Accurate and Efficient SLA Compliance Monitoring

Joel Sommers
University of Wisconsin-Madison
jsommers@cs.wisc.edu

Paul Barford
University of Wisconsin-Madison
pb@cs.wisc.edu

Nick Duffield
AT&T Labs-Research
duffield@research.att.com

Amos Ron
University of Wisconsin-Madison
amos@cs.wisc.edu

ABSTRACT

Service level agreements (SLAs) define performance guarantees made by service providers, *e.g*, in terms of packet loss, delay, delay variation, and network availability. In this paper, we describe a new active measurement methodology to accurately monitor whether measured network path characteristics are in compliance with performance targets specified in SLAs. Specifically, (1) we describe a new methodology for estimating packet loss rate that significantly improves accuracy over existing approaches; (2) we introduce a new methodology for measuring mean delay along a path that improves accuracy over existing methodologies, and propose a method for obtaining confidence intervals on quantiles of the empirical delay distribution without making any assumption about the true distribution of delay; (3) we introduce a new methodology for measuring delay variation that is more robust than prior techniques; and (4) we extend existing work in network performance tomography to infer lower bounds on the quantiles of a distribution of performance measures along an unmeasured path given measurements from a subset of paths. We unify active measurements for these metrics in a discrete time-based tool called SLAM. The unified probe stream from SLAM consumes lower overall bandwidth than if individual streams are used to measure path properties. We demonstrate the accuracy and convergence properties of SLAM in a controlled laboratory environment using a range of background traffic scenarios and in one- and two-hop settings, and examine its accuracy improvements over existing standard techniques.

Categories and Subject Descriptors: C.2.3 [Network Operations]: Network management, Network monitoring, C.2.5 [Local and Wide-Area Networks]: Internet (e.g., TCP/IP), C.4 [Performance of Systems]: Measurement Techniques

General Terms: Algorithms, Experimentation, Management, Measurement, Performance

Keywords: Active Measurement, Network Congestion, Network Delay, Network Jitter, Packet Loss, Service-Level Agreements, SLAM

1. INTRODUCTION

Network service level agreements (SLAs) detail the contractual obligations between service providers and their customers. It is increasingly common for SLAs to specify transport-level performance assurances using metrics such as packet loss, delay, delay variation, and network availability [2, 3, 4, 33]. Meeting SLA guarantees results in revenue for the ISP. However, failing to meet SLA guarantees can result in credits to the customer. The implications of not meeting SLA guarantees are therefore serious: a disruption in service can result in significant revenue loss to both the customer and provider. *SLA compliance monitoring*, assessing whether performance characteristics are within specified bounds, is therefore critical to both parties.

Compliance monitoring is a critical challenge for SLA engineering. SLAs must be designed that can be accurately and efficiently monitored, while simultaneously limiting the risk of non-compliance. For example, assuring a low loss rate might be possible only if loss rates can be estimated with sufficiently high confidence. Although passive measurements (*e.g.*, SNMP MIB counters) may provide high accuracy for a metric such as loss on a link-by-link basis, they may be insufficient for estimating the performance experienced by customer traffic. Thus, although there are situations where active measurements may be too heavyweight or may yield inaccurate results [10,31,35], they nonetheless remain a key mechanism for SLA compliance monitoring.

In this paper, we address the following questions: can SLA compliance along a path be accurately monitored with a single lightweight probe stream? and can this stream be the basis for efficient network-wide compliance monitoring? There have been a large number of active measurement methodologies proposed to estimate transport-level performance characteristics. Nonetheless, there has been little work to directly address the specific problem of SLA compliance monitoring. In this context, measurement accuracy, ability to report confidence bounds, ability to quickly adapt to changing network conditions, and ability to efficiently assess performance on a network-wide basis are of great importance.

The first contribution of this paper is the introduction of a new active measurement methodology to accurately assess whether measured network path characteristics are in compliance with specified targets. We describe a heuristic technique for estimating packet loss rate along a path that significantly improves accuracy over existing approaches. Second, we introduce a new method for measuring mean delay along a path that is more accurate than existing methodologies. We also develop a mathematical foundation for obtaining confidence intervals for the quantiles of the empirical delay distribution. Third, we introduce a new method for measuring delay variation that is more robust than prior techniques. These probe algorithms are unified in a *multi-objective* discrete-time based tool called SLAM (SLA Monitor), which was sketched in an earlier workshop paper [36]. That paper was limited to introducing SLAM's architectural framework and outlining the loss rate

SIGCOMM'07, August 27–31, 2007, Kyoto, Japan.

measurement heuristic used by SLAM.

The second contribution of this paper is to extend prior work in the area of performance tomography toward the goal of network-wide SLA compliance monitoring. In particular, we develop a methodology to infer lower bounds on the quantiles of a distribution of path performance measures using measurements from a subset of network paths.

We demonstrate the properties of SLAM in a controlled laboratory environment using a range of background traffic scenarios and using both one- and two-hop topologies. We compare SLAM's delay and loss estimation accuracy with standard IPPM probe methodologies [7, 8] of the same rate, and examine the convergence and robustness of SLAM estimates of delay, delay variation, and loss. Our experiments show that our estimates of mean delay are within one msec of the true mean delay, while the standard probe methodology [7] can suffer inaccuracies up to about a factor of two. We also show that for a confidence level of 90%, SLAM's estimated bounds on a wide range of delay quantiles, with few exceptions, include the true quantile value. We show that in a simple two-hop topology, the inferred bound on the delay distribution is tight, and close to the actual distribution. Our experiments also reveal that SLAM estimates the end-to-end loss rate with high accuracy and with good confidence bounds. For example, in a scenario using self-similar background traffic, the true loss rate over a 15 minute period is 0.08% and the SLAM estimate is 0.07%. In contrast, the standard method for estimating loss rate [8] can have errors of more than two orders of magnitude. We demonstrate the robustness of SLAM's delay variation monitoring methodology, showing how the existing standard RTP jitter metric [32] may be too sensitive to network path conditions, and that SLAM performs well in our more complex two-hop scenario.

2. RELATED WORK

General aspects and structure of SLAs have been discussed in [27, 33]. Performance assurances provided by SLAs range from network path availability, to transport-level metrics, to application-specific metrics. These guarantees may be based on a variety of statistics of the particular metric, such as the mean, median, or a high quantile, computed over various time scales. Examples of the kinds of guarantees offered by service providers are available online [2, 3, 4].

To ensure that SLA performance targets are met with high probability, service providers collect measurements either passively within the network, by injecting measurement probes into the network, or by using a combination of both [6, 13, 18, 42]. While active measurement-based compliance monitoring has received some attention in the past, *e.g.*, [18], there has been little validation in realistic environments where a reliable basis for comparison can be established. There has been limited work addressing the accuracy of some active measurement approaches; exceptions are found in [10, 31, 35]. The issue of accuracy clearly has serious implications for SLA compliance monitoring. Other efforts have been limited in focus to estimation and optimization of a single metric, *e.g.*, [16, 19]. Our work takes an active measurement approach, focusing on simultaneous, or multi-objective, measurement of transport-level performance metrics. We further differentiate our work through validation in a controlled, realistic testbed.

In general, there has been a great deal of work on active measurements of end-to-end delay, delay variation, and loss, *e.g.*, [7, 8, 11, 19, 21, 28, 29, 30, 40, 41]. IETF standardization efforts for active measurement of delay, delay variation, loss, and reordering have taken place within the IETF IPPM working group [7,8,21,30]. Regarding delay, our method for distribution quantile estimation is distinguished from the earlier work of Choi *et al.* [16] in that we do not require the quantile of interest to be specified *a priori*, and that we do not make any assumption regarding the underlying delay distribution. As a result, our method is robust to abrupt changes in underlying network conditions. Lastly, we note that our formulation of a delay variation measurement methodology stands apart from the related IPPM [21] and real-time protocol (RTP) [32] specifications in that rather than considering highly localized variations in delay (*e.g.*, between consecutive probe packets), we consider delay variations over streams of packets.

3. PATH-ORIENTED SLA COMPLIANCE MONITORING

We now describe the basic assumptions and methods for estimating delay, delay variation, and loss along a single end-to-end path. Our objective is to develop accurate, robust estimators based on a discrete-time probe process. Moreover, we seek to improve on the best known standard IPPM methodologies [7, 8, 32]. Another metric that is often part of SLA specifications is network availability. Availability can be loosely defined as the capability of the network to successfully transmit *any* end-to-end probe over an interval of time, *e.g.*, 60 seconds [26]. Although availability may be considered as a special case of loss, we have yet to examine this metric in detail.

3.1 Delay

Both mean delay and high quantiles of the empirical delay distribution are used in SLAs. We first consider estimation of mean delay along a path, which we model as a continuous function $f(t)$ whose independent variable is the time that a probe packet is sent and the dependent variable is measured one-way delay. Based on this model, a natural approach to mean delay estimation is to use Simpson's method for numerical integration. The Simpson's formulation is straightforward: once the domain of integration is partitioned, the integral of the function f over the subinterval I_j is estimated by $\frac{1}{6}(f(a_j)+f(b_j)+4f(c_j))$, with a_j, b_j the endpoints of I_j, and with c_j its midpoint. The error of the Simpson estimate is known to be $e_j = \frac{f^{(4)}(\xi_j)}{2880}|I_j|^5$, with ξ_j some point in the interval I_j. Thus, if the fourth derivative of f exists and is not too large, it is safe to state that the local error is of order 5; *i.e.*, if we double the number of samples, the error in the estimate will be reduced locally by a factor of 32, and globally by a factor of 16.

To apply Simpson's method to a discrete-time probe process for estimating mean end-to-end delay, we do the following: at time slot i, we draw a value k from a geometric distribution with parameter p_{delay}. The geometric distribution is the discrete analog of the exponential distribution and should yield unbiased samples. Probes representing the endpoints a_j and b_j are sent at time slots i and $i+2(k+1)$ with the midpoint probe sent a time slot $i+(k+1)$. At time slot $i+2(k+1)$ the next subinterval begins, thus the last probe of a given subinterval is the first probe of the next one. Simpson's estimates from each subinterval are summed to form the total area under the delay function. The mean delay estimate is then obtained by dividing the integral estimate by the number of subintervals.

