Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 02-040

Adaptive Packet Sampling for Flow Volume Measurement

Baek-young Choi, Jaesung Park, and Zhi-li Zhang

December 12, 2002

Adaptive Packet Sampling for Flow Volume Measurement

Baek-Young Choi, Jaesung Park, Zhi-Li Zhang
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
E-mail: {choiby, jpark, zhzhang}@cs.umn.edu .

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 and analyze an *adaptive, stratified* random packet sampling technique for *flow-level* traffic measurement. In particular, we address the *theoretical and practical issues* involved. Through theoretical studies and experiments, 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.

I. 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 are required for applications such as traffic profiling, usage-based accounting, traffic engineering, traffic matrix, and QoS monitoring. Traditionally, every packet traversing a measurement point is captured by a router (Figure 1) while forwarding it, or by a middlebox [36] (e.g., a measurement probe) attached to a switch interface or a link. With today's high-speed (e.g., Gbps or Tbps) links, such an approach may no longer be feasible. Because flow statistics are typically maintained by software, the processing speed cannot match the *line speed*. Furthermore, the *large number of flows* observed on today's high-speed links introduces scalability issues in traffic measurement. Capturing every packet requires too much CPU capacity, cache memory, I/O and network bandwidth, for updating, storing, and exporting flow statistics 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) and PSAMP (Packet Sampling), 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 core routers ([33], [34]).

The foremost and fundamental question regarding sampling is its *accuracy*. This is especially pertinent in the Internet, where traffic is known to fluctuate dynamically and frequently. Inaccurate packet sampling not only defeats the purpose of traffic measurement and monitoring, but worse, can lead to wrong decisions by network operators. An important related concern is the efficiency of packet sampling. Excessive oversampling should also be avoided for the measurement solution to be scalable, especially in the presence of high day/night traffic fluctuations, which are well known (see Figure 8 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 fixed sampling rate is used, does not always ensure

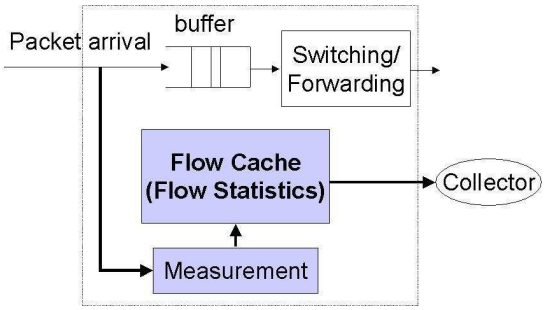


Fig. 1. Flow measurement.

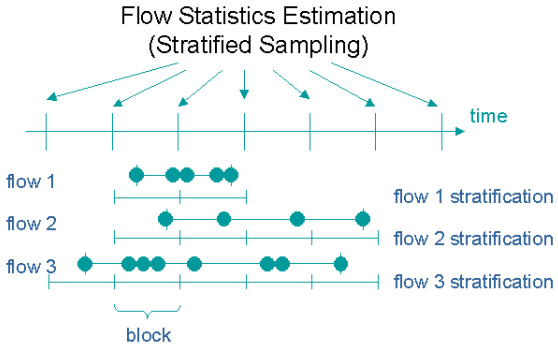


Fig. 2. Stratified random sampling.

the accuracy of estimation, and tends to oversample at peak periods when economy and timeliness are most critical [37].

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 time and stay active for a random duration; 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? How many packets does one need to sample in order to produce *flow measurement* 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? How easily can it be implemented at line speed? To answer these questions, we advance a theoretical framework and develop an *adaptive* packet sampling technique using *stratified random sampling*.

The technique is targeted for *accurate* (i.e., with *bounded* sampling errors) estimation of *large* or *elephant* flows based on sampling. That we focus only on large flows is justified by many recent studies ([19], [16], [18]) that demonstrate the prevalence of "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 employ stratified random sampling 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 network 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 paper is organized as follows. In Section II we provide an overview of the challenges in packet sampling for flow-level measurement and our proposed approaches. In Section III we formally state the flow volume estimation problem. We then analyze how sampling errors can be bounded within pre-specified accuracy parameters under dynamic traffic condition. In Section IV we discuss practical implementation issues involved. Experimental results using network traffic traces are presented in Section V. The paper is concluded in Section VI.

