Modeling Residual-Geometric Flow Sampling

Xiaoming Wang
Amazon.com
Seattle, WA 98101 USA
Email: xmwang@gmail.com

Xiaoyong Li
Texas A&M University
College Station, TX 77843 USA
Email: xiaoyong@cse.tamu.edu

Dmitri Loguinov*
Texas A&M University
College Station, TX 77843 USA
Email: dmitri@cse.tamu.edu

***Abstract*—Traffic monitoring and estimation of flow parameters in high speed routers have recently become challenging as the Internet grew in both scale and complexity. In this paper, we focus on a family of flow-size estimation algorithms we call *Residual-Geometric Sampling* (RGS), which generates a random point within each flow according to a geometric random variable and records all remaining packets in a flow counter. Our analytical investigation shows that previous estimation algorithms based on this method exhibit certain bias in recovering flow statistics from the sampled measurements. To address this problem, we derive a novel set of unbiased estimators for RGS, validate them using real Internet traces, and show that they provide an accurate and scalable solution to Internet traffic monitoring.**

I. Introduction

Recent growth of the Internet in both scale and complexity has imposed a number of challenges on network management, operation, and traffic monitoring. The main problem in this line of work is to scale measurement algorithms to achieve certain objectives (e.g., accuracy) while satisfying real-time resource constraints (e.g., fixed memory consumption and per-packet processing delay) of high-speed Internet routers. This is commonly accomplished (e.g., [5], [6], [7], [8], [9], [10], [11], [14], [15], [17], [21], [18], [19], [20], [22], [25], [30]) by reducing the amount of information a router has to store in its internal tables, which comes at the expense of deploying special estimation techniques that can recover metrics of interest from the collected samples.

In this paper, we study two problems in the general area of *measuring flow sizes* – 1) determining the number of packets transmitted by "elephant" flows [11], [15], [17], [21], [20], [22] and 2) building the distribution of flow sizes seen by the router in some time window [7], [18], [30] – coupled in a single measurement technique. The former problem arises in usage-based accounting and traffic engineering [6], [11], [12], [13], [26], while the latter has many security applications such as anomaly and intrusion detection [1], [23], [16].

Our interest falls within the family of *residual sampling*, which selects a random point A within each flow and then samples the remainder R of that flow until it ends. Denoting by L the size (in packets) of a random flow, sampled residuals R are simply $L - A$. Stochastically larger A results in fewer flows being sampled and leads to lower overhead in terms of both CPU and RAM consumption. Besides reduced overhead arising from omission of many small-size flows from counter tables, residual sampling guarantees to capture large flows with probability $1 - o(1)$ as their size $L \to \infty$. This allows ISPs to determine "heavy-hitters" and charge the corresponding customers for generated traffic.

While in P2P networks residual sampling distributes the initial point A uniformly within user lifetimes [29], flow-based estimation [11], [17] usually employs geometric A since it can be easily implemented with a sequence of independent Bernoulli variables. We call the resulting approach *Residual-Geometric Sampling* (RGS) and note that it has received some limited analytical attention in [11], [17]; however, unbiased estimation of individual flow sizes, analysis of the resulting error, asymptotically accurate recovery of flow-size distribution $P(L = i)$ from sampled residuals R, and analysis of space-CPU requirements in steady state have not been explored. We overcome these issues below.

A. Single-Flow Usage

We start with the problem of obtaining sizes of individual flows for accounting purposes. Since residual sampling requires an estimator to convert residuals into the metrics of interest, our first task is to define proper notation and desired properties for the estimation algorithm. Assume that for a flow of size L the sampling algorithm produces residual R_L, where both L and R_L are random variables. We call an estimator $e(R_L)$ *unbiased* if its expectation produces the correct flow size, i.e., $E[e(R_L)|L = l] = E[e(R_l)] = l$. Unbiased estimation allows one to average the estimated size of several flows of a given size l and accurately estimate their total contribution. We further call an estimator *elephant-accurate* if ratio $e(R_l)/l$ converges to 1 in mean-square as $l \to \infty$. Elephant-accuracy ensures that the variance of $e(R_l)/l$ tends to zero as $l \to \infty$, which means that the amount of relative error between $e(R_l)$ and l becomes negligible for large flows.

Prior work on RGS [11], [17] has suggested the following estimator:

$$e(R_L) = R_L - 1 + 1/p, \tag{1}$$

where $0 < p \leq 1$ is the parameter of geometric variable A. To understand the performance of (1), we first build a general probabilistic model for residual-geometric sampling and derive the relationship between flow size L and its residual R_L. Using this result, we prove that:

$$E[e(R_l)] = \frac{l}{1 - (1-p)^l}, \tag{2}$$

*Supported by NSF grants CNS-0720571 and CNS-1017766.

which indicates that (2) is generally biased and on average tends to overestimate the original flow size by a factor of up to $1/p$. To address this problem, we derive a different estimator:

$$\hat{e}(R_L) = R_L - 1 + 1/p - \frac{(1-p)^{R_L}}{p} \tag{3}$$

and prove that it is both unbiased and elephant-accurate. We also derive in closed-form the mean-square error $\delta_l = E[(\hat{e}(R_l)/l - 1)^2]$ for finite l, which can be used to determine when (3) approximates the true flow size with accuracy sufficient for billing purposes.

B. Flow-Size Distribution

Our second problem is estimation of the original flow-size probability mass function (PMF), which we assume is given by $f_i = P(L = i), i = 1, 2, \ldots$ We call PMF estimator q_i *asymptotically unbiased* if it converges in probability to f_i for all i as the number of sampled flows $M \to \infty$. One may be at first tempted to compute this distribution based on the values produced by either (1) or (3) for each observed flow; however, we show that such q_i almost always differ from the original distribution f_i and the bias persists as sample size $M \to \infty$. The reason for this discrepancy is that $e(.)$ and $\hat{e}(.)$ both estimate the sizes of flows *that have been sampled* by the algorithm, which are not representative of the entire population passing through the router. As longer flows are more likely to be selected by residual sampling, this approach overestimates their fraction and skews the PMF towards the tail.

