Confident Estimation for Multistage Measurement Sampling and Aggregation

Edith Cohen, Nick Duffield, Carsten Lund, Mikkel Thorup
AT&T Labs–Research, 180 Park Avenue, Florham Park, NJ 07901,USA
Email: {edith,duffield,lund,mthorup}@research.att.com

ABSTRACT

Measurement, collection, and interpretation of network usage data commonly involves multiple stage of sampling and aggregation. Examples include sampling packets, aggregating them into flow statistics at a router, sampling and aggregation of usage records in a network data repository for reporting, query and archiving. Although unbiased estimates of packet, bytes and flows usage can be formed for each sampling operation, for many applications it is crucial to know the inherent estimation error. Previous work in this area has been limited mainly to analyzing the estimator variance for particular methods, e.g., independent packet sampling. However, the variance is of limited use for more general sampling methods, where the estimate may not be well approximated by a Gaussian distribution.

This motivates our paper, in which we establish Chernoff bounds on the likelihood of estimation error in a general multistage combination of measurement sampling and aggregation. We derive the scale against which errors are measured, in terms of the constituent sampling and aggregation operations. In particular this enables us to obtain rigorous confidence intervals around any given estimate. We apply our method to a number of sampling schemes both in the literature and currently deployed, including sampling of packet sampled NetFlow records, Sample and Hold, and Flow Slicing. We obtain one particularly striking result in the first case: that for a range of parameterizations, packet sampling has no additional impact on the estimator confidence derived from our bound, beyond that already imposed by flow sampling.

Categories and Subject Descriptors

C.2.3 [**Computer–Communications Networks**]: Network Operations—*Network monitoring*; G.3 [**Probability and Statistics**]

SIGMETRICS'08, June 2–6, 2008, Annapolis, Maryland, USA.

General Terms

Measurement, Theory

Keywords

Network Measurement, Sampling, Estimation, Confidence Intervals

1. INTRODUCTION

1.1 Motivation

Network traffic measurement typically involves the following steps: (i) taking traffic measurements at an observation point such as a router or special purpose measurement device; (ii) exporting the measurements from the observation point to a collector, possible via intermediate staging servers; (iii) storage in a database that serves reporting and query functions; (iv) archiving older measurements. A large network service provider may employ 1,000's of routers and 10,000's of interfaces; consequently the volume of traffic measurements is potentially enormous.

A large class of network management applications—such as traffic engineering, capacity planning and troubleshooting—use measured traffic usage as their input data. This data takes the form of counts of packets, bytes, or the number of flows, over time periods, broken out over subsets of traffic classified according to, e.g., source, destination, applications class, and/or other features. Some traffic subsets are known in advance of the time of measurement, e.g. for routine reporting of usage by application and customer. However, for troubleshooting or exploratory studies, the traffic subsets of interest are not known in advance of measurement. The requirement to be able to aggregate measurements over arbitrary subsets and timescales precludes measurement simply using static counters in routers; an infeasible large number would be required to measure traffic at sufficiently fine granularity to service all future queries.

These considerations motivate the use of packet and flow measurement techniques currently deployed in production networks, such as NetFlow [3]. Routers summarize individual traffic flows passing through them and export a stream of summaries (flow records) to a collector. Increasingly, large network service providers employ sampling and/or aggregation during the steps (i)–(iv) above in order to control data volumes. Some specific cases are packet sampling, aggregation of sampled packets into flow records (these two steps are commonly accomplished using Sampled NetFlow[3]), and the sampling and aggregation of the resulting flow records

on their collection path. In addition to these methods currently deployed, stateful packet sampling methods have also been proposed in the literature, e.g., Sample and Hold [17]. We review these in more detail shortly.

Whenever sampling is employed in the measurement infrastructure, traffic usage must be estimated from the samples. A generic way to produce unbiased estimators is to divide the weight of each contribution to usage (e.g., from a sampled packet or flow) by its sampling probability; this is an example of the standard Horwitz-Thompson estimator [19]. When multiple stages of sampling are employed, information about the original traffic is progressively lost. Indeed, for many applications it is crucial to know the inherent error associated with estimation. Given an estimate X of a traffic volume, for engineering purposes we want to be able to set a confidence level X_+ for the true underlying traffic volume $\bar{X}$ in the following way: there must be only a known small chance that $\bar{X}$ could have exceeded X_+ yet produced the estimate X. Likewise, we wish to generate a corresponding lower level X_- which $\bar{X}$ can fall under with only some small probability. A particular version of this problem is when $X = 0$, i.e., how likely it is that $\bar{X}$ could exceed X_+ when *no traffic* is sampled.

An important example in use today is the further sampling of packet sampled NetFlow records. If the NetFlow records were not derived from sampled packets, the flow sizes would be known exactly, and hence the results of [29] would give the required confidence intervals. But when packet sampling is used prior to flow sampling, the flow size is now an estimate, i.e., a random variable, so the previous result does not apply. Our aim in this paper is to produce a general framework in which to calculate confidence intervals for arbitrary combinations of aggregation operations with (a class of) sampling operations.

1.2 Contribution and Outline

The sampling operations considered in this paper are a generalization of **threshold sampling** introduced in [10] for the size-dependent sampling of flow records; see Section 1.3 below. Threshold sampling is a form of Poisson sampling that allows for unit selection probabilities; items are selected independently with a specified probability. Threshold sampling achieves an optimal trade-off between low sample volumes and low estimation variance; see Section 2 below. It is employed or proposed for a number of internet traffic measurement systems [23, 21], and in other fields [30].

(i) In Section 2 we define generalized threshold sampling and show how a number of sampling schemes from the literature are in fact instances of it.

(ii) In Section 3 we set up the problem of unbiased estimation for multistage sampling and aggregation of network usage as a stochastic process on a tree. In our formulation (a) the leaf variables represent the weights of unsampled data; (b) threshold sampling operation are associated with each link; (c) a general node aggregates sampled variables from its child nodes, and (d) the root node carries the result of the multistage sampling operation. Within this formulation we are able to derive Chernoff bounds for the tail distribution of the estimation error. These bounds are sometimes called exponential bounds: in this case the tail probability of a given fractional estimation error falls off exponentially in the size of the usage to be estimated. The bounds supply rigorous confidence intervals for the true aggregates in terms of their estimates. Some details of the analysis are deferred to Section 7.

(iii) Section 4 applies the results to three single and multistage sampling algorithms that are currently employed and/or proposed in the literature. Three important cases that we treat are

- Threshold Sampling [10] of packet sampled NetFlow records, for which we analyze the interplay between packet sampling rate and flow sampling threshold for flow records and the resulting impact on estimation accuracy.
- Sample and Hold [17], which we realize as a multistage threshold sample; and
- Flow Slicing [23], a composition of uniform packet sampling, sample and hold, and threshold sampling.

(iv) Section 5 evaluates the performance of the bounds on a network dataset, and shows that they perform within expected accuracy. We conclude in Section 6.

1.3 A Sampling of Sampling and Aggregation

In order to set the scene for the analysis of this paper, we briefly describe the sampling and aggregation methods that will feature in this paper; they serve as examples to which our methods are applicable.

(i) *Packet Sampling:* Packets are sampled by a router or special purpose measurement device. In one variant, a report on each sampled packet is exported to a collector; examples include the sFlow router feature [25], the emerging Packet Sampling (PSAMP) standard of the IETF [7], and Trajectory Sampling [15].

In another variant, packet sampling is a precursor to the compilation of flow statistics, which cannot generally be performed at the line rate of router interfaces. Sampled NetFlow [3] operates in this manner, packet sampling being either periodic in the packet count (every N^{th} packet is selected) or stratified by count (one packet is chosen at random from every group of N successive packets).

(ii) *Aggregation of packet information into flow statistics:* Flows are sets of packets with a common property, known as the key, observed at the router within some period of time. The key commonly comprises fields from the packet header, such as source and destination IP address and TCP/UDP port numbers. Flows can be demarked using, e.g., periodic time intervals, or timeouts which can be inactive (the flow is considered terminated when the time since the last packet observed with the flow key exceeds an inactive timeout threshold) or active (the the time since the first packet observed with the flow key exceeds an active timeout threshold); other criteria may be used in addition, see [3]. When the flow terminates, the router summarizes its aggregate properties in a flow record (typically including the key, total packets and bytes seen, observation time of first and last packet) which is then exported.

(iii) *Sampling of Flow Records.* The salient empirical fact concerning flows is that a small proportion of the flows represent a large proportion of packets and bytes; see, e.g.,

[10]. For this reason, estimates of packet and bytes counts derived from uniformly sampled flow records have poor accuracy, being very sensitive to inclusion or omission of the large flows. *Threshold Sampling* was proposed to overcome this problem [10]. Flows reporting (byte or packet) size x are sampled with probability $p_z(x) = \min\{1, x/z\}$ where z is known as the threshold. Flows of size at least z are sampled with probability 1 while smaller flows are sampled with probability proportional to their size. The form of p_z can be shown to give the optimal tradeoff between the average number of flows sampled and the variance of the flow size estimator derived from the samples. *Priority sampling* is a variant of threshold sampling in which exactly k flows are selected from a population [11].

(iv) *Stateful Packet Sampling.* Several authors have proposed packet sampling and aggregation methods that maintain some degree of flow state. In Counting Samples [18] and the subsequent Sample and Hold [17], potential new flow cache entries are sampled prior to instantiation. Specifically, when a packet arrives, if a cache entry is currently maintained for its key, the entry is accordingly updated. If no entry exists, then one is instantiated with a fixed probability $1 - p$ (Counting Samples) or probability $1 - r^x$ where x is the packet size (Sample and Hold). Thus the chance to miss a flow entirely is exponentially small in the number of packets (or bytes)[1]. Further work in this direction involves dynamic adjustment of sampling probabilities and progressive resampling of aggregates in response to changing network loads and cache utilization [6, 5, 16, 22].