II. Overview: Challenges and Our Approach

A flow is a sequence of packets that share certain common properties (called *flow specification*) and have some temporal locality as observed at a given measurement point. Depending on the application and measurement

TABLE I
SUMMARY OF TRACES USED.

Name	Trace	Date	Avg Load	Duration
Π_1	OC3 Auck-II	Oct. 2001	152Kbps	4hr
Π_2	OC3 Tier-1 Backbone	Aug. 2002	49.1Mbps	30min
Π_3	OC12 Tier-1 Backbone	Aug. 2002	43.4Mbps	30min
Π_4	OC48 Tier-1 Backbone	Aug. 2002	510.9Mbps	30min
Π_5	OC12 Tier-1 Backbone	Aug. 2002	5.2Mbps	24hr
Π_6	OC12 AIX	Oct. 2001	21.6Mbps	90sec

objectives, flows may be defined in various manners such as source/destination IP addresses, port numbers, protocol or combinations thereof. They can be further grouped and aggregated into various granularity levels such as network prefixes or autonomous systems. Our analysis, providing bounded accuracy in flow volume estimation, applies to *any* kind of flow definition. For illustrational consistency, in this paper we present flow statistics and experiemental results using flows of *5-tuple* (source/destination IP addresses, port numbers and protocol number) with a $60sec$ timeout value as our basic flow definition. The 5-tuple definition is at the finest granularity using packet header traces. The traces used in this study are obtained from both public and commercial OC3, OC12, and OC48 links. The public traces are from NLANR [7] and commercial link traces are from tier-1 ISP backbone network. The trace statistics are listed in Table I.

As illustrated in Figure 1, flow measurement in routers works as follows. When a packet arrives, it is classified into a flow. If the flow state is already present in the flow cache, the corresponding flow information is updated. Otherwise, a new entry is created in the cache. When no new packet arrives within a given timeout period since the arrival of the last packet, this flow is terminated and the flow statistics are exported to a collector entity [6].

There are several challenging issues in packet sampling for flow-level measurement. In this section we provide an overview of these challenges and our proposed approach.

A. Flow Characteristics and the Impact on Packet Sampling

Clearly, flows are quite diverse in their sizes. Note that extremely small flows (e.g., with 10 or fewer packets) may not be detected at all using packet sampling, thus it would be infeasible to achieve any reasonable degree of accuracy. Figure 3 shows the cumulative probability distribution of flow sizes in terms of packet count (i.e., number of packets) for flows in the traces. The majority (80%) of the flows are small (e.g., with 10 or fewer packets), while a small percentage of them are large (e.g., with more than 10^5 or 10^6 packets).

Fortunately, for many traffic monitoring and measurement 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 4 where we order the flows based on their packet counts, and plot the cumulative probability, they account for total traffic (in terms of packet count). We see that less than 20% of the top-ranked flows are responsible for more than 80% of the total traffic different links. The aforementioned phenomenon has been referred to as the "elephants and mice phenomenon" in the literature, and has been observed at various granularities such as point-to-multipoint router level [19], network prefix level [16] and inter-AS level [18]. The observation suggests that meaningful traffic monitoring and measurement objectives (e.g., for traffic engineering or profiling) can often be achieved by concentrating only on a relatively small percentage of large (i.e., elephant) flows.

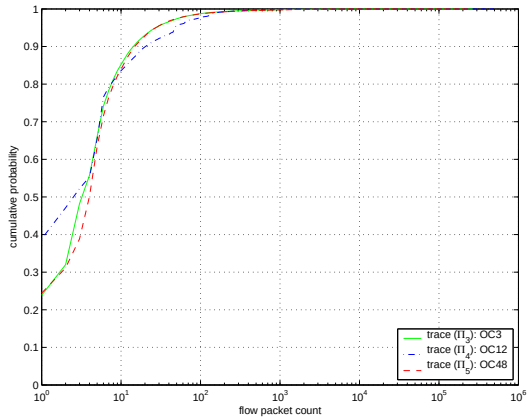


Fig. 3. Many small flows.

