

A Factor Analytic Approach to Inferring Congestion Sharing Based on Flow Level Measurements

Dogu Arifler, *Member, IEEE*, Gustavo de Veciana, *Senior Member, IEEE*, and Brian L. Evans, *Senior Member, IEEE*

Abstract—Internet traffic primarily consists of packets from elastic flows, i.e., Web transfers, file transfers, and e-mail, whose transmissions are mediated via the Transmission Control Protocol (TCP). In this paper, we develop a methodology to process TCP flow measurements in order to analyze throughput correlations among TCP flow classes that can be used to infer congestion sharing in the Internet. The primary contributions of this paper are: 1) development of a technique for processing flow records suitable for inferring congested resource sharing; 2) evaluation of the use of factor analysis on processed flow records to explore which TCP flow classes might share congested resources; and 3) validation of our inference methodology using bootstrap methods and nonintrusive, flow level measurements collected at a single network site. Our proposal for using flow level measurements to infer congestion sharing differs significantly from previous research that has employed packet level measurements for making inferences. Possible applications of our method include network monitoring and root cause analysis of poor performance.

Index Terms—Factor analysis, inference of congestion sharing, network measurement.

I. INTRODUCTION

ROOT CAUSE analysis of poor network performance using voluminous amounts of measurements collected by network monitoring tools and devices is a challenging problem for network engineers and researchers. Poor network performance, such as excessive delays during file downloads, is often due to a congested resource such as an overloaded server, an overutilized customer access link, or a backbone router that is congested as a result of link failure or misconfigured routing in carrier's network.

Determining which network flows are sharing congested resources might be the first step in analyzing and eliminating the causes of poor network performance. For instance, information on congestion sharing might be used by content providers to replicate content at other locations to reduce the load on the congested portions of the network, and by service providers to diagnose problems and direct traffic sharing a bottleneck onto dis-

joint routes. However, deciding which network flows are sharing congested resources in the Internet is usually difficult without access to the complete routing information for the network. In general, network managers have information only about their network domain, and have little or no information about the properties of the other domains.

A significant portion of the Internet Protocol (IP) traffic consists of packets from *elastic flows* [1] or “document” traffic, i.e., Web transfers, file transfers, and e-mail, whose transfers are mediated via Transmission Control Protocol (TCP) (see for example, [2]). In this paper, we describe the application of a statistical method called *factor analysis* that can be used to analyze voluminous amounts of flow level TCP network traffic measurements collected at a single measurement site to infer which classes of TCP flows are sharing congested resources. We develop a conditional sampling strategy on the time series of *flow class throughput* data. Our sampling strategy generates samples that are used in the construction of the flow class throughput correlation matrix that is suitable for inferring congested resource sharing. We show that the correlation structure of flow class throughputs obtained by flow level measurements for a number of TCP flow classes can often be captured by a fewer number of *latent factors*. The latent factors represent congested resources and can be used to infer which flow classes are sharing resources in the network. We also empirically investigate the effect of filtering out small and large flows on our inferences of congestion sharing. Our methodology is validated using two sets of TCP measurements collected at a border router. The statistical accuracy of the inferences is assessed using bootstrap confidence intervals.

A. Flows, Flow Records, and Flow Classes

Since network flows are of main interest in this paper, we first define *flows*, *flow records*, and *flow classes*. Although there is no standard definition of a *flow*, a commonly accepted definition of an IP flow [3] is a unidirectional sequence of packets, which are close to each other in time and share a common identifier such as a common source and destination address. For instance, packets corresponding to a file download constitute a flow.

State-of-the-art networking equipment that runs traffic monitoring tools (such as NetFlow [4], sFlow [5], and Argus [6]) is capable of generating *flow records*. A flow record contains the source and destination IP addresses, TCP or User Datagram Protocol (UDP) port numbers, IP protocol type, type of service fields in IP headers, start and end times, and the number of packets and bytes in a flow. A major problem in flow measurement is the lack of scalability: at very high speed routers, the number of flows to be measured might easily exceed millions

Manuscript received February 29, 2004; revised March 2, 2005 and November 2, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor J. Byers. This work was supported by The State of Texas Advanced Technology Program under Project 003658-0614-2001, and in part by the National Science Foundation under Grant CNS-0435307.

D. Arifler was with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712-0240 USA. He is now with the Department of Computer Engineering, Eastern Mediterranean University, Famagusta, Cyprus (e-mail: dogu.arifler@emu.edu.tr).

G. de Veciana and B. L. Evans are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712-0240 USA (e-mail: gustavo@ece.utexas.edu; bevans@ece.utexas.edu).

Digital Object Identifier 10.1109/TNET.2006.890103

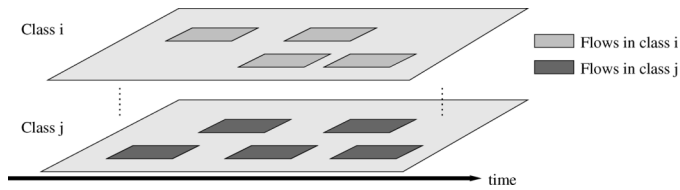


Fig. 1. A set of flow records can be arranged in time and classified into classes (shown at different planes) as illustrated. The length of the rectangle indicates the duration of the corresponding flow. Classes i and j might correspond to two different sets of flows corresponding to two different source-destination IP address prefix pairs.

per hour. Therefore, at high link speeds, the flows [7] and/or the packets within a flow [8] may be sampled in order to keep up with the link speeds. The network node, such as a router, performing record generation usually exports these records to a data warehouse for further processing.

We define an *IP flow class* as a collection, or aggregation, of flows that are emitted successively and in parallel, and that have a common attribute. For example, we can refer to all flows sharing common source and destination IP address prefixes as a flow class (see Fig. 1). A Web browsing session, in which a user visits a number of pages at a Web site and triggers a number of object downloads at each page, generates flows that can also be treated as a flow class.

B. Related Work

The simplest approach to detecting shared resources is to use a utility such as `traceroute` that tracks the route that a packet follows from its source to its destination. Such utilities, however, require the cooperation of routers in the network on the path of the flow. Owners of the carrier networks are often unwilling to provide information about their networks, and hence the use of such utilities is not always viable. Savage, Cardwell, and Anderson [9] describe a “locality” based approximation for detecting shared paths by looking at the destination addresses of flows. Their approximation is based on the assumption that flows destined to a particular host or network address generally follow the same path, and hence visit the same bottleneck in the network.

Harfoush, Bestavros, and Byers [10] use packet-pair probing for determining whether two flows originating from the same source share a bottleneck. Their technique is based on correlating end-to-end packet loss measurements to identify flows that share “similar network conditions”. The main disadvantage of their technique is the requirement of cooperating senders. Rubenstein, Kurose, and Towsley [11] develop an end-to-end technique based on packet loss or delay observations to infer whether or not two flows are experiencing congestion on a common set of network resources. Their methodology is based on the observation that losses or delays experienced by two packets passing through the same bottleneck exhibit some degree of positive correlation. Their technique assumes that the flows share a common endpoint; i.e., either the sources or the destinations of packets are co-located and collaborating, which may have limited applicability. Kim *et al.* [12] propose a wavelet-based approach to detecting shared congestion. In order to determine whether two flows are sharing a bottleneck,

they correlate probe packets’ delay sequences generated by applying wavelet denoising to original delay sequences of flows. They show that wavelet denoising filters out delay fluctuations that are not due to shared congestion. Unlike [10] and [11], their method is not limited to flows that share a common endpoint, provided that the synchronization offset between delay sequences is less than 1 second.

Rabbat, Nowak, and Coates [13] propose sending packet probes from two sources to infer whether a subgraph of a graph formed from the paths connecting two sources to two receivers is shared. The methodology is based on the assumption that probe packets arrive at a receiver in the order in which they reach the node where paths from two sources join. Generalization of their method to more than two sources may not be scalable due to probing traffic overhead.

Katabi, Bazzi, and Yang [14] develop iterative techniques that minimize entropy-based cost functions to cluster flows that share a bottleneck into groups. Their method is based on the observation that correct clustering minimizes the entropy of inter-packet spacing within clusters with an empirical distribution measured by an observer. The main advantage of their method is that it does not require sending probe traffic into the network and does not require cooperating senders; i.e., it is passive. However, they also indicate that their technique is robust only when the observer can monitor more than 15% of the traffic from the bottleneck link, and hence is not practical when the observer is an end-receiver.

C. Contributions and Organization

Our work on inferring congestion sharing differs significantly from the previous work in that we consider *flow level* instead of *packet level* statistics. We rely on *passive*, flow level TCP measurements (flow records) collected at a network node (e.g., router, gateway, server), although it is possible to take an active approach by sending probing flows into the network. Furthermore, while many other previous methods that infer resource sharing are limited to determining whether particular flow class *pairs* share bottlenecks, our methodology considers a set of flow classes *simultaneously*.

This paper is an extended version of [15], which proposes using flow level throughput measurements to infer resource sharing and presents results based on known, analytical fluid models, and of [16] and [17], which describe application of factor analysis to real TCP data to infer which flow classes share congested paths. In this paper, we include additional validation results based on part of an extensive set of TCP simulations. We also discuss the effects of flow sizes on inference results based on our empirical findings.