Denote by M_i the number of sampled flows with $R_L = i$ and define a new estimator:

$$\tilde{q}_i = \frac{M_i - (1-p)M_{i+1}}{Mp + (1-p)M_1}. \tag{4}$$

Using the general model of RGS derived later in the paper, we prove that $\tilde{q}_i$ tends to f_i in probability as $M = \sum_i M_i \to \infty$ and obtain the amount of error $|\tilde{q}_i - f_i|$ for finite M. We also provide asymptotically unbiased estimators for the total number of flows n:

$$\tilde{n} = M + \frac{1-p}{p} M_1 \tag{5}$$

and the number of flows n_i with exactly i packets:

$$\tilde{n}_i = \frac{M_i - (1-p)M_{i+1}}{p}, \tag{6}$$

where $\tilde{n}/n \to 1$ and $\tilde{n}_i/n_i \to 1$, both in probability, as $M \to \infty$. We call the resulting combination (3)-(6) *Unbiased Residual-Geometric Estimators* (URGE).

C. Performance Evaluation

To reduce RAM overhead, our implementation periodically discards flow records if the corresponding flows have completed (i.e., FIN, RST packets detected) or if no packets from these flows arrive within some timeout τ. Unfortunately, no analytical results are available on the number of flows $M(t)$ that a router needs to track in steady state or the amount of RAM needed to keep the counters. We overcome this problem by deriving $E[M(t)]$ in equilibrium and showing that it can be orders of magnitude smaller than both the total number of flows n and the number of sampled flows M.

We finish the paper by evaluating URGE with real Internet traces obtained from NLANR [24] and CAIDA [3]. Our experiments reveal that the proposed algorithm produces very accurate estimation of flow metrics and thus allows one to perform more aggressive sampling (i.e., smaller probability p) of the monitored traffic. We also discover in experiments that URGE works very well even on short traces, which makes it suitable for monitoring small customer networks and individual protocols.

II. Related Work

In this section, we review several sampling algorithms in the area of traffic monitoring. In particular, we classify existing work into two categories: *packet sampling* and *flow sampling*, where the former makes per-packet and the latter per-flow decisions to sample incoming traffic.

A. Packet Sampling

Sampled NetFlow (SNF) [25] is a widely used technique in which incoming packets are sampled with a fixed probability p. The general goal of SNF is to obtain the PMF of flow sizes; however, [14] shows that it is impossible to accurately recover the original flow-size distribution from sampled SNF data. Estan *et al.* [10] propose *Adaptive NetFlow* (ANF), which adjusts the sampling probability p according to the size of the flow table; however, ANF's bias in the sampled data is equivalent to that in SNF and is similarly difficult to overcome in practice.

Instead of using one uniform probability for all flows as in [10], [25], another direction in packet sampling is to compute $p_i(c)$ for each flow i based on its currently observed size c. This approach has been studied by two independent papers, *Sketch-Guided Sampling* (SGS) [20] and *Adaptive Non-Linear Sampling* (ANLS) [15]. A common feature of these two methods is to sample a new flow with probability 1 and then monotonically decrease $p_i(c)$ as c grows. Both methods must maintain a counter for each flow present in the network and are difficult to scale due to the high RAM/CPU usage.

B. Flow Sampling

In *flow thinning* [14], each flow is sampled independently with probability p and then all packets in sampled flows are counted. Hohn *et al.* [14] show that flow thinning is able to accurately estimate the flow size distribution; however, this method typically misses $1 - p$ percent of elephant flows and thus does not support applications such as usage-based accounting and traffic engineering [6], [11], [12], [13], [26]. For highly skewed distributions with a few extremely large flows and many short ones (which is typical for Internet links), this method may also take a long time to converge.

To address these problems of flow thinning, Estan *et. al.* [11] introduce a size-dependent flow sampling algorithm called *Sample-and-Hold* (S&H), which is proposed to identify

elephant flows. For each packet from a new flow, the algorithm creates a flow counter with probability p; once a flow is sampled, all of its subsequent packets are then counted. It is easy to verify that S&H samples a flow with size l with probability $1-(1-p)^l$, which quickly approaches 1 as l grows. Creating a unifying analytical model for this approach and understanding the properties of samples it collects is the main topic of this paper.

Another direction of size-dependent flow sampling has been explored by Duffield *et al.* in [5], [6], [8], which present another size-dependent flow measurement method called *Smart Sampling*. Their approach selects each flow of size L with probability $p(L) = \min(1, L/z)$, where z is some constant. Since this method requires flow size L before deciding whether to sample it or not, it can only be applied off-line.

Kompella *et. al.* [17] examine a method called *Flow Slicing* (FS), which combines SNF and S&H with a variant of smart sampling. Other non-sampling methods include *exact counting* [27], [31] and *lossy counting* [18], [22], which are orthogonal to our work.

III. Underlying Model

In this section, we build a general probabilistic model of Sample-and-Hold [11] and establish the necessary analytical foundation for the results that follow. Due to limited space, omitted proofs, simulations, and various implementation details can be found in the extended technical report [28].

A. Sample-and-Hold

Consider a sequence of packets traversing a router and assume that its flow-measurement algorithm checks each packet's flow identifier x in some RAM table. If x is found in the table, the corresponding counter is incremented by 1; otherwise, with probability p a new entry for x is created in the table (with counter value 1) and with probability $1-p$ the packet is ignored.

To model this process, we first need several definitions. Assume that flow sizes are i.i.d. random variables and define *geometric age* A_L to be the number of packets discarded from the front of a flow with size L before it is sampled (see Fig. 1). Let G be a shifted geometric random variable with success probability p, i.e., $P(G=j) = (1-p)^j p$. It thus follows that A_L is simply:

$$A_L = \min(G, L). \tag{7}$$

Now define *geometric residual* R_L to be the final counter value of a flow of size L conditioned on the fact that it has been sampled (i.e., $A_L < L$):

$$R_L = L - A_L, \tag{8}$$

which is also illustrated in Fig. 1. From the perspective of traffic monitoring in this paper, geometric residual R_L is the only quantity collected during measurement and available to an estimation algorithm. Since this approach belongs to the class of residual-sampling techniques [29] and specifically uses geometric age, this paper calls S&H by a more mathematically-specific name *Residual-Geometric Sampling* (RGS).

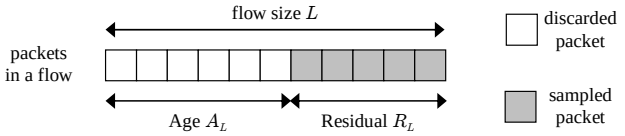


Fig. 1. Residual-geometric sampling of a flow with size L.

Assume that L has a PMF $f_i = P(L=i)$, where $i = 1, 2, \ldots$, and denote by $p_s = P(A_L < L)$ the probability that a random flow is sampled. Then, we have the following result.