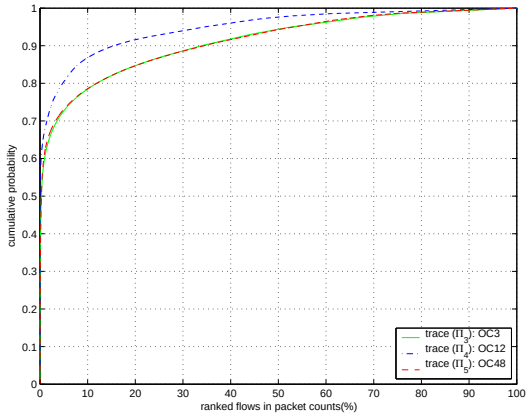


Fig. 4. Elephants-mice behavior.

This motivates us to develop a packet sampling technique to *accurately* estimate *elephant* flows. Such a technique reduces the per-packet processing overhead such as classification and flow statistics update. In addition, many small flows may not be detected by sampling, leading to a reduction in flow cache size.

An elephant flow can be defined in various ways, e.g., in terms of a packet count, a byte count (i.e., number of bytes), or even some measure of burstiness. In this paper, we define it in terms of a packet count. Packet count is an important measure of a flow, since many tasks in a router are done on per-packet basis such as packet classification, flow statistic update, and routing decision. A set of flows with large packet count contains flows with large byte count, as is evident in Figure 5, where we observe that flows with large byte count also have large packet count. This is because a flow consists of packets whose sizes cannot be arbitrarily large[1]. However, as can be seen in the figure, the opposite is not always true.[2] Figure 6 shows packet count, byte count and average packet size of top-ranked flows over time. The flows are ranked in packet count over the trace. It shows that the average packet size varies among flows. Thus flow ranks in packet count may not be equivalent to ranks in byte count. Therefore, it is important to measure also byte count of large packet count flows. Traffic variability *within* a flow is another important factor that we must consider. From Figure 6, we see that the rate of a flow (i.e., the number of packets/bytes generated by a flow per unit of time) varies during the flow duration. This gives a difficulty of defining an elephant flow. A flow's classification may change over time as its rate changes, if classified as an elephant or a mice over a fixed interval for all simultaneous flows.

Based on the above observations, we classify flows based on *proportion of packet count over a time interval that encompasses the flow duration*. The proportion tells us how bursty a flow is compared to other simultaneous flows. Since the proportion is defined over the time interval enclosing a flow, it eliminates the rate fluctuation impact. A flow is referred to as an *elephant* flow in our study if its *packet count* proportion is larger than a pre-specified threshold. (We will provide a more rigorous definition of an elephant flow in Section III.) This definition of elephant flows captures flow characteristics of packet count, byte count as well as burstiness [3].

Note that by relying only on packet count in the definition of elephant flows, the resulting sampling technique

[1] The maximum packet size is limited by MTU (Maximum Transmission Unit) on a link. Also note that the linear lines in Figure 5 are due to the dominant packet sizes ($40, 570, 1500 bytes$)

[2] As an extreme example, while the network game traffic flows tend to have large packet counts, their average byte counts are small. Empirical measurement of 'Quake' client traffic in [24] shows that the mean packet size is around only 24 bytes (standard deviation of around 1 byte). Thus the resulting byte count of such flows is not very large, while their packet count ranges from 14000 to 39000.

[3] While packet count and byte count of a flow are closely related, we observe that flow durations are less correlated to them. Due to space limitation, we do not present these results here.