The outline of the rest of the paper is as follows. Section II first provides an insight into the reason why one expects to see correlated throughputs among congestion sharing flows, and then describes methods for constructing a flow class throughput correlation matrix from flow records. We review the basics of factor analysis and discuss its application in our context. Section III describes results based on TCP simulations and demonstrates the effectiveness of our methods. Section IV analyzes real TCP data to make inferences and presents a bootstrap

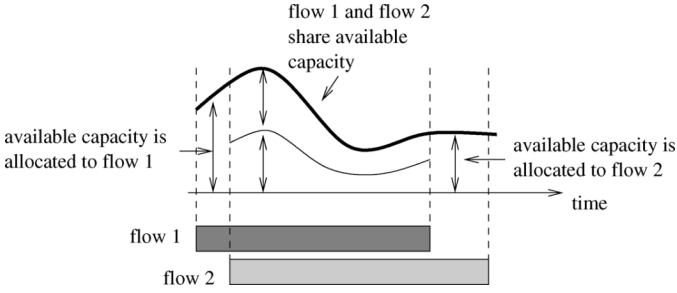


Fig. 2. Sharing of instantaneous available resource capacity by two temporally overlapping flows. The length of the rectangle corresponds to the flow’s duration. In this particular example, both flows have similar packet round trip times and packet loss rates, and the instantaneous bandwidth sharing is roughly fair during the time period over which they overlap.

method to obtain confidence intervals on parameters of interest for inferring congestion sharing. Section V concludes the paper.

II. METHODS

In this section, we describe how to process flow records in order to capture correlations among measured flow throughputs. The size, start and end times of a flow f are available from its record and are denoted by v_f , s_f , and e_f , respectively. We define the *throughput* or *perceived throughput*¹ for a flow as the amount of data it carries (in bits) divided by the duration of the flow (in seconds). A key premise of this section is that elastic flows that are overlapping long enough on the same congested resource tend to have positively correlated perceived throughputs.

A. Model for Congestion at a Bottleneck

First, we formalize the existence of correlation between perceived throughputs of temporally overlapping elastic flows that share a congested resource. We then investigate the effect of the flow durations on the correlation.

It has been shown that TCP flows that are operating in additive-increase multiplicative-decrease mode of congestion control share the capacity of a bottleneck link roughly fairly when the flows have similar round-trip times and packet loss rates [18], [21], [22]. As an example, Fig. 2 illustrates how two temporally overlapping TCP flows share available bottleneck capacity in an idealized model. However, in general, TCP flows take some time to discover the congestion state, or the available capacity, of the network, and especially, very small flows (due to TCP’s Slow Start) may not have an opportunity to “learn” the capacity available to them during their sojourn. As a consequence, the throughput perceived by short flows will not reflect the congestion state of the network during their sojourn.²

We consider a simple model of congestion level seen by a flow³: $\{B(i)\}$ is a first-order autoregressive (AR(1)) process [23] that represents the instantaneous bandwidth (or capacity)

¹Other authors refer to perceived throughput as realized throughput [18], file transfer throughput [19], or flow rate [20].

²We also note that throughput of a flow is limited by the sizes of sender’s and receiver’s buffers. In that case, sender’s or receiver’s buffer is the bottleneck for that flow.

³We assume that the congestion process is roughly independent of the flow; i.e., the flow makes only a small contribution to the overall congestion.

available to each flow at the bottleneck. $\{B(i)\}$, whose mean is denoted by μ_B , is then defined by

$$B(i) - \mu_B = \alpha (B(i-1) - \mu_B) + Z(i)$$

where $\{Z(i)\} \sim N(0, \sigma_Z^2)$, $|\alpha| < 1$, and $Z(i)$ is uncorrelated with $B(j)$ for each $j < i$.⁴ For now, let us assume that s_f and e_f are discrete times as well. A given flow f carries an amount of data equal to

$$V_f = \sum_{i=s_f}^{e_f} B(i).$$

In the discrete-time AR(1) model, the duration of a flow f is given by $d_f = e_f - s_f + 1$. For simplicity, consider two flows f_1 and f_2 with given start and end times, and suppose that $s_{f_1} = 0$ and $s_{f_1} \leq s_{f_2}$ without loss of generality. The perceived throughputs of these flows are

$$Y_{f_1} = \frac{\sum_{i=0}^{e_{f_1}} B(i)}{d_{f_1}} + \frac{W_{f_1}}{d_{f_1}} \quad \text{and} \quad Y_{f_2} = \frac{\sum_{j=s_{f_2}}^{e_{f_2}} B(j)}{d_{f_2}} + \frac{W_{f_2}}{d_{f_2}}$$

where $W_{f_1}, W_{f_2} \sim N(0, \sigma_W^2)$ are included to model the “noisy” throughputs perceived by short flows, and are independent of each other, $\{B(i)\}$, and $\{Z(i)\}$. In this context, a “noisy” throughput means that the throughput perceived by a short flow is not a typical one for the class to which the flow belongs due to its inability to discover the congestion state of the network. For flows with long sojourn times, the “noise” terms become negligible. The autocorrelation function of $\{B(i)\}$ is denoted by $\gamma(h)$, and is equal to $\frac{\sigma_Z^2 \alpha^h}{(1 - \alpha^2)}$ for $h \geq 0$. The correlation between Y_{f_1} and Y_{f_2} is

$$\text{Corr}(Y_{f_1}, Y_{f_2}) = \frac{1}{d_{f_1} d_{f_2} \sigma_{Y_{f_1}} \sigma_{Y_{f_2}}} \sum_{i=0}^{e_{f_1}} \sum_{j=s_{f_2}}^{e_{f_2}} \gamma(|j - i|) \quad (1)$$

where $\sigma_{Y_{f_1}}$ and $\sigma_{Y_{f_2}}$ are the standard deviations of throughputs of f_1 and f_2 , respectively. We note that the correlation in (1) does not change even in the case of unequal sharing of bottleneck capacity.⁵

The standard deviation of the throughput of f with $s_f = 0$ is given by

$$\sigma_{Y_f} = \sqrt{\text{Cov}(Y_f, Y_f)} = \sqrt{\frac{1}{d_f^2} \left(\sum_{i=0}^{e_f} \sum_{j=0}^{e_f} \gamma(|j - i|) + \sigma_W^2 \right)}. \quad (2)$$

⁴For similar approaches to modeling available bandwidth at a bottleneck, refer to [24] and [25]. Note that in our AR(1) model, available bandwidth can become negative. However, we introduce this simple model only to provide an insight into the nature of flow throughput correlations without attempting an exact traffic model in any way.

⁵Unequal sharing can occur in practice when TCP flows have different round-trip times [18], [22]. In such cases, bottleneck capacity will be shared in proportion to weights that are inversely proportional to associated round-trip times. For instance, instantaneous bandwidths available for flows f_1 and f_2 could be $w_1 B(i)$ and $w_2 B(i)$, respectively, where w_1 and w_2 are weights. However, it is straightforward to show that the correlation between Y_{f_1} and Y_{f_2} is insensitive to w_1 and w_2 .

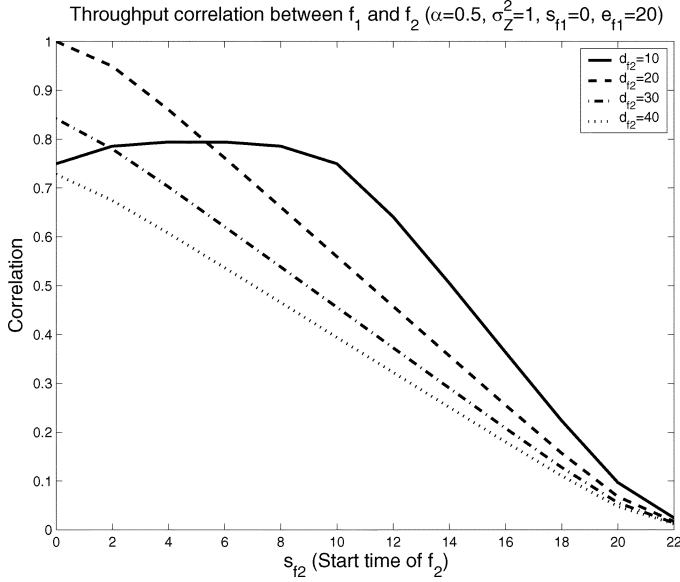


Fig. 3. The effect of flow duration and temporal overlap on the correlation in (1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 0$. Flow 1 starts at time 0 and ends at time 20.

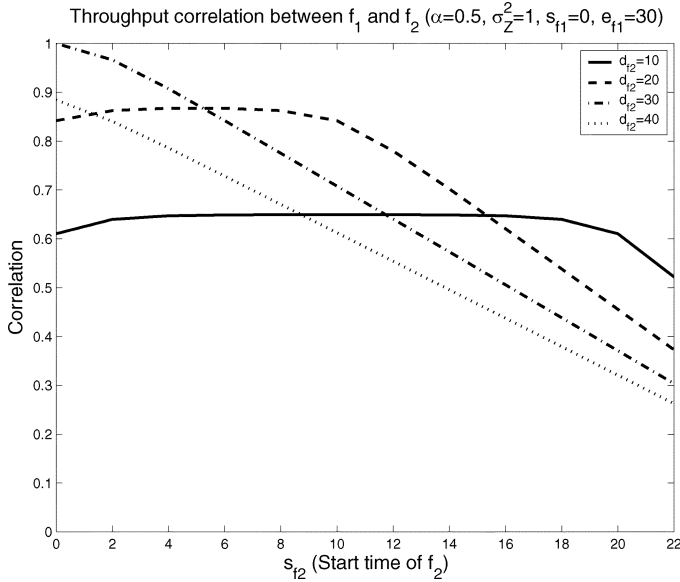


Fig. 4. The effect of flow duration and temporal overlap on the correlation in (1) between throughputs of f_1 and f_2 that share a congested resource. The correlation values shown are for $\sigma_W^2 = 0$. Flow 1 starts at time 0 and ends at time 30.

From (2), one can conclude that perceived throughputs of long flows have smaller standard deviation than those of short flows since d_f^2 dominates [26]. These results agree with the observations reported for the perceived throughputs of small and large flows in [18] and [22].

