Baek-Young Choi · Zhi-Li Zhang

Adaptive Random Sampling for Traffic Volume Measurement

Abstract Traffic measurement and monitoring are an important component of network management and traffic engineering. With high-speed Internet backbone links, efficient and effective packet sampling techniques for traffic measurement and monitoring are not only desirable, but also increasingly becoming a necessity. Since the utility of sampling depends on the *accuracy* and *economy* of measurement, it is important to *control* sampling error. In this paper, we propose an *adaptive* packet sampling technique for *flow-level* traffic measurement with *stratification approach*. We employ and advance sampling theory in order to ensure the accurate estimation of large flows. With real network traces, we demonstrate that the proposed sampling technique provides unbiased estimation of flow size with *controllable error bound*, in terms of both packet and byte counts for *elephant* flows, while avoiding excessive oversampling.

Keywords Network Monitoring · Traffic Measurement · Packet Sampling · Flow

---

Baek-Young Choi
Department of Computer Science and Electrical Engineering
University of Missouri, Kansas City
5100 Rockhill Road
Kansas City, MO 64110, USA Tel.: +1-816-235-2750
Fax: +1-816-235-5159
E-mail: choiby@umkc.edu

Zhi-Li Zhang
Department of Computer Science
University of Minnesota, Twin Cities

1 Introduction

Traffic measurement and monitoring serve as the basis for a wide range of IP network operations, management and engineering tasks. Particularly, *flow-level* measurement is required for applications such as traffic profiling, usage-based accounting, traffic engineering, traffic matrix, and QoS monitoring. With today's high-speed (e.g., Gbps or Tbps) links, monitoring every packet traversing a measurement point may no longer be feasible. Because flow statistics are typically maintained by software, the processing speed cannot match the line speed. Capturing every packet requires too much *processing capacity, cache memory, and I/O and network bandwidth*, in order to *update, store, and export* flow records. Packet sampling has been suggested as a scalable alternative to address this problem. Both the Internet IETF (Internet Engineering Task Force) working groups, IPFIX (IP Flow Information Export) [14] and PSAMP (Packet Sampling) [18], have recommended the use of packet sampling. Static sampling method such as "1 out of k" is being used by Cisco and Juniper for high-speed backbone routers [16, 19].

The fundamental question regarding sampling is its *accuracy*. An inaccurate packet sampling not only defeats the purpose of traffic measurement and monitoring, but worse, can lead to wrong decisions by network operators. Particularly, when it comes to accounting, users would not make monetary commitment based on erroneous and unreliable data. Efficiency of packet sampling is also an important concern. Excessive oversampling should also be avoided for the measurement solution to be *scalable*, especially in the presence of well known high day/night traffic fluctuations (see Figure 6 for example). Therefore, it is important to *control the accuracy* of estimation in order to *balance the trade-off between the utility and overhead of measurement*. Given the dynamic nature of network traffic, *static* sampling, where a fixed sampling rate is used, does not always ensure the accuracy of estimation, and tends to oversample at peak periods when economy and timeliness are most critical.

Packet sampling for *flow-level measurement* is a particularly challenging problem. One issue is the diversity of flows; flows can vary drastically in their volumes. The dynamics of flows is another issue; flows arrive at random times and stay active for random durations. The rate of a flow (i.e., the number of packets generated by a flow per unit of time) may also vary over time, further complicating the matter of packet sampling.

How can we ensure accuracy of measurement of *dynamic* flows with a *pre-specified error bound*? How to decide on a sampling rate to avoid excessive oversampling while ensuring accuracy? How to perform sampling procedure and estimate flow volume? To answer these questions, we advance a theoretical framework and develop an *adaptive* packet sampling technique using *stratified random sampling*.

The technique is aimed for *accurate* estimation of *large* or *elephant* flows using packet sampling. That we focus only on large flows is justified by many recent studies ([2, 10, 11]) which demonstrate the prevalence of the "elephant and mice phenomenon" for flows defined at various levels of granularity: a small percentage of flows typically accounts for a large percentage of the total traffic. Therefore, for many monitoring and measurement applications, accurate estimation of flow statistics for elephant flows is often sufficient. We propose a random sampling with time stratification approach to circumvent the issues caused by flow dynamics. Through theoretical analysis, we establish the properties of the proposed adaptive stratified random sampling technique for flow-level measurement. Using real packet traffic traces, we demonstrate that the proposed technique indeed produces the desired accuracy of flow volume estimation, while at the same time achieving significant reduction in the amount of packet samples and flow cache size.

The remainder of the section is organized as follows. In Section 2 we outline related works and the challenges in packet sampling for flow measurement. In Section 3 we provide an overview of our approach and formally state the problem. In Section 4, we analyze how sampling errors can be bounded within pre-specified accuracy parameters under dynamic traffic conditions. Experimental results using network traffic traces are presented in Section 5. The paper is summarized in Section 6.