With the above formulation, the subintervals are not of equal lengths (the lengths form a geometric distribution). Thus, we can either directly apply Simpson's method to estimate the mean delay, or we can apply relative weights to the subintervals according to their lengths. In our results described below, we use weighted subintervals which we found to give more accurate results, though the absolute differences were small.

There are several considerations in using this approach. First,

probes may be lost in transit. We presently discard subintervals where probe loss occurs. Second, while the assumption that delay largely behaves as a smooth function seems reasonable, it may be more accurate to account for random spikes in delay by modeling the process as the sum of two processes, one smooth and one random. For example, if the function $f(t)$ is written as $f_1(t)+f_2(t)$, with f_1 smooth and f_2 random, then our numerical integration does much better on f_1 and slightly worse on f_2 as compared to straight averaging. The Simpson's approach should be effective for this model as well: if the values of the random part are quite small compared to the smooth part, then our estimate should be better than simple averaging (*i.e.*, the sampling method advocated in RFC 2679 [7]). Note that there is little risk in using Simpson's method: even if delay is a completely random process (which is not likely), the variance of the Simpson's rule estimator for mean delay is increased only slightly as compared to simple averaging.

Distribution-Free Quantile Estimation. Besides using mean delay as the basis of service-level guarantees, ISPs also use high quantiles of the delay distribution, such as the 95th percentile [16].

Let $\{x_i : i = 1,\ldots,n\}$ be n independent samples drawn at random from a common distribution F, sorted in increasing order. For simplicity, assume F is continuous. Let Q_p denote the p^{th} quantile of that distribution, *i.e.*, the unique solution of $F(Q_p) = p$.

We wish to obtain confidence intervals for Q_p based on $\{x_i\}$. One approach would be to start with the empirical distribution function: $\widehat{F}(x) = n^{-1}\#\{i : x_i \le x\}$ and use a quantile estimate of the form $\widehat{Q}_p = \max\{x : \widehat{F}(x) \le p\}$. Analysis of the variance of this estimator might give us asymptotic confidence intervals as n becomes large. Instead, we seek rigorous probabilistic bounds on Q_p that hold for all n.

Now $\{x_k \le x\}$ is the event that at least k of the samples are less than or equal to x, an event which has probability $G(n,F(x),k)$, where $G(n,p,k) = \sum_{j\ge k} p^j(1-p)^{n-j}\binom{n}{j}$. Taking $x = Q_p$ we have $\mathsf{Pr}[x_k \le Q_p] = G(n,p,k)$.

Based on the x_i, we now wish to determine a level $X^+(n,p,\varepsilon)$ that the true quantile Q_p is guaranteed to exceed only with some small probability ε. Thus, we chose $X^+(n,p,\varepsilon) = x_{K^+(n,p,\varepsilon)}$ with $K^+(n,p,\varepsilon) = \min\{k : G(n,p,k) \le \varepsilon\}$.

Similarly, $\mathsf{Pr}[x_k \ge Q_p] = 1 - G(n,p,k)$. Based on the x_i, we now wish to determine a level $X^-(n,p,\varepsilon)$ that the true quantile Q_p is guaranteed to fall below only with some small probability ε. Thus, we chose $X^-(n,p,\varepsilon) = x_{K^-(n,p,\varepsilon)}$ with $K^-(n,p,\varepsilon) = \max\{k : 1 - G(n,p,k) \le \varepsilon\}$.

Put another way, $K^+(n,p,\varepsilon)$ is the $1-\varepsilon$ quantile of the binomial $B_{n,p}$ distribution, while $K^-(n,p,\varepsilon)$ is the ε quantile of the binomial $B_{n,p}$ distribution. The $K^\pm$ can be computed exactly; examples are given in Table 1.

1: Example quantile Indices $K^\pm$ for various sample sizes n, and quantiles p. Confidence level is $1-\varepsilon = 90\%$. Also shown is the reference quantile index $K^0 = np$. — indicates that no upper bound K^+ was available, which can occur when the top atom has mass greater than the desired significance level, *i.e.*, $p^n > \varepsilon$.

	Quantile								
	50			90			99		
n	K^-	K^0	K^+	K^-	K^0	K^+	K^-	K^0	K^+
100	44	50	57	86	90	95	98	99	—
1000	480	500	521	888	900	913	986	990	995
10000	4936	5000	5065	8961	9000	9039	9887	9900	9914

3.2 Delay Variation

Characterizing delay variation in a complex setting and in a compact and robust way is a challenging problem. In looking for a suitable model for delay variation (DV), we found that the notion itself is defined in multiple ways. For example, IPPM RFC 3393 [21] refers on the one hand to the variation of delay with respect to some reference metric, such as the average or minimum observed delay, and on the other hand to the dynamics of queues along a path or at a given router. DV samples in RFC 3393 are defined as the difference in one-way delays of packet i and packet j, $D_i - D_j$. These two packets may be consecutive packets of a probe stream, but they need not be. A statistic of interest identified by the RFC is the empirical distribution of DV samples, the mean of which is sometimes used in SLAs. Maximum DV is also of importance, as it may be useful for sizing playout buffers for streaming multimedia applications such as voice and/or video over IP [24].

An alternative definition of delay variation is found in the Real-time Protocol (RTP) standard, RFC 3550 [32]. It uses an exponentially weighted moving average over the absolute one-way delay differences, $j(i) = j(i-1) + (|D_i - D_{i-1}| - j(i-1))/16$, where D_i is the one-way delay of packet i, and $j(0) = 0$. The RTP jitter value is intended for use as a measure of congestion. Rather than being used as a meaningful absolute value, it is meant to be used as a mechanism for qualitative comparison of multiple RTP stream receivers, or at different points of time at a single receiver. We posit that a DV estimator that can capture dynamic conditions has more direct relevance to applications and is therefore more meaningful to SLAs.

Building on these notions of delay variation, we consider a stream of probes of length k, *e.g.*, 100 probes. We denote the time difference between two probes i and j when they are sent as $s_{i,j}$ and the time difference between the same two probes when they are received as $r_{i,j}$. We construct a matrix M where each cell $M_{i,j}$ contains the ratio $r_{i,j}/s_{i,j}$. Thus, $M_{i,j}$ is 1 if the spacing between probes i and j does not change; is greater than 1 if the measured spacing increases; or is less than 1 if the measured spacing decreases as the probes traverse the network path. (Ratio $r_{i,j}/s_{i,j}$ is defined as 1 for $i = j$ and it is defined as 0 if probe i or j is lost.) Note that computing the above ratio $r_{i,j}/s_{i,j}$ with respect to consecutive probes in the stream gives a more accurate description of the instantaneous nature of DV while probes farther apart give a description of DV over longer time intervals.

Next, we compute the eigenvalues of this matrix M, resulting in a vector e of length k, with values sorted from largest to smallest. If the probe stream traverses the network undisturbed, we would expect that matrix M would consist entirely of 1s, with the largest eigenvalue as k and all other eigenvalues as 0; we denote the vector of these "expected" eigenvalues as e'. We subtract e' from e, taking the L_1 norm of the resulting vector: $\sum_{i=1}^{k} |e_i - e'_i|$. We refer to this L_1 norm as our *DV matrix metric*. As with RTP, it is not intended to be meaningful in an absolute sense but useful for relative comparisons over time.

The DV matrix formulation relies on and is motivated by the fact that we have a notion of what is *expected* in the absence of turbulence along the path, *i.e.*, that probe spacings should remain undisturbed. By looking at the eigenstructure of the DV matrix, we extract, in essence, the *amount of distortion* from what we expect.

3.3 Loss

The loss metric specified by SLAs is *packet loss rate*: the number of lost packets divided by total number of arriving packets over a given time interval. As identified in [35], the difficulty in estimating the end-to-end loss rate is that it is unclear how to measure *demand* along a path (*i.e.*, the denominator used in calculating the loss rate) particularly during congestion periods. Thus, we propose a heuristic approach as outlined in an earlier workshop paper [36].

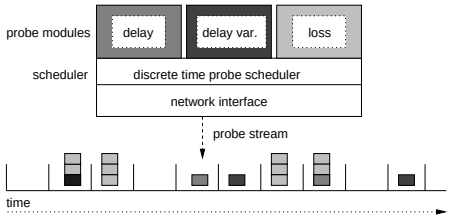
We start with the methodology in [35], which initiates a probe pair at a given time slot with probability p_{loss} for estimation of the end-to-end frequency of congestion episodes $\widehat{F}$ and the mean duration of congestion episodes $\widehat{D}$. In this approach, each probe consists of three packets, sent back-to-back. We measure the loss rate $\widehat{l}$ of the probes *during congestion episodes*. Since the methodology of [35] does not identify individual congestion episodes, we take an empirical approach, treating consecutive probes in which at least one packet is lost as indication of a congestion episode (*i.e.*, similar to [41]). We assume that the end-to-end loss rate L is stationary and ergodic. Given an estimate of the frequency of congestion $\widehat{F}$, we estimate the end-to-end loss rate as $\widehat{L} = \widehat{F}\widehat{l}$.

The key assumption of this heuristic is that we treat the probe stream as a *marker flow*, *viz.*, that the loss rate observed by this flow has a meaningful relationship to other flows along the path. We note again that the probes in [35] consist of multiple packets (3 by default), which has some similarity to a TCP stream where delayed ACKs cause a sender to release two closely-spaced packets. While we do not claim that the probe stream is, in general, the same as a TCP stream, our results below demonstrate that such an assumption may be reasonable in this context.