In order to illustrate the behavior of (1) with different flow durations and different amounts of temporal overlap between the two flows, we set $\alpha = 0.5$, $\sigma_Z^2 = 1$, and $\sigma_W^2 = 0$ (no noise), and in Figs. 3 and 4, we exhibit the correlation as a function of s_{f_2} for different d_{f_2} values when $e_{f_1} = 20$ and $e_{f_1} = 30$. Based

on the AR(1) model, we can draw the conclusion that the correlation between perceived throughputs of congestion sharing elastic flows is largely determined by the amount of temporal overlap between flows relative to the (product of) durations and standard deviations of flows. The throughput correlation is high when the two flows temporally overlap, and then decreases with increasing s_{f_2} (i.e., as the amount of overlap decreases). Furthermore, the correlation between overlapping flows decreases as the duration of the first flow is increased (see Fig. 4).

As a consequence, for a set of flow records, we expect throughput samples associated with long flows that have large amounts of temporal overlap to result in high throughput correlations. However, note that the occurrences of such samples are rare since the Internet is currently dominated by short flows [27]–[29]. Furthermore, throughput samples associated with long flows overlapping with short flows will give a lower value for throughput correlation. On the other hand, throughput samples associated with short flows are noisy, and will not exhibit high throughput correlation. Therefore, leaving out long and very short flows may be desirable when estimating throughput correlations that are due to congestion sharing. Since flows with long durations will typically be large (in size),⁶ we study the effect of different size thresholds to filter out large flows, and similarly, consider the impact of different size thresholds for omitting small flows. Unlike the duration of a flow, the size of a flow is invariant regardless of the capacity of links. Hence, flow size is the proper flow attribute to consider for filtering out flows.

B. Flow Class Throughput

From the record of a flow f , its perceived throughput is determined by $y_f = v_f/d_f$, where $d_f = e_f - s_f$ and, for now, $e_f > s_f$. Let us denote the set of flow records by \mathcal{F} . Each flow $f \in \mathcal{F}$ belongs to a flow class $c \in \mathcal{C}$. The function $\phi : \mathcal{F} \rightarrow \mathcal{C}$ associates a flow with a flow class. We let $\mathcal{F}_c(n) = \{f \in \mathcal{F} : \phi(f) = c \text{ and } s_f \leq n\Delta t \text{ and } (n-1)\Delta t < e_f\}$ denote the set of flows that belong to class c and are *active* at discrete time n (of length Δt). Recall that a flow class is defined as a collection of flows that have common attributes, and hence there can be more than one flow from a given class at a given time. As such, we define the *throughput of a flow class* $c \in \mathcal{C}$ as an average over the flows in that class (class average) that are active at a discrete time n at a measurement point:

$$y_c(n) = \begin{cases} \frac{1}{|\mathcal{F}_c(n)|} \sum_{f \in \mathcal{F}_c(n)} y_f, & \text{if } |\mathcal{F}_c(n)| > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

C. Factor Analysis of the Flow Class Throughput Correlation Matrix

We will assume that $\{y_c(n)\}$ is a realization of the ergodic random process of the throughput of flow class c . Let $f_{\mathbf{Y}}$ denote the joint pdf of the random vector,⁷ $\mathbf{Y} = (Y_{c_1}, Y_{c_2}, \dots, Y_{c_p})^T$, of flow class throughputs for p flow classes at stationarity; i.e.,

⁶Based on the processor sharing approximation of TCP bandwidth sharing at a bottleneck, we assume that the flows of different sizes experience the same slowdown.

⁷A random vector is a vector whose elements are random variables.

$\mathbf{Y} \sim f_{\mathbf{Y}}$. It is possible that $Y_{c_i} = 0$ for any $i = 1, \dots, p$; i.e., no flow from class c_i is active. Since we are eventually interested in the correlation structure among class throughputs at times when they are simultaneously active (or when flows overlap), we next define $\mathcal{E} = \{Y_{c_i} > 0, \forall c_i \in \mathcal{C}\}$. Also, define a random vector of flow class throughputs conditioned on \mathcal{E} , $\mathbf{Y}^* = (Y_{c_1}^*, Y_{c_2}^*, \dots, Y_{c_p}^*)^T$, with a joint pdf $f_{\mathbf{Y}^*|\mathcal{E}}$, i.e., $\mathbf{Y}^* \sim f_{\mathbf{Y}^*|\mathcal{E}}$. Denote the mean vector of \mathbf{Y}^* by $\boldsymbol{\mu}_{\mathbf{Y}^*} = (\mu_{c_1}, \mu_{c_2}, \dots, \mu_{c_p})^T$.

The idea underlying factor analysis is to consider a representation for a p -element random vector, \mathbf{Y}^* in our case, in terms of a random vector of m ($m \leq p$) *common* factors $\mathbf{F} = (F_1, F_2, \dots, F_m)^T$, and a random vector of p *unique* factors $\mathbf{U} = (U_{c_1}, U_{c_2}, \dots, U_{c_p})^T$. We assume that centered and normalized \mathbf{Y}^* can be expressed as

$$\mathbf{D}(\mathbf{Y}^* - \boldsymbol{\mu}_{\mathbf{Y}^*}) = \mathbf{A}\mathbf{F} + \mathbf{U} \quad (4)$$

where \mathbf{A} denotes a deterministic $p \times m$ *loading matrix*, and \mathbf{D} is $p \times p$ diagonal matrix consisting of the reciprocals of standard deviations of elements of \mathbf{Y}^* . The following additional assumptions are usually made [30]: $\mathbb{E}[\mathbf{F}] = \mathbf{0}$, $\text{Cov}(\mathbf{F}) = \mathbb{E}[\mathbf{F}\mathbf{F}^T] = \mathbf{I}$ (orthogonal factors), $\mathbb{E}[\mathbf{U}] = \mathbf{0}$, $\text{Cov}(\mathbf{U}) = \mathbb{E}[\mathbf{U}\mathbf{U}^T] = \boldsymbol{\Psi} = \text{diag}(\psi_1, \dots, \psi_p)$ (a diagonal matrix), and $\text{Cov}(\mathbf{U}, \mathbf{F}) = \mathbf{0}$. The assumption that $\boldsymbol{\Psi}$ is diagonal means that all covariances among class throughputs are accounted by the factors. Then, using (4), one can write the correlation matrix \mathbf{R} of \mathbf{Y}^* :

$$\mathbf{R} = \text{Corr}(\mathbf{Y}^*) = \mathbf{A}\mathbf{A}^T + \boldsymbol{\Psi}. \quad (5)$$

The elements of the loading matrix \mathbf{A} , Λ_{ij} , capture the degree of correlation exhibited between a given factor (corresponding column in \mathbf{A}) and variable (corresponding row in \mathbf{A}). Estimates $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\Psi}}$ for \mathbf{A} and $\boldsymbol{\Psi}$ can be determined by using the *principal component method* as follows (see [30] for more details). First, the (positive definite) correlation matrix in (5) is expressed as

$$\mathbf{R} = \lambda_1 \boldsymbol{\xi}_1 \boldsymbol{\xi}_1^T + \lambda_2 \boldsymbol{\xi}_2 \boldsymbol{\xi}_2^T + \dots + \lambda_p \boldsymbol{\xi}_p \boldsymbol{\xi}_p^T$$

where $(\lambda_i, \boldsymbol{\xi}_i)$ are the eigenvalue-eigenvector pairs such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p > 0$. $\hat{\mathbf{A}}$ and $\hat{\boldsymbol{\Psi}}$ can be determined by taking the largest m eigenvalues, and by approximating \mathbf{R} as

$$\begin{aligned} \mathbf{R} &\approx \hat{\mathbf{A}}\hat{\mathbf{A}}^T + \hat{\boldsymbol{\Psi}} \\ &= (\sqrt{\lambda_1}\boldsymbol{\xi}_1, \dots, \sqrt{\lambda_m}\boldsymbol{\xi}_m) \\ &\quad \times (\sqrt{\lambda_1}\boldsymbol{\xi}_1, \dots, \sqrt{\lambda_m}\boldsymbol{\xi}_m)^T \\ &\quad + \begin{pmatrix} \hat{\psi}_1 & 0 & \dots & 0 \\ 0 & \hat{\psi}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{\psi}_p \end{pmatrix} \end{aligned} \quad (6)$$

so that $\hat{\Lambda}_{i1}^2 + \hat{\Lambda}_{i2}^2 + \dots + \hat{\Lambda}_{im}^2 + \hat{\psi}_i = \hat{h}_i^2 + \hat{\psi}_i = 1$ for $i = 1, \dots, p$, where \hat{h}_i^2 is called the *communality*, and $\hat{\psi}_i$ is called the *specific variance*. The communality represents the portion of the normalized variance of $Y_{c_i}^*$ that is accounted by the m

common factors, while $\hat{\psi}_i$ reflects the portion of the normalized variance due to a factor that is unique to $Y_{c_i}^*$.

D. Selection of the Number of Factors and Explanatory Power

When using the principal component method to “factor” the correlation matrix *without any assumptions on the distribution of variables*, there are only ad hoc heuristics for determining the sufficiency of the number of factors m in the model. In exploratory studies, one common approach to determine m is proposed by Kaiser [31]. Kaiser’s rule proposes selecting factors whose variances (λ_i) are greater than 1. The intuition behind this rule is that a factor that has a variance less than 1 contains less information than a (normalized) original variable does. The number of factors m used in the model needs to account for a “reasonable” proportion of total variance (a measure of overall variability), which is given by the trace of the correlation matrix (p). If the proportion of total variance captured by the common factors, $1 - \sum_{i=1}^p \psi_i/p$, is “high”, then we say that the factors have a strong *explanatory power*. Our experiments with simulation data suggest that Kaiser’s rule generally retains fewer factors than expected. Therefore, we select m based on the number of eigenvalues that are greater than 0.9. In contrast, we have found that Kaiser’s rule produces good results with real TCP data with the following modification: We select m based on the number of eigenvalues whose confidence intervals contain 1 or lie above 1.

E. Interpretation of Factor Loadings

For our application, the factors represent shared congested resources that cause variation in class throughputs. Among the loadings for a given class c_i , i.e., $\hat{\Lambda}_{i1}, \hat{\Lambda}_{i2}, \dots, \hat{\Lambda}_{im}$, the loading(s) with the largest magnitude(s) are treated as significant loadings. The flow classes that have the largest loading (in magnitude) on a common factor are identified as classes that are likely to share a congested resource in the network: Most of the variations in the throughput for those classes are accounted by a common factor or congested resource.