2 Related Work and Background

2.1 Related Works

Statistical sampling of network traffic was first used in [4] for measuring traffic on the NSFNET backbone in the early 1990's. Claffy *et al.* evaluated classical event and time driven *static* sampling methods to estimate statistics of distributions of packet size and inter-arrival time. Hash based sampling proposed in [5] employs the same hashing function at all links in a network, to sample same set of packets at different links, and to infer statistics on the spatial relations of the network traffic. The study in [15] presents an algorithm to bound flow packet count estimation error of the top k largest flows under a static traffic model. A size-dependent flow sampling method proposed in [6] addresses the issue of reducing the bandwidth needed for the transmission of traffic measurement to a management center for later analysis. For the purpose of usage-based charging, flows are probabilistically sampled depending on their sizes, assuming flow statistics are known a priori. In [9], a probabilistic packet sampling method is used to identify large byte count flows. Once a packet from a flow is sampled or identified, all the subsequent packets belonging to the flows are sampled. The problem of estimating flow distributions using packet sampling has been studied in [7] and [13]. Statistical inference technique and protocol level information are used to estimate the shape of the flow distribution in [7]. Authors in [13] compares packet sampling and flow sampling, and show flow sampling performs better in recovering flow distribution. Our work differs from the above, in that we use packet sampling to estimate *large* flows *accurately* under *dynamic traffic conditions*.

Table 1 Summary of traces used.

Trace	Link Speed	Duration	Avg Load
Π_1	OC3 Auck-II	4hr	152Kbps
Π_2	OC3 Tier-1 Backbone	30min	49.1Mbps
Π_3	OC12 Tier-1 Backbone	30min	43.4Mbps
Π_4	OC48 Tier-1 Backbone	30min	510.9Mbps
Π_5	OC12 Tier-1 Backbone	24hr	5.2Mbps

2.2 Background

In this paper, we present flow statistics and experimental results using flows of 5-tuple (source/destination IP addresses, port numbers and protocol number) with a $60sec$ timeout value, for illustrational consistency. Our technique, however, providing bounded accuracy in flow volume estimation, applies to *any* kind of flow definition. The packet traces used in this study are obtained from a public measurement infrastructure [17] and a commercial tier-1 ISP backbone network. The trace statistics are listed in Table 1.

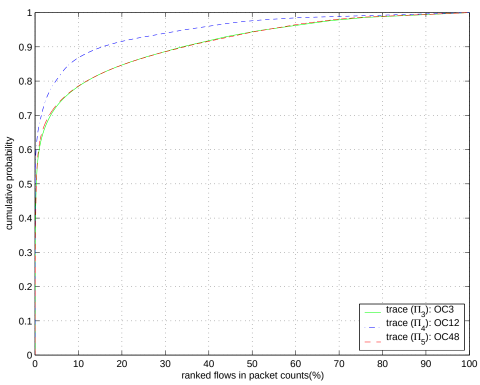

Fig. 1 Elephants-mice behavior.

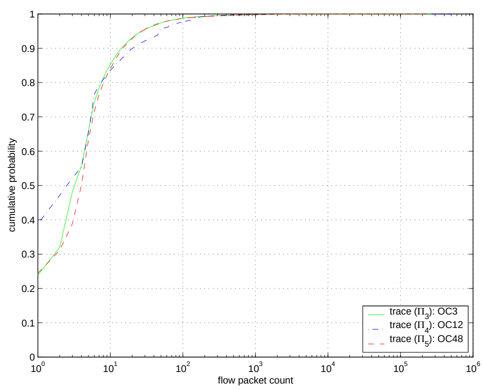

Fig. 2 Many small flows.

As a first step toward designing a sampling technique, let us observe and discuss flow characteristics and their impacts on packet sampling.

First, flows are diverse in their sizes. Note that extremely small flows (e.g., with 10 or fewer packets) may not be detected at all using any packet sampling; thus, it would be infeasible to achieve any reasonable degree of accuracy. Fortunately, for many traffic engineering applications, it is sufficient to provide an accurate estimate of flow sizes for only *large* flows. This is due to the fact that the small percentage of large flows typically accounts for a large percentage of total traffic. This is evident in Figure 1 where we order the flows based on their packet counts, and plot the cumulative probability they account for the total traffic (in terms of packet count). We see that less than $10 \sim 20\%$ of the top-ranked flows are responsible for more than 80% of the total traffic different links, which is referred as the "elephants and mice phenomenon". This motivates us to develop a packet sampling technique to *accurately* estimate *elephant* flows. Such a packet sampling technique reduces the per-packet processing overhead such as classification and flow statistics update. Meanwhile, packet sampling may also relieve the per-flow overhead. Figure 2 shows the cumulative probability distribution of flow sizes in terms of packet count (i.e., number of packets) for flows in the traces. The many of the flows (over 80% of flows) are small (e.g., with 10 or fewer packets), while a small percentage of them are large. This implies that many small flows may not be detected by packet sampling, leading to a reduction in flow cache size.