Lemma 1: Probability p_s that a flow is selected by RGS is:

$$p_s = E[1-(1-p)^L] = 1 - \sum_{i=1}^{\infty} f_i (1-p)^i. \tag{9}$$

Next, let $h_i = P(R_L = i)$ be the PMF of geometric residual R_L. The following lemma expresses h_i in terms of f_i.

Lemma 2: The PMF of geometric residual R_L is:

$$h_i = \frac{p \sum_{j=i}^{\infty} f_j (1-p)^{j-i}}{p_s}. \tag{10}$$

The result of Lemma 2 is fundamental as most of the results in this paper are conveniently derived from (10).

B. Fixed Flow Size

We next analyze a special case of residual sampling where the original flow size is fixed at $L = l$. Note that residuals are now R_l instead of R_L since the original flow size is no longer a random variable. Recall that the goal of single-flow size estimation is to obtain l from R_l for each sampled flow. The next corollary follows from (10) and gives the distribution and expectation of geometric residual R_l.

Corollary 1: Given flow size $L = l$, the PMF of R_l is:

$$P(R_l = i) = \frac{(1-p)^{l-i} p}{1-(1-p)^l} \tag{11}$$

and its expectation is:

$$E[R_l] = \frac{l}{1-(1-p)^l} + 1 - 1/p. \tag{12}$$

Next, we apply the results obtained in this section to analyze existing estimation methods that have been proposed for RGS.

IV. Analysis of Existing Methods

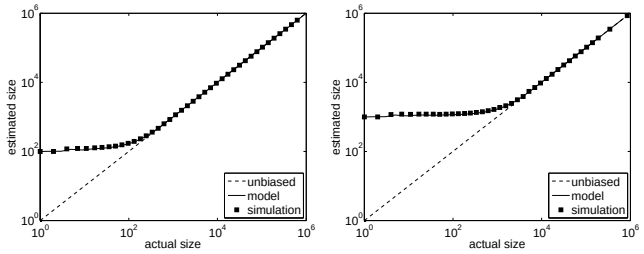
In this section, we examine prior approaches [11], [17] to estimating single-flow usage and whether their results can be generalized to recover the PMF of L.

A. Single-Flow Usage

To evaluate single-flow estimators, we use the following definition that is commonly used in statistics [2].

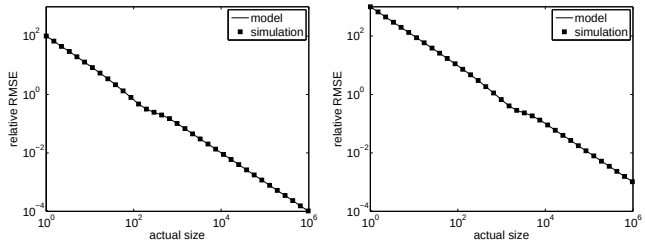
Definition 1: Estimator $e(R_l)$ is called *unbiased* if $E[e(R_l)] = l$ for all $l \geq 1$.

Unbiased estimation is a key property of an estimator as it allows accurate estimation of the total contribution from a sufficiently large pool of flows (e.g., one customer network). However, since large flows are typically rare, one commonly



(a) $p = 0.01$ (b) $p = 0.001$

Fig. 2. Expectation of estimator (13) in simulations and its model (14).



(a) $p = 0.01$ (b) $p = 0.001$

Fig. 3. RRMSE of (13) in simulations and its model (16).

faces an additional requirement to estimate their size *with just a single sample* $e(R_l)$, which is formalized in the next definition.

Definition 2: Estimator $e(R_l)$ is called *elephant-accurate* if $e(R_l)/l \to 1$ in mean-square as $l \to \infty$.

Elephant-accuracy guarantees that the amount of relative error between $e(R_l)$ and l decays to zero as $l \to \infty$. As before, suppose that a flow of size l produces a counter with value R_l. Recall that [11], [17] suggest the following estimator:

$$e(R_l) = R_l - 1 + 1/p, \tag{13}$$

where p is the probability of residual-geometric sampling. The next result directly follows from (12).

Theorem 1: Expectation $E[e(R_l)]$ is given by:

$$E[e(R_l)] = \frac{l}{1-(1-p)^l}. \tag{14}$$

Note that (14) indicates that (13) is generally biased, especially when lp is small. Indeed, for $lp \approx 0$, we have $1-(1-p)^l \approx lp$ and $E[e(R_l)] \approx 1/p$ regardless of l, which shows that in such cases $E[e(R_l)]$ *carries no information about the original flow size*. However, as $l \to \infty$, it is straightforward to verify that the bias in $e(R_l)$ vanishes exponentially, which is consistent with the analysis in [17], which has only considered the case of $l \to \infty$.

To see the extent of bias in (13) and verify (14), we apply residual-geometric sampling to flows of size l ranging from 1 to 10^6 packets, feed the measured sizes to (13), and average the result after 1000 iterations for each l. Fig. 2 plots the obtained $E[e(R_l)]$ along with model (14). The figure indicates that (14) indeed captures the bias and that (13) tends to overestimate the size of short flows *even in expectation*, where smaller sampling probability p leads to more error.

To quantify the error of individual values $e(R_l)$ in estimating flow size l and to understand elephant-accuracy, denote by $Y_l = e(R_l)/l$ and define the *Relative Root Mean Square Error* (RRMSE) to be:

$$\delta_l = \sqrt{E[(Y_l - 1)^2]}. \tag{15}$$

Note that $\delta_l \to 0$ indicates that $Y_l \to 1$ in mean-square and thus implies elephant-accurate estimation. The next result derives δ_l in closed form.

Theorem 2: The RRMSE of (13) is given by:

$$\delta_l = \sqrt{\frac{1-p-l(l-1)p^2(1-p)^l-(1-p)^{l+1}}{l^2p^2(1-(1-p)^l)}}. \tag{16}$$

Observe from (16) that for flows with size $l = 1$, the relative error is $\sqrt{1-p}/p$, but as $l \to \infty$, $\delta_l \to 0$ and the estimator is elephant-accurate. Fig. 3 plots (16) against simulations, indicating a close match. The figure also shows that the RRMSE starts from $1/p$ and decreases towards zero as $\Theta(1/l)$ as $l \to \infty$.

B. Flow-Size Distribution

We now investigate whether $e(R_L)$ defined in (13) can be used to estimate the actual flow-size distribution $\{f_i\}_{i=1}^{\infty}$. Denote by $q_i = P(e(R_L) = i)$ the PMF of estimated sizes among the sampled flows. To understand our objectives with approximating the PMF of L, the following definition is in order.