(v) *Aggregation of Flow Records* Flow records (possibly sampled) may be aggregated over longer collection windows (e.g. minutes or hours) for routing reporting or archiving.

1.4 Related Work

Prior studies have examined estimation error associated with individual sampling methods. In some cases, estimator variances are used to derive confidence intervals based on a Gaussian approximation. From the Central Limit Theorem, this can be a reasonable approach for simple sampling of a large number of packets. However, the variance is of limited use for more general sampling methods, where such approximations may not be so good. A number of authors have analyzed estimator variance to determine the ramifications of packet sampling for measurement-based applications. The impact for the problem of ranking flows by volume is considered in [20]; for passive performance measurement in [31], together with other applications of Trajectory Sampling in [8]. The impact of sampling on the estimation of packet size distributions was considered in [4]. The effect on security applications has been considered by a number of authors: see, e.g., [2, 24, 26].

Stateful sampling methods for aggregating packets into flows have been proposed in recent years. Work in this area often incorporates an analysis of estimation error either through analysis of variance; we list in particular [5, 6, 16, 17, 18, 22]. The original design of Sample and Hold proposed estimating the actual flow bytes by the sampled flow bytes [17]. This yields a biased estimator whose undercount of the actual flow bytes follows a truncated geometric distribution. In fact there is no unbiased estimator of the actual bytes that is a function of the sampled bytes alone. A more recent paper [23], which derived unbiased estimators for Sampled and Hold, analyzed their variance. Estimation variance for Threshold and Priority Sampling as applicable to sampling of flow records, has been examined in [1, 10, 11, 14, 27, 28].

Closest in approach to the current paper are the following. [29] used Chernoff bounds to derive confidence intervals on the true traffic volumes, based on estimates derived through threshold and priority sampling. [13] introduced the notion of a generalized threshold and discussed its behavior under the limited case of aggregation and threshold sampling. [9] showed how the Law of Total Variance can be used propagate estimator variance in multistage sampling and aggregation.

[1]We emphasize the chance to sample a flow with Sample and Hold is the same as for normal packet sampling with the same probability. But Sample and Hold has an advantage in reporting more information than sampled NetFlow.

2. GENERALIZED THRESHOLD SAMPLING

In this section we define generalized threshold sampling, and show how it encompasses a number of sampling schemes in current use and proposed in the literature.

2.1 Terminology

Consider a **weight** x, i.e., a nonnegative possibly random variable. In (standard) **threshold sampling**, a weight x is sampled with probability $p_z(x) = \min\{1, x/z\}$. The corresponding unbiased estimate of x is $\widehat{x} = (\chi/p_z(x))x = \chi \max\{x, z\}$, where χ is the indicator function for selection, i.e., $\chi = 1$ w.p. $p_z(x)$ and 0 otherwise. In [10] the probability p_z is shown to minimize the *cost* $C_z = \mathsf{E}[\chi] + z^{-2}\,\mathsf{Var}\,\widehat{x}$, i.e., a linear combination of expected number of samples and variance estimate. It is desirable to keep both these small; p_z implements their optimal trade-off.

Generalized threshold sampling admits more general sampling probabilities. In fact, for applications we need to generalize even further to the multidimensional case; we write $\boldsymbol{x} = (x^{(1)}, \ldots, x^{(d)}) \in [0, \infty)^d$. For example, $(x^{(1)}, x^{(2)})$ may denote packets and bytes reported in a flow record. We do not assume that all possible values of x are allowed. For the flow record example, protocol conventions concerning packets sizes impose constraints between $x^{(1)}$ and $x^{(2)}$. In some cases, the sampling properties may be determined entirely by a subset of the $x^{(i)}$, the remaining variables are just additive auxiliary variables. For example: flow sampling can be performed on the basis of byte values $x^{(2)}$; but the packets $x^{(1)}$ can also be estimated. Generally, we will denote the set of allowed $\boldsymbol{x}$ by Ω.

A **probability function** is a map $p : [0, \infty)^d \to [0, 1]$ such that $p(\boldsymbol{x}) = 0$ implies $\boldsymbol{x} = 0$. Denote by $\Omega_p \subset \Omega$ the allowed $\boldsymbol{x}$ on which the probability function is strictly less than 1:

$$\Omega_p = \{\boldsymbol{x} \in \Omega : p(\boldsymbol{x}) < 1\} \tag{1}$$

With each probability function p we associate a vector of **generalized thresholds** $\boldsymbol{\tau}_p = (\tau_p^{(1)}, \ldots, \tau_p^{(d)})$:

$$\tau_p^{(i)} = \sup_{\boldsymbol{x} \in \Omega_p} \frac{x^{(i)}}{p(\boldsymbol{x})} \tag{2}$$

Generalized threshold sampling entails sampling $\boldsymbol{x}$ with probability $p(\boldsymbol{x})$, where p is a probability function for which the thresholds $\tau_p^{(i)} < \infty$. Finally, we will find it useful to

define

$$\delta_p^{(i)} = \sup\{x^{(i)} : \boldsymbol{x} \in \Omega_p\} \tag{3}$$

Clearly $\delta_p^{(i)} \leq \tau_p^{(i)}$.

2.2 Examples

(i) *Standard threshold sampling:* $p(x) = p_z(x) := \min\{1, x/z\}$ is an example with $\delta = \tau = z$.

(ii) *Uniform Sampling* For uniform sampling with probability p of weights whose values are unbounded, then clearly the sampling threshold is infinite $\tau = \sup_{x>0} x/p = +\infty$. However, if there is an a priori upper bound $x_{\max}$ on x, then $\tau = x_{\max}/p$. An example of this is sampling of IP packets, where x is the packet size, bounded above by the network maximum transmission unit (MTU). An MTU of 1500 bytes is currently common.

(iii) *Flow Slicing:* Flow Slicing [23] includes an extension of threshold sampling to operate with a multifactor aggregate flow descriptor $\boldsymbol{x} = (x^{(1)}, x^{(2)}, x^{(3)})$, these being the aggregate numbers of bytes, packets and flows possessing a TCP SYN flag that match a given key. The sampling probability is $p(\boldsymbol{x}) = \min\{1, \sum_{i=1}^{3} x^{(i)}/z^{(i)}\}$ for some $z^{(i)} > 0$. Thus $\tau_p^{(i)} \leq z^{(i)}$. Equality is possible if $x^{(j)} = 0$ is allowed in Ω. On the other hand, known constraints between variables can constrain the thresholds. Suppose we know the minimum possible packet size M_{min} and the MTU, which we denote by $M_{\max}$. Then

$$x^{(2)} M_{min} \leq x^{(1)} \leq x^{(2)} M_{\max} \tag{4}$$

We analyze estimation error for Flow Slicing further in Section 4.3.

2.3 Estimation and Bounded Uncertainty

The thresholds $\boldsymbol{\tau}_p = (\tau_p^{(i)})$ play a role in simple bounds on the uncertainty of the usual Horwitz-Thompson estimators of $\boldsymbol{x}$. Let α denote a random variable uniformly distributed on $(0, 1]$. The sampling operator associated with the probability function p is a random function $S_p : [0, \infty)^d \to [0, \infty)^d$ where

$$S_p(\boldsymbol{x}) = \frac{\boldsymbol{x}}{p(x)} I(p(\boldsymbol{x}) \geq \alpha) \tag{5}$$

where $I(A)$ is the usual indicator function of the event A. $\{p(\boldsymbol{x}) \geq \alpha\}$ is the event that the weight $\boldsymbol{x}$ is sampled. If it is sampled, then the estimate of each component $x^{(i)}$ is formed by dividing by the sampling probability $p(\boldsymbol{x})$. It is elementary that $\mathsf{E}[S_p(\boldsymbol{x})] = \boldsymbol{x}$, i.e, $\widehat{\boldsymbol{x}} = S_p(\boldsymbol{x})$ is an unbiased estimator of $\boldsymbol{x}$.

In the estimation context, we can interpret the $\tau_p^{(i)}$ as follows. They bound possible values of the estimates $\widehat{x}^{(i)}$ can take when they are not equal to $x^{(i)}$. Thus, roughly speaking, they are the largest possible uncertain values of the $\widehat{x}^{(i)}$. This interpretation can be extended a little further. The following bounds on the variance of $\widehat{x}^{(i)}$ are easy to establish:

$$\mathsf{Var}(\widehat{x}^{(i)}) = (x^{(i)})^2 (p(\boldsymbol{x})^{-1} - 1) \leq \tau_p^{(i)} x^{(i)} \tag{6}$$

When $\tau_p^{(i)}$ is unbounded, so is the corresponding variance. Thus the finiteness condition on $\tau_p^{(i)}$ is natural when we want to consider estimation with bounded variance.

3. ANALYSIS OF MULTISTAGE SAMPLING AND AGGREGATION

In this section we represent multistage sampling and aggregation of network measurements by a stochastic process on a tree, whose values represent unbiased estimates after each stage of aggregation. We establish Chernoff bounds for this stochastic process that exponentially bound the tail probabilities of the estimation error.

3.1 Sampling Trees.

A generalized threshold sampling tree is a tuple $(V, E, P, \boldsymbol{X})$ where (V, E) is a tree with node (or vertex) set V and edges E, $P = \{p_k : k \in V\}$ is a set of probability functions, and the sampling process $\boldsymbol{X} = \{\boldsymbol{X}_k : k \in V\}$ is a vertex-indexed family of weights in $[0, \infty)^d$ as defined below.

In a generalized threshold sampling tree we associate nodes with aggregation and edges with sampling. $c(k)$ is the set of child nodes of node k. $R \subset V$ is the set of leaf nodes, i.e., those with no children. $d(k)$ is the set of descendants of k, not including k itself. $a(k)$ is the set of ancestors of k, not including k itself, i.e., $a(k) = \{j : k \in d(j)\}$. R_k is the set of leaves descended through k, i.e., $R_k = R \cap d(k)$. The root node of the tree will be denoted by 0.