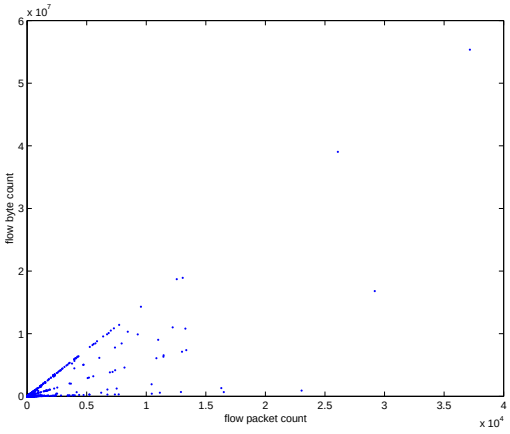


Fig. 5. Correlation of Flow byte count and packet count (trace Π_1).

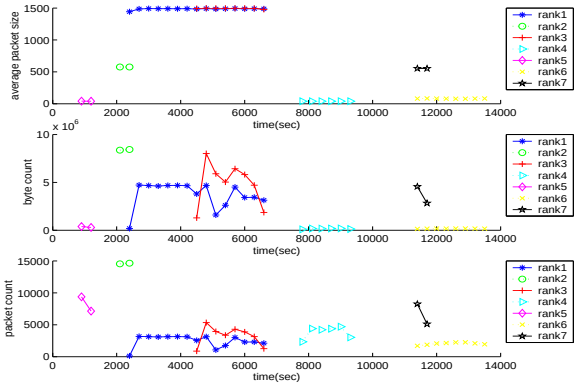


Fig. 6. Flow dynamics (ranked in packet count) (trace Π_1).

is *size-independent*. To identify flows with large byte count (or large burstiness), we only need to identify them from sampled flows with large packet counts. In contrast, *size-dependent sampling* requires computation of sampling probability for each object (either a packet or a flow) [31], [30], thus incurring additional per-object processing overhead.

B. Adaptive Stratified Random Sampling

As pointed out in [37], in order to achieve both the sampling accuracy and efficiency at the same time, it is important to *adapt* sampling rate according to changes in the traffic. Under dynamically changing traffic conditions, *static* sampling rate may lead to *inaccurate undersampling* or *excessive oversampling*. Such problems become more acute considering long term daily scale, where day time traffic rate differs significantly from night time as shown Figure 8.

Because flows are *dynamic* in their arrival time and active duration (as seen in Figure 6), it is very hard to define a sampling frame (i.e., a sampling interval) that is valid for *all* elephant flows, while allowing us to adjust the sampling rate in accordance with the changing traffic condition to ensure estimation accuracy. We tackle this problem by using *stratified* random sampling. As illustrated in Figure 2, it uses *predetermined, non-overlapping* time blocks called *strata*. For each block, it samples packets with the same probability (i.e., via simple random sampling). At the end of each block, flow statistics are estimated. Then, naturally, a flow's volume is 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 while adjusting sampling rate according to dynamical traffic changes.

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. As will be shown in Section III, in order to achieve a desired accuracy, at least a certain number of packets must be sampled during the sampling frame that encompasses an elephant flow duration. The sampling rate 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 flow's rate variability over multiple blocks.

C. Related Works

Statistical sampling of network traffic was first used in [15] 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. In [28], the authors applied a random packet sampling to evaluate the ATM end-to-end QoS such as cell transfer delay. Hash based sampling proposed in [29] employs the same hashing function at all links in a network to sample same set of packets at different links and in order to infer statistics on the spatial relations of the network traffic.

The study in [22] 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 [30] 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 [31], 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. However, by truncating preceding packets, it underestimates byte counts of flows. Furthermore, the approximation used in computing sampling probability may lead to different sampling probabilities for the same byte count flows.

Our work differs from the above, in that we use packet sampling to estimate flow volume in terms of both packet and byte counts within a *pre-specified error bound under dynamic traffic conditions*.

III. Theoretical framework for adaptive random sampling

In this section, first we formally define an elephant flow and formulate flow estimation problem. For the defined elephant flows, we analyze the minimum number of samples required using simple random sampling within a time unit in order to bound sampling errors. We then describe how to determine sampling probability and how the accuracy is achieved for flows of arbitrary lengths using stratified random sampling. Finally, we establish the statistical properties of the proposed technique.