On the other hand, there are challenges involved in packet sampling for flow measurement. First, packet rate of a flow, i.e., the byte/packet counts over time, varies a lot within a flow's lifetime. This variable packet rate within a flow make it difficult to define elephant flows [2]. Furthermore, flows are dynamic in their arrival time, duration, and rate over time (the number of packets/bytes generated by a flow per unit of time); flows arrive at random time, and stay active for a random duration. furthermore, the rate of a flow varies during the flow duration. In order to achieve both sampling accuracy and efficiency at the same time, it is important to *adapt* the sampling rate according to changes in the traffic. Such problems become more acute in case of attack or long-term daily scale, where day time traffic rate differs significantly from night time as shown Figure 6. However, it is hard to decide a *sampling interval* and an optimal *sampling probability* on an interval, due to the dynamic flow arrival and duration.

3 Our Approach and Problem Formulation

In this section, we first outline our approach to tackle the challenges of flow measurement with sampling. We then state our flow measurement problem formally.

In this paper, we define a large or elephant flow in terms of a *packet count*. Thus, sampling decision will not be made based on the packet content such as packet size. The formal definition will be given later. Packet count is an important resource usage measure of a flow in a router, since many tasks in a router are done on per-packet basis such as packet classification, flow statistic update, and routing decision. Moreover, a set of flows with large packet count *contains* flows with large byte count, as is evident in Figure 3, where we observe that flows with large byte count also have large packet count. It is because a flow consists of packets whose sizes cannot be arbitrarily large. The maximum packet size is limited by MTU (Maximum Transmission Unit) on a path.[1] This suggests that a sampling scheme without using packet size captures flows with large byte count as well. Note that by relying only on packet count in the definition of elephant flows, the sampling technique is content-independent which does not incur per packet processing overhead.

[1] Note that the linear lines in Figure 3 are due to the dominant packet sizes (40, 570, and 1500 bytes).

For accuracy and efficiency of sampling, a sampling interval should be determined, for which a sampling rate is adjusted in accordance with the changing traffic condition. However, it is very hard to define a sampling interval that is valid for *all* elephant flows, due to dynamic arrival and duration of flows. We tackle this problem with a time stratification approach. As illustrated in Figure 4, we divide time into predetermined, non-overlapping intervals called *strata* or a *block*. For each block, we samples packets with the same probability (i.e., via simple random sampling). At the end of each block, flow statistics are estimated. Then, a flow's volume is naturally summarized into a single estimation record at the end of the last time block enclosing the flow. Notice that from each flow's point of view, its duration is divided or *stratified* in a fixed time. The predetermined time blocks enable us to estimate flow volume, without knowing dynamic flow arrival times and their durations.

We classify flows based on *proportion of packet count over a time interval that encompasses the flow duration*; i.e., a flow is referred to as an *elephant* flow in our study if its packet count proportion is larger than a pre-specified threshold (for example, 1% of total traffic). The proposed definition of elephant flow captures large packet and byte count flows as well as high rate (or bursty) flows. Furthermore, it removes the difficulty dealing with intra-flow rate variability. Flows with *packet count over a certain threshold* captures naturally high byte count flows as well as high packet count flows, as discussed before. The *proportion* tells us how *bursty* a flow is or rate of a (entire) flow, compared to other simultaneous flows. Since the proportion is defined over *the time interval enclosing a flow*, it eliminates the intra-flow rate fluctuation issue.

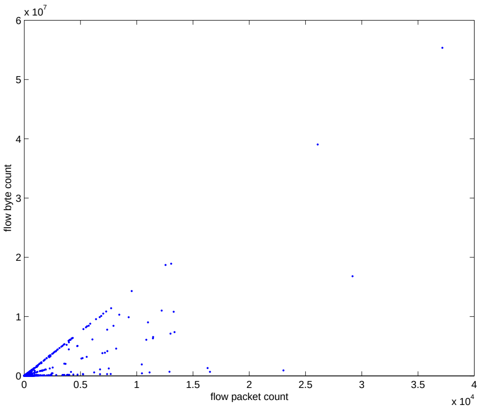

Fig. 3 Correlation of flow byte count and packet count (trace Π_1).

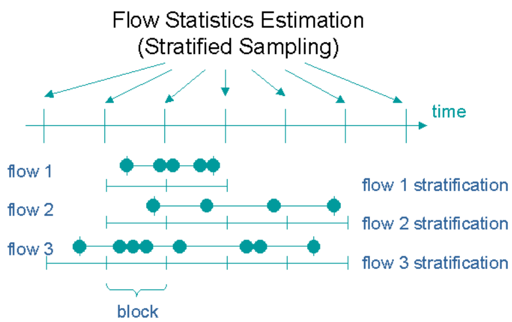

Fig. 4 Our approach.

A time block is the minimum time scale over which an elephant flow (packet count proportion) is identified. It is also the minimum time scale over which the sampling rate can be adjusted. The sampling rate of a block is set to collect the required number of samples in a block in order to bound the estimation error of the smallest (threshold) elephant flow. Given the arbitrary length of elephant flow duration, the sampling frame for a flow could be one block or a series of consecutive blocks in the stratified sampling. We prove the accuracy of flow estimation is bounded for the defined elephant flows with the proposed technique, regardless of the flow's rate variability over multiple blocks.