An edge (j, k) with terminal node k is associated with a probability function p_k. The sampling process $\boldsymbol{X}$ is determined as follows. Each leaf node $k \in R$ comes equipped with some fixed $\boldsymbol{X}_k \geq 0$. For all other nodes $k \in V \setminus R$, the aggregate $\boldsymbol{X}_k$ is defined through the componentwise sum (see Figure 1)

$$\boldsymbol{X}_k = \sum_{j \in c(k)} S_{p_j}(\boldsymbol{X}_j) \tag{7}$$

Note that we view the tree as a deterministic object in the sense that its topology is independent of the process $\boldsymbol{X}_k$. Thus even if $\boldsymbol{X}_k = 0$ because none of the weights $\boldsymbol{X}_j$ descended from k survived sampling, we do not delete or otherwise omit the branch descended from k from consideration.

Each $\boldsymbol{X}_k$ is an unbiased estimator of the total weight

$$\bar{\boldsymbol{X}}_k = \sum_{i \in R_k} \boldsymbol{X}_i \tag{8}$$

at leaves descended from k. Our aim is to establish Chernoff bounds on the difference. Without loss of generality we focus on the statistics of $\boldsymbol{X}_0 - \bar{\boldsymbol{X}}_0$.

3.2 Estimation Error Bounds

It is important to note that although a specific tree topology represents the multistage sampling and aggregation of a specific set of packets or flows, the analysis that we now present gives bounds which are independent of the topology and hence the details of the set of packets under study. In fact, the exponential error estimation bound below depends only on (a) the total (possibly multidimensional) usage of the traffic under study $\bar{\boldsymbol{X}}_0$, the measured usage $\boldsymbol{X}_0$ and a worst case threshold $\bar{\boldsymbol{\tau}}_0 = (\bar{\tau}_0^{(1)}, \ldots, \bar{\tau}_0^{(d)})$ that is a function only of the sampling operations used on the tree, and is, hence, presumably known in any given application. For clarity we denote the thresholds $\boldsymbol{\delta}_{p_k}$ and $\boldsymbol{\tau}_{p_k}$ by $\boldsymbol{\delta}_k$ and $\boldsymbol{\tau}_k$ respectively.

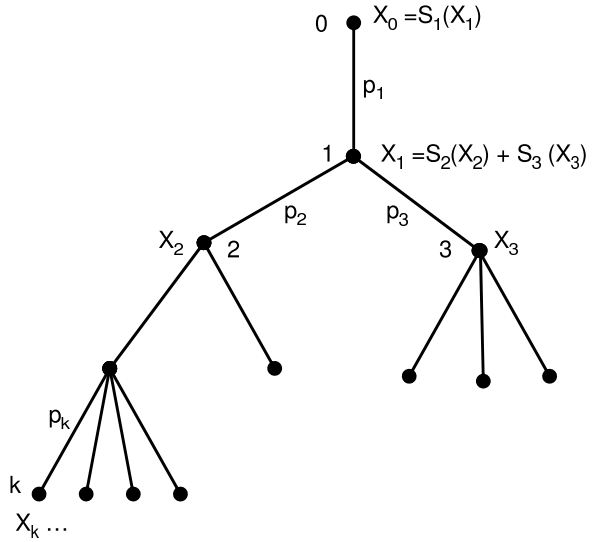

Figure 1: Illustration of topology with weights.

Define

$$K(\sigma) = \frac{e^{\sigma}}{(1+\sigma)^{1+\sigma}} \quad (9)$$

$$\bar{\tau}_k^{(i)} = \max_{j \in d(k)} \tau_j^{(i)} \quad (10)$$

Theorem 1 below is our main technical result. It states that we can apply standard type Chernoff bounds at any node k based on $\bar{\tau}_k$ from (10) regardless of the complexity of the subtree descending from k. The proof of Theorem 1 is deferred to Section 7.

THEOREM 1. *Let $\sigma > 0$. For each $i \in \{1, \ldots, d\}$,*

$$\mathsf{Pr}[X_0^{(i)} \geq (1+\sigma)\bar{X}_0^{(i)}] \leq K(\sigma)^{\bar{X}_0^{(i)}/\bar{\tau}_0^{(i)}} \quad (11)$$

$$\mathsf{Pr}[X_0^{(i)} \leq (1-\sigma)\bar{X}_0^{(i)}] \leq K(-\sigma)^{\bar{X}_0^{(i)}/\bar{\tau}_0^{(i)}} \quad (12)$$

The form of the bounds can be interpreted as follows. The probability of a given fractional error σ falls off exponentially in a scale factor. The scale factor obtained by dividing the size $\bar{X}_0^{(i)}$ of the usage to be estimated, by the threshold $\bar{\tau}_0^{(i)}$. Thus usage which is large compared with the threshold is easier to estimate accurately. Note that the governing threshold $\tau_0^{(i)}$ does not depend on aggregation operations, only the maximum threshold $\bar{\boldsymbol{\tau}}_0$ of sampling operations over all tree nodes.

3.3 Confidence Intervals from Estimates

The bounds in Section 3.2 can be inverted to determine confidence intervals for the true leaf total $\bar{\boldsymbol{X}}_0$, based on a particular value $\boldsymbol{x}$ of the estimate $\boldsymbol{X}_0$. Given particular outcome $\boldsymbol{X}_0 = \boldsymbol{x}$ and a probability $\varepsilon \in (0, 1]$, we seek as confidence limits those values $\boldsymbol{X}_{\pm}(\varepsilon, \boldsymbol{x}, \bar{\tau}_0)$ of $\bar{\boldsymbol{X}}_0$ that obey $X_-^{(i)}(\varepsilon, x^{(i)}, \bar{\boldsymbol{\tau}}_0^{(i)}) < x^{(i)} < X_+^{(i)}(\varepsilon, x^{(i)}, \bar{\boldsymbol{\tau}}_0^{(i)})$ and for which the probability to observe $x^{(i)}$ with $\bar{X}_0^{(i)}$ being greater than $X_+(\varepsilon, x^{(i)}, \bar{\tau}_0^{(i)})$ is less than ε, and likewise the probability to observe $x^{(i)}$ with $\bar{X}_0^{(i)}$ being less than $X_-(\varepsilon, x^{(i)}, \bar{\tau}_0^{(i)})$ will be less than ε.

Finding confidence intervals from estimated quantities is a well known task in general. For the current setting we take an approach similar to the related work in [29]. We write $\theta = \{\boldsymbol{X}_k : k \in R\}$; we express the underlying dependence $\boldsymbol{X}$ on θ by through its distribution Pr_θ. For each i let

$$B^{(i)}(x, \varepsilon) = \{\theta : \bar{X}_0^{(i)} \geq x, K(x/\bar{X}_0^{(i)} - 1)^{\bar{X}_0^{(i)}/\bar{\tau}_0^{(i)}} \leq \varepsilon\} \quad (13)$$

Then

$$\mathsf{Pr}_\theta[B^{(i)}(X_0^{(i)}, \varepsilon)] \quad (14)$$

$$= \mathsf{Pr}_\theta[X_0^{(i)} \leq \max\{x : \theta \in B^{(i)}(x, \varepsilon)\}] \quad (15)$$

$$= \max_{x:\ \theta \in B^{(i)}(x,\varepsilon)} \mathsf{Pr}_\theta[X_0^{(i)} \leq x] \leq \varepsilon \quad (16)$$

where the last inequality follows from (12). A similar argument follows using the upper bound (11) and hence we have proved:

THEOREM 2.

$$\mathsf{Pr}[\bar{X}_0^{(i)} \geq X_+(\varepsilon, X_0^{(i)}, \bar{\tau}_0^{(i)})] \leq \varepsilon \quad (17)$$

$$\mathsf{Pr}[\bar{X}_0^{(i)} \leq X_-(\varepsilon, X_0^{(i)}, \bar{\tau}_0^{(i)})] \leq \varepsilon \quad (18)$$

where $X_-(\varepsilon, x, \tau) < X_+(\varepsilon, x, \tau)$ are the solutions X to

$$K(x/X - 1)^{X/\tau} = \varepsilon \quad (19)$$

The roots $X_\pm(\varepsilon, x, \tau)$ can be written more compactly as

$$X_\pm(\varepsilon, x, \tau) = x\Xi_\pm(e^{-1}\varepsilon^{\tau/x}) \quad (20)$$

where for $y < 1/e$, $\Xi_-(y) < \Xi_+(y)$ are the solutions ξ to

$$\xi e^{-\xi} = y \quad (21)$$

3.4 Uniform Sampling

Here we show how uniform sampling can fit into our framework. This is crucial to the analysis in Section 4 of multistage sampling applications that employ uniform sampling in some stages. With uniform sampling $p(x) = p < 1$, the threshold τ is infinite unless the auxiliary variable is essentially bounded, i.e., bounded with probability 1. Thus when $p_k(x) = \pi_k$, independent of x, then

$$\tau_k = \operatorname{ess\,sup}(X_k)/\pi_k \quad (22)$$

In general, such a bound may not be particularly useful, since the maximum possible value of X_k may be far larger than the typical value, especially when X_k is the result of multiple successive sampling operations. However at leaf nodes k, X_k is deterministic, and in this case we have $\tau_k = X_k/\pi_k$. We apply this bound to the analysis of estimation errors arising from the sampling of packet sampled NetFlow records and Flow Slicing in Section 4.1 below.

4. APPLICATIONS

We now show how the methods of the previous section apply in three cases: threshold sampling of packet sampled flow records, Sample and Hold, and Flow Slicing.