3.4 Multi-Objective Probing

We use the term *multi-objective* probing to refer to simultaneous estimation of multiple performance metrics using a single probe stream. The individual discrete-time algorithms described above operating at the same time may schedule probes to be sent at the same time slot. Such requests can be accommodated by tagging probes according to the relevant estimator. Thus, a single probe stream can be used for concurrent estimation of packet loss, delay, delay variation, and other quantities, thereby reducing the impact of measurement traffic on the network.

The basic architecture of our multi-objective probe scheduler is depicted in Figure 1. The main component of the architecture is a discrete-time scheduler that provides callback and probe scheduling mechanisms. Probe modules implement the various path-oriented estimation methods described above. This design allows for logical separation among multiple, simultaneously operating measurement methods and for optimizations of network bandwidth.



1: Multi-objective probe scheduler architecture. Algorithmic modules interact with a discrete-time probe scheduler to perform estimation of delay, delay variation, and loss characteristics.

4. TOWARD NETWORK-WIDE SLA COMPLIANCE MONITORING

The previous section described a set of methodologies for efficient per-path monitoring. SLA compliance monitoring, however, requires accurate and efficient measurement on a network-wide basis. However, the measurement overhead of sending probes over a full n^2 mesh of paths is highly undesirable. In this section, we describe the mathematical foundation that enables economical monitoring over a subset of network paths. This new methodology enables greater flexibility for specifying performance assurances in terms of quantiles of a distribution, while attaining a high level of measurement efficiency.

4.1 Routing Matrices, Measurement, and Linear Dependence

Let $G = (V,E)$ be a directed graph comprising vertices (nodes) V and directed edges (links) $(v_1,v_2) \in E \subset V \times V$. Let R be a set of paths (routes) *i.e.*, each $r \in R$ is an ordered set of $n > 0$ contiguous links $(v_0,v_1),(v_1,v_2),\ldots,(v_{n-1},v_n)$. The *routing matrix* A associated with R is the incidence matrix of the links in the routes, namely, $A_{re} = 1$ if link e occurs in route r and zero otherwise.

We now describe what we term the *scalar additive network performance model*. Let $x : E \to \mathbb{R}$ be a function on the links. This naturally gives rise to the path function $y : R \to \mathbb{R}$ defined as $y_r = \sum_{e \in r} x_e = \sum_{e \in E} A_{re} x_e$. This relation is a prototype for additive network performance models. Two examples are:

Network Delay: The latency of packet traversing the path r is the sum of the latencies incurred on each link of the path. This may be understood either as the x_e being individual measurements, or as x_e being mean latencies. This is the example on which we focus in this paper.

Network Loss: In this model, x_e is the log transmission probability of traversing link e; if there is no spatial correlation between link losses we can write y_r as the log transmission probability along the path r.

Performance Tomography.

Two classes of inference problems arising from the framework above have been studied recently. In *link performance tomography* the aim is to infer the distribution of the link variable x_e given only path measurements y_r. Variants of this problem have been studied, mostly depending on exploiting correlations between measurement on different paths, *e.g.*, either at the packet level, *e.g.*, by using multicast probes [12, 25] or groups of unicast probes [23, 39], or more generally of distinct packet streams that experience common performance impairments [9, 22].

A second class of problem has more recently attracted attention [14, 15, 17]: given a set of path performance measures across intersecting paths, is it possible to infer the whole set of measures if only a subset is known? Clearly there is some relation between the two problems in the sense that if all link performance measures could be inferred from a subset of path measures, then the remaining path measures could be determined simply.

For scalar additive performance measures, the second problem has a simple expression in terms of the routing matrix A. Suppose that the matrix A is not of full (row) rank, *i.e.*, the set of row vectors is not linearly independent. Consequently there exists a minimal set of paths $S \subsetneq R$ which span in the sense that such that every row of $a_r = \{A_{re} : e \in E\}$ of A can be expressed as a linear combination of the $\{a_r : r \in S\}$. For the scalar additive performance model, this translates to saying that all $\{y_r : r \in R\}$ can be determined from the subset $\{y_r : r \in S\}$. Recent work on this problem has focused on understanding how the dimension of the set S depends on network topology. Chen *et al.* [15] concluded that the number of paths in S grows as $O(\#V)$ (*i.e.*, linear in the number of network nodes $\#V$) in a real router-level topology, or at worst like $O(\#v \log \#V)$ in some simulated topologies.

Distributional Path Performance Measures.

In this work we extend the computational approach described above to infer distributions of a set of path performance measures

from a subset. We assume in a given network setting the existence of the set $S \subsetneq R$ with the properties detailed above has been established. This means in particular that for every network path in R, every link in this path is traversed by some path in the subset R, and below we show how the distributions of delay in path in R can be inferred from only those in S. This inference depends on the assumption that any packet traversing a given link will experience the same delay distribution, even if the actual delays differ. The proofs of the results are relatively straightforward but have been omitted due to space limitations and will appear in a future technical report.

There are two challenges in trying to extend the scalar approach to distributions. The first is dependence among link measurements. Dependence is not an issue in the linear algebra of mean quantities since the average of a linear combination of random variables is equal to same linear combination of respective averages even when the variables are dependent. Working with distributions is more complex, for example the distribution of a sum of random variables is not equal to the convolution of their distributions unless the random variables are independent. A second complexity is algebraic: there is no simple subtraction operation for distributions. For example, if X and Y are independent random variables and $X = Y$ in distribution, it is not the case that $X - Y$ is identically zero.

4.2 Delay Distributional Inference

We suppose routing (and hence the matrix A) is static over a measurement interval. On each path r a stream of measurement packets labeled $i = 1,2,\ldots,n_r$ is launched along the path. Packet i incurs a latency X_{re}^i on traversing the link $e \in r$. The latency of the packet on the path is $Y_r^i = \sum_{e \in r} X_{re}^i$.

To motivate the following, consider the star topology network in Figure 3b in which source nodes v_1, v_2 and destination nodes v_3, v_4 are linked through a central node v_c. Denote the edges by $e_1 = (v_1, v_c)$, $e_2 = (v_2, v_c)$, $e_3 = (v_c, v_3)$ and $e_4 = (v_c, v_4)$. We consider the 4 paths $r_1 = (e_1, e_3)$, $r_2 = (e_1, e_4)$, $r_3 = (e_2, e_3)$ and $r_4 = (e_2, e_4)$. Let X_n be the delay on link e_n, and Y_n the delay on path r_n. Clearly, $Y_1 + Y_4 =^d Y_2 + Y_3$. Assume that the distributions of Y_2, Y_3 and Y_4 are known; we focus on inferring that of Y_1.

Our major statistical assumption is that all X_{re}^i are independent. We remark that the opposite type of assumption, *i.e.*, the identity of certain link variables, has been employed for multicast performance tomography (and some unicast variants) to describe the propagation of multicast packets. The identity assumption is natural in that case, since it reflects either the delay encountered by a single multicast packet or a train of closely spaced unicast packets prior to branching to distinct endpoints.

In the present case, we can consider two types of dependence. In the first case we consider dependence between different measurements. Provided probe packets are dispatched at intervals longer than the duration of a network congestion event, then probes on the same path or on intersecting paths are unlikely to exhibit delay dependence, even if individual packets experience the *distribution* of congestion events similarly on the same link. Thus, is seems reasonable to model the Y_{re}^i as independent. The second case to consider is dependence among the individual link delays X_{re}^i on a given path r. Violation of this property might occur in packet streams traversing a set of links congested by the same background traffic. As far as we are aware, there are no live network or testbed studies that have investigated this property. Dependence was found in a network simulation study, but was pronounced only in a small network configuration with few traffic streams [25]. For this reason we believe that link delay correlation need not be significant in a large network with a diverse traffic.

For $r \in R$ let $\{b_{rr'} : r' \in S\}$ be the coefficients of the spanning set $\{a_{r'} : r' \in S\}$ in the expression of a_r, *i.e.*,

$$a_r = \sum_{r' \in S} b_{rr'} a_{r'} \tag{1}$$

Let $S_r^+ = \{r' \in S : b_{rr'} > 0\}$ and $S_r^- = \{r' \in S : b_{rr'} < 0\}$.

LEMMA 1. *Assume $\{a_{r'} : r' \in S \subsetneq R\}$ is a minimal spanning set. For each $r \in R$ there exist positive integers d_r and $\{d_{rr'} : r' \in S\}$ such that*

$$d_r a_r + \sum_{r' \in S_r^-} d_{rr'} a_{r'} = \sum_{r' \in S_r^+} d_{rr'} a_{r'} \tag{2}$$

For each $r \in R, e \in E$ let $X_{re}^{(i)}, i = 1,2,\ldots$ denote the sum of i independent copies of a single delay on link e, *e.g.*, X_{re}^1; likewise let $Y_r^{(i)}$ denote the sum of i independent copies of Y_r^i. The symbol $=^d$ will denote equality in distribution.

THEOREM 1.

$$Y_r^{(d_r)} + \sum_{r' \in S_r^-} Y_{r'}^{(d_{rr'})} =^d \sum_{r' \in S_r^+} Y_{r'}^{(d_{rr'})} \tag{3}$$

One can already see in Theorem 1 a basic feature of our results that follows merely from the partition of S into S_r^- and S_r^+. Suppose we are primarily interested in determining whether Y_r often takes some large value. Suppose measurements tell us that some of the $\{Y_{r'} : r' \in S_r^+\}$ tend to take large values, but that none of the $\{Y_{r'} : r' \in S_r^-\}$ do. Then we know from the equality (3) that Y_r must also tend to take large values. If none of the $\{Y_{r'} : r' \in S\}$ tend to take large values, then neither does Y_r. But when some $Y_{r'}$ for r' in both S_r^+ and S_r^- tend to take large values, then it is difficult to draw conclusions about Y_r. These observations prefigure our later results on distributional bounds for Y_r.

Distributions and Inversion.