Note that the loading matrix is determined only up to an orthogonal rotation matrix $\mathbf{\Gamma}$. If $\mathbf{A}^* = \mathbf{A}\mathbf{\Gamma}$, then

$$\mathbf{R} = \mathbf{A}^* \mathbf{A}^{*T} + \boldsymbol{\Psi} = \mathbf{A}\mathbf{\Gamma}\mathbf{\Gamma}^T\mathbf{A}^T + \boldsymbol{\Psi} = \mathbf{A}\mathbf{A}^T + \boldsymbol{\Psi}.$$

Throughout this paper, we apply a rotation to the loading matrix to obtain a better description of the factors by using a common method in factor analysis called *varimax* rotation [30]. Varimax rotation attempts to find a rotation matrix $\mathbf{\Gamma}$ such that the squares of the loadings on each factor are as spread out as possible. More specifically, $\mathbf{\Gamma}$ is chosen to maximize

$$\sum_{k=1}^m \left[\sum_{j=1}^p \Lambda_{jk}^{*4} - \frac{1}{p} \left(\sum_{j=1}^p \Lambda_{jk}^{*2} \right)^2 \right].$$

This criterion tends to drive squared loadings towards either zero or one, away from intermediate values. Hence, deciding which loadings are significant is easier with \mathbf{A}^* . Fig. 5 illustrates how \mathbf{A}^* can be interpreted to identify congestion sharing flow classes.

	Factor 1	Factor 2	
$\Lambda^* =$	Λ_{11}^*	Λ_{12}^*	Class 1
	Λ_{21}^*	Λ_{22}^*	Class 2
	Λ_{31}^*	Λ_{32}^*	Class 3
	Λ_{41}^*	Λ_{42}^*	Class 4
	Λ_{51}^*	Λ_{52}^*	Class 5

Fig. 5. An example (rotated) loading matrix. Suppose that there are five flow classes, and we have identified two common factors and boxed the factor loading with the largest magnitude in each row. We can then infer that classes 1, 2, and 5 share one congested resource (Factor 1), and 3 and 4 share another (Factor 2).

F. Squared Error Loss

In addition to explanatory power which exhibits the ability of the factors to account for the variance in data, we need a metric to assess the efficacy of factor loadings in *distinguishing* which factors have the most effect on throughputs, and to investigate various flow filtering schemes to improve the ability to infer congestion sharing in controlled simulations (in which we know the topology). We define *squared error loss*⁸ as

$$L := \|\mathbf{\Lambda}^0 - \text{abs}(\hat{\mathbf{\Lambda}})\|^2 = \sum_{i=1}^p \sum_{j=1}^m (\Lambda_{ij}^0 - |\hat{\Lambda}_{ij}|)^2 \quad (7)$$

where $\Lambda_{ij}^0 = 1$ if the flow class c_i shares the factor j , and $\Lambda_{ij}^0 = 0$ otherwise, which correspond to “ideal” loadings in a matrix $\mathbf{\Lambda}^0$. When there is only one factor ($m = 1$), we set $\Lambda_{i1}^0 = 1$ for all i . The notation $\|\cdot\|$ denotes the Euclidian norm, and for a matrix, it is given by the square root of the sum of squares of each element in the matrix. The function $\text{abs}(\cdot)$ returns a matrix whose elements are the absolute values of the corresponding elements in the input matrix. Note that squared error loss penalizes large deviations from the ideal more than small deviations.

G. Estimation of the Correlation Matrix

One drawback of the development until this point is that all of the flows must be active at a given time to contribute an observation of random vector \mathbf{Y}^* when we estimate first- and second-order statistics of flow class throughputs. Therefore, when there are only a few flows belonging to a flow class under consideration, only a few class throughput observations are available for statistical analysis. To address this problem, we estimate the *elements* of the correlation matrix of flow class throughputs using separate bivariate analyses: In order to estimate an element (a correlation) of the matrix, we use the class throughput observations at times when the flow class *pair* is active. Such an element-wise approach is employed in multivariate statistics when there are “missing values” for one or more variables in a significant number of observation vectors (see for example, [30] and [32]). We adopt this approach to compute correlations since there are a lot of sampling instants when not all of the variables can be manipulated simultaneously: Instead of having missing

⁸If comparisons *across* different bottleneck configurations (different number of classes and different number of congested resources) are desired, one can divide squared error loss by $p \times m$, the number of elements in the loading matrix.

TABLE I
KEY PARAMETERS

p	Number of flow classes
$\rho_{c_i c_j}$	Correlation between throughputs of c_i and c_j
m	Number of shared congested resources (factors)
Λ_{ij}	Correlation between throughput of c_i and resource j
Ψ_i	Specific variance of throughput of c_i
λ_i	i th eigenvalue of \mathbf{R}
$1 - \sum_{i=1}^p \psi_i / p$	Explanatory power of m factors
L	Squared error loss

values, simply, no flow from a given flow class is active at that instant. Since the correlation matrix constructed in this way may not always be positive definite, the matrix can be adjusted to make it positive definite. For example, as proposed in [33], a constant may be added to the nonpositive eigenvalues of \mathbf{R} to make them positive.

Accordingly, we develop the following conditional sampling strategy: Define the event $P(n, c_i, c_j) = \{y_{c_i}(n) > 0 \text{ and } y_{c_j}(n) > 0, \text{ for } c_i, c_j \in \mathcal{C}\}$; i.e., both classes c_i and c_j are active at (discretized) time n . Let $N(T, c_i, c_j)$ be the number of discrete times over which both flow classes c_i and c_j are active, where T denotes the measurement period. *Pairwise* mean and variance are given by

$$\mu_{c_i, c_j} = \lim_{T \rightarrow \infty} \frac{1}{N(T, c_i, c_j)} \sum_{n=1}^T y_{c_i}(n) \mathbf{1}_{P(n, c_i, c_j)}$$

$$\sigma_{c_i, c_j}^2 = \lim_{T \rightarrow \infty} \frac{1}{N(T, c_i, c_j)} \sum_{n=1}^T (y_{c_i}(n) - \mu_{c_i, c_j})^2 \mathbf{1}_{P(n, c_i, c_j)}$$

where $\mathbf{1}_E$ is the standard indicator function, and is equal to 1 if E is true and 0, otherwise. The correlation of throughputs of flow classes c_i and c_j is given by

$$\rho_{c_i, c_j} = \lim_{T \rightarrow \infty} \frac{\sum_{n=1}^T (y_{c_i}(n) - \mu_{c_i, c_j})(y_{c_j}(n) - \mu_{c_j, c_i}) \mathbf{1}_{P(n, c_i, c_j)}}{N(T, c_i, c_j) \sigma_{c_i, c_j} \sigma_{c_j, c_i}} \quad (8)$$

Hence, a correlation matrix can be approximated using correlations (from separate bivariate analyses) between variables:

$$\mathbf{R} \approx (\rho_{c_i c_j}), \quad i, j = 1, \dots, p \quad (9)$$

where $\rho_{c_i c_j}$ is given by (8) (and $\rho_{c_i c_i} = 1$). Also, in Table I, we tabulate the key parameters used to infer congestion sharing for easy reference.

H. Bootstrap Methods

Since the distribution of real throughput data is unknown, we resort to the *bootstrap* method [34] to compute the bias-corrected and accelerated (BC_a) confidence intervals whenever we wish to assess the statistical accuracy of eigenvalues and factor loadings, which are key parameters for making congestion sharing inferences. The bootstrap is a computer-based method that depends on *resampling* a given set of data consisting of N samples B times. A bootstrap sample is a random sample of size N drawn with replacement from the original sample. Corresponding to each bootstrap replication, an estimate for the parameter of interest $\hat{\theta}^*(b)$ is computed for $b = 1, \dots, B$. One can then use the independent replications of $\hat{\theta}$ to obtain

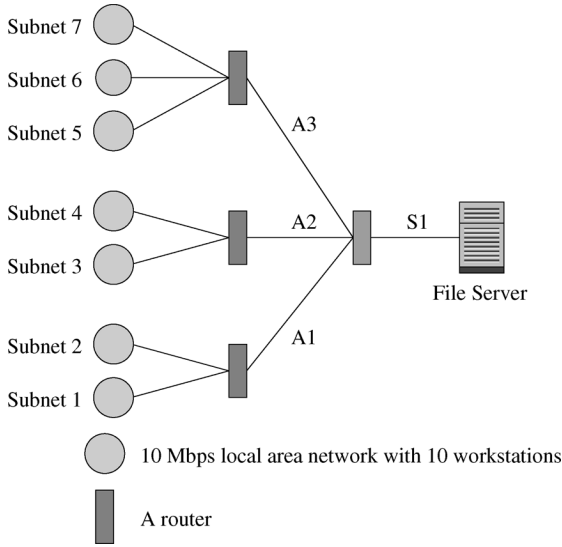


Fig. 6. Tree topology used in OPNET TCP simulations.

confidence intervals. The BC_a confidence intervals are very close to the exact intervals of θ . The recommended number of replications required for computing the BC_a confidence intervals is at least 1000 [34]. A detailed discussion of bootstrap methods can be found in [34].

III. TCP SIMULATIONS

In this section, we present results based on part of an extensive set of OPNET Modeler [35] simulations. The primary aim of this section is to validate the methods introduced in Section II for identifying congestion sharing flow classes in a controlled environment in which the routes from the sources to destinations are known exactly. The effectiveness of factor analytic methods in identifying such flow classes under different traffic conditions and different network configurations is evaluated.