A. Elephant Flow Definition and Problem Formulation

In this paper, a flow is referred as an elephant flow if its packet count proportion is larger than a pre-specified threshold over a flow encompassing time interval (for example, 0.1%). For those elephant flows, the proposed sampling technique estimates *flow packet count as well as byte count with controlled accuracy*.

First, we formally give a definition of an elephant flow used in this paper.

Definition 1—Elephant Flow: Consider a discretized time interval that contains an entire duration of flow f. Suppose the interval consists of L consecutive (time) blocks where 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*.

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

Determining flow rate over a certain time scale of interest is reasonable for practical issues. A straightforward computation of flow rate, i.e., flow size divided by its duration may not be meaningful, particularly for very short flows. For example, the rate for single packet flows is not well-defined since its duration is considered to be zero.

TABLE II
NOTATION.

	Explanation
m_h	total number of arriving packets in block h
n_h	total number of sampled packets in block h
p^f	proportion of flow f in packet counts
$\hat{p}^f$	estimated proportion of flow f in packet counts ($r.v.$)
n^f	number of sampled packets of flow f ($r.v.$)
m^f	total number of packets of flow f
$\hat{m}^f$	estimated packet count of flow f ($r.v.$)
v^f	byte count of flow f
$\hat{v}^f$	estimated byte count of flow f
S^f	squared coefficient of variation (SCV) of packet sizes of a flow f
p^θ	elephant flow threshold (in packet count proportion)
S^θ	threshold in SCV of elephant flow packet sizes

On the other hand, flows with two packets sent back-to-back would give the highest flow rate which is equivalent to a link rate. For long lived flows, classifying flows over a duration which might be slightly larger than its actual duration has only minimal impact on the class characteristic.

The time scale over which flows are classified can be determined by a certain engineering purpose. total traffic during a flow's of a flow as an elephant or a mouse can be done by just keeping one counter of 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 with 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:

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

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 $1-\eta$. Given this formulation of the bounded error sampling problem, the question is *what is the minimum number of packets that must be sampled randomly so as to guarantee the prescribed accuracy for diverse and dynamic flows*. We address this question in the following subsection.

We have chosen to bound relative error, since it gives generic accuracy regardless of load, link or characteristic. However, we will also discuss about bounding absolute error at the end of this section. The notations used here are summarized in Table II.

B. 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 sampling fraction is small. Then, the sampling distribution of sample mean for *random samples* has a normal distribution with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$, *regardless of the distribution of population*, from the Central Limit Theorem. μ and σ are population mean and 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. [4]

In this subsection, we derive the required number of samples to provide the pre-specified accuracy using simple random sampling. We extend it to

We explain how the accuracy is achieved for flows active over multiple blocks in the next subsection.

1) Flow Packet Count Estimation: Using a simple random sampling, a flow packet count is estimated as following: 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 Y 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 $Y_i = 1$ if ith packet belongs to the flow f, and $Y_i = 0$ otherwise. Then the number of packets belonging to the flow f is

$$m^f = \sum_{i=1}^{m} Y_i \tag{4}$$

The flow proportion of packets is computed by to the total packet count during the interval

$$p^f = \frac{m^f}{m} = \frac{\sum_{i=1}^{m} Y_i}{m} \tag{5}$$

Let $Y_1, Y_2, \ldots, Y_n$ be n random samples, and n^f packets of them belong to flow f. The sample proportion of flow f is therefore defined as

$$\hat{p}^f = \frac{n^f}{n} = \frac{\sum_{j=1}^{n} Y_j}{n} \tag{6}$$

Within a time block, a simple random sampling in which a sampling probability is fixed is used. Then, from the Central Limit Theorem of *random samples* [5], 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,

$$\hat{p}^f \approx p^f + \frac{\sqrt{p^f(1-p^f)}}{\sqrt{n}} Y_p \tag{7}$$

where Y_p is a random number of a standard normal distribution ($\sim N(0,1)$) and the subscript p stands for packet count.