Below is the formal definition of an elephant flow.

Definition 1 (Elephant Flow) *Consider a quantized time interval that contains an entire duration of flow f. Suppose the interval consists of L consecutive (time) blocks where total m_i packets are seen in block i ($i = 1 \ldots L$). Let m^f packets belong to flow f out of total m packets. If the proportion of flow packet count p^f is greater than a threshold p^θ, then we call the flow an* elephant.

$$\frac{m^f}{m} = \frac{\sum_{h=1}^{L} m_h^f}{\sum_{h=1}^{L} m_h} = p^f \geq p^\theta \tag{1}$$

The online identification of a flow as an elephant or a mouse can be done by just keeping one counter of the total packet for blocks of a flow duration. When the flow expires, the packet count proportion of the flow over the total packet counts during the blocks indicates whether the flow is an elephant or not. If it is indicated as an elephant, the flow volume estimation should be accurate within the pre-specified error bound.

Our objective is to bound the relative error of packet count estimation, $\hat{m}^f$ and byte count estimation, $\hat{v}^f$ for the elephant flows, i.e., given *prescribed* error tolerance level, $\{\eta, \varepsilon\}$, (where $(1 - \eta)$ and ε are referred as *reliability* and *precision* respectively, and $0 \leq \eta \leq 1$), flow packet count and byte count estimation error have to be bounded respectively as:

$$Pr\left\{\left|\frac{\hat{m}^f - m^f}{m^f}\right| > \varepsilon\right\} \leq \eta \text{ and } Pr\left\{\left|\frac{\hat{v}^f - v^f}{v^f}\right| > \varepsilon\right\} \leq \eta \tag{2}$$

where $p^f \geq p^\theta$ for flow f. In other words, we want the relative error in flow volume estimation using random sampling to be bounded by ε with a high probability $1 - \eta$.

4 Theoretical Framework of Adaptive Random Sampling for Flow Volume Measurement

In this section, we analyze the minimum number of samples required within a time block to bound sampling errors, and describe how to determine sampling probability. We then discuss the optimal sampling probability and our prediction approach to approximate optimal sampling. Finally we describe the statistical properties of the proposed technique and how the accuracy is achieved for flows of arbitrary lengths.

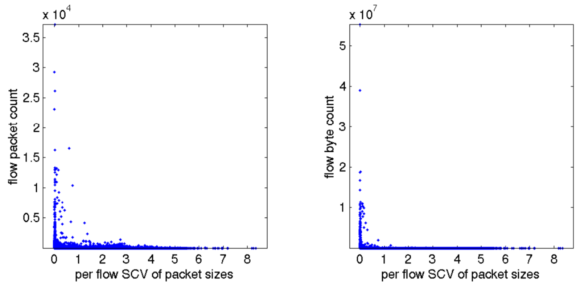

Fig. 5 Flow packet count and byte count vs. SCV.

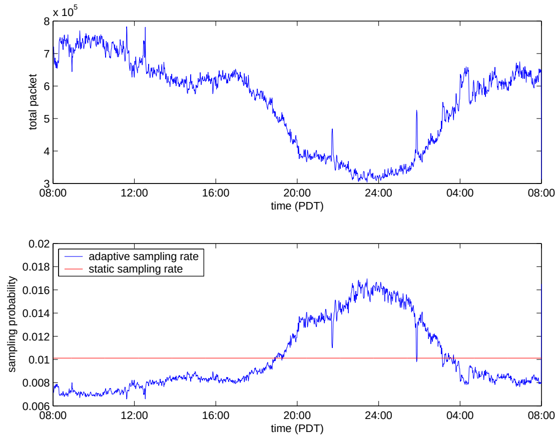

Fig. 6 Traffic load, total packet count and sampling probability (trace Π_5, $\{\eta, \varepsilon\} = \{0.1, 0.1\}$, $\{p^\theta, S^\theta\} = \{0.01, 0.2\}$).

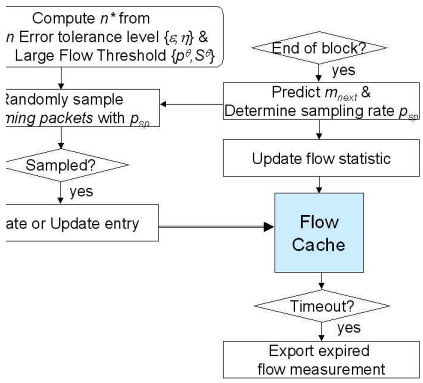

Fig. 7 Adaptive random samping for flow volume measurement.