Each simulation in this section corresponds to 2 hours of file download activity. That is, the measurement period T is 2 hours when computing the correlation in (8). The length of the discrete interval Δt (see Section II-B) is chosen as 1 second. During simulations, we record the request time, size (in bytes) and duration (in seconds) of each file transfer. File transfer requests arrive according to a Poisson process, and file sizes are selected from a lognormal distribution [27]–[29] with a mean file size of 16 kB and a standard deviation of 131 kB. On each bottleneck link, we include background traffic to model the effects of additional traffic from other users or applications.

A. Networks With Tree Topologies

While actual networks rarely look like trees, tree topologies have been frequently used in studies that infer which flows share congested resources. Trees may provide a good abstraction for logical topologies. For example, each tree branch can represent a link that may potentially become a bottleneck in the actual network, while overprovisioned links are not included in the tree.

Consider the tree topology shown in Fig. 6. Users download files from a server using the File Transfer Protocol (FTP). Access links are denoted by A1, A2, and A3, and the link con-

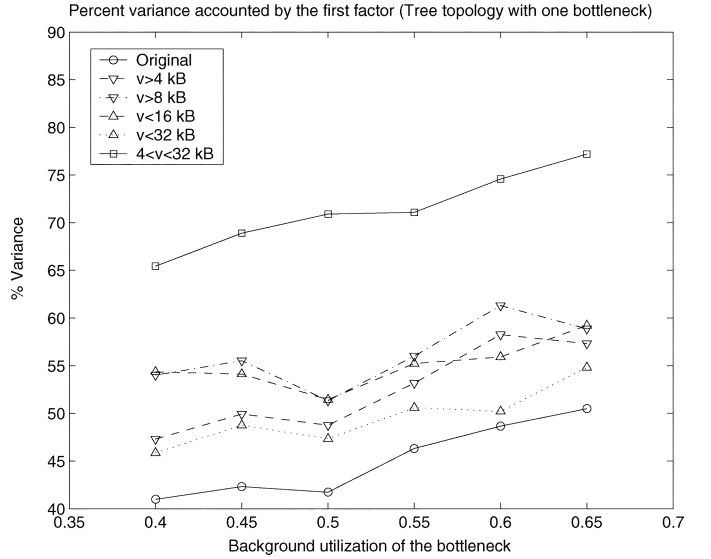


Fig. 7. Percent variance under different background traffic conditions on the single bottleneck. The total load due to classes 1–7 on S1 is kept at 30%.

necting the FTP server to the network is denoted by S1. Seven classes of flows are defined according to their local subnet addresses. Each subnet is a 10 Mbps local area network that has 10 workstations.

In order to be able to capture the throughput correlation between flow classes successfully, we consider the effect of flow sizes on correlation estimates. The results in Section II-A suggested that removing small and large flows from flow records under consideration might improve our ability to capture positive throughput correlation among temporally overlapping, congestion sharing flows.

In the subsequent discussion, we will analyze the effect of retaining flows satisfying the following size conditions: $\{f \in \mathcal{F} : v_f > 4 \text{ kB}\}$, $\{f \in \mathcal{F} : v_f > 8 \text{ kB}\}$, $\{f \in \mathcal{F} : v_f < 16 \text{ kB}\}$, $\{f \in \mathcal{F} : v_f < 32 \text{ kB}\}$, and $\{f \in \mathcal{F} : 4 < v_f < 32 \text{ kB}\}$. We refer to the results based on the entire set of flows (without filtering) as the “original”. Note that these thresholds are not intended for establishing criteria for selecting or defining large or small flows in the Internet. Rather, our focus will be on analyzing the effect of removing certain sized flows on flow class throughput correlations.

1) *Single Bottleneck: Effect of Background Traffic:* Consider the case in which link S1 (1.544 Mbps) in Fig. 6 is the bottleneck and access links A1, A2, and A3 are overprovisioned (44.736 Mbps). The users belonging to classes 1–7 generate a total load of 30% on the bottleneck link and the bottleneck’s background traffic utilization is varied from 40% to 65% to demonstrate the ability to identify this bottleneck under different background traffic conditions.

Using the modified Kaiser’s rule described in Section II-D, we correctly determine that there is one significant factor for each utilization factor considered. Fig. 7 shows that the explanatory power of this single factor increases as congestion increases on the bottleneck. Fig. 8 illustrates that at higher utilization levels of the bottleneck link, squared error loss is smaller. Both Figs. 7 and 8 show the effect of filtering out small flows, large flows, and small and large flows simultaneously on the

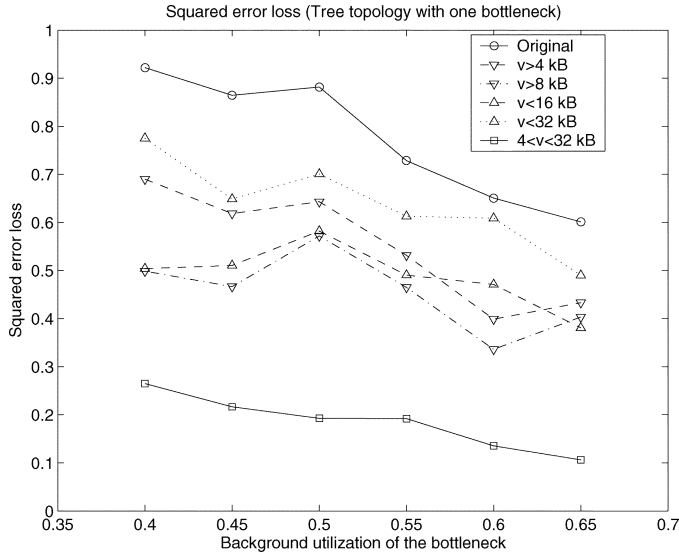


Fig. 8. Squared error loss under different background traffic conditions on the single bottleneck. The total load due to classes 1–7 on S1 is kept at 30%.

percentage of variance accounted by the significant factor and on the squared error loss. We see that filtering out small flows or large flows improves the explanatory power of the factor and decreases the squared error loss. Note that increasing (decreasing) the lower (higher) filtering threshold of flow sizes has significant benefits on these measures. However, increasing (decreasing) the lower (higher) threshold retains fewer flows and decreases the statistical accuracy of estimates. An important observation is that omitting *both* small and large flows simultaneously significantly improves the explanatory power of the factor and decreases the squared error loss. Retaining flows whose sizes are between 4 kB and 32 kB is a compromise between reliability of inference for resource sharing and statistical accuracy of estimates.

2) *Single Bottleneck: Effect of Class Loads:* Consider again the case in which link S1 (1.544 Mbps) is the bottleneck and access links A1, A2, and A3 are overprovisioned (44.736 Mbps). We investigate the ability to identify this bottleneck for different total loads (20%, 30%, 40%) generated by users belonging to classes 1–7 on the bottleneck link S1. Background traffic utilizes 50% of link S1.

Using the modified Kaiser’s rule, we successfully determine that there is one significant factor when total class loads on the bottleneck link are greater than or equal to 20%. Fig. 9 shows that the explanatory power of this factor increases as congestion increases on the bottleneck. Fig. 10 illustrates that at higher utilization levels, squared error loss is smaller. Again, omitting both small and large flows simultaneously significantly improves the explanatory power of the factor and decreases the squared error loss.

3) *Single Bottleneck: Effect of Nonstationary Traffic:* We also investigate the effect of having nonstationary background traffic for the single bottleneck case described in the previous subsections. The background traffic utilization of the bottleneck link changes between 60% and 40% every 20 minutes over the

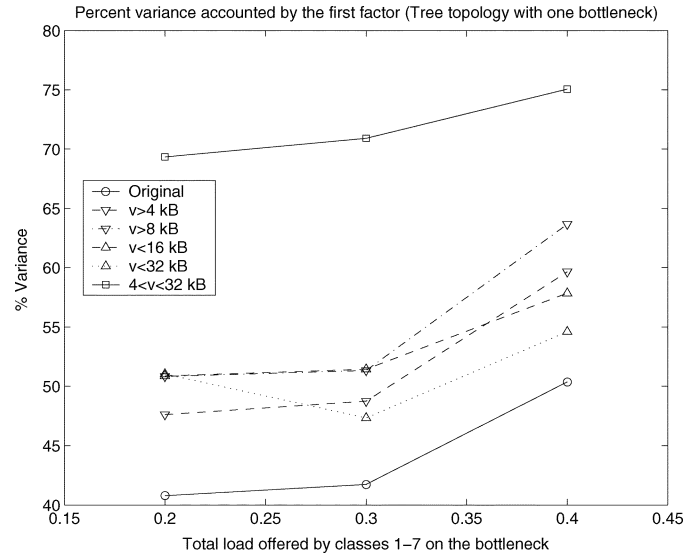


Fig. 9. Percent variance under different total loads from classes 1–7 on the single bottleneck S1. The utilization of S1 due to background traffic is kept at 50%.

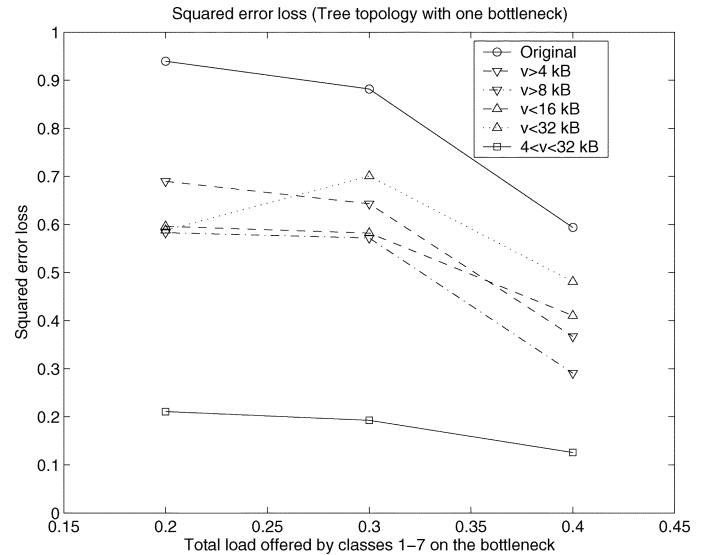


Fig. 10. Squared error loss under different total loads from classes 1–7 on the single bottleneck S1. The utilization of S1 due to background traffic is kept at 50%.

period of 2 hours. Using the modified Kaiser’s rule, we successfully determine that there is one significant factor with explanatory power 71%. For this particular scenario, nonstationarity of the network traffic does not seem to affect determination of resource sharing. Further analysis may be required to assess the impact of nonstationarity of network traffic on inference results.