4.1 Threshold Sampling of Packet–Sampled Flows

The sampling topology for threshold sampling of packet sampled flow records is illustrated in Figure 2. At the bottom are shown the individual packets, grouped in flows prior to sampling. The weight $X_{i,j}$ is the byte size of packet j in

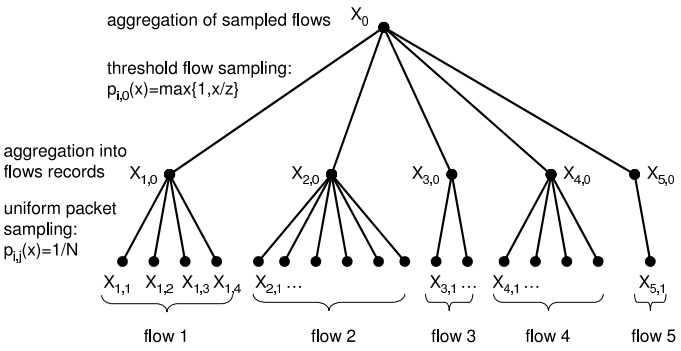

Figure 2: Illustration of topology with weights for threshold sampling of packet sampled flows.

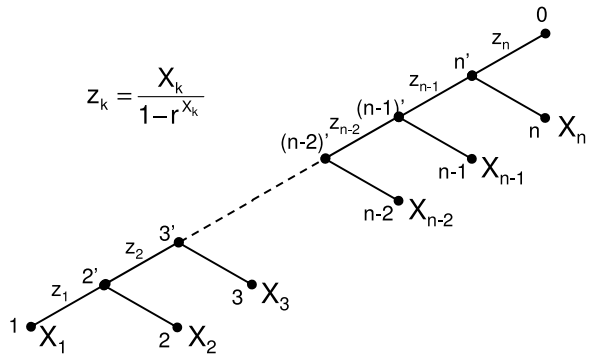

Figure 3: Sampling and Aggregation Tree for Sample and Hold. Each leaf link k corresponds to a packet of byte size X_k. Links (k', k) a trivial sampling with probability one. Links $((k+1)', k')$ are threshold sampling with threshold $z_k = X_k/p_k$ where $p_k = 1 - r^{X_k}$ the probability to sample packet k.

flow i. Each packet is sampled independently with probability $1/N$. As remarked in Section1.3(ii), packet sampling is commonly implemented as periodic or stratified. In practice, we do not expect differences between these regimes and independent sampling to significantly affect the conclusions of our analysis,; the sampling properties of flows embedded in background traffic on high speed links has been found to be largely independent of the exact method; see [12].

The packets sampled from a flow are aggregated into a flow record with estimated byte size $X_{i,0} = \sum_j' NX_{i,j}$ where $\sum'$ indicates the sum is over the random set of selected packets. Each flow record is then threshold sampled with threshold z, the results of which are aggregated at node 0. In network measurement applications, a subset of interesting flow records is usually selected based on their key; we regard Figure 2 as representing the passage of all packets in a flow matching a given key through multistage sampling and aggregation. Estimation of aggregate traffic over a certain period would involve aggregation over a number of such trees, one per flow. We do not consider this final aggregation over matching flow records in our analysis because each such tree has the same sampling parameters and so the governing threshold $\bar{\tau}$ is the same for many such trees as for one. This illustrates one strength of our approach: we do not need to know the specific sampling tree topology in advance. Although the relevant tree would depend on the flow keys of interest, we need only know the maximum threshold for sampling operations in the tree.

Using the approach described in Section 3.4 then the threshold for the packet sampling step is $NM_{\max}$, where $M_{\max}$ is the network MTU. Thus the overall threshold is

$$\bar{\tau}_0 = \max\{NM_{\max}, z\} \tag{23}$$

The form of this bound is quite interesting, since it means that the accuracy bound is actually independent of the packet sampling rate provided $NM_{\max} < z$. Likewise, it is independent of the flow sampling threshold z provided $NM_{\max} > z$.

We can also estimate the number of packets, extending to the two dimensional weights $(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)})$ representing (bytes, packets), using the same flow sampling probability $p(\boldsymbol{x}) = p_z(x^{(1)})$. Then the threshold $\tau^{(2)}$ for packet sampling is N, and for flow sampling is

$$\sup_{\boldsymbol{x}: x^{(1)} < z} x^{(2)}/p_z(x^{(1)}) = \sup_{\boldsymbol{x}: x^{(1)} < z} x^{(2)}/(x^{(1)}/z) \le z/M_{min} \tag{24}$$

where M_{min} is the minimum packet size. Thus the overall threshold for packet number estimation is

$$\bar{\tau}_0^{(2)} = \max\{N, z/M_{min}\} \tag{25}$$

We emphasize that the disjoining of the space of sampling parameters in regions where the confidence intervals depend only on one of the sampling parameters $(1/N, z)$ is specific to the confidence intervals derived from our Chernoff bounds. This does not exclude the possibility of tighter confidence intervals with a more complex dependence on the parameters.

4.2 Sample and Hold

We now render Sample and Hold as a sampling tree; see Figure 3. Leaf links k corresponds to a packet of byte size X_k. Links (k', k) have trivial sampling with probability one. Links $((k+1)', k')$ are threshold sampling with threshold $z_k = X_k/p_k$ where $p_k = 1 - r^{X_k}$ the probability to sample packet k. (Here $1 - r$ can be thought of as a per-byte sampling probability).

Following [23], the total byte weight $\bar{X}_0 = \sum_{k=1}^n X_k$ of bytes has an unbiased estimator

$$\widehat{X}_0 = \frac{X_{\widehat{k}}}{p_{\widehat{k}}} + \sum_{i=1+\widehat{k}}^{n} X_i \tag{26}$$

where $\widehat{k}$ is the index of the first selected packet. We now confirm that the sampling and aggregation tree of Figure 3 reproduces Sample and Hold with the same estimator X_0, equal to $\widehat{X}_0$ in distribution.

THEOREM 3. *(i) $X_{m'} \ge z_m$ for $\widehat{k} \le m \le n$. Hence X_0 and $\widehat{X}_0$ have the same distribution.*

(ii) The unbiased estimator X_0 for sample and hold obeys the bounds of Theorem 1 with threshold $\bar{\tau}_0 = \max_k X_k/(1 - r^{X_k})$.

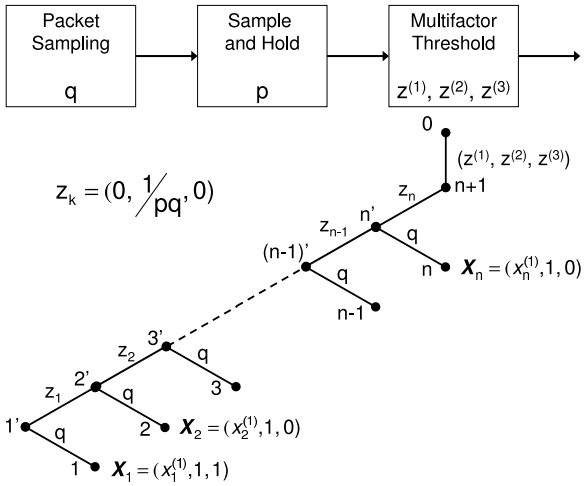

Figure 4: Flow Slicing. (Top) Schematic of Sampling Composition. (Bottom) Representation as Sampling Tree

PROOF. (i) Since no packet has been sampled before packet $\widehat{k}$, $X_{\widehat{k}'} = X_{\widehat{k}}$. Threshold sampling with threshold $z_{\widehat{k}} = X_{\widehat{k}}/p_{\widehat{k}} > X_{\widehat{k}}$ yields $\max\{z_{\widehat{k}}, X_{\widehat{k}}\} = X_{\widehat{k}}/p_{\widehat{k}} = z_{\widehat{k}'}$, the corresponding probability being $p_{z_{\widehat{k}}}(X_{\widehat{k}}) = p_{\widehat{k}}$. We now proceed by induction. Suppose $X_{m'} \geq z_m$ when $\widehat{k} < m \leq \ell - 1$. Then

$$X_{\ell'} = \frac{X_{\widehat{k}}}{p_{\widehat{k}}} + \sum_{m=\widehat{k}+1}^{\ell} X_m \geq \frac{X_{\widehat{k}}}{p_{\widehat{k}}} + X_\ell \tag{27}$$

Thus, to show that $X_{\ell'} \geq z_\ell = X_\ell/p_\ell$ it suffices to show that $\Gamma(x) \geq \Gamma(y) - y$ for any $x, y > 0$ and $r \in (0,1)$ where $\Gamma(x) = x/(1-r^x)$. This follows since $\Gamma'(x) = \gamma(q^x)$ where $\gamma(z) = (1 - z + z\log(z))/(1-z)^2$. Using the standard bound $1/z - 1 \leq \log z \leq z - 1$ we find $0 \leq \gamma(z) \leq 1$. Hence integrating the corresponding bounds $\Gamma'(x) \geq 0$ and $\Gamma'(y) - 1 \leq 0$ we find $\Gamma(x) \geq \Gamma(0^+) = -1/\log(r) \geq \Gamma(y) - y$. Applying the terminal case $\ell = n$ we find that $X_0 =^d \widehat{X}_0$, as required. The bound (ii) then follows using the maximum threshold z_k. □

The foregoing adapts to packet counting (as in Counting Samples): replace X_k/p_k with $1/p$, where p is the uniform packet sampling probability. The corresponding unbiased estimate of the number of packets, (ii) holds with $\bar{\tau}_0 = 1/p$.

4.3 Flow Slicing

Flow Slicing [23] is a multistage sampling and aggregation scheme which composes independent packet sampling, sample and hold, and threshold sampling on multidimensional flow descriptors. A prime motivation is that the use of resources in the measurement infrastructure (flow cache lookup rate, flow cache occupation, export bandwidth) can be independently controlled by adjusting the sampling parameters of the separate stages. The three-dimensional weights are $\boldsymbol{x} = (x^{(1)}, x^{(2)}, x^{(3)})$ where $x^{(1)}$ and $x^{(2)}$ are the numbers of bytes and packets in a flow, and $x^{(3)}$ is the observed number of TCP SYN packets. It is assumed that all flows are TCP flows, with only the first packet having the TCP SYN flag set. Thus for the first packet of a flow, the weight is $\boldsymbol{x} = (x^{(1)}, 1, 1)$ while for any other packet it is $\boldsymbol{x} = (x^{(1)}, 1, 0)$. Estimating the number of flows from measured sampled SYN packet was proposed in [12].

