach for researchers in that at least they are based on a some real world datasets; this is certainly more justifiable than an approach that assumes no knowledge (i.e., uniform distributions). We recognize that two datasets is still a small number of datasets, however the paucity of available traffic matrix datasets limits this exploration. We thus encourage the community to try to obtain more traffic matrix measurements from interesting networks. In addition to ISP and university networks, it would be interesting to uncover the statistical properties of TMs from enterprise networks.

We introduced the problem of constructing a TM to match a topology, and discussed the notion of a TM being well-matched to a topology. The idea of being *well-matched* can include hard constraints (such as guaranteeing no link capacities are exceeded) and soft requirements (e.g., not assigning large flows to nodes intended for failure recovery). We provided two solutions, an ILP to formalize a version of the problem that poses link capacities as a constraint, and targets to minimize the maximum load property in the objective function. This solution will find a feasible placement, if one exists. Our second solution uses a heuristic algorithm that focuses on the softer requirements that come from common carrier practices. Both of these solutions were able to correctly identify that majority of node pairs that can act as hosts for large OD flows.

There are other interesting approaches that could be pursued for generating realistic OD flow rates. One is to make use of the gravity model, or information theoretic approach, discussed in [5]. Another would be to build upon the properties of the traffic fanout behavior of nodes as outlined in [11]. The time series OD flow modeling employed in [16] could prove useful for generating dynamic TMs. The true objective of the placement problem is to reproduce the placement of OD flows found in actual TMs. This is hard to quantify and thus future research for this problem could focus on identifying better metrics to capture the ”match” between a traffic matrix and a topology.

8. REFERENCES

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

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

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

[4] The Abilene Network of Internet2 Community, “Abilene Observatory”, http://abilene.internet2.edu/.

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

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

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

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

[9] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft and C. Diot, “IGP Link Weight Assignment for Transient Link Failures”, *The 18th International Teletraffic Congress (ITC)*, Germany, September 2003.

[10] A. Soule, A. Nucci, E. Leonardi, R. Cruz, and N. Taft, “How to Identify and Estimate the Top Largest Traffic Matrix Elements in a Changing Environment”, *Proceedings of ACM Sigmetrics*, New York, June 2004.

[11] K. Papagiannaki, N. Taft, and A. Lakhina, ”A Distributed Approach to Measure IP Traffic Matrices”. *Proceedings of ACM IMC*. October 2004.

[12] B. Fortz and M. Thorup, “Optimizing OSPF/IS-IS Weights in a Changing World”. *IEEE Journal on Selected Areas in Communications*, vol. 20, no 4., 2002.

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

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

[15] E. W. Zegura, “GT-ITM: Georgia Tech Internetwork Topology Models (Software)”, *Georgia Tech*, http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm/tar.gz, 1996.

[16] A. Gunnar, M. Johansson, and T. Telkamp, ”Traffic Matrix Estimation on a Large IP Backbone - A Comparison on Real Data”, *Proceedings of ACM IMC*, October, 2004.

[17] A. Medina, A. Lakhina, I. Matta, and J. Byers, “BRITE: Boston University Representative Internet Topology Generator”, *Boston University*, http://cs-www.bu.edu/brite, April, 2001.

[18] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, ”Traffic Matrices: Balancing Measurement, Inference and Modeling”, *Proceedings of ACM Sigmetrics*, Banf, Canada, June, 2005.

[19] N. Spring, R. Mahajan and D. Wetherall, “Measuring ISP topologies with RocketFuel”, *Proceedings of ACM SIGCOMM 2002*, Pittsburgh PA, August, 2002.

[20] John A. Rice, ”Mathematical Statistics and Data Analysis”, *Wadsworth & Brooks/Cole*. Chapter 9. 1988.

[21] A. Nucci, A. Sridharan and N. Taft, “The Problem of Synthetically Generating IP Traffic Matrices: Initial Recommendations”,*Sprint Technical Report RR05-ATL-1498*, May 2005. Available at http://ipmon.sprintlabs.com.

[22] A. Nucci, N. Taft, C. Barakat and P. Thiran, ”Controlled Use of Excess Backbone Bandwidth for Providing New Services in IP-over-WDM Networks”, *IEEE Journal of Selected Areas in Communications*. Nov. 2004.

[23] D. Applegate and E. Cohen, ”Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs”, *Proceedings of ACM SIGCOMM*, August, 2003.

[24] M. Roughan, M. Thorup and Y. Zhang, ”Traffic Engineering with Estimated Traffic Matrices”, *Proceedings of ACM IMC*. Miami FL, May 2003.