4) *Three Bottlenecks:* Consider the case in which links A1, A2, and A3 (each 1.544 Mbps) are bottlenecks and link S1 is overprovisioned (44.736 Mbps). We investigate the ability to identify these bottlenecks and *associate* each flow class with a bottleneck for different loads (10%, 15%, 20%) generated by *each* class.

The background utilization on A1 and A2 (each serving two subnets) is set to 50%. The background utilization on

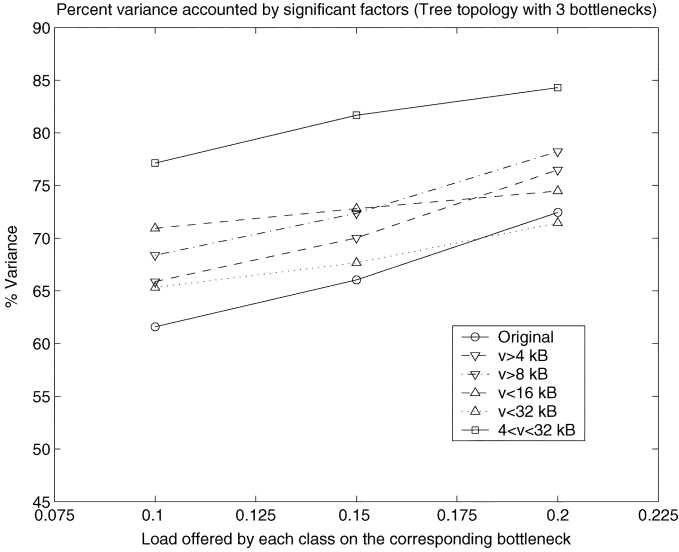


Fig. 11. Percent variance under different loads offered by each of the classes 1–7 for the three-bottleneck scenario. The total utilization factors of each of the bottlenecks are the same in each offered load case, and are 70%, 80%, and 90%, respectively.

A3 (serving three subnets) is adjusted so as to keep the total utilization on each bottleneck (A1, A2, and A3) the same; i.e., the background utilization on A3 is 40%, 35%, and 30% corresponding to 10%, 15%, and 20% loads offered by each class, respectively. From the generated records, we successfully determine that there are three significant factors when loads generated by each class are greater than or equal to 10%. Furthermore, we correctly identify which flow classes share congested resources: We find that the throughputs of flow classes from subnets 1 and 2 have the largest loading with factor 1, the throughputs of flow classes from subnets 3 and 4 have the largest loading with factor 2, and the throughputs of flow classes from subnets 5, 6, and 7 have the largest loading with factor 3. These factors are interpreted as the bottleneck access links A1, A2, and A3, respectively. Fig. 11 shows that the explanatory power of the three factors increases as congestion increases on the bottlenecks. Fig. 12 illustrates that at higher utilization factors, the factor loadings distinguish which factor a flow class is most associated with more easily. As in the previous cases, omitting both small and large flows simultaneously significantly improves the explanatory power of the factors and decreases the squared error loss.

B. Interaction of Coupled Flow Classes

We also show how factor analysis identifies two coupled bottlenecks in an example scenario given in Fig. 13. Users belonging to subnet 1 (class 1) download files from server 1, users belonging to subnet 2 (class 2) download files from server 2, and users belonging to subnet 3 (class 3) download files from server 3. Class 1 offers a load of 20% on the bottleneck link 1 and 2. Class 2 offers a load of 40% on the bottleneck link 1. Class 3 offers a load of 40% on the bottleneck link 2. The load due to background traffic on the bottlenecks 1 and 2 is set to 20%.

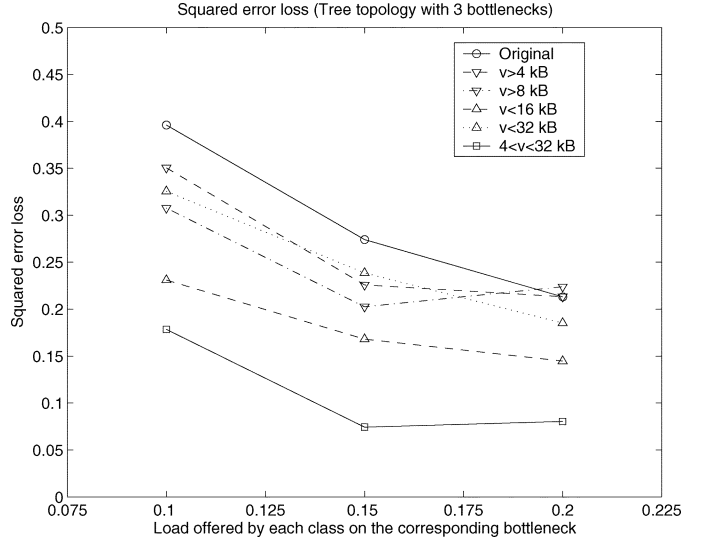


Fig. 12. Squared error loss under different loads offered by each of the classes 1–7 for the three-bottleneck scenario. The total utilization factors of each of the bottlenecks are the same in each offered load case, and are 70%, 80%, and 90%, respectively.

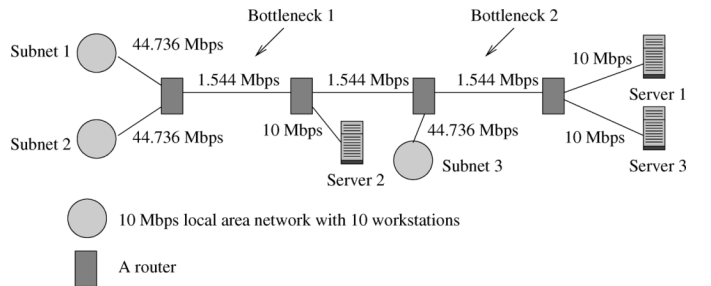


Fig. 13. Linear network topology with coupled bottleneck links used in OPNET TCP simulations.

After filtering out flows whose sizes are smaller than 4 kB or greater than 32 kB, we find that there are two significant factors. Then, we estimate factor loadings of four class throughputs based on two significant factors. We first estimate factor loadings and specific factors based on (6), and then use varimax rotation [30] on $\hat{\Lambda}$:

$$\hat{\Lambda}^* = \begin{pmatrix} \boxed{0.5011} & \boxed{0.7272} \\ \boxed{0.9570} & \boxed{-0.0041} \\ -0.1143 & \boxed{0.9214} \end{pmatrix}$$

(significant loadings boxed) and

$$\hat{\Psi} = \text{diag}(0.2200, 0.0841, 0.1380).$$

The explanatory power of the two factors is 85%. From the results, one can see that the throughputs of classes 2 and 3 are captured by only one factor, i.e., the bottleneck link that each traverses. For class 1, one can argue that both loadings are significant, and hence the throughputs of class 1 can be explained by two factors, i.e., the two bottlenecks the flows belonging to class 1 visit. This example scenario shows the effectiveness of factor analysis in identifying *multiple* bottlenecks in a linear network.

TABLE II
DESCRIPTION OF NETFLOW DATASETS COLLECTED AT
UT AUSTIN'S BORDER ROUTER

	Date	Period	TCP records
Dataset2002	11/6/2002	12:58 PM - 2:07 PM	5,173,385
Dataset2004	1/21/2004	12:58 PM - 1:26 PM	4,440,697

IV. ANALYSIS OF ACTUAL TCP FLOW RECORDS

In this section, we apply factor analysis to TCP flow class throughput correlation matrices that are constructed using actual TCP flow records collected by networking equipment. With actual TCP flow measurements, a validation of the inferences of flow classes sharing congestion is extremely hard, if not impossible, since routing information about all the domains that flows visit and the congestion status of the servers that provide the incoming traffic are not available. However, bootstrap confidence intervals can be used to demonstrate the statistical accuracy of the inferences.

A. Description of Datasets

We use NetFlow [4] records collected at the border router of The University of Texas at Austin (UT Austin) on November 6, 2002, between 12:58 PM and 2:07 PM CST, and on January 21, 2004, between 12:58 PM and 1:26 PM CST. The records that are collected in 2002 are referred to as Dataset2002, and those that are collected in 2004 are referred to as DataSet2004. Dataset2002 consists of 5 173 385 TCP flow records out of a total of 5 866 602 flow records. Dataset2004 consists of 4 440 697 TCP flow records out of a total of 6 556 674 flow records. The records contain both the incoming and outgoing traffic from UT Austin. The IP addresses belonging to UT Austin were made anonymous to protect privacy. Table II summarizes these datasets.

We assume that over a one-hour period, flow class throughputs can be modeled as stationary processes. Furthermore, we assume that the packets from a given TCP flow follow the same route.⁹ Such assumptions, although idealized, are not completely unrealistic for our one-hour long flow measurements.