Let $\mathscr{Y}_r$ denote the common distribution of the Y_r^i, and $\widetilde{\mathscr{Y}_r}$ its Laplace transform, *i.e.*, $\widetilde{\mathscr{Y}_r}(s) = \int_0^\infty \mathscr{Y}_r(dy) e^{-sy}$. Let $*$ denote convolution. In terms of distributions, (3) becomes

$$\mathscr{Y}_r^{*d_r} \underset{r' \in S_r^-}{*} \mathscr{Y}_{r'}^{*d_{rr'}} = \underset{r' \in S_r^+}{*} \mathscr{Y}_{r'}^{*d_{rr'}} \tag{4}$$

To what extent can we solve these convolution equations? In Laplace transform space we obtain from (4):

$$\widetilde{\mathscr{Y}_r}^{d_r} \prod_{r' \in S_r^-} \widetilde{\mathscr{Y}_{r'}}^{d_{rr'}} = \prod_{r' \in S_r^+} \widetilde{\mathscr{Y}_{r'}}^{d_{rr'}} \tag{5}$$

Given empirical estimates of $\{\mathscr{Y}_{r'} : r' \in S\}$ one can in principle use numerical Laplace transform inversion to recover all $\mathscr{Y}_r$. This is an approach we intend to pursue in a subsequent work. In this paper, we use (4) directly in order to obtain bounds on the distributions $\mathscr{Y}_r$.

Convolution Bounds.

Let V_i, $i = 1,2,\ldots,n$ be independent random variables and set $V = \sum_{i=1}^n V_i$ be their sum. Let $Q_p(V_i)$ denote the p-quantile of V_i, *i.e.*,

$$\mathsf{Pr}[V \le x] \ge p \Leftrightarrow Q_p(V) \le x \tag{6}$$

The following result formalizes the perhaps obvious statement that if you know that $V_1 \le x$ a fraction p of the time, and $V_2 \le y$ a fraction q of the time, then you can conclude that $V_1 + V_2$ is less than $x + y$ no less than a fraction pq of the time.

THEOREM 2. *Let $V_i, i = 1,2,\ldots,n$ be independent random variables with sum $V = \sum_{i=1}^{n} V_i$, and let $p_i \in (0,1]$ with $p = \prod_{i=1}^{n} p_i$.*

$$Q_p(V) \leq \sum_{i=1}^{n} Q_{p_i}(V_i) \tag{7}$$

Network Quantile Bounds.

THEOREM 3. *Denote $Y_r^{\pm} = \sum_{r' \in S_r^{+}} Y_{r'}^{(d_{rr'})}$*

(i) $Q_p(Y_r) \geq (d_r)^{-1} Q_{p^{d_r}}(Y_r^{(d_r)})$.

(ii) $Q_p(Y_r^{(d_r)}) \geq Q_{pq}(Y_r^{+}) - Q_q(Y_r^{-})$

(iii) $Q_p(Y_r) \geq (d_r)^{-1} \sup_{q \in (0,1]} (Q_{p^{d_r} q}(Y_r^{+}) - Q_q(Y_r^{-}))$

Theorem 3 provides a lower bound on the quantiles, or, equivalently, an upper bound on the cumulative distribution. Thus, it underestimates the frequency with which a given level is exceeded. This may or may not be desirable if the measured quantiles are to be used for detecting SLA violations (*i.e.*, raising alarms). On the one hand false positives will be reduced, while at the same time some high quantiles may be underestimated. Following a network example below, we describe how knowledge of the topology of measured paths may be used to adjust alarm thresholds in order to mitigate the effects of quantile underestimation.

Computation of Quantiles.

We use the measured end-to-end latencies on the paths $r \in S$, the $\Omega_r = \{Y_r^i : i = 1,2,\ldots,n_r\}$, to estimate the required quantiles on the RHS of Theorem 3(iii). To compute the distribution of $Y_r^{\pm}$ we might construct the sets of values $\{\sum_{r' \in S_r^{\pm}} \sum_{i=1}^{d_{rr'}} y_{rr'} : y_{rr'} \in \Omega_r\}$. However, this gives rise to $n_r^{\pm} = \prod_{r' \in S_r^{\pm}} n_{r'}^{d_{rr'}}$ member of each set, which may require prohibitively large amounts of memory. Instead, memory can be controlled by discretizing the distributions before convolution.

Discrete Mass Distributions and Their Convolution.

A positive discrete mass distribution is specified by a tuple $(\varepsilon, n, m = \{m_i : i = 0,\ldots,n\})$ where ε is the bin width, with a mass m_i in bin $[i\varepsilon,(i+1)\varepsilon)$ for $i = 0,1,\ldots,n-1$, and mass m_n in $[n\varepsilon,\infty)$. Two such distributions (ε,n,m) and (ε',n',m') the have convolution

$$(\varepsilon,n,m) * (\varepsilon',n',m') = (\varepsilon+\varepsilon', 1+(n-1)(n'-1), m'') \tag{8}$$

where $m_j'' = \sum_{i=0}^{j} m_i m_{j-i}'$. Given ε, n, an set of measurements $\{Y_r^i : i = 1,2,\ldots,n_r\}$ gives rise to a empirical discrete mass distribution (ε,n,m) with $m_i = \#\{Y_r^i : Y_r^i \in [i\varepsilon,(i+1)\varepsilon)\}$ for $i = 0,1,\ldots,n-1$ and $m_n = \#\{Y_r^i : Y_r^i \geq n\varepsilon\}$. The distribution of each $\{\sum_{r' \in S_r^{\pm}} \sum_{i=1}^{d_{rr'}} y_{rr'} : y_{rr'} \in \Omega_r\}$ is then estimated by taking the grand convolution over $r' \in S_r^{\pm}$ of the $d_{rr'}$-fold convolutions of the empirical mass distribution generated from each $\#\{Y_{r'}^i : Y_{r'}^i \in [i\varepsilon,(i+1)\varepsilon)\}$. A target resolution ε in the final distribution is achieved by choosing resolutions ε' for the constituent distribution that sum to ε, for example, $\varepsilon' = \varepsilon / \sum_{r' \in S_r^{\pm}} d_{rr'}$. Finally, we normalize to a probability distribution by dividing each mass element by $n_r^{\pm}$. We call the resulting variables $\widehat{Y}_r^{\pm}$, and use them in place of the $Y_r^{\pm}$ in Theorem 3.

Network Example.

In the above formalism, we have $S_1^{+} = \{2,3\}$, $S_1^{-} = \{4\}$ with $d_{12} = d_{13} = d_{14} = 1$ and $Y_1^{+} = Y_2 + Y_3$ and $Y_1^{-} = Y_4$. Suppose now that X_i are exponentially distributed with distinct means μ_i. Then Y_1^{+} has a mixed exponential distribution with PDF

$$y_1^{+}(x) = \sum_{i=1}^{4} \frac{e^{-x/\mu_i} \mu_i^2}{\prod_{j \in \{1,2,3,4\}, j \neq i} (\mu_i - \mu_j)} \tag{9}$$

while Y_1^{-} has a mixed exponential distribution with PDF

$$y_1^{-}(x) = \frac{e^{-x/\mu_2} - e^{-x/\mu_4}}{\mu_2 - \mu_4} \tag{10}$$

For the optimization of Theorem 3, elementary calculus shows that when $Y_r^{\pm}$ have densities $y_r^{\pm}$, the stationary points of $q \mapsto Q_{p^{d_r} q}(Y_r^{+}) - Q_q(Y_r^{-}))$ obey

$$y_r^{+}(Q_{p^{d_r} q}(Y_r^{+})) = p^{d_r} y_r^{-}(Q_{p^{d_r} q}(Y_r^{-})) \tag{11}$$

We use the above expression to compute the bounds and consider four cases. For cases (a)–(c) we plot the actual CDF on the unmeasured path, together with the CDF bound in Figure 2.

(a) *Homogeneous Delay.* $m_1 = 1.0$, $m_2 = 1.1$, $m_3 = 1.2$, $m_4 = 1.3$. The delay on path r_1 is somewhat underestimated, but then large delays only very rarely occur.

(b) *High Delay on Unmeasured Path, Low Delay Elsewhere.* $m_1 = 10$, $m_2 = 1.1$, $m_3 = 1.2$, $m_4 = 1.3$. The low delays on links not included in the unmeasured path allow fairly close estimation of the delay distribution on r_1.

(c) *High Delay on Unmeasured Path, Some High Delay Elsewhere.* $m_1 = 10$, $m_2 = 11$, $m_3 = 1.2$, $m_4 = 1.3$. Although elevation of delay on r_1 is detected, the amount is somewhat underestimated due to the presence of high delay on one of the measured paths; this parallels the remarks following Theorem 1.

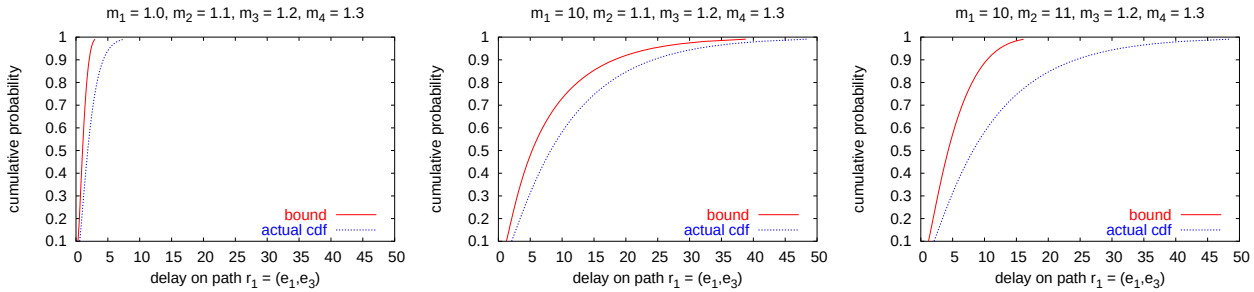
(d) *Low Delay on Unmeasured Path, Some High Delay Elsewhere.* $m_1 = 1.0$, $m_2 = 11$, $m_3 = 1.2$, $m_4 = 1.3$. The results are similar to the homogeneous case; the presence of high delay elsewhere in the network does not further perturb the delay bound.