[4] It is important to understand that a *randomizing eliminates correlation.* For example in [35], randomizing technique is used to destroy correlation for the purpose of investigating the impact of long range dependence on the queueing performance.

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

$$Pr\left\{\left|\frac{\hat{mp^f}-mp^f}{mp^f}\right|>\varepsilon\right\}=Pr\left\{\left|\frac{\hat{p}^f-p^f}{\sigma_{\hat{p}^f}}\right|>\frac{p^f\sqrt{n}\varepsilon}{\sqrt{p^f(1-p^f)}}\right\}\approx 2\left(1-\Phi\left(\frac{\sqrt{p^f}\sqrt{n}\varepsilon}{\sqrt{(1-p^f)}}\right)\right)\leq\eta \quad (8)$$

where $\Phi(\cdot)$ is the cumulative distribution function (c.d.f) of the standard normal distribution.

By solving the inequality in Eq. (8) 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

$$n\geq n^{*,p}=\left\lceil z_p\cdot\left(\frac{1-p^f}{p^f}\right)\right\rceil \quad (9)$$

where $z_p = \left(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon}\right)^2$.

Notice that with elephant threshold of packet count proportion p^θ, $\left(\frac{1-p^\theta}{p^\theta}\right)$ can be set as a constant $C_\theta = \left(\frac{1-p^\theta}{p^\theta}\right)$. Then,

$$n\geq n^{*,p}=\lceil z_p\cdot C_\theta\rceil \quad (10)$$

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. (10) 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.

2) 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 follows:

$$v^f=m^f\mu^f=mp^f\mu^f \quad (11)$$

where μ^f is the actual average packet size of flow f. Similarly the estimated flow byte count $\hat{v}^f$ is

$$\hat{v}^f=\hat{m}^f\hat{\mu}^f=m\hat{p}^f\hat{\mu}^f \quad (12)$$

where $\hat{\mu}^f$ is the estimated average packet size of flow f.

Notice that two levels of uncertainties are involved for flow byte count estimation, namely the estimations of flow proportion and flow average packet size.

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 [4]: Let $X_1, X_2, \ldots$ be independent, identically distributed random variables with mean μ and variance σ^2, and for each positive n, let F_n be a random variable assuming positive integers as values; it needs not be independent of the X_m's. Let $W_n=\sum_{i=1}^{F_n}X_i$. Suppose as $n\to\infty$, $\frac{F_n}{n}$

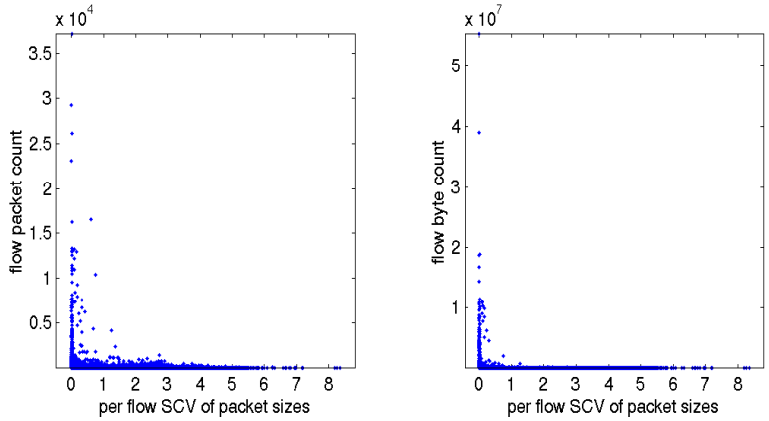


Fig. 7. Flow packet count and byte count vs. SCV.