B. Methodology

In NetFlow records, the start time of a flow is the time of arrival of the first packet in the flow, and the end time is the time of arrival of the last packet in the flow. Since the time between the first and the last packet is zero, flow throughput is not defined for flows consisting of one packet. Hence, one-packet flows will be omitted. From the premises of Section II-A that are validated by performed simulations in Section III, we filter out all flow records whose sizes are smaller than one threshold or larger than another threshold in order to better capture the throughput correlations among flow classes. Based on empirical investigations and datasets at hand, we select to filter out flows whose sizes are less than 8 kB or greater than 64 kB. The

⁹This assumption is supported by the empirical measurements in [36].

choices for these thresholds are based on some practical, empirical considerations: For example, Estan and Varghese [7] define “small” flows as those that send less than 0.1% of the link capacity during a given measurement interval, say 1 second. For instance, for a (bottleneck) OC-1 (optical carrier level 1) link of 51.84 Mbps, a small flow will be one that transports less than 7 kB. When choosing the upper threshold value for filtering out flows, we took into account the measurement studies that find that 50 kB Web objects (carried by TCP) are becoming common in the Internet [37]. Therefore, we can consider a flow whose size is larger than 64 kB as “large”. In addition, in the Internet, packets belonging to flows that consist of only a few packets can sometimes arrive back to back (or with a very small inter-packet spacing). In this case, it is unreasonable to assume that such large flow throughputs are typical for that flow class [20]. Hence, we will also omit all flows whose durations are shorter than one second.¹⁰

We choose to analyze incoming traffic (flow records with source IP addresses) associated with AOL and HotMail, since one can reasonably expect that traffic belonging to these content providers potentially experience congestion at their source due to high demand for their content. We define two flow classes for traffic from each provider: AOL1 and AOL2 (class 1 and class 2) from AOL, and HotMail1 and HotMail2 (class 3 and class 4) from Microsoft Corporation. Assignment of flows into AOL1 or AOL2 (and similarly for HotMail1 and HotMail2) is performed by randomly splitting all flows from AOL (and HotMail) into two sets. The cumulative distribution functions in Figs. 14 and 15 provide some insight into the flow size distributions of the chosen flows from the datasets. Note that the measurement period T used in computing the correlation in (8) is 69 minutes for Dataset2002 and 28 minutes for Dataset2004. The length of the discrete interval Δt (see Section II-B) is chosen as 1 second.¹¹

C. Statistical Accuracy of Inferences

First, we describe how to choose the number of significant factors. We estimate 95% BC_a confidence intervals for four eigenvalues of the class throughput correlation matrix \mathbf{R} in (9). The results are shown in Table III for Dataset2002, and in Table IV for Dataset2004.

Once the confidence intervals for the eigenvalues are estimated, the modified Kaiser's rule for real data can be used to choose the significant eigenvalues: the eigenvalues whose confidence intervals lie below 1 are designated as *insignificant*. Therefore, from Tables III and IV, there are two significant factors; i.e., four classes share two different network infrastructures. The explanatory power of the two factors is 72% in the case of Dataset2002 and 63% in the case of Dataset2004.

After establishing the number of significant factors, we estimate factor loadings of four class throughputs based on two significant factors. We first estimate factor loadings and

¹⁰Flows can also be categorized according to their duration. Brownlee and Claffy [37] term flows whose durations are less than 2 seconds as “dragonflies”.

¹¹We find that using finer scale intervals does not affect the reported results. Using intervals that are longer than 1 second will reduce the number of samples of flow class throughputs, thereby producing less accurate results.

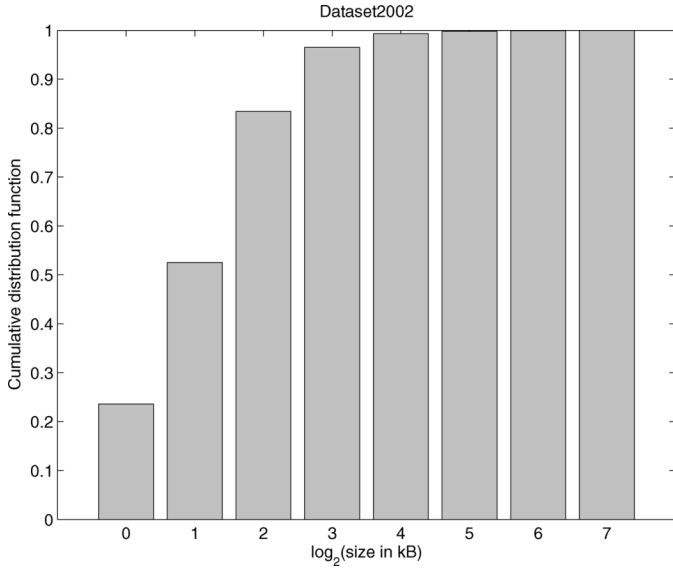


Fig. 14. Cumulative distribution function of flow sizes in kB under consideration from Dataset2002.

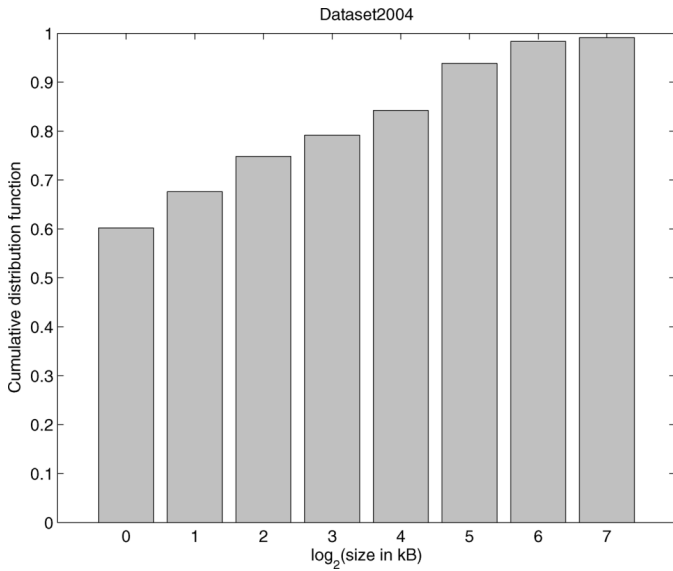


Fig. 15. Cumulative distribution function of flow sizes in kB under consideration from Dataset2004.

specific factors based on (6), and then use varimax rotation on $\hat{\mathbf{A}}$. For DataSet2002

$$\hat{\mathbf{A}}^* = \begin{pmatrix} \boxed{0.7933} & 0.0711 \\ \boxed{0.7289} & -0.1315 \\ -0.0842 & \boxed{0.9088} \\ 0.0501 & \boxed{0.9240} \end{pmatrix}$$

(significant loadings boxed) and

$$\hat{\Psi} = \text{diag}(0.3656, 0.4514, 0.1669, 0.1437).$$

TABLE III
MEAN OF BOOTSTRAP REPLICATIONS AND 95% CONFIDENCE INTERVALS FOR EIGENVALUES OF \mathbf{R} BASED ON DATASET2002

Eigenvalue	Mean	Interval estimate
λ_1	1.7274	(1.5457, 1.7900)
λ_2	1.1562	(1.0861, 1.3206)
λ_3	0.8344	(0.7058, 0.9150)
λ_4	0.2785	(0.2194, 0.4458)

TABLE IV
MEAN OF BOOTSTRAP REPLICATIONS AND 95% CONFIDENCE INTERVALS FOR EIGENVALUES OF \mathbf{R} BASED ON DATASET2004

Eigenvalue	Mean	Interval estimate
λ_1	1.4287	(1.3646, 1.4786)
λ_2	1.0780	(1.0237, 1.1603)
λ_3	0.9094	(0.8230, 0.9690)
λ_4	0.5856	(0.5413, 0.6379)

TABLE V
MEAN OF BOOTSTRAP REPLICATIONS AND 95% CONFIDENCE INTERVALS FOR FACTOR LOADINGS BASED ON DATASET2002

Loading	Mean	Interval estimate
Λ_{11}^*	0.7944	(0.7567, 0.8252)
Λ_{12}^*	0.0761	(0.0036, 0.1688)
Λ_{21}^*	0.7250	(0.6360, 0.7884)
Λ_{22}^*	0.1331	(0.0235, 0.2401)
Λ_{31}^*	0.0836	(0.0241, 0.1436)
Λ_{32}^*	0.9110	(0.8564, 0.9362)
Λ_{41}^*	0.0535	(0.0042, 0.1294)
Λ_{42}^*	0.9250	(0.8806, 0.9483)

For Dataset2004

$$\hat{\mathbf{A}}^* = \begin{pmatrix} \boxed{0.8378} & -0.0451 \\ \boxed{0.8411} & 0.0044 \\ 0.0200 & \boxed{-0.7415} \\ 0.0260 & \boxed{-0.7351} \end{pmatrix}$$

(significant loadings boxed) and

$$\hat{\Psi} = \text{diag}(0.2961, 0.2926, 0.4497, 0.4589).$$

Next, 95% BC_a confidence intervals for absolute values of eight rotated factor loadings are computed. When computing confidence intervals for factor loadings, one needs to take into account sign reversals of loadings and changes in the order of factors across bootstrap samples [38]. As such, we compute the confidence intervals of the absolute values of loadings. We rearrange the order of factors if such reordering results in a smaller $\|\hat{\mathbf{A}}^* - \hat{\mathbf{A}}^*(b)\|$, where $\hat{\mathbf{A}}^*$ is estimated using (6) and varimax rotation, and $\hat{\mathbf{A}}^*(b)$ is the estimate for $\hat{\mathbf{A}}^*$ using the b th bootstrap replication. The results are given in Table V for DataSet2002 and in Table VI for DataSet2004.

By inspecting the significant loadings on the loading matrix, we can conclude that classes 1 and 2 (flows belonging to AOL) share factor 1, and classes 3 and 4 (flows belonging to Hot-Mail) share factor 2 with 95% confidence. In this case, factor 1 would be interpreted as the networking infrastructure belonging to AOL, and factor 2 would be the networking infrastructure belonging to Microsoft Corporation.