If this delay bound estimates are to be used for raising alarms based on crossing threshold levels, it may be desirable to adjust alarm thresholds based on the spatial distribution of measured path delays. Specifically, case (c) above illustrates that when higher delays are encountered on a path in S_r^{-}, a lower alarm threshold may be used in order to compensate for the partial "obscuring" of the delay on the unmeasured path. In situations exemplified by cases (a) and (b), no adjustment to the threshold is needed, since there are no measured paths with high delay (so in particular, none in S_r^{-}).

5. EXPERIMENTAL TESTBED

We implemented a tool to perform multi-objective probing, called SLAM (SLA MONITOR). SLAM sends UDP packets in a one-way manner between a sender and receiver. It consists of about 2,000 lines of C++, including code to implement the loss, delay, and delay variation probe modules. The implementation is extensible and can accommodate other discrete-time probe algorithms. In this section, we describe the controlled laboratory environment in which we evaluated SLAM. We considered two topologies, shown in Figure 3. Each setup consisted of commodity workstation end hosts and commercial IP routers.

The first topology (Figure 3a) was set up in a dumbbell-like configuration. We used 10 workstations on each side of the bottleneck



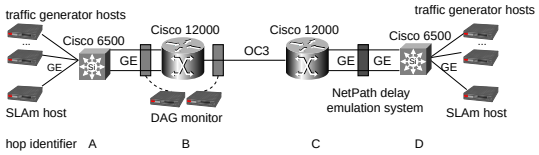
2: Example bounds on the inferred delay distribution. (a) Left: homogeneous delay; (b) Center: high delay on unmeasured path; (c) Right: high delay on unmeasured path and some others.

OC3 for producing background traffic and one workstation at each side to run SLAM. Background traffic and probe traffic flowed over separate paths through a Cisco 6500 enterprise router (hop A) and was multiplexed onto a bottleneck OC3 (155 Mb/s) link at a Cisco GSR 12000 (hop B). Packets exited the OC3 via another Cisco GSR 12000 (hop C) and passed to receiving hosts via a Cisco 6500 (hop D). NetPath [5] was used between hops C and D to emulate propagation delays for the background traffic hosts in the testbed. We used a uniform distribution of delays with a mean of 50 msec, minimum of 20 msec, and maximum of 80 msec. The bottleneck output queue at the Cisco GSR at hop B was configured to perform tail drop with a maximum of about 50 msec of buffer space.

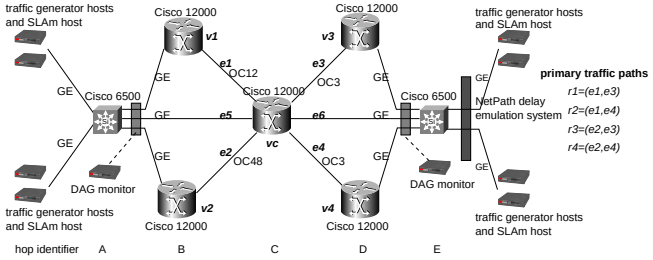
The second topology (Figure 3b) was set up in a star-like configuration. We used 12 hosts on each side of the setup (6 at top, 6 at bottom) to generate traffic over links e_1 (OC12–622 Mb/s), e_2 (OC48–2.488 Gb/s), e_3 (OC3) , and e_4 (OC3) making up the star. An additional host configured at each corner ran SLAM. Aggregation routers (Cisco 6500's at hops A and E) were configured to direct traffic over four primary configured paths, r_1–r_4, as shown in the figure. In addition, traffic flowed over path (e_1, e_2) to create sufficient load on e_1 to include queueing delay and loss. SLAM probes flowed over the four primary traffic paths to monitor delay, loss, and delay variation. SLAM was also configured to monitor paths (e_1, e_6), (e_2, e_6), (e_5, e_3), and (e_5, e_4). Only probe traffic traversed links e_5 and e_6, thus it was assumed that these additional probe measurements were sufficient to separately measure characteristics on links e_1, e_2, e_3, and e_4. As with the dumbbell topology, NetPath [5] was used to emulate propagation delays for the background traffic hosts in the testbed. We used a uniform distribution of delays with a mean of 50 msec, minimum of 20 msec, and maximum of 80 msec. Each queue was configured to perform tail drop. Using the notation $(r, e) = B$ to denote the output queue at router r on to link e in msec, buffer size configurations were follows: $(v_1, e_1) \approx 25$ msec, $(v_2, e_2) \approx 12.5$ msec, $(v_c, e_3) \approx 50$ msec, and $(v_c, e_4) \approx 100$ msec.

Each workstation used in our experiments had a Pentium 4 processor running at 2GHz or better, with at least 1 GB RAM and an Intel Pro/1000 network interface card and was configured to run either FreeBSD 5.4 or Linux 2.6. The SLAM hosts were configured with a default installation of FreeBSD 5.4. The SLAM workstations used a Stratum 0 NTP server configured with a TrueTime GPS card for synchronization. We used the software developed by Corell *et al.* [20] to provide accurate timestamps for SLAM. All management traffic for the two topological configurations flowed over separate network paths (not pictured in either figure).

A critical aspect of our laboratory environment is the ability to measure a reliable basis for comparison for our experiments. For the dumbbell topology, optical splitters were attached to the links



(a) Dumbbell topology. Probes and cross traffic are multiplexed onto a bottleneck OC3 (155Mb/s) link where queueing delay and loss occurs.



(b) Star topology. Probes and cross traffic follow paths r_1, r_2, r_3, and r_4, shown in the figure.

3: Laboratory testbeds.

between hops A and B and to the link between hops B and C and synchronized Endace DAG 4.3 (Gigabit Ethernet) and 3.8 (OC3) passive monitoring cards were used to capture packet traces entering and leaving the bottleneck node. For the star topology, optical splitters were attached to the Gigabit ethernet links entering the core star topology (just after hop A), and exiting the star (just before hop E). We used synchronized DAG 4.3 passive monitoring cards to capture packet traces entering and leaving the star setup. By comparing packet header information, we were able to identify which packets were lost along each path. Furthermore, these cards provide synchronization of better than one microsecond allowing precise delay measurement through the bottleneck router.

We used four background traffic scenarios for experiments using the dumbbell setup. For the first scenario, we used Iperf [38] to produce constant-bit rate (CBR) UDP traffic for creating a series of approximately constant duration (about 65 msec) loss episodes, spaced randomly at exponential intervals with a mean of 10 seconds over a 10 minute period. We found that short loss episodes were difficult to consistently produce with Iperf, thus the duration we used was a compromise between a desire for short episodes and the ability to predictably produce them. The second scenario consisted of 100 long-lived TCP sources run over a 10 minute period. For the final two scenarios, we used Harpoon [34] with a heavy-tailed file size distribution to create self-similar traffic approximating a mix of web-like and peer-to-peer traffic commonly seen in

today's networks. We used two different offered loads of 60% and 75% of the bottleneck OC3. Since good performance cannot be guaranteed when resources are oversubscribed, SLAs often contain clauses to allow discarding performance measurements if utilization exceeds a given threshold [33]. Thus, we chose these offered loads to reflect relatively high, yet acceptable average loads in light of this practice. Experiments using the self-similar traffic scenario were run for 15 minutes. For all scenarios, we discarded the first and last 30 seconds of the traces.

For the star setup, we used three background traffic scenarios in our experiments. For the first scenario, we used Iperf [38] to produce CBR UDP traffic over the four primary traffic paths to create a series of approximately constant duration loss episodes at (v_c, e_3) and (v_c, e_4). We used an additional Iperf flow over path (e_1, e_2) to produce a series of loss episodes at (v_1, e_1). All loss episodes were spaced at exponential intervals with a mean of 10 seconds, and the test duration was 10 minutes. The second scenario consisted of long-lived TCP sources configured to use all four primary traffic paths plus path (e_1, e_2). There were at least 100 traffic sources configured to use each path, and the test duration was 10 minutes. In the third scenario, we used Harpoon [34] with a heavy-tailed file size distribution to create self-similar traffic as in scenarios three and four for the dumbbell topology. Traffic sources were configured to produce approximate average loads of 65% on link e_1, 15% on link e_2, 75% on link e_3, and 60% on link e_4, and the test duration was 15 minutes. For all scenarios, we discarded the first and last 30 seconds of the traces. Finally, we note that while maximum queueing delays at (v_2, e_2) were non-zero for all three traffic scenarios, no loss occurred at (v_2, e_2).

6. EVALUATION

We now describe the experimental evaluation of SLAM using the testbed described above. We examine the accuracy of SLAM's delay and loss estimates, comparing its results with estimates obtained using standard IPPM methodologies [7, 8], which are based on Poisson-modulated probes. We also compare the DV matrix metric with other standard methodologies [21, 32].

6.1 SLAM Measurement Overhead

Two important implementation decisions were made in the SLAM probe sender. First, the scheduler must accommodate estimation techniques that use multi-packet probes, such as the loss rate estimation method we use. Second, the scheduler must arbitrate among probe modules that may use different packet sizes. At present, the smallest packet size scheduled to be sent at a given time slot is used.

An effect of the implementation decision for probe packet sizes is that the overall bandwidth requirement for the multi-objective stream is less than the aggregate bandwidth requirement for individual probe modules if used separately. One concern with this implementation decision is the issue of packet size dependence in the measurement technique. For delay and delay variation, packet sizes should be small to keep bandwidth requirements low. For delay variation, the packet size should closely match that used by a codec referred to in the G.107 and related standards so that the E-model formulas can be directly used [1]. We use 48 bytes at an interval of 30 msec in our evaluation below, which approximates the G.723.1 codec. For delay, another concern is the relative difference between end-to-end transmission and propagation delays. In situations where propagation delay is large relative to transmission delay, the packet size can be small since the transmission delays along a path contribute little to the overall delay. In cases where the opposite situation holds, packet sizes should be large enough to estimate delays experienced by packets of average size. In our evaluation described below, we use 100 byte packets for delay estimation. For loss estimation packet sizes, the key consideration is that multi-packet probes should admit accurate instantaneous indications of congestion. In previous work [35], a packet size of 600 bytes was used and was found to be a reasonable balance between limiting measurement impact while still obtaining accurate congestion indications. We verified this previous finding and leave a detailed analysis for future work.