Definition 3: An estimator $\{q_i\}_{i=1}^{\infty}$ of PMF $\{f_i\}_{i=1}^{\infty}$ is called *asymptotically unbiased* if q_i converges in probability to f_i for all i as the number of sampled flows $M \to \infty$.

The next theorem follows directly from (10).

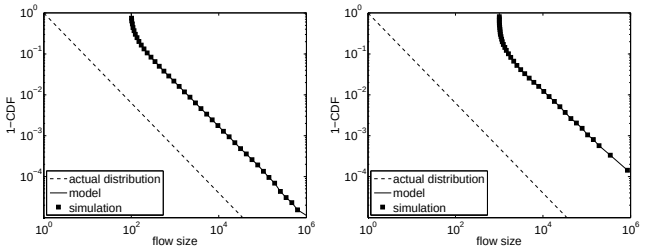
Theorem 3: The PMF of flow sizes estimated from (13) is given by:

$$q_i = \frac{\sum_{j=y(i)}^{\infty} f_j (1-p)^{j-y(i)} p}{p_s}, \tag{17}$$

where $y(i) = \lceil i + 1 - 1/p \rceil$ and p_s is in (9).

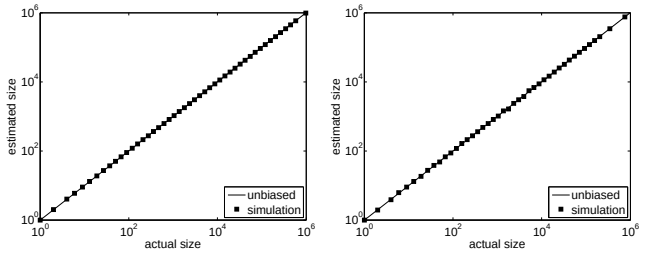
The result in (17) indicates that each q_i is different from f_i regardless of the sampling duration and thus cannot be used to approximate the flow-size distribution. We verify (17) with a simulated packet stream with 5M flows, where flow sizes follow a power-law distribution $P(L \le i) = 1 - i^{-\alpha}$ for $i = 1, 2, \ldots$ and $\alpha = 1.1$. Fig. 4 plots the CCDF of random variable $e(R_L)$ obtained from simulations as well as model (17), both in comparison to the tail of the actual distribution. The figure shows that (17) accurately predicts the values obtained from simulations and that PMF $\{q_i\}$ is indeed quite different from $\{f_i\}$.

So far, our study of existing methods in residual-geometric sampling has shown that they are not only generally biased, but also unable to recover the flow-size distribution from residuals R_L. This motivates us to seek better estimation approaches, which we perform next.



(a) $p = 0.01$ (b) $p = 0.001$

Fig. 4. Distribution $\{q_i\}$ in simulations and its model (17).



(a) $p = 0.01$ (b) $p = 0.001$

Fig. 5. Expectation of estimator (19) in simulations.

V. URGE

This section proposes a family of algorithms called *Unbiased Residual-Geometric Estimators* (URGE), proves their accuracy, and verifies them in simulations.

A. Single-Flow Usage

For estimating individual flow sizes, we first consider an estimator directly implied by the result in (12). Notice that solving (12) for l and expressing flow size l in terms of $E[R_l]$, we get:

$$l = u - \frac{1}{\log(1-p)} W\Big(u(1-p)^u \log(1-p)\Big), \qquad (18)$$

where $u = E[R_l] + 1/p - 1$ and $W(z)$ is Lambert's function (i.e., a multi-valued solution to $We^W = z$) [4]. Thus, a possible estimator can be computed from (18) with $E[R_l]$ replaced by the measured value of geometric residual R_l. However, there are two reasons that (18) is a bad estimator of flow sizes. First, Lambert's function $W(z)$ has no closed form solution and has to be numerically solved using tools such as Matlab. Second, it can be verified (not shown here for brevity) that (18) is not an unbiased estimator. Instead, we define a new estimator:

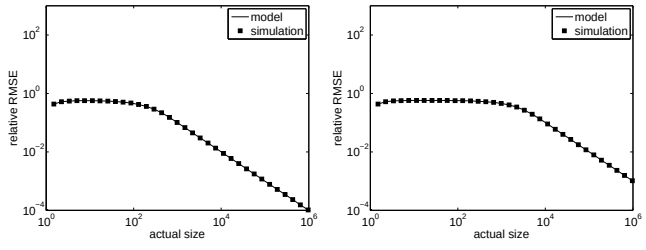
$$\hat{e}(R_l) = R_l - 1 + 1/p - \frac{(1-p)^{R_l}}{p}. \qquad (19)$$

and next show that it is unbiased.

Lemma 3: Estimator $\hat{e}(R_l)$ in (19) is unbiased, i.e.,

$$E[\hat{e}(R_l)] = l. \qquad (20)$$

We plot in Fig. 5 simulation results obtained from (19). The figure indicates that $\hat{e}(R_l)$ accurately estimates actual sizes for all flows in both cases of p. Next, we derive the RRMSE of URGE.



(a) $p = 0.01$ (b) $p = 0.001$

Fig. 6. RRMSE of (19) in simulations and model (21).

Theorem 4: The RRMSE of (19) is given by:

$$\hat{\delta}_l = \sqrt{\frac{1-p+lp(p-2)(1-p)^l - (1-p)^{2l+1}}{l^2p^2(1-(1-p)^l)}}. \qquad (21)$$

It is easy to verify from (21) that URGE has zero RRMSE for $l = 1$ or $l \to \infty$, confirming its elephant-accuracy. We plot $\hat{\delta}_l$ obtained from simulations along with the model in Fig. 6, which shows that (21) accurately tracks the actual relative error. From Figures 5-6, it is clear that $\hat{e}(R_l)$ significantly improves the accuracy of estimating small flow sizes compared to $e(R_l)$. In practice, (21) can be used to determine threshold l_0, which leads to desired bounds on error for all $l \geq l_0$ and allows ISPs to use $e(R_l)$ instead of l.

B. Flow-Size Distribution

It is worth mentioning that while (19) produces unbiased estimation of flow sizes, $\hat{e}(R_L)$ is not suitable for computing the flow-size distribution, as we show below. Denote by $\hat{q}_i = P(\hat{e}(R_L) = i)$ the PMF of $\hat{e}(R_L)$. Then, we have the following result.