We illustrate the parameters and sampling tree for flow slicing in Figure 4. Packets are independently sampled with probability q then passed to Sample and Hold where sampling is per packet with probability p[2]. The resulting flows undergo multifactor threshold sampling with thresholds $(z^{(1)}, z^{(2)}, z^{()})$ for bytes, packets and flows, i.e., the sampling probability is $p(\boldsymbol{x}) = \min\{1, \sum_{i=1}^{3} x^{(i)}/z^{(i)}\}$. We assume that

$$\frac{M_{min}}{z_1} + \frac{1}{z_2} + \frac{1}{z_3} < 1 \tag{28}$$

since otherwise the last sampling stage is trivial, with $p(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \neq 0$.

Let $s \in \{0, 1\}$ denote a packet SYN flag. Packet sampling of a packet $(x^{(1)}, 1, s)$ yields a weight $(x^{(1)}/q, 1/q, s/q)$. Following Section 4.2 we represent Sample and Hold as threshold sampling with packet threshold $1/pq$, i.e., the size of weight to be sampled $(1/q)$ divided by the sample and hold packet sampling probability p. We can represent this as multifactor threshold sampling with thresholds $(0, 1/pq, 0)$. The verification that this sampling tree reproduced Sample and Hold is somewhat easier than in the byte sampling case treated in Section 4.2. After a first packet $\widehat{k}$ is selected by Sample and Hold, the threshold $z^{(2)} = 1/pq$ does not exceed $X^{(2)}_{j'}$ for any $j > \widehat{k}$. Hence, any subsequent packet weight that survives the initial independent packet sampling is selected by Sample and Hold with probability 1.

We now bound the overall thresholds $\bar{\boldsymbol{\tau}}_0 = (\bar{\tau}_0^{(1)}, \bar{\tau}_0^{(3)}, \bar{\tau}_0^{(3)})$ for Flow Slicing. First, following Section 2.2(ii), the thresholds $\boldsymbol{\tau}$ for the initial independent packet sampling are bounded componentwise by $(M_{\max}, 1, 1)/q$. Following Section 4.2, the thresholds for SH are bounded by $(M_{\max}, 1, 1)/(pq)$. For multidimensional flow sampling, when $p(\boldsymbol{x}) < 1$ we have the trivial bound $\boldsymbol{\tau} \leq (z^{(1)}, z^{(2)}, z^{(3)})$ from Section 2.2(iii). However, the constraints between a flow's total number of packets and bytes allow us to do better for the first two components. Using (4),

$$\begin{aligned} \frac{\boldsymbol{x}}{p(\boldsymbol{x})} &\leq \left(\frac{x^{(1)}}{\frac{x^{(1)}}{z^{(1)}} + \frac{x^{(2)}}{z^{(2)}}}, \frac{x^{(2)}}{\frac{x^{(1)}}{z^{(1)}} + \frac{x^{(2)}}{z^{(2)}}}, \frac{1}{\frac{x^{(1)}}{z^{(1)}} + \frac{x^{(2)}}{z^{(2)}} + \frac{1}{z^{(3)}}} \right) \\ &\leq \left(\frac{1}{\frac{1}{z^{(1)}} + \frac{1}{z^{(2)} M_{\max}}}, \frac{1}{\frac{M_{min}}{z^{(1)}} + \frac{1}{z^{(2)}}}, z^{(3)} \right) \end{aligned} \tag{29}$$

Summarizing, the overall byte, packet and SYN thresholds for flow slicing are:

$$\bar{\tau}_0^{(1)} = \max\{\frac{M_{\max}}{pq}, \frac{z^{(1)}}{1 + \frac{z^{(1)}}{z^{(2)} M_{\max}}}\} \tag{30}$$

$$\bar{\tau}_0^{(2)} = \max\{\frac{1}{pq}, \frac{z^{(2)}}{1 + \frac{z^{(2)} M_{min}}{z^{(1)}}}\} \tag{31}$$

$$\bar{\tau}_0^{(3)} = \max\{\frac{1}{pq}, z^{(3)}\} \tag{32}$$

Note that without the inclusion of $x^{(3)}$ in the multifactor threshold sampling probability, the effective threshold for SYN count estimation is infinite, i.e., there would be no useful bound.

[2]so actually this is Counting Samples

application	bytes	% of traffic	# flows	% flows	max flow size	average	min
all	4265677642	100.00	85680	100.00	3372865057	49786	28
ftp	3394832734	79.58	727	0.84	3372865057	4669646	40
web	80120429	1.87	7787	9.08	3139196	10289	40
mail	5387032	0.12	1495	1.74	1326756	3603	40
dns	4083277	0.09	40767	47.58	621812	100	40

Table 1: Summary statistics of flows for selected applications: ftp, web,mail, dns, and all traffic

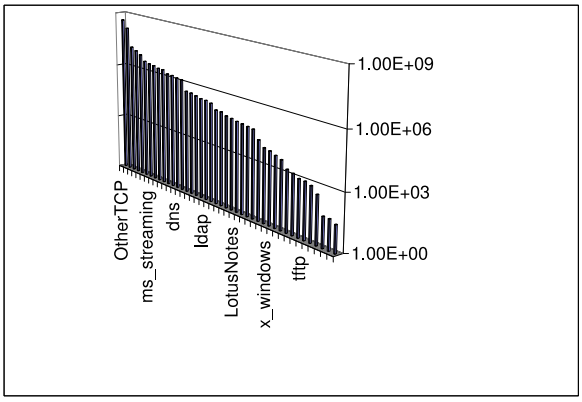

Figure 5: Flow Data: True Byte Volume by Application Class

5. EVALUATION

5.1 Threshold Sampling of Packet–Sampled Flows

We evaluate the performance of our bounds on a dataset of 85,680 flow records, collected using *unsampled* NetFlow, exported from a router. The distribution of bytes reported in the flow records was quite heavy-tailed with a single record containing 78% of the total weight. Packets were classified by application type based on TCP/UDP port number[3]. This data was used previously in the study [29], from which we reproduce a table of statistics of selected applications; see Table 1. The set of applications were chosen in order to obtain a spectrum of different statistical properties over the applications. For example, although less than 1% of the flows are for the ftp application, they represent most of the byte weight. Conversely, nearly half the flows are for dns, yet they represent less than 0.1% of the byte weight.

In contrast to [29] we now include the effects of packet sampling. We considered packet sampling rates of $1/N$ with $N = 10,\ 100$ and $1,000$ and threshold sampling with threshold $z = 5,000,\ 50,000$ and $500,000$. For each application, for each pair of parameters $(1/N, z)$ taking the above values, we performed $2,500$ independent estimates X_0 of the true byte size $\bar{X}_0 = \sum_{i,j} X_{i,j}$, the sum being over all flows i and packets j within each flow; see Figure 2.

First, we investigated conformance with confidence intervals defined in Section 3.3. The true byte volumes $\bar{X}_0$ for each application class are ordered in Figure 5. For each class, we generated the confidence intervals $X_{\pm}(\varepsilon, X_0, \bar{\tau}_0)$ for each of the 2,500 estimates X_0 of $\bar{X}_0$ in that class, using $\varepsilon = 5\%$. We then compiled statistics of violation of the limits. The proportions of runs in which $\bar{X}_0 > X_+(\varepsilon, X_0, \bar{\tau}_0)$ are displayed in Figure 6(upper); the proportions of runs in which $\bar{X}_0 < X_-(\varepsilon, X_0, \bar{\tau}_0)$ are displayed in Figure 6(lower).

For a confidence limit based on the true distribution (rather than a bound), we would expect the confidence limits to be violated in a proportion $\varepsilon = 5\%$ of the runs. In fact, the true proportion of violations is less than this in all cases, being about 3% at most. Note that in many cases there is no violation at all. Thus our exponential bound is somewhat conservative. We believe this is manageable in the sense that it leads to overestimation of estimation errors, rather than underestimation.

The results presented in Figure 6 concern the single confidence level of 5%. In order to test conformance over the bound with the observed distribution of the estimates we constructed quantile-quantile plots of the estimates against the distribution bounds. This is done as follows. For each application, we ordered the experimental estimates as $x_1 \leq x_2 \leq \ldots \leq x_{2500}$. Thus x_i is an estimate of the q_i^{th} quantile of X_0, where $q_i = (i-1)/2499$. For $q_i < 1/2$, we let q_i play the role of ε in (12), and seek a lower bound for the q_i^{th} quantile of as the largest x for which we know that

$$\mathsf{Pr}_{\bar{X}_0}[X_0 < x] \leq q_i \tag{33}$$

Thus we seek such a value y_i as the root in $[0, \bar{X}_0)$ to the equation

$$q_i = K(y_i/\bar{X}_0 - 1)^{\bar{X}_0/\bar{\tau}_0} \tag{34}$$

One can verify that $y \mapsto= \log K(y/X - 1)$ is concave on $(0, \infty)$, taking maximum value 0 at $y = X$ and approaching 1 as $x \searrow 0$. Hence when $q_i > e^{-\bar{X}_0/\bar{\tau}_0}$, (34) has a unique root y_i in $[0, \bar{X}_0)$; otherwise we take $y_i = 0$. When $q_i > 1/2$, we let q_i play the role of $1 - \varepsilon$ in the upper bound (11) and seek an upper bound y_i for the q_i^{th} quantile as the root in $(\bar{X}_0, \infty)$ to the equation

$$1 - q_i = K(y_i/\bar{X}_0 - 1)^{\bar{X}_0/\bar{\tau}_0} \tag{35}$$

As before one sees that the root is unique.