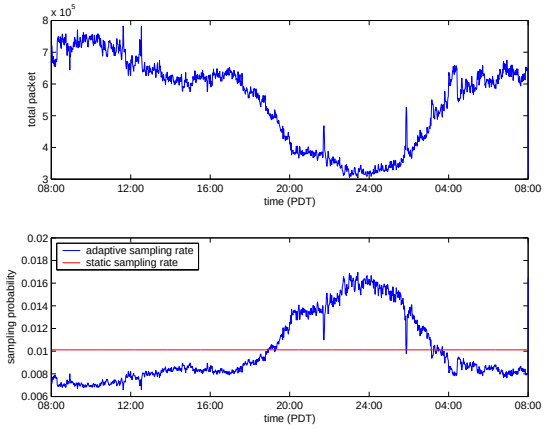


Fig. 8. Traffic load, total packet count and sampling probability (trace Π_5, $\{\eta, \varepsilon\} = \{0.1, 0.1\}$, $\{p^\theta, S^\theta\} = \{0.01, 0.2\}$).

converges to 1 almost surely. Then as $n \to \infty$,

$$\frac{W_n - F_n \mu}{\sigma \sqrt{n}} \tag{13}$$

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

$$\hat{v}^f = m p^f \mu^f + m \left[\frac{\sqrt{p^f}}{\sqrt{n}} \left(\mu^f \sqrt{1 - p^f} Y_p + \sigma^f Y_b \right) \right] \tag{14}$$

where $Y_b, Y_p \sim N(0,1)$. (The proof can be found in the Appendix.)

From the above Eq. (14), the relative error of flow byte count is summarized as

$$\frac{\hat{v}^f - v^f}{v^f} = \frac{\frac{\sqrt{p^f}}{\sqrt{n}} \left(\mu^f \sqrt{1 - p^f} Y_p + \sigma^f Y_b \right)}{p^f \mu^f} = \frac{1}{\sqrt{n p^f}} \left(\sqrt{1 - p^f} Y_p + \frac{\sigma^f}{\mu^f} Y_b \right) \sim N\left(0, \frac{1 - p^f + S^f}{n p^f} \right) \tag{15}$$

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

$$n \geq n^{*,b,f} = \left\lceil z_p \cdot \left(\frac{1 - p^f + S^f}{p^f} \right) \right\rceil \tag{16}$$

where $S^f = (\sigma^f / \mu^f)^2$ is the *squared coefficient of variation* (SCV) of packet sizes of flow f.

Eq. (16) 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. It also tells that larger number of samples are needed for flow byte count estimation compared with the one for flow packet count estimation from Eq. (10) as long as the packet sizes of a flow are not uniform ($S^f > 0$).

Our observation shown in Figure 7 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 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

$$n \geq n^{*,b} = \lceil z_p \cdot B_\theta \rceil \tag{17}$$

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

With the required number of samples computed, we will describe how to decide sampling probability to ensure the number of samples under dynamic traffic conditions and how the accuracy is achieved for flows over multiple blocks in the following subsections.

C. Optimal Sampling Probability and Prediction of Total Packet Count

The optimal sampling probability of a block to produce n^* samples would be

$$p_{sp} = \frac{n^*}{m_h} \tag{18}$$

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.

In any case, we cannot accurately choose sampling rate when the population size (total packet count of the observation time block) is unknown. We can compute the sampling probability at the beginning of a block by predicting the total packet count. We employ an AR (Auto-Regressive) model 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. In particular, using the AR model, the model parameters can be obtained by solving a set of simple linear equations [3], making it suitable for *online implementation*.

The network traffic predictability has been studied in [26] and [37]. The total packet count prediction with AR model is justified by empirical studies using real network traffic traces in [37]. The predictability may depend on the time scale (block size) of observation. We observed strong positive correlations for a wide range of time scale from $1min$ to $30min$ for long traces and $1sec$ to $10sec$ for short ($90sec$) traces.

We now briefly describe how the total packet count m_h of the hth block can be estimated based on the past packet counts using the AR(u) model. Using the AR(u) model [3], m_h can be expressed as

$$m_h = \sum_{i=1}^{u} a_i m_{h-i} + e_h \tag{19}$$

where a_i, $i = 1, \ldots, u$, are the model parameters, and e_h is the *uncorrelated* error (which we refer to as the *prediction error*).