Lemma 4: PMF of $\hat{e}(R_L)$ is given by:

$$\hat{q}_i = \frac{1}{p_s} \sum_{j=y(i)}^{\infty} (1-p)^{j-y(i)} f_j p, \qquad (22)$$

where p_s is in (9), function $y(i)$ is:

$$y(i) = \lceil i + 1 - 1/p - \omega \rceil, \qquad (23)$$

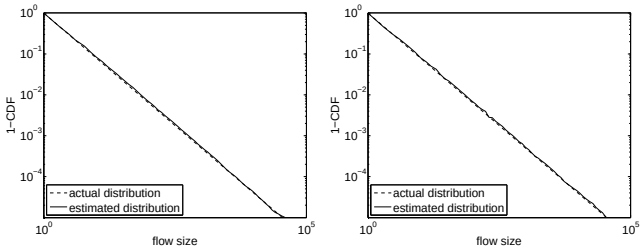
and $\omega = W\left(-(1-p)^{i+1-1/p} \log(1-p)\right)$.

Notice from (22)-(23) that distribution $\hat{q}_i$ does not even remotely approximate the original PMF f_i. This problem is fundamental since residual sampling exhibits bias towards larger flows and even if we could recover L from R_L exactly, the distribution of sampled flow sizes would not accurately approximate that of all flows passing through the router.

We thus explore another technique for estimating the flow-size distribution. Before doing that, we need the next lemma.

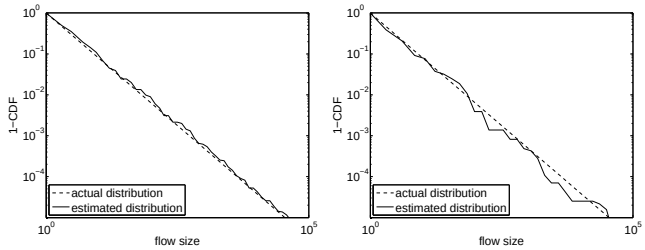
Lemma 5: The flow size distribution f_i can be expressed using the PMF of geometric residuals $\{h_i\}$ in (10) as:

$$f_i = \frac{h_i - (1-p)h_{i+1}}{p + (1-p)h_1}. \qquad (24)$$



(a) $p = 0.01$, $M = 194,208$ (b) $p = 0.001$, $M = 26,233$

Fig. 7. Estimator (25) in simulations.



(a) $p = 10^{-4}$, $M = 3,090$ (b) $p = 10^{-5}$, $M = 337$

Fig. 8. Estimator (25) in simulations with very small p.

This result leads to a new estimator for the flow-size distribution:

$$\tilde{q}_i = \frac{M_i - (1-p)M_{i+1}}{Mp + (1-p)M_1}, \tag{25}$$

where M is the total number of sampled flows and M_i is the number of them with the geometric residual equal to i. Since $M_i/M \to h_i$ in probability as $M \to \infty$ (from the weak law of large numbers), we immediately get the following result.

Corollary 2: The estimator in (25) is asymptotically unbiased.

We next verify the accuracy of $\tilde{q}_i$ in simulations with 5M flows in the same setting as in the previous section. We plot in Fig. 7 the CCDF estimated from (25) along with the actual distribution. The figure shows that $\tilde{q}_i$ accurately follows the actual distribution for both cases of p.

C. Convergence Speed

We next examine the effect of sample size M on the convergence of estimator $\tilde{q}_i$. To illustrate the problems arising from small M, we study (25) with $p = 10^{-4}$ and 10^{-5} in simulations with the same 5M flows. The estimator obtained $M = 3,090$ flows for $p = 10^{-4}$ and just $M = 337$ for $p = 10^{-5}$. Fig. 8 indicates that while the estimated curves under both choices of p still approximate the trend of the original distribution, they exhibit different levels of noise. As the next result indicates, small p leads to a small sample size M and thus more noise in the estimated values.

Corollary 3: Suppose that M flows are selected by residual-geometric sampling from a total of n flows. Then, the expected value of M is given by:

$$E[M] = np_s = nE[1 - (1-p)^L]. \tag{26}$$

To shed light on the choice of proper p for RGS, we show how to determine the minimum M that would guarantee a certain level of accuracy in $\tilde{q}_i$. Define $\tilde{h}_i = M_i/M$ to be an estimate of $h_i = P(R_L = i)$. The next lemma follows from Lemma 5 and Corollary 2 and indicates that the accuracy of $\tilde{q}_i$ directly depends on whether $\tilde{h}_i$ approximates h_i accurately.

Lemma 6: Suppose that $|\tilde{h}_j - h_j| \le \eta h_j$ holds with probability $1 - \xi$ for $j \in [1, i+1]$ and small constants η and ξ. Then, there exists a constant ζ:

$$\zeta = \frac{\eta(p + 2\eta(1-p)h_1)}{p + (1-p)(1-\eta)h_1} \tag{27}$$

such that $\zeta \to 0$ as $\eta \to 0$ and $P(|\tilde{q}_i - f_i| \le \zeta f_i) = 1 - \xi$.

Next, we obtain a bound on M from the requirement that $\tilde{h}_i$ be bounded in probability within a given range $[h_i(1-\eta), h_i(1+\eta)]$.

Theorem 5: For small constants η and ξ, $|\tilde{h}_i - h_i| \le \eta h_i$ holds with probability $1 - \xi$ if sample size M is no less than:

$$M \ge \frac{(1-h_i)}{h_i \eta^2} \left(\Phi^{-1}(1 - \xi/2)\right)^2, \tag{28}$$

where $\Phi(x)$ is the CDF of the standard Gaussian distribution $\mathcal{N}(0,1)$.

For example, to bound $\tilde{h}_i$ within 10% percent of h_i (i.e., $\eta = 0.1$) with probability $1 - \xi = 95\%$ for all $h_i \ge 10^{-2}$, the following must hold:

$$M \ge \frac{(1 - 10^{-2}) \times 1.96^2}{10^{-2} \times 0.1^2} \approx 3.8 \times 10^4, \tag{29}$$

which indicates that $M = 38$K flows must be sampled to achieve target accuracy. If we reduce η to 1%, increase $1-\xi$ to 99%, and require the approximation to hold for all $h_i \ge 10^{-3}$, then M must be at least 66M flows. Converting η into ζ using (27), one can establish similar bounds on the deviation of $\tilde{q}_i$ from f_i.

D. Estimation of Other Flow Metrics

Besides flow sizes and the flow-size distribution, URGE also provides estimators for the total number of flows and the number of them with size i. Before introducing these estimators, we need the next lemma.

Lemma 7: The expected number of flows with sampled residuals $R_L = i$ is:

$$E[M_i] = E[M]h_i = nh_i p_s, \tag{30}$$

where h_i is the PMF of geometric residuals R_L and p_s is given by (9).