4.1 Required Number of Samples

Our approach and analysis framework are based on random sampling. The assumptions we make in the analysis are: sample size n is reasonably large ($>$ 30 packets) and the population size m is large enough compared to the sample size ($m \gg n$) so that the sampling fraction is small. Then, the sampling distribution of the sample mean for *random samples* has a normal distribution with the mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$, *regardless of the distribution of population*, from the Central Limit Theorem. μ and σ are the population mean and the standard deviation, respectively. Recall that the requirement of samples being i.i.d (independent and identically distributed) for the condition of the theorem is simply achieved by *random* sampling from the *common* population. [2]

Using a simple random sampling, a flow packet count is estimated as follows: consider a unit time interval that contains an *entire duration* of flow f, in which m packets are seen. From these, n packets are *randomly sampled* ($n < m$), and n^f packets belong to flow f. Then the packet count of flow f, m^f is estimated by $\hat{m}^f$ using the sample proportion $\hat{p}^f$:

$$\hat{m}^f = m \cdot \frac{n^f}{n} = m \cdot \hat{p}^f \tag{3}$$

A proportion may be considered to be a special case of the mean where a variable I takes on only the values 0 and 1. For example, suppose we wish to find the proportion of a particular flow f. Let there be m packets, and let $I_i = 1$ if ith packet belongs to the flow f, and $I_i = 0$ otherwise. Then the number of packets belonging to the flow f is $m^f = \sum_{i=1}^{m} I_i$. The flow proportion of packets is computed by to the total packet count during the interval $p^f = \frac{m^f}{m} = \frac{\sum_{i=1}^{m} I_i}{m}$. Let $\hat{I}_1, \hat{I}_2, \ldots, \hat{I}_n$ be n random samples, and n^f packets of them belong to flow f. The sample proportion of flow f is therefore defined as

$$\hat{p}^f = \frac{n^f}{n} = \frac{\sum_{j=1}^{n} \hat{I}_j}{n} \tag{4}$$

Within a time block, a fixed sampling probability is used. Then, from the Central Limit Theorem of *random samples* [1], as the sample size $n \to \infty$, the sample mean $\hat{p}^f$ approaches the population mean p^f and variance $\sigma^2_{\hat{p}^f} = p^f(1-p^f)/n$ regardless of the distribution of population. Thus, the sample proportion can be written with its mean and variance,

$$\hat{p}^f \approx p^f + \frac{\sqrt{p^f(1-p^f)}}{\sqrt{n}} Y_p \tag{5}$$

where $Y_p \sim N(0,1)$, and the subscript p stands for packet count.

Now Eq. (2) can be rewritten as follows:

$$P\left\{\left|\frac{\hat{mp^f} - mp^f}{mp^f}\right| > \varepsilon\right\} = P\left\{\left|\frac{\hat{p}^f - p^f}{\sigma_{\hat{p}^f}}\right| > \frac{p^f\sqrt{n}\varepsilon}{\sqrt{p^f(1-p^f)}}\right\}$$
$$\approx 2\left(1 - \Phi\left(\frac{\sqrt{p^f}\sqrt{n}\varepsilon}{\sqrt{(1-p^f)}}\right)\right) \le \eta \tag{6}$$

[2] It is important to understand that a *randomizing eliminates correlation*. For example, in [8], the randomizing technique is used to destroy correlation for the purpose of investigating the impact of long range dependence on the queueing performance.

where $\Phi(\cdot)$ is the cumulative distribution function (c.d.f) of the standard normal distribution. By solving the inequality in Eq. (6) with respect to n, we can derive the minimum required number of samples $n^{*,p}$ to estimate flow packet count within the given error tolerance level

$$n \geq n^{*,p} = \left\lceil z_p \cdot \left(\frac{1-p^f}{p^f} \right) \right\rceil = \lceil z_p \cdot C_\theta \rceil \quad (7)$$

where $z_p = \left(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon} \right)^2$, and $C_\theta = \left(\frac{1-p^\theta}{p^\theta} \right)$ are contant. With at least $n^{*,p}$ number of random samples, simple random sampling can provide *pre-specified accuracy* $\{\eta, \varepsilon\}$ for *any* flows whose proportion is larger than a pre-defined elephant threshold p^θ. Eq. (7) concisely *relates* the *minimum number of packet samples* to the estimation *accuracy* and the *elephant flow threshold*. Moreover, given accuracy and elephant flow threshold, it shows that the amount of measurement needed remains *constant* regardless of the traffic fluctuation, which makes the technique *truely scalable* as opposed to static sampling.

4.1.1 Flow Byte Count Estimation

For the defined elephant flows, we also aim to measure flow byte count accurately, in addition to flow packet counts. The actual byte count of a flow f is expressed as $v^f = m^f \mu^f = mp^f \mu^f$, where μ^f is the actual average packet size of flow f. Similarly the estimated flow byte count $\hat{v}^f$ can be written as

$$\hat{v}^f = \hat{m}^f \hat{\mu}^f = m\hat{p}^f \hat{\mu}^f \quad (8)$$

where $\hat{\mu}^f$ is the estimated average packet size of flow f. Eq. (8) tells us that there are two levels of uncertainties are involved for flow byte count estimation, namely the estimations of flow proportion ($\hat{p}^f$) and flow's average packet size ($\hat{\mu}^f$).