The quantile-quantile plots then use the points (x, y_i). We illustrate these bounds for a selection of applications in Figure 7; the solid vertical and horizontal lines show the true value; the line $y = x$ is also shown. The applications are ftp, www, mail, and dns. These were chosen in order to give a range or packet and flow size distributions. In all cases, we see the bound is, as expected, mostly conservative in the sense that $y_i > x_i$ for $x_i > \bar{X}_0$ and $y_i < x_i$ when $x_i < \bar{X}_0$. Some slight deviation from this rule arises for two reasons. Firstly, the empirical median is not exactly equal to the true value $\bar{X}_0$, and secondly, for clarity we have plotted only 1 in

[3]The well known problems with such classification do not concern us since we are interested only in the accuracy of byte estimation, not the semantics of application type.

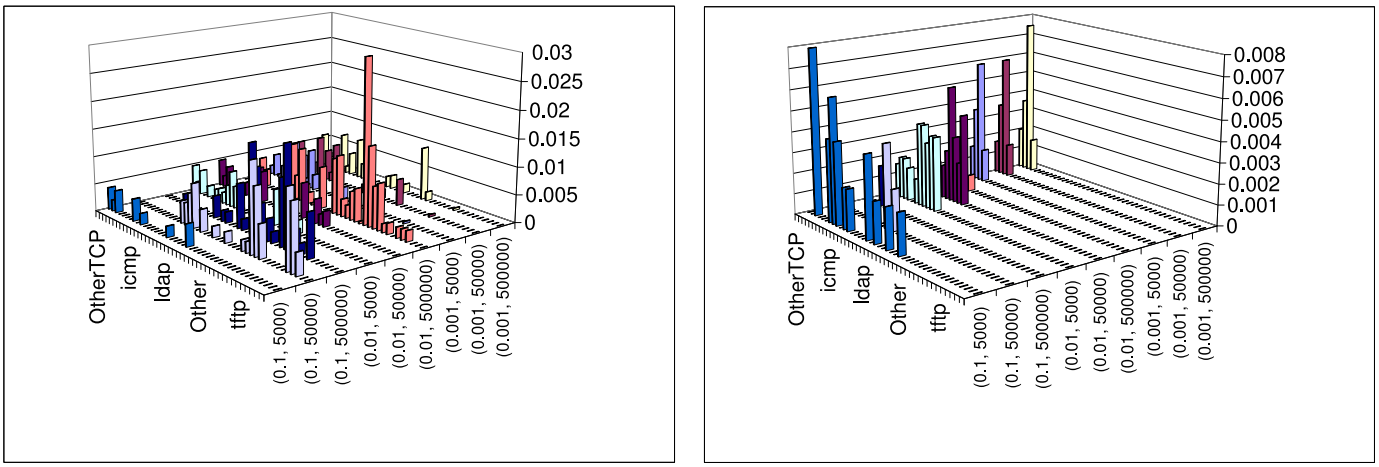

Figure 6: Threshold Sampling of Packet Sampled Flows. Performance of Confidence Intervals by Application Class, according to sampling parameters $(1/N, z)$. (Top) proportion of experiments in which 5% upper limit is violated. (Bottom) proportion of experiments in which 5% lower limit is violated. Observe that all violations occur at a rate less than 5% per class.

every 77 quantiles and hence the jump from the upper and lower bounding regimes in the plots is not exactly around the median.

More interesting is the variation according to the sampling parameters $(1/N, z)$. Recalling the MTU of 1500 bytes, then $1/N = 0.001$, the $NM > z$ for all $z \in \{5000, 50000, 5000000\}$ and hence the packet sampling error dominates the bound and correspondingly the curve for $1/N = 0.001$ roughly coincide. On the other hand, the curves for $(0.1, 500000)$ and $(0.01, 500000)$ roughly coincide, since $z > NM$ is both cases, and the flow sampling error dominates. Finally, the size of the typical error is larger for larger N and z, as expected.

6. CONCLUSIONS

This paper has addressed the problem of obtaining rigorous confidence intervals for estimates of network usage derived from multistage sampling and aggregation schemes of the type commonly in use in production communications networks. We proposed the notion of generalized threshold sampling which encompasses as examples a number of sampling schemes in use and in the literature. The power of the bound lies in its relative simplicity, being an exponential function of the maximum generalized threshold of any sampling stage. Our multistage scheme covers three important examples: sampling of NetFlow aggregates of packet samples, Sample and Hold, and Flow Slicing which we represent as a multistage sampling trees.

Future work will attempt to encompass the constrained sampling schemes, in particular priority sampling [11] and adaptive and stepped versions of NetFlow and Sample and Hold [5]. A further challenge is the incorporation of uniform sampling at nodes other than the leaf nodes; although we have provided a bound for multistage sampling with arbitrary occurrence of uniform sampling, we expect that a tighter bound is possible.

7. MULTISTAGE SAMPLING BOUNDS: MATHEMATICAL DETAIL

7.1 Bounding Functions and Their Estimates

In establishing bounds for exponential moments of $S_p(\boldsymbol{x})$ we shall employ a bounding function which captures the interpretation of $\boldsymbol{\delta}_p, \boldsymbol{\tau}_p$ as thresholds. Define $f : \mathbb{R} \times [0,\infty)^3 \to [0,\infty)$ by

$$f(\theta, x, \delta, \tau) = \begin{cases} 1 + x(e^{\theta\tau} - 1)/\tau, & x < \delta \\ e^{\theta x}, & x \geq \delta \end{cases} \tag{36}$$

and then its d-dimensional analog $h : \mathbb{R} \times [0,\infty)^{3d} \to [0,\infty)^d$:

$$h(\theta, \boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{\tau}) = (f(\theta, x^{(1)}, \delta^{(1)}, \tau^{(1)}), \ldots f(\theta, x^{(d)}, \delta^{(d)}, \tau^{(d)})) \tag{37}$$

Here we extend by continuity the function $(e^{\theta\tau} - 1)/\tau$ to the value θ as $\tau \to 0$. We will sometimes refer to the components of h as $(h^{(i)})$. Inequalities involving h will always be understood componentwise. The main interpretation of h as bounding exponential moments comes in Theorem 4(iii) below. The properties under aggregation and sampling estimation are in parts (i) and (ii) respectively; (iii) follows from (ii) as a special case.

THEOREM 4. *(i) Let* $\boldsymbol{x} = \sum_{j=1}^n \boldsymbol{x}_j \in [0,\infty)^d$ *with* $x_j^{(i)} \geq 0$. *Then for each* i

$$f(\theta, x^{(i)}, \delta^{(i)}, \tau^{(i)}) \leq \prod_{j=1}^n f(\theta, x_j^{(i)}, \delta^{(i)}, \tau^{(i)}) \tag{38}$$

and hence, componentwise in h,

$$h(\theta, \boldsymbol{x}, \boldsymbol{\delta}, \boldsymbol{\tau}) \leq \prod_{j=1}^n h(\theta, \boldsymbol{x}_j, \boldsymbol{\delta}, \boldsymbol{\tau}) \tag{39}$$

(ii) $\mathsf{E}[h(\theta, S_p(\boldsymbol{x}), \boldsymbol{\delta}, \boldsymbol{\tau})] \leq h(\theta, \boldsymbol{x}, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_p\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_p\})$, *componentwise, where the maximum is also componentwise.*

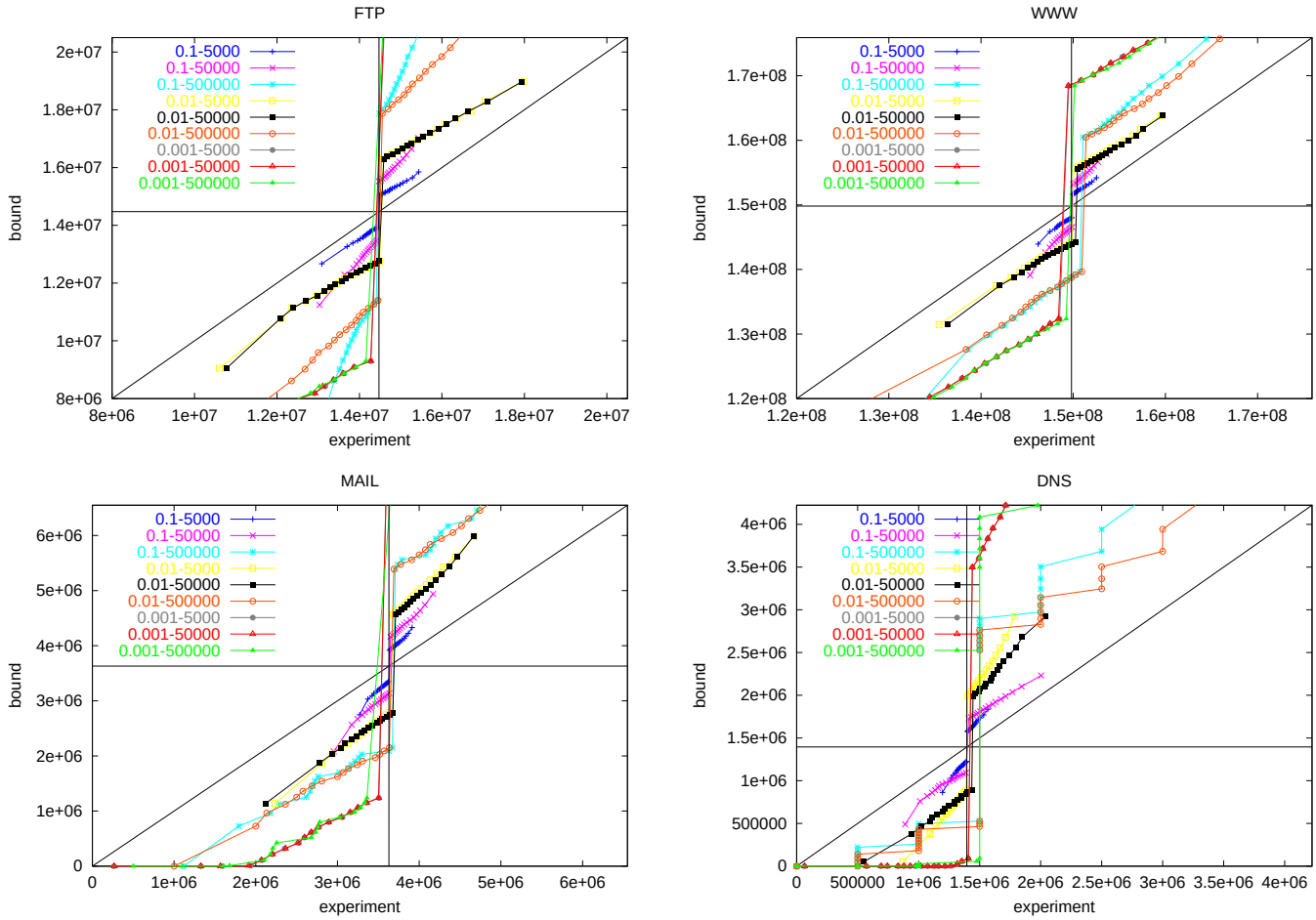

Figure 7: Quantile-Quantile plots for bounds vs. empirical distributions. Top Left: ftp. Top right: www. Bottom Left: mail. Bottom right: dns