Based on this, we next develop two estimators and prove their accuracy. Let $\tilde{n}$ be an estimator of the total number of flows n observed in the measurement window $[0, T]$:

$$\tilde{n} = M + \frac{(1-p)}{p} M_1 \tag{31}$$

and $\tilde{n}_i$ be an estimator of the number of flows n_i with size i:

$$\tilde{n}_i = \frac{M_i - (1-p)M_{i+1}}{p}. \tag{32}$$

Then, the next result shows that both of these estimators are asymptotically unbiased.

Lemma 8: Ratios $\tilde{n}/n$ and $\tilde{n}_i/n_i$ converge to 1 in probability as $M \to \infty$.

Note that [17] provided a similar estimator as (31) and proved $E[\tilde{n}] = n$ using a different approach from ours; however, our results are stronger as they show convergence in probability and additionally address estimation of n_i. Simulations verifying (31)-(32) are omitted for brevity.

E. Active Flows

In a typical implementation of URGE, one needs a flow table to keep a mapping between flow identifiers and associated counters. An important element of any sampling algorithm is to ensure that the table keeps only *active* flows, which can be accomplished by periodic sweeps through RAM and removal of all flows that have completed. Such a strategy together with RGS can achieve a significant reduction in the table size.

To understand how much benefit removal of dead flows provides to memory consumption, we next derive the expected number of active flows at any time t and their fraction sampled by the algorithm. Assume a measurement window $[0, T]$, where T is given in packets seen by the router. For each flow j, let inter-packet delays within the flow be given by a random variable Δ_j, which counts the number of packet arrivals from *other* flows between adjacent packets of j. Denoting by $\Delta = E[\Delta_j]$, we have the following result.

Lemma 9: Assuming stationary flow arrivals in $[0, T]$ and $T \to \infty$, the expected number of active flows $N(t)$ at time t is given by:

$$E[N(t)] = \Delta + 1. \tag{33}$$

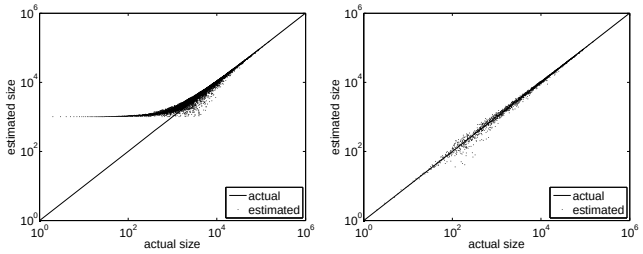
Our baseline reduction in flow volume comes from geometric sampling in previous sections and reduces the number of flows by a factor of $r_1 = n/E[M]$. Now additionally define ratio $r_2 = n/E[N(t)] = T/(\Delta + 1)E[L]$ and observe that longer observation windows (i.e., larger T), smaller flow sizes (i.e., smaller $E[L]$), and denser arrivals (i.e., smaller Δ) imply more savings of memory. In fact, $T \to \infty$ results in $r_2 \to \infty$ if the other parameters are fixed. However, even more reduction is possible by discarding dead flows in RGS. Denote by $M(t)$ the number of sampled flows that are still alive at t and consider the next result.

Lemma 10: Assuming the flow arrival process is stationary in $[0, T]$ and $T \to \infty$, the expected number of *active* sampled flows at time t is given by:

$$E[M(t)] = (\Delta + 1)\Big(1 - \frac{1-p}{pE[L]}p_s\Big), \tag{34}$$

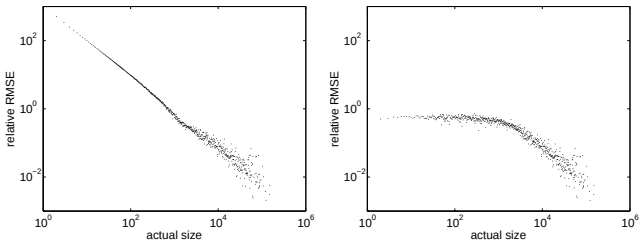
where p_s in (9) is the fraction of all flows sampled by RGS.

Define $r_3 = n/E[M(t)]$ and notice that it increases not only as T grows, but also when p decreases. Performing a self-check using Jensen's inequality, observe that $0 \le p_s/pE[L] \le 1$ and therefore $E[M(t)] \le E[N(t)]$, which means that the former indeed always results in more reduction in table size. Simulations with heavy-tailed flows (omitted due to limited space) show that the model is very accurate.



(a) $E[e(R_l)]$ (b) $E[\hat{e}(R_l)]$

Fig. 9. Estimating single-flow usage in the FRG trace with $p = 0.001$.



(a) δ_l (b) $\hat{\delta}_l$

Fig. 10. RRMSE of single-flow usage in the FRG trace with $p = 0.001$.

VI. Performance Evaluation

In this section, we evaluate our models using several Internet traces in Table I from NLANR [24] and CAIDA [3]. Trace FRG was collected from a gigabit link between UCSD and Abilene in 2006. We extracted from it additional traces with only Web, DNS, and NTP flows (also seen in the table). Additionally, we use three traces from CAIDA: LARGE – a one-hour trace from an OC48 link, MEDIUM – a one-minute trace from a OC192 link, and SMALL – a 7-minute trace from a gigabit link.

As the table shows, URGE typically sees a reasonably large number of flows M over the entire interval $[0, T]$; however, the number of active flows $N(t)$ and those constantly kept in memory $M(t)$ is much smaller. For the FRG trace, for example, $E[M]$ is 15 times smaller than n, while $E[N(t)]$ is 81 and $E[M(t)]$ is 658 times smaller. In general, NLANR traces benefit more from the removal of dead flows than CAIDA data, because former was collected over two consecutive days and thus had a larger observation window T, which led to larger ratios r_2 and r_3. The same reasoning also explains the fact that the LARGE trace exhibits much higher benefit from removing dead flows than MEDIUM or SMALL traces.

A. Estimation Accuracy

First, we examine the problem of estimating the total number of flows n in $[0, T]$ and size-one flows n_1 in this interval. The fifth and eighth columns of Table II list the absolute error of models (31) and (32), respectively. The table indicates that these estimates are commonly within 2.5% of the correct value.