The flow byte count estimation can be quantified with the help of the following two lemmas, which are the consistency of sample proportion, and an extension of the Central Limit Theorem for a sum of a random number of random variables, respectively:

Lemma 1 $\frac{n^f}{n \cdot p^f} \to 1$ *almost surely as* $n \to \infty$ *by the strong law of large numbers.*

Lemma 2 (p369, problem 27.14 in [3]) *Let* $X_1, X_2, \ldots$ *be independent, identically distributed random variables with mean* μ *and variance* σ^2*. For each positive* n*, let* F_n *be a random positive integer variable. It does not need to be independent of the* X_m*'s. Let* $W_n = \sum_{i=1}^{F_n} X_i$*. Suppose then as* $n \to \infty$*,* $\frac{F_n}{n}$ *converges to* 1 *almost surely. Then as* $n \to \infty$*,*

$$\frac{W_n - F_n \mu}{\sigma \sqrt{n}}$$

converges in distribution to a $N(0,1)$ *random variable.*

Applying these lemmas, the byte count of a flow can be approximated with the sum of two normal random variables as

$$\hat{v}^f = mp^f\mu^f + m\left[\frac{\sqrt{p^f}}{\sqrt{n}}\left(\mu^f\sqrt{1-p^f}Y_p + \sigma^f Y_b\right)\right] \tag{9}$$

where $Y_b, Y_p \sim N(0,1)$.

Then, the required number of samples for flow byte count estimation can be obtained similarly to the flow packet count estimation,

$$n \geq n^{*,b,f} = \left\lceil z_p \cdot \left(\frac{1-p^f+S^f}{p^f}\right)\right\rceil \tag{10}$$

where $S^f = (\sigma^f/\mu^f)^2$ is the *squared coefficient of variation* (SCV) of packet sizes of flow f. Eq. (10) reveals that the required number of samples for a flow byte count estimation is related to the *variability of packet sizes of a flow* as well as packet count proportion and accuracy.

Our observation shown in Figure 5 sheds light on the problem of flow byte count estimation. Even though the variability of packet sizes (SCV) of a flow ranges widely in general (from 0.00007 to 8), it is very limited for large flows. This means large flows tend to have packets of similar sizes. One can effectively give a reasonable bound on the SCV of elephant flows, around $0.2(<1)$ for example. Therefore, the number of required samples to bound estimation error for flow byte count can be obtained by

$$n \geq n^{*,b} = \lceil z_p \cdot B_\theta \rceil \tag{11}$$

where $B_\theta = \left(\frac{1-p^\theta+S^\theta}{p^\theta}\right)$.

4.2 Optimal Sampling Probability and Prediction of Total Packet Count

With the required number of samples computed, the optimal sampling probability of a block to produce n^* ($n^{*,p}$ or $n^{*,b}$) samples would be

$$p_{sp} = \frac{n^*}{m_h} \tag{12}$$

where m_h is the total number of packets in a block h. n^* can be $n^{*,p}$ only for flow packet count or $n^{*,b}$ for flow byte count as well.

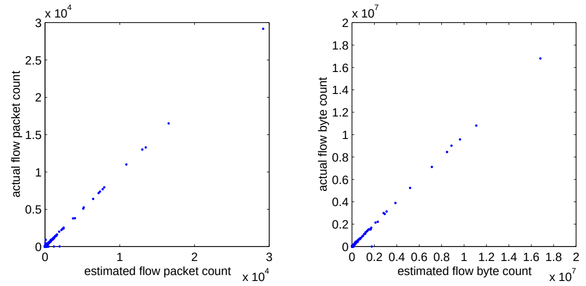

Fig. 8 Actual vs. estimated flow volume (trace $\mathit{\Pi}_1$, $\{\eta, \varepsilon\} = \{0.1, 0.1\}$).

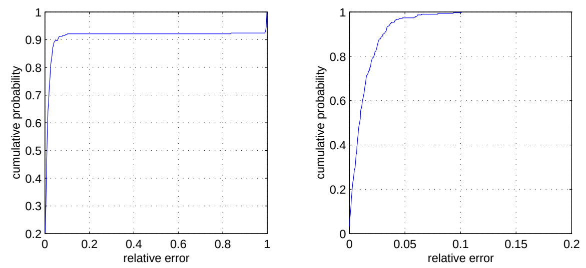

Fig. 9 Relative error of elephant flows (trace $\mathit{\Pi}_1$,$\{\eta, \varepsilon\} = \{0.1, 0.1\}$).

However, we cannot accurately choose the sampling rate when the population size (total packet count of the observation time block) is unknown. We compute the sampling probability at the beginning of a block by predicting the total packet count. We employ an the AR (Auto-Regressive) model [12] for predicting the total traffic packet count m, as compared to other time series models, since it is easier to understand and computationally more efficient making it suitable for *online implementation*.