(iii) $\mathsf{E}[\exp(\theta S_p(\boldsymbol{x}))] \le h(\theta, \boldsymbol{x}, \boldsymbol{\delta}_p, \boldsymbol{\tau}_p)$, *componentwise.*

The proof of Theorem 4 will require the following lemma:

LEMMA 1. *(i) For all* $\theta \in \mathbb{R}$, $z \to (e^{\theta z} - 1)/z$ *is nondecreasing.*

(ii) For all $\theta \in \mathbb{R}$ *and* $z, \delta, \tau \ge 0$, $e^{\theta z} \le f(\theta, z, \delta, \tau)$.

PROOF. (i) The derivative of $z \mapsto (e^{\theta z} - 1)/z$ is $(1 + e^{\theta z}(\theta z - 1))/z^2$ which is nonnegative since $e^{-y} \ge 1 - y$ for any $y \in \mathbb{R}$, which rearranges to $1 + e^y(y - 1) \ge 0$.

(ii) From definition (36) we have equality if $x \ge \delta$. Otherwise we have $x < \delta \le \tau$ and the result the follows from part (i) of this Lemma. □

PROOF OF THEOREM 4. (i) First assume $x \ge \delta$. Then $f(x) = e^{\theta x} = \prod_j e^{\theta x_j}$. The result the follows from Lemma 1(ii).

Henceforth assume $0 \le x < \delta$. Observe $1 + x(e^{\theta\tau} - 1)/\tau \le \prod_{j=1}^n (1 + x_j(e^{\theta\tau} - 1)/\tau)$. For an inductive proof of the preceding statement: assume $\{a_j : j = 1, 2, \ldots\}$ with either all $a_j > 0$ or all $a_j \in [-1, 0]$. If $\prod_{j=1}^n (1 + a_j) \ge 1 + \sum_{j=1}^n a_j$ then $\prod_{j=1}^{n+1}(1 + a_j) \ge (1 + a_{n+1})(1 + \sum_{j=1}^n a_j) = 1 + \sum_{j=1}^{n+1} a_j + a_{n+1} \sum_{j=1}^n a_j) \ge 1 + \sum_{j=1}^{n+1} a_j$.

Thus $f(\theta, x, \delta, \tau) \le \prod_{j=1}^n g(\theta, x_j, \delta, \tau, x)$, where

$$g(\theta, x_j, \delta, \tau, x) = \begin{cases} 1 + x_i(e^{\theta\tau} - 1)/z, & x < \delta \\ e^{\theta x_i}, & x \ge \delta \end{cases} \tag{40}$$

Since the $x_j \ge 0$, $x_j \ge \delta$ implies $x \ge \delta$ and hence $g(\theta, x_j, \delta, \tau) = e^{\theta x_j}$. On the other hand, if $x_j < \delta$ then $x_j \le \tau$ and by Lemma 1, $e^{\theta x_j} \le 1 + x_j(e^{\theta\tau} - 1)/\tau$. This establishes that $g(\theta, x_j, \delta, \tau, x) \le f(\theta, , x_j, \delta, \tau)$, and the result follows.

(ii) Consider the first component of $\mathsf{E}[h(\theta, S_p(\boldsymbol{x}), \boldsymbol{\delta}, \boldsymbol{\tau})]$ and for brevity denote $x = x^{(1)}$, $\delta = \delta^{(1)}$ and $\tau = \tau^{(1)}$.

$$\begin{aligned} &\mathsf{E}[h^{(1)}(\theta, S_p(\boldsymbol{x}), \boldsymbol{\delta}, \boldsymbol{\tau})] \\ &= (1 - p(\boldsymbol{x}))f(\theta, 0, \delta, \tau) + p(\boldsymbol{x})f(\theta, x/p(\boldsymbol{x}), \delta, \tau) \\ &= 1 + p(\boldsymbol{x})\left(f(\theta, x/p(\boldsymbol{x}), \delta, \tau) - 1\right) \qquad (41) \\ &= \begin{cases} 1 + x\frac{e^{\theta\tau} - 1}{\tau}, & x/p(\boldsymbol{x}) < \delta \\ 1 + p(\boldsymbol{x})(e^{\theta x/p(\boldsymbol{x})} - 1), & x/p(\boldsymbol{x}) \ge \delta \end{cases} \qquad (42) \end{aligned}$$

In the second case of (42), if $x < \delta_p$, $x/p(\boldsymbol{x}) \le \tau_p$ and so by Lemma 1 $p(\boldsymbol{x})(e^{\theta x/p(\boldsymbol{x})} - 1) \le x(e^{\theta\tau_p} - 1)/\tau_p$. On the other and, if $x \ge \delta_p$, $p(\boldsymbol{x}) = 1$ and so $1 + p(\boldsymbol{x})(e^{\theta x/p(\boldsymbol{x})} - 1) = e^{\theta x}$.

Hence

$$\begin{aligned}
&\mathsf{E}[h^{(1)}(\theta, S_p(\boldsymbol{x}), \boldsymbol{\delta}, \boldsymbol{\tau})] \\
&\leq \begin{cases} 1 + x\dfrac{e^{\theta \max\{\tau,\tau_p\}} - 1}{\max\{\tau,\tau_p\}}, & x < \max\{\delta,\delta_p\} \\ e^{\theta x}, & x \geq \max\{\delta,\delta_p\} \end{cases} && (43)\\
&= f(\theta, x, \max\{\delta,\delta_p\}, \max\{\tau,\tau_p\}) && (44)
\end{aligned}$$

(iii) follows as a special case of (ii) since $h(\theta, S_p(\boldsymbol{x}), 0, 0) = \exp(\theta S_p(\boldsymbol{x}))$. □

7.2 Bounding Exponential Moments of Sampling Processes.

When k is a descendant of j let $\boldsymbol{\tau}_{j,k} = (\tau_{j,k}^{(1)}, \ldots, \tau_{j,k}^{(d)})$ denote the componentwise maximum of the thresholds $\boldsymbol{\tau}_{k'}$ on the path from j to k, excluding $\boldsymbol{\tau}_j$, i.e.,

$$\tau_{j,k}^{(i)} = \max\{\tau_k^{(i)}, \max_{k' \in a(k) \cap d(j)} \tau_{k'}^{(i)}\}. \quad (45)$$

The thresholds $\boldsymbol{\delta}_{j,i}$ are defined similarly. Similarly to (10) we define

$$\bar{\boldsymbol{\delta}}_k = \max_{j \in d(k)} \boldsymbol{\delta}_j \quad (46)$$

Define

$$F(\theta, x, \tau) = \exp(x(e^{\theta\tau} - 1)/\tau) \quad (47)$$

Theorem 5. *(i)*

$$\begin{aligned}
&\mathsf{E}[h(\theta, \boldsymbol{X}_k, \boldsymbol{\delta}, \boldsymbol{\tau}) | \{\boldsymbol{X}_j : j \in c(k)\}] \\
&\quad \leq \prod_{j \in c(k)} h(\theta, \boldsymbol{X}_j, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_j\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_j\}) \quad (48)
\end{aligned}$$

(ii) $\mathsf{E}[h(\theta, \boldsymbol{X}_k, \boldsymbol{\delta}, \boldsymbol{\tau})] = h(\theta, X_k, \boldsymbol{\delta}, \boldsymbol{\tau})$ *if* $k \in R$, *and otherwise*

$$\begin{aligned}
&\mathsf{E}[h(\theta, \boldsymbol{X}_k, \boldsymbol{\delta}, \boldsymbol{\tau})] \\
&\quad \leq \prod_{j \in R_k} h(\theta, \boldsymbol{X}_j, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_{k,j}\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_{k,j}\}) \quad (49)
\end{aligned}$$

(iii) For each $i = \{1, \ldots, d\}$,

$$\mathsf{E}[e^{\theta X_0^{(i)}}] \leq \prod_{k \in R} f(\theta, X_k^{(i)}, \bar{\delta}_0^{(i)}, \tau_0^{(i)}) \leq F(\theta, \bar{X}_0^{(i)}, \bar{\tau}_0^{(i)}) \quad (50)$$

Proof. (i)

$$\begin{aligned}
&\mathsf{E}[h(\theta, \boldsymbol{X}_k, \boldsymbol{\delta}, \boldsymbol{\tau}) | \boldsymbol{X}_{j'}, j' \in c(k)] \\
&\quad = \mathsf{E}[h(\theta, \sum_{j \in c(k)} S_j(\boldsymbol{X}_j), \boldsymbol{\delta}, \boldsymbol{\tau}) | \boldsymbol{X}_{j'}, j' \in c(k)] && (51)\\
&\quad \leq \mathsf{E}[\prod_{j \in c(k)} h(\theta, S_j(\boldsymbol{X}_j), \boldsymbol{\delta}, \boldsymbol{\tau}) | \boldsymbol{X}_{j'}, j' \in c(k)] && (52)\\
&\quad = \prod_{j \in c(k)} \mathsf{E}[f(\theta, S_j(\boldsymbol{X}_j), \boldsymbol{\delta}, \boldsymbol{\tau}) | \boldsymbol{X}_j] && (53)\\
&\quad \leq \prod_{j \in c(k)} h(\theta, \boldsymbol{X}_j, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_j\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_j\}) && (54)
\end{aligned}$$

(51) → (52) uses Lemma 1(ii); (52) → (53) uses independence of sampling; (53) → (54) uses Theorem 4(i).