In the experiments below, we fix SLAM probe parameters as shown in Table 2. In prior work, $p_{loss} = 0.3$ was found to give good loss characteristic estimates [35]. We verified the results regarding the setting of the parameter p_{loss} but omit detailed results in this paper. We experimented with a range of values for p_{delay} from 0.01 to 0.5 (mean probe intervals from 5 msec to about 500 msec) and found that estimation accuracy for SLAM is virtually unchanged over the range of parameter settings except those below about 0.02 (above about 200 msec mean probe spacing). We do not include detailed results in this paper due to space limitations. For delay variation, we used a packet size of 48 bytes sent at periodic intervals of 30 msec. We used a stream length k of 100 probes in computing the DV matrix metric.

2: SLAM parameters used in evaluation experiments. For all experiments, we set the discrete time interval for the scheduler to be 5 msec.

Loss		Delay		Delay Variation	
Packet size	p_{loss}	Packet size	p_{delay}	Packet size	Interval
600 bytes	0.3	100 bytes	0.048	48 bytes	30 msec

With the parameters of Table 2, the bandwidth savings due to multi-objective probing is about 100 Kb/s. Separately, the loss probe stream is about 490 Kb/s, the delay probe stream is about 20 Kb/s, and the delay variation is about 60 Kb/s: a sum of about 570 Kb/s. With SLAM, the probe stream is actually about 470 Kb/s. Note that for the dumbbell topology, the SLAM parameters used in our experiments result in only about 0.3% of the bottleneck OC3 consumed for measurement traffic. For the star topology, three SLAM streams traverse links e_3 and e_4 (namely, for link e_3, paths r_1, r_3 and (e_5, e_3) are monitored, resulting in three streams traversing e_3). The measurement traffic consumption on these OC3 links is still less than 1% of the capacity.

6.2 Delay

Table 3 compares the true delay measured using the DAG-collected passive traces with the mean delay estimate produced by SLAM and the estimates produced using standard RFC 2679 [7] (Poisson-modulated probes), sent at the same rate. Values are shown for each traffic scenario and are averages over full experiment duration. Note that the differences in true values are due to inherent variability in traffic sources, but the results are representative of tests run with different random seeds. First, we see in Table 3a that the SLAM results are close to the true values. We also see that while results for the standard stream are close for the CBR and long-lived TCP traffic scenarios, they are less accurate for the more realistic self-similar traffic scenarios, with with relative errors ranging from about 25% to 120%. Second, we see that in Table 3b that the SLAM results are close to the true values, though somewhat less accurate than for the simple dumbbell topology. The accuracy of the mean delay estimate for the RFC 2679 stream varies over the range of traffic scenarios and paths, but is generally better than in the dumbbell topology. A possible explanation for this behavior is that the increased level of aggregation of traffic sources in the star topology leads to an improvement in mean delay estimates.

Figure 4 shows true mean delay and the SLAM-estimated mean

3: Comparison of mean delay estimation accuracy for SLAM and RFC 2679 (Poisson) streams using the (a) dumbbell and (b) star testbed topologies. Values are in seconds and are averages over the full experiment duration.

(a) Delay accuracy using the dumbbell topology.

Probe stream → **Traffic scenario ↓**	SLAM		RFC 2679 (Poisson)	
	true	estimate	true	estimate
CBR	0.0018	0.0018	0.0018	0.0022
Long-lived TCP	0.0387	0.0386	0.0386	0.0391
Harpoon self-similar (60% load)	0.0058	0.0059	0.0071	0.0092
Harpoon self-similar (75% load)	0.0135	0.0135	0.0060	0.0132

(b) Delay accuracy using the star topology.

Probe stream → **Traffic scenario (route) ↓**	SLAM		RFC 2679 (Poisson)	
	true	estimate	true	estimate
CBR (r_1)	0.0066	0.0064	0.0066	0.0047
CBR (r_2)	0.0087	0.0075	0.0087	0.0056
CBR (r_3)	0.0053	0.0048	0.0053	0.0036
CBR (r_4)	0.0073	0.0063	0.0073	0.0043
Long-lived TCP (r_1)	0.0598	0.0601	0.0598	0.0612
Long-lived TCP (r_2)	0.1168	0.1172	0.1162	0.1189
Long-lived TCP (r_3)	0.0362	0.0364	0.0362	0.0364
Long-lived TCP (r_4)	0.0936	0.0936	0.0936	0.0935
Harpoon self-similar (r_1)	0.0508	0.0503	0.0542	0.0505
Harpoon self-similar (r_2)	0.0108	0.0112	0.0123	0.0112
Harpoon self-similar (r_3)	0.0414	0.0417	0.0446	0.0428
Harpoon self-similar (r_4)	0.0019	0.0027	0.0028	0.0024

delay over the duration of experiments using CBR traffic (top) in the dumbbell topology, and for self-similar traffic on route r_1 in the star topology. Results for other experiments are consistent with plots shown in Figure 4. True delay estimates are shown for 10 second intervals and estimates for SLAM are shown for 30 second intervals. We see that in each case after an initial convergence period, the SLAM estimate tracks the true delay quite well.

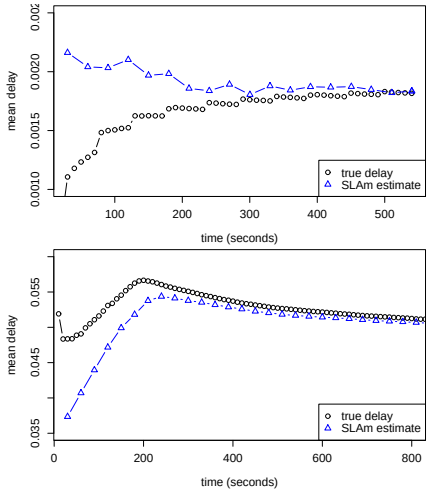
Distribution-Free Quantile Estimation. Figure 5 compares the true delay distribution with the SLAM-estimated delay distribution with 90% confidence bounds. Representative plots are shown for the long-lived TCP traffic scenario in the dumbbell topology (Figure 5a) and for the CBR UDP traffic scenario in the star topology (Figure 5b). We see that for these vastly different traffic and topological setups that the delay distribution is estimated quite well and that with few exceptions, the confidence bounds include the true delay distribution for the range of estimated quantiles shown.

Delay Distribution Inference. We now examine the problem of inferring the delay distribution along a path given measured delay distributions along a subset of paths. Specifically, given measurements along paths r_2, r_3, and r_4, we wish to infer the delay distribution for path r_1.

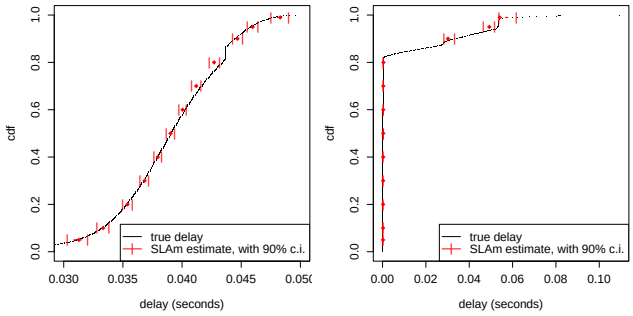
Figure 6 shows representative results for two traffic scenarios considered using the star topology. For these results, we used a bin width ε of 100 μsec for the input discrete mass distributions. The computed bound and the actual CDF measured using SLAM are shown for the CBR UDP traffic (top) and self-similar TCP traffic (bottom). We see that for each traffic scenario the computed bound is relatively tight, with the closest qualitative match for the more realistic self-similar traffic scenario. The skewed distribution arising from the CBR UDP traffic scenario results in an underestimation of the high delay values along path r_1. For the self-similar TCP traffic scenarios, the delay distributions are somewhat smoother (though not homogeneous along paths in the star topology), and the resulting bounds are tighter.

6.3 Delay Variation

Evaluation of measured delay variation is complicated by the fact



4: Comparison of true mean delay with SLAM estimates over time. True mean delays are plotted using 10 second intervals. SLAM estimates are plotted using 30 second intervals. Plots shown for CBR traffic in the dumbbell topology (top), and self-similar traffic on route r_1 in the star topology (bottom).



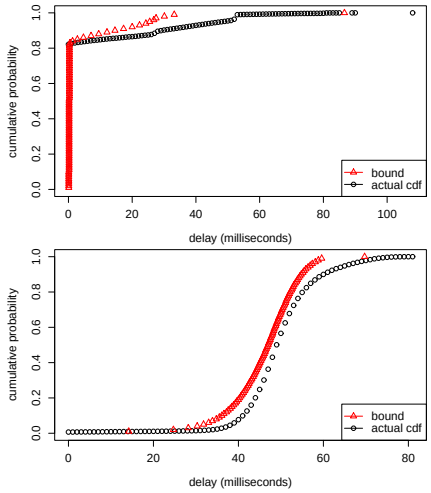
(a) Long-lived TCP sources, dumbbell topology.

(b) Constant-bit rate UDP sources, star topology, route r_1.

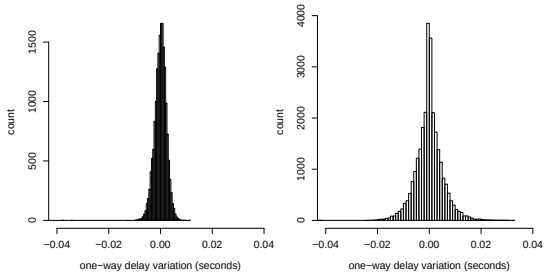
5: Delay distribution quantile estimates, with 90% confidence interval.

that there is no clear basis by which to compare estimates. As discussed in § 3, there are multiple definitions of delay variation, for example in the RTP standard RFC 3550 and in the IPPM standard RFC 3393. Therefore, we focus on a comparative analysis among these two IETF standards and our DV matrix formulation.