TABLE VI
MEAN OF BOOTSTRAP REPLICATIONS AND 95% CONFIDENCE INTERVALS FOR
FACTOR LOADINGS BASED ON DATASET2004

Loading	Mean	Interval estimate
Λ_{11}^*	0.8370	(0.8223, 0.8544)
Λ_{12}^*	0.0478	(0.0034, 0.1312)
Λ_{21}^*	0.8402	(0.8254, 0.8580)
Λ_{22}^*	0.0303	(0.0000, 0.1296)
Λ_{31}^*	0.0458	(0.0000, 0.0731)
Λ_{32}^*	0.7395	(0.6314, 0.7879)
Λ_{41}^*	0.0512	(0.0004, 0.0969)
Λ_{42}^*	0.7316	(0.6207, 0.7737)

D. Discussion of Results

The potential power of this inference technique in root cause analysis may be illustrated by considering the results in Tables V and VI. For example, suppose that the users belonging to classes AOL1 and AOL2 at UT Austin were experiencing poor performance (excessive download times). Treating the external network as a “black box” (i.e., no knowledge about the utilization factors of access links or routing information of outside network), network managers could infer that poor performance was not due to the access links connecting UT Austin to the Internet, because the flow classes did not have one common factor that would indicate a bottleneck shared by all classes. The network managers could then hypothesize that the cause for poor performance was either at the content provider’s server or a corresponding bottleneck link visited by pairs of flow classes (1, 2 and 3, 4) in the Internet.

V. CONCLUSION

Our proposed approach for inferring congestion sharing based on flow records can serve as a tool for network monitoring and root cause analysis of poor performance, and differs drastically from the packet based methodologies employed so far to infer network-internal characteristics. We believe that the use of factor analysis in analyzing network behavior is a novel idea. A distinctive feature of our work is the consideration of the correlation structure of conditionally sampled random processes (flow class throughputs) whose samples are taken when the classes are active at the sampling instant.

ACKNOWLEDGMENT

The authors would like to thank Mr. J. Wang and Prof. W. C. Bard for providing the flow records collected at UT Austin’s border router.

REFERENCES

- [1] S. Shenker, “Fundamental design issues for the future Internet,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1176–1188, Sep. 1995.
- [2] IP Monitoring Project. Sprint Corp. [Online]. Available: <http://ipmon.sprintlabs.com>
- [3] N. Brownlee, C. Mills, and G. Ruth, “Traffic flow measurement: Architecture,” IETF RFC 2722, Oct. 1999.
- [4] NetFlow. Cisco Systems, Inc. [Online]. Available: <http://www.cisco.com>
- [5] sFlow. sFlow.org [Online]. Available: <http://www.sflow.org>
- [6] Argus. QoSient, llc [Online]. Available: <http://www.qosient.com/argus>
- [7] C. Estan and G. Varghese, “New directions in traffic measurement and accounting,” in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2001, pp. 75–80.
- [8] N. Duffield, C. Lund, and M. Thorup, “Properties and prediction of flow statistics from sampled packet streams,” in *Proc. ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2002, pp. 159–171.

- [9] S. Savage, N. Cardwell, and T. Anderson, “The case for informed transport protocols,” in *Proc. IEEE Workshop on Hot Topics in Operating Systems*, Mar. 1999, pp. 58–63.
- [10] K. Harfoush, A. Bestavros, and J. Byers, “Robust identification of shared losses using end-to-end unicast probes,” in *Proc. IEEE Int. Conf. Network Protocols*, Nov. 2000, pp. 22–36.
- [11] D. Rubenstein, J. Kurose, and D. Towsley, “Detecting shared congestion of flows via end-to-end measurement,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 381–395, Jun. 2002.
- [12] M. S. Kim, T. Kim, Y. Shin, S. S. Lam, and E. J. Powers, “A wavelet-based approach to detect shared congestion,” in *Proc. ACM Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications*, Aug. 2004, pp. 293–306.
- [13] M. Rabbat, R. Nowak, and M. Coates, “Network tomography and the identification of shared infrastructure,” in *Proc. IEEE Asilomar Conf. Signals, Systems and Computers*, Nov. 2002, pp. 34–38.
- [14] D. Katabi, I. Bazzi, and X. Yang, “A passive approach for detecting shared bottlenecks,” in *Proc. IEEE Int. Conf. Computer Communications and Networks*, 2001, pp. 174–181.
- [15] D. Arifler, G. de Veciana, and B. L. Evans, “Network tomography based on flow level measurements,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 2004, vol. 2, pp. 437–440.
- [16] —, “Inferring path sharing based on flow level TCP measurements,” in *Proc. IEEE Conf. Communications*, Jun. 2004, vol. 4, pp. 2054–2059.
- [17] D. Arifler and B. L. Evans, “Factor analysis of network flow throughput measurements for inferring congestion sharing,” in *Proc. 13th Eur. Signal Processing Conf.*, Antalya, Turkey, Sep. 2005.
- [18] L. Massoulié and J. W. Roberts, “Bandwidth sharing and admission control for elastic traffic,” *Telecommun. Syst.*, vol. 15, pp. 185–201, Jun. 2000.
- [19] A. A. Kherani and A. Kumar, “Stochastic models for throughput analysis of randomly arriving elastic flows in the Internet,” in *Proc. IEEE INFOCOM*, 2002, vol. 2, pp. 1014–1023.
- [20] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, “On the characteristics and origins of Internet flow rates,” in *Proc. ACM Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2002, vol. 32, pp. 309–322.
- [21] D. Chiu and R. Jain, “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks,” *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, 1989.
- [22] S. B. Fred. T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts, “Statistical bandwidth sharing: A study of congestion at flow level,” in *Proc. ACM Conf. Appl., Tech., Arch., and Protocols for Computer Communications*, Aug. 2001, pp. 111–122.
- [23] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, 2nd ed. New York: Springer-Verlag, 2002.
- [24] E. Altman, F. Baccelli, and J. Bolot, “Discrete-time analysis of adaptive rate control mechanisms,” in *High Speed Networks and Their Performance*. Amsterdam: North-Holland, 1994, pp. 121–140.
- [25] E. Altman, T. Basar, and R. Srikant, “Robust rate control for ABR sources,” in *Proc. IEEE INFOCOM*, 1998, vol. 1, pp. 166–173.
- [26] D. Arifler, “Network tomography based on flow level measurements,” Ph.D. dissertation, The University of Texas, Austin, 2004.
- [27] M. E. Crovella and A. Bestavros, “Self-similarity in World Wide Web traffic: Evidence and possible causes,” *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.
- [28] A. B. Downey, “The structural causes of file size distributions,” in *Proc. IEEE Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Aug. 2001, pp. 361–370.
- [29] M. Mitzenmacher, “A brief history of generative models for power law and lognormal distributions,” in *Proc. Allerton Conf. Communications, Control, and Computing*, Oct. 2001, pp. 182–191.
- [30] A. C. Rencher, *Multivariate Statistical Inference and Applications*. New York: Wiley, 1998.
- [31] H. F. Kaiser, “The application of electronic computers to factor analysis,” *Educ. Psychol. Meas.*, vol. 20, pp. 141–151, 1960.
- [32] S. J. Devlin, R. Gnanadesikan, and J. R. Kettenring, “Robust estimation of dispersion matrices and principal components,” *J. Amer. Statist. Assoc.*, vol. 76, no. 374, pp. 354–362, Jun. 1981.
- [33] —, “Robust estimation and outlier detection with correlation coefficients,” *Biometrika*, vol. 62, no. 3, pp. 531–545, 1975.
- [34] D. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. London, U.K.: Chapman & Hall, 1993.
- [35] OPNET Modeler 9.0. OPNET Technologies, Inc. [Online]. Available: <http://www.opnet.com>
- [36] V. Paxson, “End-to-end routing behavior in the Internet,” *IEEE/ACM Trans. Netw.*, vol. 5, no. 5, pp. 601–615, Oct. 1997.

- [37] N. Brownlee and K. Claffy, "Understanding Internet traffic streams: Dragonflies and tortoises," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 110–117, Oct. 2002.
- [38] Z. V. Lambert, A. R. Wildt, and R. M. Durand, "Approximating confidence intervals for factor loadings," *Multivariate Behavioral Res.*, vol. 26, no. 3, pp. 421–434, 1991.



Dogu Arifler (S'95–M'05) received the B.S.E.E., M.S., and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin in 1997, 1999, and 2004, respectively.

He worked as a software engineer at National Instruments in Austin, TX, between 1999 and 2001. Currently, he is an Assistant Professor in the Department of Computer Engineering at the Eastern Mediterranean University, Famagusta, Cyprus. His research interests include network performance analysis and applications of statistical methods to

networking.

Dr. Arifler was a recipient of the Cyprus Fulbright Commission's full scholarship award during 1993–1997. He has been a member of the ACM since 2005.



Gustavo de Veciana (S'88–M'94–SM'01) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California at Berkeley in 1987, 1990, and 1993, respectively.

He is currently a Professor at the Department of Electrical and Computer Engineering at the University of Texas at Austin and the Director of the Wireless Networking and Communications Group (WNCG). His research focuses on the design, analysis and control of telecommunication networks.

His current interests include measurement, modeling and performance evaluation, wireless and sensor networks, and architectures and algorithms to design reliable computing and network systems.

Dr. de Veciana has been an editor for the IEEE/ACM TRANSACTIONS ON NETWORKING. He was a recipient of the General Motors Foundation Centennial Fellowship in Electrical Engineering and a 1996 National Science Foundation CAREER Award, co-recipient of the IEEE William McCalla Best ICCAD Paper Award for 2000, and co-recipient of the Best Paper Award in *ACM Transactions on Design Automation of Electronic Systems*, January 2002–2004.



Brian L. Evans (M'87–SM'97) received the B.S. degree in electrical engineering and computer science from the Rose-Hulman Institute of Technology, Terre Haute, IN, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, in 1988 and 1993, respectively.

From 1993 to 1996, he was a post-doctoral researcher at the University of California, Berkeley. In 1996, he joined the faculty at The University of Texas at Austin, where he holds the Mitchell

Professorship in Electrical and Computer Engineering.

Prof. Evans has published more than 140 peer-reviewed conference and journal papers. He is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and a member of the Design and Implementation of Signal Processing Systems Technical Committee of the IEEE Signal Processing Society. He is the recipient of a 1997 U.S. National Science Foundation CAREER Award.