(ii) holds trivially for leaf nodes k. We establish the general case inductively. Suppose (ii) holds for all children k of a node ℓ.

$$\begin{aligned}
&\mathsf{E}[h(\theta, \boldsymbol{X}_\ell, \boldsymbol{\delta}, \boldsymbol{\tau})] \\
&\quad = \mathsf{E}[\mathsf{E}[h(\theta, \boldsymbol{X}_\ell, \boldsymbol{\delta}, \boldsymbol{\tau}) | \boldsymbol{X}_k : k \in c(\ell)]] && (55)\\
&\quad \leq \prod_{k \in c(\ell)} \mathsf{E}[h(\theta, \boldsymbol{X}_k, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_k\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_k\})] && (56)\\
&\quad \leq \prod_{k \in c(\ell)} \prod_{i \in R_k} h(\theta, \boldsymbol{X}_i, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_k, \boldsymbol{\delta}_{k,i}\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_k, \tau_{k,i}\}) && (57)\\
&\quad = \prod_{i \in R_\ell} h(\theta, \boldsymbol{X}_i, \max\{\boldsymbol{\delta}, \boldsymbol{\delta}_{\ell,i}\}, \max\{\boldsymbol{\tau}, \boldsymbol{\tau}_{\ell,i}\}) && (58)
\end{aligned}$$

(55) → (56) uses Theorem 4(ii); (56) → (57) is the assumption on $c(\ell)$; (57) → (58) is just a rearrangement.

The first inequality in (iii) is just the componentwise version of (ii) in the special case $\boldsymbol{\delta} = \boldsymbol{\tau} = 0$ since $h(\theta, \boldsymbol{x}, 0, 0) = (e^{\theta x^{(i)}})$. The second inequality (iii) then follows from Lemma 1 and the fact that for $\tau \geq 0$,

$$f(\theta, x, \delta, \tau) \leq F(\theta, x, \tau), \quad (59)$$

(extending by continuity to $\tau = 0$). This follows since neither $1 + x(e^{\theta\tau} - 1)/\tau$ nor $e^{\tau x}$ exceed $F(\theta, x, \tau)$. □

Proof of Theorem 1. It suffices to prove for the root node $k = 0$. The Chernoff Upper Bound for $X_0^{(i)}$ follows from Theorem 5(iii):

$$\begin{aligned}
&\mathsf{Pr}[X_0^{(i)} \geq (1+\sigma)\bar{X}_0^{(i)}] && (60)\\
&\quad \leq \inf_{\theta \geq 0} \mathsf{E}[e^{\theta X_0^{(i)}}] e^{-(1+\sigma)\theta \bar{X}_0^{(i)}} && (61)\\
&\quad \leq \inf_{\theta \geq 0} \exp(\bar{X}_0^{(i)}(\frac{e^{\theta \bar{\tau}_0^{(i)}} - 1}{\bar{\tau}_0^{(i)}}) - (1+\sigma)\theta) && (62)\\
&\quad = K(\sigma)^{\bar{X}_0^{(i)}/\bar{\tau}_0^{(i)}} && (63)
\end{aligned}$$

The lower bound is similar. □

8. REFERENCES

[1] N. Alon, N. Duffield, C. Lund, and M. Thorup. Estimating arbitrary subset sums with few probes. In *Proc. 24th ACM Symp. on Principles of Database Systems (PODS)*, pages 317–325, 2005.

[2] D. Brauckhoff, B. Tellenbach, A. Wagner, and M. May. Impact of traffic sampling on anomaly detection metrics. In *Proc. Internet Measuement Conference (IMC 2006)*, Rio de Janeiro, Brazil, October 25–27 2006.

[3] Cisco. White paper—netflow services and applications. http://www.cisco.com/warp/public/cc/pd/iosw /ioft/neflct/tech/napps_wp.htm.

[4] K. C. Claffy, G. C. Polyzos, and H.-W. Braun. Application of Sampling Methodologies to Network Traffic Characterization. *Computer Communication Review*, 23(4):194–203, October 1993. appeared in Proceedings ACM SIGCOMM'93, San Francisco, CA, September 13–17, 1993.

[5] E. Cohen, N. Duffield, H. Kaplan, C. Lund, and M. Thorup. Sketching unaggregated data streams for subpopulation-size queries. In *Proc. 26th ACM Symp. on Principles of Database Systems (PODS)*, 2007.

[6] E. Cohen, N. Grossaug, and H. Kaplan. Processing top-k queries from samples. In *Proceeding CoNEXT'06*, Lisbon, Portugal, December 4–6 2006.

[7] N. Duffield. (Ed.) A Framework for Packet Selection and Reporting. Internet Draft, June 2007. Work in Progress.

[8] N. Duffield and M. Grossglauser. Trajectory Sampling with Unreliable Reporting. In *INFOCOM*, 2004.

[9] N. Duffield and C. Lund. Predicting resource usage and estimation accuracy in an ip flow measurement collection infrastructure. In *ACM SIGCOMM Internet Measurement Workshop*, 2003. Miami Beach, Fl, October 27-29, 2003.

[10] N. Duffield, C. Lund, and M. Thorup. Charging from sampled network usage. In *ACM SIGCOMM Internet Measurement Workshop*, 2001. San Francisco, CA, November 1-2, 2001.

[11] N. Duffield, C. Lund, and M. Thorup. Flow sampling under hard resource constraints. In *Proc. ACM IFIP Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance)*, pages 85–96, 2004.

[12] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Trans. Netw.*, 13(5):933–946, 2005.

[13] N. Duffield, C. Lund, and M. Thorup. Optimal combination of sampled network measurements. In *Proc. 5th ACM SIGCOMM Internet Measurement Workshop (IMC)*, page to appear, 2005.

[14] N. Duffield, C. Lund, and M. Thorup. Sampling to estimate arbitrary subset sums. Technical Report cs.DS/0509026, Computing Research Repository (CoRR), 2005. `http://arxiv.org/abs/cs.DS/0509026`. Preliminary journal version of [11].

[15] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *IEEE/ACM Transactions on Networking*, 9(3):280–292, June 2001.

[16] C. Estan, K. Keys, D. Moore, and G. Varghese. Building a better netflow. In *Proceedings of the ACM SIGCOMM 04*, New York, NY, June 12–16 2004.

[17] C. Estan and G. Varghese. New directions in traffic measurement and accounting. In *Proc. ACM SIGCOMM '2002*, Pittsburgh, PA, August 2002.

[18] M. Gibbons and Y. Matias. New sampling-based summary statistics for improving approximate query answers. In *Proceedings ACM SIGMOD'98*, Seattle, Washington, June 2–4 1998.

[19] D. Horvitz and D. Thompson. A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc.*, 47, 663–685 1952.

[20] J. Jedwab, P. Phaal, and B. Pinna. Traffic estimation for the largest sources in a network, using packet sampling with limited storage. Technical Report HPL-92-3, HP Laboratories, Bristol, March 1992. url: http://www.hpl.hp.com/techreports/92/HPL-92-35.html.

[21] T. Johnson, S. Muthukrishnan, and I. Rozenbaum. Sampling algorithms in a stream operator. In *Proc. ACM SIGMOD*, pages 1–12, 2005.

[22] K. Keys, D. Moore, and C. Estan. A robust system for accurate real-time summaries of internet traffic. In *Proceedings ACM SIGMETRICS'05*, Banff, Alberta, Canada, June 6–10 2005.

[23] R. R. Kompella and C. Estan. The power of slicing in internet flow measurement. In *Proc. ACM Internet Measurement Conference*, Berkeley, CA, October 2005.

[24] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang. Is sampled data sufficient for anomaly detection? In *Proc. Internet Measuement Conference (IMC 2006)*, Rio de Janeiro, Brazil, October 25–27 2006.

[25] P. Phaal, S. Panchen, and N. McKee. Inmon corporation's sflow: A method for monitoring traffic in switched and routed networks. RFC 3176, September 2001. url:http://www.ietf.org/rfc/rfc3176.txt.

[26] J. Reeves and S. Panchen. Traffic monitoring with packet-based sampling for defense against security threats. In *Proceedings of Passive and Active Measurement Workshop (PAM 2002)*, Fort Collins, CO, USA, March 25–26 2002.

[27] M. Szegedy. The DLT priority sampling is essentially optimal. In *Proc. 38th STOC*, pages 150–158, 2006.

[28] M. Szegedy and M. Thorup. On the variance of subset sum estimation. Technical Report cs.DS/0702029, Computing Research Repository (CoRR), 2007. `http://arxiv.org/abs/cs.DS/0702029`.

[29] M. Thorup. Confidence intervals for priority sampling. In *Proc. ACM SIGMETRICS/Performance 2006*, pages 252–263, Saint-Malo, France, June 26–30 2006.

[30] J. Turian and I. D. Melamed. Computational challenges in parsing by classification. In *HLT-NAACL Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, New York, NY, 2006.

[31] T. Zseby. Deployment of sampling methods for sla validation with non-intrusive measurements. In *Proceedings of Passive and Active Measurement Workshop (PAM 2002)*, Fort Collins, CO, USA, March 25–26 2002.