We first look at the `one-way-ipdv` metric of RFC 3393. Each `one-way-ipdv` sample is produced by choosing consecutive packets of a probe stream identical to the SLAM stream (48 byte packets sent at 30 msec intervals). Histograms of `one-way-ipdv` samples for the long-lived TCP traffic scenario (left) and for the self-similar traffic scenario at 60% offered load (right) in the dumbbell topology are shown in Figure 7. The plots show that while there is a narrower range of values for the long-lived TCP source scenario the shapes of each distribution are qualitatively similar. The narrow range for the long-lived TCP scenario arises because the queue is often close to full. Also, the left tail of the long-lived TCP plot and both left and right tails of the self-similar plot show that there are some large `one-way-ipdv` values. Beyond simple qualitative observations of these plots, however, it is not clear how queuing *dynamics* along the path are captured by this metric since it only captures local differences in delays. It is also not clear how one might infer application performance, *e.g.*, for a VoIP stream, since large



6: Computed bounds for the delay distribution on path r_1, given measured delay distributions for paths r_2, r_3, and r_4. Results are shown for the UDP CBR scenario (top), and self-similar TCP traffic (bottom).



7: Histograms of RFC 3393 `One-way-ipdv` samples for the long-lived TCP traffic scenario (left), and for the self-similar self-similar traffic scenario at 60% offered load (right) using the dumbbell topology. Each `One-way-ipdv` sample is produced by choosing consecutive packets of a periodic stream.

values of `one-way-ipdv` do not necessarily translate into packet losses because of underbuffering at an application playout buffer.

Figure 8a plots 60 second periods of the RTP jitter metric along with a time series of queuing delays (top) and the DV matrix metric along with a time series of queuing delays (bottom). The background traffic used for these plots is the self-similar traffic at a 60% offered load using the dumbbell topology. We calculate the two metrics using a probe stream identical to the SLAM stream. In these plots we observe first that although the RTP jitter and DV matrix metrics are calculated in very different ways, they have similar qualitative characteristics over time with the DV matrix exhibiting a somewhat smoother profile.

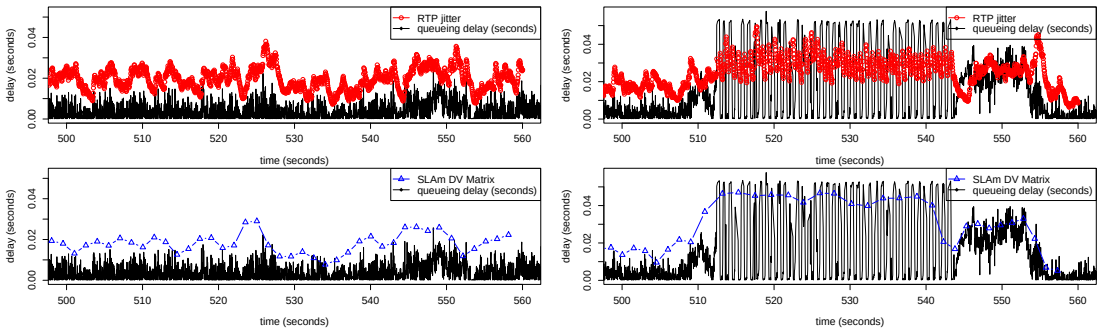
In order to expose additional aspects of the RTP and DV matrix metrics, we introduced a CBR traffic source that was sent in addition to the self-similar traffic at a 60% load, also using the dumbbell topology. Over periods of approximately 30 seconds, the CBR source alternated between on/off periods, each of about 500 msec. The addition of the CBR source results in a period of oscillation of the queue between full and empty as shown in Figure 8b. As with Figure 8a, the top plot shows the RTP jitter metric along with a time series of queuing delays and the bottom plot shows the DV matrix metric along with the same time series of queuing delays. We observe in these two plots that at the onset of the CBR on/off bursts, the RTP jitter metric oscillates in a similar way as the queue. The DV matrix metric, however, remains smooth and at an increased level, suggesting that relative to the other DV matrix measurements over this 60 second time interval, queuing turbulence along the path is greatest during the period of CBR bursts. In contrast, over the CBR burst period the RTP jitter values are often smaller than many other jitter values during the trace segment. Also, relative to the range of jitter values observed over the 60 second segment, the jitter values during the CBR burst period do not stand out—they stand out only in their oscillatory behavior. This effect is explained by the fact that although an EWMA filter with a small value for α is used (1/16) in the RTP jitter formulation, the view is still of individual delay variations rather than the behavior over a longer interval of time. Although the CBR traffic source we used to reveal this behavior is somewhat pathological, our observations in this context are consistent with the behavior of the RTP and DV matrix values during periods of queuing turbulence in other traffic scenarios and topologies/paths (not shown due to space limitations).

Finally, we examine the performance of the DV matrix metric in the star topology. A desirable property of a method for measuring delay variation is that, in a multihop setting, it should report a maximum over all the links comprising the path. In Figure 9, we plot the DV matrix metric for links e_1 and e_4 which make up path r_2 for the CBR UDP traffic scenario. Plots for other traffic scenarios and routes are qualitatively similar to Figure 9. Observe that the DV matrix value reported over the path over time is generally the maximum reported for the individual links. These results are encouraging. First, the DV matrix methodology appears to yield reliable measures of delay variation over a single hop. Second, the performance of the DV matrix metric in the two-hop star topology appears to be robust. In the future we plan to examine its sensitivity to different matrix sizes and in more complex multihop settings.

6.4 Loss

Table 4 compares the true loss rate measured using the passive traces (true values) with the loss rate estimates of SLAM and the standard RFC 2680 [8] (Poisson-modulated) probe stream sent at the same rate. Values are shown for each of the traffic scenarios, and for the two topologies and are average loss rates over the duration of each experiment. Note that differences in true values are due to inherent variability in traffic sources. Considering both results for the dumbbell topology (Table 4a) and for the star topology (Table 4b), we see that the standard stream yields very poor estimates of the true loss rate, and that the estimates produced by SLAM are close to the true values. Moreover, in all but a few cases, the RFC 2680 probe estimates are off by more than an order of magnitude—a significant relative error. For a number of experiments, the Poisson estimates are close to zero—a phenomenon consistent with earlier experiments [35] and primarily due to the fact that single packet probes generally yield poor indications of congestion along a path. (Note that these accuracy improvements are consistent with experiments described in [35].) The estimates produced by SLAM are significantly better, with a maximum relative error occurring in the case of the open-loop CBR background traffic for both the dumbbell and star topologies.

Figure 10 shows the true loss rate and SLAM-estimated loss rate over the duration of experiments using long-lived TCP traffic in the dumbbell topology (top) and self-similar traffic on route r_2 in the star topology (bottom). True loss rate estimates are shown for 10 second intervals and estimates for SLAM are shown for 30 second intervals. Results for other experiments are consistent with plots in Figure 10. The upper and lower bars for SLAM indicate estimates of one standard deviation above and below the mean using the variance estimates derived from [37]. For the SLAM estimates we see



(a) Time series plots of 60 second periods of the RTP jitter metric along with a time series of queuing delays (top) and the DV matrix metric along with a time series of queuing delays (bottom). Background traffic is the self-similar traffic at a 60% offered load.

(b) Time series plots of 60 second periods of the RTP jitter metric along with a time series of queuing delays (top) and the DV matrix metric along with a time series of queuing delays (bottom). Background traffic is created using periodic intervals of CBR UDP traffic that are sent in on/off bursts each of approximately 500 msec in addition to continuous self-similar traffic at a 60% offered load.

8: A comparison of the behavior of the RTP (RFC 3550) jitter metric and the DV matrix metric using the dumbbell topology.

4: Comparison of loss rate estimation accuracy for SLAM and RFC 2680 (Poisson) streams using the (a) dumbbell and (b) star testbed topologies. Values are average loss rates over the full experiment duration.

(a) Loss accuracy using the dumbbell topology.

Probe stream → **Traffic scenario ↓**	SLAM		RFC 2680 (Poisson)	
	true	estimate	true	estimate
CBR	0.0051	0.0073	0.0051	0.0017
Long-lived TCP	0.0163	0.0189	0.0163	0.0062
Harpoon self-similar (60% load)	0.0008	0.0007	0.0017	0.0000
Harpoon self-similar (75% load)	0.0049	0.0050	0.0055	0.0000

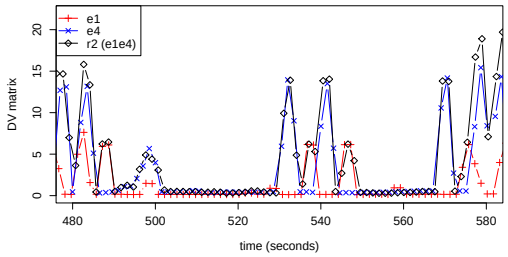
(b) Loss accuracy using the star topology.

Probe stream → **Traffic scenario (route) ↓**	SLAM		RFC 2680 (Poisson)	
	true	estimate	true	estimate
CBR (r_1)	0.0391	0.0370	0.0391	0.0087
CBR (r_2)	0.0339	0.0334	0.0339	0.0064
CBR (r_3)	0.0458	0.0359	0.0458	0.0068
CBR (r_4)	0.0390	0.0371	0.0390	0.0089
Long-lived TCP (r_1)	0.0081	0.0078	0.0092	0.0008
Long-lived TCP (r_2)	0.0463	0.0446	0.0433	0.0104
Long-lived TCP (r_3)	0.0021	0.0024	0.0028	0.0006
Long-lived TCP (r_4)	0.0479	0.0478	0.0442	0.0072
Harpoon self-similar (r_1)	0.0170	0.0205	0.0289	0.0058
Harpoon self-similar (r_2)	0.0008	0.0006	0.0069	0.0000
Harpoon self-similar (r_3)	0.0192	0.0178	0.0219	0.0036
Harpoon self-similar (r_4)	0.0005	0.0006	0.0002	0.0000

the narrowing of variance bounds as an experiment progresses, and that the true loss rate is usually within these bounds. We also see that SLAM tracks the loss rate over time quite well, with its estimated mean closely following the true loss mean.