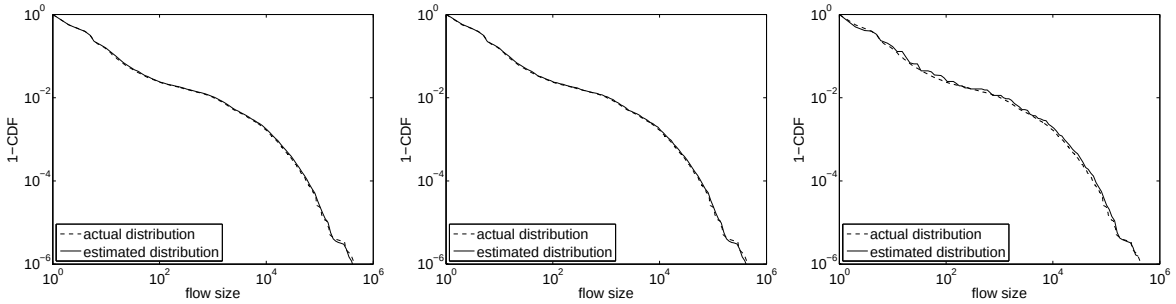
We next evaluate the performance of URGE in estimating single-flow usage. Fig. 9 plots the expectation of estimated

TABLE I
REDUCTION IN THE NUMBER OF FLOWS USING RESIDUAL SAMPLING WITH $p = 0.01$ AND DIFFERENT TYPES OF PERIODIC REMOVAL OF DEAD FLOWS

source	trace	total flows n	total pkts $nE[L]$	sampling only		removal only		both	
				$E[M]$	r_1	$E[N(t)]$	r_2	$E[M(t)]$	r_3
NLANR	FRG	$1,756,702$	$131,821,685$	$117,995$	15	$21,645$	81	$2,669$	658
	Web	$239,174$	$6,497,894$	$26,051$	9	$9,698$	24	985	240
	DNS	$120,446$	$292,977$	$2,073$	44	600	152	19	$4,797$
	NTP	$382,489$	$720,447$	$4,086$	54	$3,036$	73	77	$2,887$
CAIDA	LARGE	$9,653,609$	$117,250,415$	$519,144$	19	$262,525$	37	$21,590$	447
	MEDIUM	$2,317,369$	$43,837,666$	$139,316$	17	$281,137$	8	$53,903$	43
	SMALL	$200,910$	$2,179,574$	$12,862$	16	$44,414$	5	$5,948$	34

TABLE II
PERFORMANCE OF URGE WITH $p = 0.001$

source	trace	# of flows			# of size-one flows		
		actual (n)	estimated ($\hat{n}$)	error	actual (n_1)	estimated ($\hat{n}_1$)	error
NLANR	FRG	$1,756,702$	$1,736,261$	1.16%	768,742	749,958	2.44%
	Web	$239,174$	$253,996$	6.2%	13,686	13,922	1.72%
	DNS	$120,446$	$124,176$	3.1%	76,607	78,045	1.88%
	NTP	$382,489$	$375,326$	1.87%	281,370	279,096	0.8%
CAIDA	LARGE	$9,653,609$	$9,717,315$	0.66%	$4,535,449$	$4,630,037$	2.09%
	MEDIUM	$2,317,369$	$2,278,984$	1.66%	$1,299,343$	$1,273,989$	1.95%
	SMALL	$200,910$	$202,604$	0.84%	93,575	95,106	1.64%



(a) $p = 0.01$ (b) $p = 0.001$ (c) $p = 0.0001$

Fig. 11. Estimating the flow size distribution using URGE in the FRG trace.

flow sizes (averaged over 100 iterations) along with the actual values obtained from the FRG trace using $p = 0.001$. The figure shows that the estimator $e(R_l)$ from previous work tends to overestimate the sizes of small flows, while URGE's estimator $\hat{e}(R_l)$ accurately follows the actual values. We also compare the relative errors of the two studied methods in Fig. 10, which indicates that URGE has RRMSE bounded by 1 for all flows, while $e(R_l)$ exhibits very large δ_l for small and medium flows, which is an increasing function of $1/p$.

For the flow-size distribution, we first examine three values of p to compare its effect on the accuracy of URGE in the FRG trace. Fig. 11 indicates that estimation for all three values of p are very consistent and all of them follow the actual distribution accurately. In our experiments with $p = 0.0001$, URGE recovered the original PMF $\{f_i\}$ using only $M = 7,616$ total flows out of $n = 1.75$M.

Finally, we apply URGE with $p = 0.001$ to NLANR traces of different traffic types and plot in Fig. 12 the estimated distributions along with the actual ones. As the figure shows, the flow statistics of different applications can be accurately estimated by URGE. We observe a similar match in our experiments with three CAIDA traces as shown in Fig. 13.

VII. CONCLUSION

In this paper, we proved that previous methods based on residual-geometric sampling had certain bias in estimating single-flow usage and were unable to recover the flow-size distribution from the sampled residuals. To overcome this limitation, we proposed a novel modeling framework for analyzing residual sampling and developed a set of algorithms that were able to perform accurate estimation of flow statistics, even under the constraints of small router RAM size, short trace duration, and low CPU sampling overhead.

REFERENCES

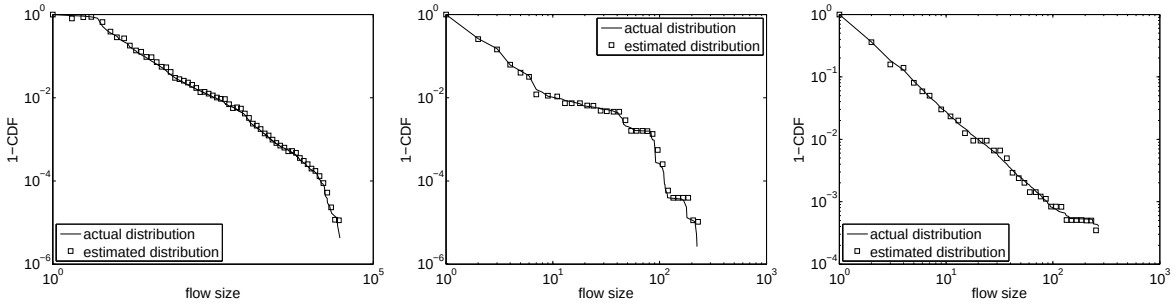
[1] D. Brauckhoff, B. Tellenbach, A. Wagner, A. Lakhina, and M. May, "Impact of Traffic Sampling on Anomaly Detection Metrics," in *Proc. ACM IMC*, Oct. 2006, pp. 159–164.

[2] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Duxbury/Thomson Learning, 2002.

[3] Cooperative Association for Internet Data Analysis (CAIDA). [Online]. Available: http://www.caida.org/.