Through empirical studies, we have found that AR(1) with a small number of past total packet count values (around 5) is sufficient to yield a good prediction. For the currently active flows, their statistics are updated using the sampling rate at the end of a block h as follows:

$$\hat{m}_h^f = \hat{m}_{h-1}^f + \frac{m_h}{n_h}\hat{n}_h^f, \; \hat{v}_h^f = \hat{v}_{h-1}^f + \frac{m_h}{n_h}\hat{n}_h^f \hat{\mu}_h^f \tag{13}$$

Figure 7 shows the flow chart of the adaptive random sampling procedure.

4.3 Accuracy of Stratified Random Sampling: Statistical Properties

Now we discuss the statistical properties of the proposed technique. Consider a flow whose enclosing interval consists of L number of blocks. Then, *from the flow's point of view*, $n^* \cdot L$ packets are sampled for the L blocks and for each block a simple random sampling is used, which is equivalent to a *stratified random sampling with an equal number of samples per stratum*. Stratified random sampling is known to provide *unbiased estimators* for the population mean, total, and proportion, in that their expectations are equal to the values of population ($E(\hat{p}^f) = p^f, E(\hat{v}^f) = v^f$). The technique is also *consistent*, since the estimation approaches the population parameter as the number of samples increases. i.e., $\hat{p}^f \to p^f$ as $n \to \infty$ (or m).

Efficiency of a sampling describes how closely a sampling distribution is concentrated around the population parameter. For consistent estimators, efficiency can be measured by the *variance*, where a smaller variance is preferred. An estimator of a smaller variance gives more *accurate* estimation, given the same number of samples. Notice that the analysis and the required number of samples are based on simple random sampling. We compare the variance of proposed stratified random sampling with simple random sampling. Due to space limitation, we simplify the discussion of the variance to a *mean* (such as $\hat{p}^f$ or $\hat{\mu}^f$) estimation. The variance of *total* estimation ($\hat{m}^f$ or $\hat{v}^f$) is easily found by using the results of a variance of *mean* estimation (e.g., $Var(\hat{m}^f) = Var(m\hat{p}^f) = m^2 Var(\hat{p}^f)$). The variance of simple random sampling with n samples is $Var(\hat{X}_{sim,n}) = \frac{\sigma^2}{n}$, where σ^2 is the population variance [20]. Thus, the accuracy (or variance) of a simple random sampling depends on the variance of the actual population (σ^2) and a sample size (n).

Now, we consider flows with arbitrary duration which stay active $L(\geq 1)$ blocks. We establish the following theorem to show that the proposed stratified random sampling provides the pre-specified error tolerance.

Lemma 3 *The variance of* stratified *random sampling with an* equal *number of* n^* *samples for each* L *strata is smaller than the variance of* simple *random sampling with* n *samples.*

$$Var(\hat{p}_{str(eq),n^*L}) \leq Var(\hat{p}_{sim,n^*}) \tag{14}$$

Proof

$$Var(\hat{p}_{str,n\cdot L}) = \frac{L}{m^2}\frac{\sum_{i=1}^{L} m_i^2\sigma_i^2}{nL} = \frac{1}{mn}\sum_{i=1}^{L} m_i\sigma_i^2 \cdot \frac{m_i}{m}$$

$$\leq \frac{1}{mn}\sum_{i=1}^{L} m_i\sigma_i^2 \leq Var(\hat{X}_{sim,n})$$

where σ_i^2 is a population variance in block i.

Therefore, the accuracy in estimation of a flow with arbitrary duration satisfies the given bound with the proposed stratified random sampling.

5 Experimental Results

In this section, we evaluate the performance of our adaptive packet sampling technique using real network traces.

We first validate the accuracy of the proposed sampling technique. In Figure 8, all the estimated flow volumes using sampled packets are compared to the actual flow volumes, to show the performance qualitatively. It can be observed that small volume flow estimations (indirectly, low proportion flows) are more off from the actual volumes, while high volume flow estimations (elephants) tend to be more close to the actual volumes.

Table 2 Sampling fraction and flow cache size reduction (Trace Π_5).

Parameters	Sampling fraction		Flow cache reduction	
$\{\eta, \varepsilon, p^\theta, S^\theta\}$	Adapt.	Static	Adapt.	Static
$\{.15, .15, .01, .0\}$	.0043	.0010	.291	.547
$\{.15, .15, .01, .2\}$	.0086	.0223	.433	.697
$\{.1, .1, .01, .0\}$	.0062	.0192	.343	.581
$\{.1, .1, .01, .2\}$	.0101	.0311	.459	.711

Figure 9 shows the cumulative probability of the relative error estimating elephant flows. There are a few flows shortened due to timeout. Their relative error are close to 1, because 1 or 2 packet flows from the broken flows are compared the original flow (left plot). However, after removing statistics of those flows (right plot), the cumulative probability of relative error being less than ε is higher than $1 - \eta$. For elephants whose threshold is higher than the threshold, the achieved accuracy is supposed to be better. Thus the statistical accuracy among *all* elephants becomes better than the specified, as in Figure 9.