7. DISCUSSION AND CONCLUSIONS

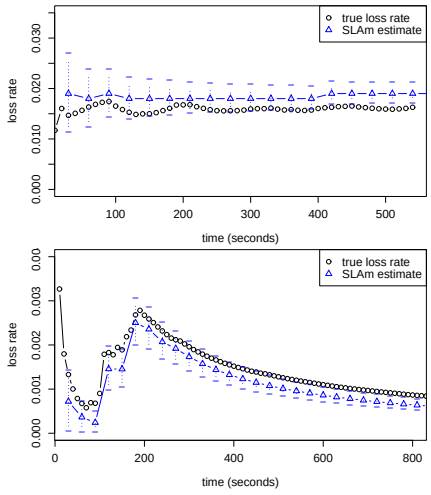
We believe that SLAM represents a significant step forward for SLA compliance monitoring using active measurements. However, there are a number of issues that remain. First, there are additional issues to consider in the network-wide setting. For example, a deployment strategy must be developed to coordinate probe streams so that links internal to the network are not carrying "too much" measurement traffic. Another key question is: given a daily (or based on some other time scale) budget of probes that may be used to monitor compliance with a SLA, what are the considerations for



9: Performance of the DV matrix in a two-hop setting (r_2) using the star topology. Time series plot shown for CBR UDP traffic scenario. Curves show DV matrix metric for route r_2 and separately for links e_1 and e_4 that comprise r_2.

optimizing the probe process? Should the probing period be over a relatively long time scale (*e.g.*, the entire interval of interest), thus potentially limiting the accuracy of estimates, or should the probing period be over a shorter time scale, potentially improving estimation accuracy but at the cost of not probing over the entire interval, thus potentially missing important events? We have assumed in this paper that perfect accuracy is the goal for compliance monitoring. However, for some SLAs, a tradeoff (if it is predictable) between accuracy and measurement overhead may be appropriate. Next, our examples of distributional inference have focussed on delay. We plan to more closely examine loss in the future. Finally, while measuring availability in a simple path-oriented scenario is rather straightforward, simple application of performance tomography to infer network-wide availability may not be sufficient in the face of routing changes.

In summary, this paper introduces a new methodology for SLA compliance monitoring using active measurements, including new methods for measuring end-to-end packet loss, mean delay, and delay variation. We propose a new method for obtaining confidence intervals on the empirical delay distribution. We also describe a new methodology for inferring lower bounds on the quantiles of a distribution of a performance metric along a path in a network-wide setting from a subset of known paths. We implemented these measurement methods in a tool called SLAM that unifies the various probe streams resulting in lower overall probe volume. We evaluated the capabilities of the tool in a controlled laboratory en-



10: Comparison of true loss rate with SLAM estimates over time. True loss rates are plotted using 10 second intervals. SLAM estimates are plotted using 30 second intervals. Plots shown for long-lived TCP traffic in the dumbbell topology (top) and self-similar traffic on route r_2 in the star topology (bottom). The upper and lower bars for SLAM indicate estimates of one standard deviation above and below the mean using the variance formulation of [37].

vironment using a range of traffic conditions and in one- and two-hop settings. Our results show that SLAM's delay and loss rate estimates are much more accurate than estimates obtained through standard probe methodologies. Furthermore, we illustrated the convergence and robustness properties of the loss, delay, and delay variation estimates of SLAM which make it useful in an operational setting.

Acknowledgments

We thank the anonymous reviewers and our shepherd Anees Shaikh for their feedback. This work is supported in part by NSF grant numbers CNS-0347252, CNS-0627102, CNS-0646256 and CCR-0325653 and by Cisco Systems. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or of Cisco Systems.

8. REFERENCES

[1] ITU-T Recommendation G.107, The E-model, a computational model for use in transmission planning, March 2005.

[2] AT&T Managed Internet Service (MIS). `http://new.serviceguide.att.com/mis.htm`, 2007.

[3] NTT Communications Global IP Network Service Level Agreement (SLA). `http://www.us.ntt.net/support/sla/network/`, 2007.

[4] Sprint NEXTEL service level agreements. `http://www.sprint.com/business/support/serviceLevelAgreements.jsp`, 2007.

[5] S. Agarwal, J. Sommers, and P. Barford. Scalable network path emulation. In *Proceedings of IEEE MASCOTS '05*, September 2005.

[6] M. Aida, N. Miyoshi, and K. Ishibashi. A scalable and lightweight QoS monitoring technique combining passive and active approaches. In *Proceedings of IEEE INFOCOM '03*, March 2003.

[7] G. Almes, S. Kalidindi, and M. Zekauskas. A one-way delay metric for IPPM. IETF RFC 2679, September 1999.

[8] G. Almes, S. Kalidindi, and M. Zekauskas. A one way packet loss metric for IPPM. IETF RFC 2680, September 1999.

[9] D. Arifler, G. de Veciana, and B. L. Evans. 'network tomography based on flow level measurements. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Montreal, Canada, May 17-21 2004.

[10] P. Barford and J. Sommers. Comparing probe- and router-based packet loss measurements. *IEEE Internet Computing*, September/October 2004.

[11] J. Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of ACM SIGCOMM '93*, September 1993.

[12] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network internal loss characteristics. *IEEE Trans. on Information Theory*, 45(7):2462–2480, 1999.

[13] M.C. Chan, Y.J. Lin, and X. Wang. A scalable monitoring approach for service level agreements validation. In *IEEE International Conference on Network Protocols (ICNP)*, pages 37–48, 2000.

[14] Y. Chen, D. Bindel, and R. Katz. Tomography-based overlay network monitoring. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '03*, October 2003.

[15] Y. Chen, D. Bindel, H. Song, and R.H. Katz. An algebraic approach to practical and scalable overlay network monitoring. In *Proceedings of ACM SIGCOMM '04*, 2004.

[16] B.Y. Choi, S. Moon, R. Cruz, Z.-L. Zhang, and C. Diot. Practical delay monitoring for ISPs. In *Proceedings of ACM CoNEXT '05*, 2005.

[17] D.B. Chua, E.D. Kolaczyk, and M. Crovella. Efficient estimation of end-to-end network properties. In *Proceedings of IEEE INFOCOM '05*, 2005.

[18] L. Ciavattone, A. Morton, and G. Ramachandran. Standardized active measurements on a tier 1 IP backbone. *IEEE Communications*, 41(6):90–97, June 2003.

[19] R. Cole and J. Rosenbluth. Voice over IP Performance Monitoring. *ACM SIGCOMM Computer Communcation Review*, April 2001.

[20] E. Corell, P. Saxholm, and D. Veitch. A user friendly TSC clock. In *Proceedings of Passive and Active Measurement Conference*, March 2006.

[21] C. Demichelis and P. Chimento. IP packet delay variation metric for IP performance metrics (IPPM). IETF RFC 3393, November 2002.

[22] N. Duffield. Network Tomography of Binary Network Performance Characteristics. *IEEE Transactions on Information Theory*, 52, 2006.

[23] N. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. Inferring link loss using striped unicast probes. In *Proceedings of IEEE INFOCOM '01*, April 2001.

[24] Y. Liang, N. Farber, and B. Girod. Adaptive playout scheduling and loss concealment for voice communication over IP networks. *IEEE Transactions on Multimedia*, 5(4), December 2003.

[25] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. *IEEE/ACM Transactions on Networking*, 10(6):761–775, 2002.

[26] J. Mahdavi and V. Paxson. IPPM metrics for measuring connectivity. IETF RFC 2678, September 1999.

[27] J. Martin and A. Nilsson. On service level agreements for IP networks. In *IEEE INFOCOM '02*, 2002.

[28] A. Pasztor and D. Veitch. A precision infrastructure for active probing. In *Passive and Active Measurement Workshop*, 2001.

[29] V. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, University of California Berkeley, 1997.

[30] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP performance metrics. IETF RFC 2330, 1998.

[31] M. Roughan. Fundamental bounds on the accuracy of network performance measurements. In *ACM SIGMETRICS*, June 2005.

[32] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. IETF RFC 3550, July 2003.

[33] A. Shaikh and A. Greenberg. Operations and Management of IP Networks: What Researchers Should Know. Tutorial Session, ACM SIGCOMM '05. August, 2005.

[34] J. Sommers and P. Barford. Self-configuring network traffic generation. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '04*, 2004.

[35] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. In *Proceedings of ACM SIGCOMM '05*, 2005.

[36] J. Sommers, P. Barford, N. Duffield, and A. Ron. A Framework for Multi-objective SLA Compliance Monitoring. In *Proceedings of IEEE INFOCOM (minisymposium)*, May 2007.

[37] J. Sommers, P. Barford, N. Duffield, and A. Ron. A geometric approach to improving active packet loss measurement. *To appear, IEEE/ACM Transactions on Networking*, 2008.

[38] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf 1.7.0 – the TCP/UDP bandwidth measurement tool. `http://dast.nlanr.net/Projects/Iperf`. 2007.

[39] Yolanda Tsang, Mark Coates, and Robert Nowak. Passive unicast network tomography using em algorithms. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1469–1472, Salt Lake City, Utah, May 2001.

[40] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modeling of temporal dependence in packet loss. In *Proceedings of IEEE INFOCOM '99*, March 1999.

[41] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the constancy of Internet path properties. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop '01*, November 2001.

[42] T. Zseby. Deployment of sampling methods for SLA validation with non-intrusive measurements. In *Proceedings of Passive and Active Measurement Workshop*, 2001.