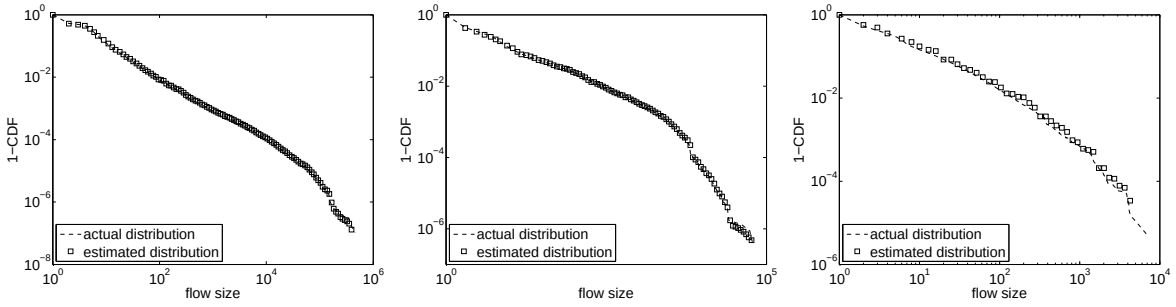
[4] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert W Function," *Advances in Computational Mathematics*, vol. 5, pp. 329–359, 1996.

[5] N. Duffield and C. Lund, "Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure," in *Proc. ACM IMC*, Oct. 2003, pp. 179–191.



(a) Web (b) NTP (c) DNS

Fig. 12. Estimating the flow size distribution using URGE in NLANR traces with $p = 0.001$.



(a) LARGE (b) MEDIUM (c) SMALL

Fig. 13. Estimating the flow size distribution using URGE in CAIDA traces with $p = 0.001$.

[6] N. Duffield, C. Lund, and M. Thorup, "Charging from Sampled Network Usage," in *Proc. ACM IMW*, Nov. 2001, pp. 245–256.

[7] N. Duffield, C. Lund, and M. Thorup, "Estimating Flow Distributions from Sampled Flow Statistics," in *Proc. ACM SIGCOMM*, Aug. 2003, pp. 325–336.

[8] N. Duffield, C. Lund, and M. Thorup, "Flow Sampling under Hard Resource Constraints," in *Proc. ACM SIGMETRICS*, Jun. 2004, pp. 85–96.

[9] N. Duffield, C. Lund, and M. Thorup, "Learn More, Sample Less: Control of Volume and Variance in Network Measurement," *IEEE Trans. Inform. Theory*, vol. 51, no. 5, pp. 1756–1775, May 2005.

[10] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a Better Netflow," in *Proc. ACM SIGCOMM*, Aug. 2004, pp. 245–256.

[11] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting," in *Proc. ACM SIGCOMM*, Aug. 2002, pp. 323–336.

[12] W. Fang and L. Peterson, "Inter-AS Traffic Patterns and their Implications," in *Proc. IEEE GLOBECOM*, Dec. 1999, pp. 1859–1868.

[13] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving Traffic Demands for Operational IP Networks: Methodology and Experience," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 265–280, Jun. 2001.

[14] N. Hohn and D. Veitch, "Inverting Sampled Traffic," in *Proc. ACM IMC*, Oct. 2003, pp. 222–233.

[15] C. Hu, S. Wang, J. Tian, B. Liu, Y. Cheng, and Y. Chen, "Accurate and Efficient Traffic Monitoring Using Adaptive Non-linear Sampling Method," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 421–429.

[16] K. Ishibashi, R. Kawahara, T. Mori, T. Kondoh, and S. Asano, "Effect of Sampling Rate and Monitoring Granularity on Anomaly Detectability," in *Proc. IEEE Global Internet Symposium*, May 2007, pp. 25–30.

[17] R. R. Kompella and C. Estan, "The Power of Slicing in Internet Flow Measurement," in *Proc. USENIX/ACM IMC*, Oct. 2005, pp. 105–118.

[18] A. Kumar, M. Sung, J. Xu, and J. Wang, "Data Streaming Algorithms for Efficient and Accurate Estimation of Flow Size Distribution," in *Proc. ACM SIGMETRICS*, Jun. 2004, pp. 177–188.

[19] A. Kumar, M. Sung, J. Xu, and E. Zegura, "A Data Streaming Algorithm for Estimating Subpopulation Flow Size Distribution," in *Proc. ACM SIGMETRICS*, Jun. 2005, pp. 61–72.

[20] A. Kumar and J. Xu, "Sketch Guided Sampling – Using On-Line Estimates of Flow Size for Adaptive Data Collection," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–11.

[21] A. Kumar, J. Xu, J. Wang, O. Spatschek, and L. Li, "Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 1762–1773.

[22] Y. Lu, A. Montanari, B.Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter Braids: A Novel Counter Architecture for Per-Flow Measurement," in *Proc. ACM SIGMETRICS*, Jun. 2008, pp. 121–132.

[23] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, "Is Sampled Data Sufficient for Anomaly Detection?" in *Proc. ACM IMC*, Oct. 2006, pp. 165–176.

[24] National Laboratory for Applied Network Research (NLANR). [Online]. Available: http://moat.nlanr.net/.

[25] Cisco Sampled NetFlow. [Online]. Available: http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12s_sanf.html.

[26] R. Pan, L. Breslau, B. Prabhakar, and S. Shenker, "Approximate Fairness through Differential Dropping," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 33, no. 2, pp. 23–39, Apr. 2003.

[27] S. Ramabhadran and G. Varghese, "Efficient Implementation of a Statistics Counter Architecture," in *Proc. ACM SIGMETRICS*, Jun. 2003, pp. 261–271.

[28] X. Wang, X. Li, and D. Loguinov, "Modeling Residual-Geometric Flow Sampling (extended version)," Texas A&M University, Tech. Rep. 2010-12-2, Dec. 2010. [Online]. Available: http://irl.cs.tamu.edu/publications/.

[29] X. Wang, Z. Yao, and D. Loguinov, "Residual-Based Measurement of Peer and Link Lifetimes in Gnutella Networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 391–399.

[30] L. Yang and M. Michailidis, "Sampled Based Estimation of Network Traffic Flow Characteristics," in *Proc. IEEE INFOCOM*, May 2007, pp. 1775–1783.

[31] Q. Zhao, J. Xu, and Z. Liu, "Design of a Novel Statistics Counter Architecture with Optimal Space and Time Efficiency," in *Proc. ACM SIGMETRICS*, Jun. 2006, pp. 323–334.