Next, we examine the efficiency of the adaptive sampling in terms of reduction in packet measurement and flow cache size. A sampling fraction which is the ratio of the number of samples over the total number of packets, determines the resource usage efficiency. We compare the sampling fraction from a trace using adaptive sampling and static sampling. The average sampling rate of the adaptive sampling method is used for the sampling rate of the static sampling method. In adaptive sampling, higher accuracy requires a larger number of samples, while larger block size decreases the sampling rate. As shown in Table 5, the sampling fraction is higher for the static sampling scheme for various block sizes and accuracy parameters. Since the processing overhead is a function of the number of packets sampled, the advantage of the adaptive sampling is clear.

Table 5 also shows flow cache size reduction for both adaptive and static sampling with various traces and various sampling rates. Flow cache size reduction is computed in terms of average flow cache size in case of using sampling compared to one without sampling for the trace. Note that flow statistics should be kept for a longer time when sampling is used, since the timeout value inversely increases with sampling rate. Nontheless Table 5 illustrates that packet sampling indeed reduces flow cache size for all cases of sampling rate and traces, and the adaptive sampling performs better than static sampling.

6 Summary

In this paper, we have addressed the problem of flow volume measurement using a packet sampling. Since a small percentage of flows are observed to account for a large percentage of the total traffic, we focused on the accurate measurement of elephant flows. We proposed an adaptive sampling method that adjusts the sampling rate so as to bound the error in flow volume estimation without excessive oversampling. It enables us to control the accuracy of estimation, in turn, the tradeoff of the utility and overhead of measurement. Based on stratification approach, the proposed method divides time into strata, and within each stratum, samples packets randomly at a rate determined to collect the minimum number of samples needed to achieve the desired accuracy. Thus, sampling rates are adapted inversely to total traffic amount. Unlike static sampling where the amount of samples increases proportionally with the amount of traffic, our scheme collects a fixed number of samples on each stratum, regardless of total traffic. Therefore, the proposed scheme becomes truely scalable as opposed to static sampling. The technique can be applied to any granularity of flow definition. Through analysis and experimentation we have shown that the proposed method provides accurate and unbiased estimation of byte and packet counts of elephant flows without excessive oversampling.

Acknowledgements We thank Jaesung Park for his valuable help in the initial stage of this work.

References

1. Berry, D., Lindgren, B.: Statistics theory and Methods, 2nd edn. Duxbury Press, ITP (1996)
2. Bhattacharyya, S., Diot, C., Jetcheva, J., Taft, N.: Pop-level and access-link-level traffic dynamics in a tier-1 pop. In: Proceedings of ACM SIGCOMM Internet Measurement Workshop. San Francisco (2001)
3. Billingsley, P.: Probability Measures. Wiley-Interscience Publication (1995)
4. Claffy, K., Polyzos, G.C., Braun, H.W.: Application of sampling methodologies to network traffic characterization. In: Proceedings of ACM SIGCOMM, pp. 13–17. San Francisco, CA (1993)
5. Duffield, N., Grossglauser, M.: Trajectory sampling for direct traffic observation. In: Proceedings of ACM SIGCOMM (2000)
6. Duffield, N., Lund, C., Thorup, M.: Charging from sampled network usage. In: Proceedings of ACM SIGCOMM Internet Measurement Workshop (2001)
7. Duffield, N., Lund, C., Thorup, M.: Estimating flow distributions from sampled flow statistics. In: Proceedings of ACM SIGCOMM'03 (2003)
8. Erramilli, A., Narayan, O., Willinger, W.: Experimental queuing analysis with long-range dependent packet traffic. IEEE/ACM Transactions on Networking **4**(2), 209–223 (1996)
9. Estan, C., Varghese, G.: New directions in traffic measurement and accounting. In: Proceedings of ACM SIGCOMM (2002)
10. Fang, W., Peterson, L.: Inter-as traffic patterns and their implications. In: Proceedings of IEEE Globecom. Rio, Brazil (1999)
11. Feldmann, A., Greenberg, A., C. Lund, N.R., Rexford, J., True, F.: Deriving traffic demands for operational ip networks: Methodology and experience. IEEE/ACM Transactions on Networking pp. 265–279 (2001)
12. Gottman, J.: Time-series analysis. Cambridge University Press (1981)
13. Hohn, N., Veitch, D.: Inverting sampled traffic. In: Proceedings of ACM SIGCOMM Internet Measurement Conference (2003)
14. IPFIX: Internet Engineering Task Force, IP Flow Information Export. http://www.ietf.org/html.charters/ipfix-charter.html

15. Jedwab, J., Phaal, P., Pinna, B.: Traffic estimation for the largest sources on a network, using packet sampling with limited storage. Tech. Rep. HPL-92-35, HP Labs Technical Report (1992)
16. Juniper packet sampling. http://www.juniper.net
17. National Laboratory for Applied Network Resaerch (NLANR): PMA Traces Archive. http://moat.nlanr.net
18. PSAMP: Internet Engineering Task Force Packet Sampling working group. https://ops.ietf.org/lists/psamp
19. Cisco sampled netflow. http://www.cisco.com
20. Yamane, T.: Elementary Sampling Theory. Prentice-Hall, Inc. (1967)