The model parameters a_i, $i = 1, \ldots, u$, can be determined by solving a set of linear equations in terms of v past values of m_i's, where $v \geq 1$ is a configurable parameter independent of u, and is typically referred to as the memory size.

Let $\hat{m}_h$ denote the *predicted* packet count of the hth block. Using the the AR(u) prediction model, we have

$$\hat{m}_h = \sum_{i=1}^{u} a_i m_{h-i}. \tag{20}$$

In predicting the *total* packet count m, we assume the *actual* packet count of a block is known at the end of the block. Note that having the actual total packet count is reasonable to assume in current routers and does not change the nature of the adaptive random sampling technique we propose.

Using the AR prediction model, at the end of each block, the model parameters (a_i) are computed [3]. The complexity of the AR prediction model parameter computation is only $O(v)$ where v is the memory size. Through empirical studies, we have found that AR(1) with a small memory size (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:

$$\hat{m}_h^f = \hat{m}_{h-1}^f + \frac{m_h}{n_h}\hat{n}_h^f \tag{21}$$

$$\hat{v}_h^f = \hat{v}_{h-1}^f + \frac{m_h}{n_h}\hat{n}_h^f\hat{\mu}_h^f \tag{22}$$

Figure 9 shows the flow chart of the adaptive random sampling procedure.

D. Accuracy of Stratified Random Sampling: Statistical Properties

In our proposed sampling method, we collect *equal* number of random samples (n^*) for *each* stratum on average. 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 equal number of samples per stratum*. In the previous subsection, we have shown that simple random sampling with n^* samples provides the prescribed accuracy for the estimation of flows whose duration fall within a block. Here, we first explore statistical properties of the stratified random sampling. Then we show how a stratified random sampling with equal number of n^* samples per stratum also gives at least the prescribed accuracy for flows who live for L blocks.

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 of stratified random sampling 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 value of the population (population parameter). For consistent estimators, efficiency can be measured by the *variance*, where a smaller variance is preferred. An estimator of a smaller variance would give more *accurate* estimation, given the same number of samples. Mean square error (MSE) is a frequently used metric to compare estimators. Let X be a random variable of the population (in general) and $\hat{X}$ be the estimated mean of the population. (In case of proportion, X takes on 1 if a packet belongs to flow f and 0 otherwise.) As in the following equation, the variance itself becomes MSE for an unbiased estimator.

$$MSE(\hat{X}) = E(\hat{X} - \mu)^2 = Var(\hat{X}) + bias^2 \tag{23}$$

Thus it is important to study variance carefully. Notice that the analysis and the required number of samples in the previous subsections is based on simple random sampling. We first compare the variance of proposed stratified random sampling with simple random sampling. 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 ($\hat{p}^f$ or $\hat{\mu}^f$). For example,

$$Var(\hat{m}^f) = Var(m\hat{p}^f) = m^2 Var(\hat{p}^f) \tag{24}$$

Hence, we simplify the discussion of the variance to a *mean* estimation.

The variance of simple random sampling with n samples is

$$Var(\hat{X}_{sim,n}) = \frac{\sigma^2}{n} \tag{25}$$

where σ^2 is the population variance [1]. 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). The following theorem states that our proposed sampling bounds the variance of relative error by the pre-specified accuracy parameters.

Theorem 3: Using n^* random packets, the variance of the relative error in estimating flow packet count and byte count is bounded above by the pre-specified accuracy:

$$Var\left(\frac{\hat{m}^f - m^f}{m^f}\right) \leq \frac{1}{z_p} \tag{26}$$

$$Var\left(\frac{\hat{v}^f - v^f}{v^f}\right) \leq \frac{1}{z_p} \tag{27}$$

where $z_p = \left(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon}\right)^2$.

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

Theorem 4: The variance of *stratified* random sampling with *equal* number of n^* samples for each L strata is smaller than the variance of *simple* random sampling with n samples.

$$Var(\hat{X}_{str(eq),n^*L}) \leq Var(\hat{X}_{sim,n^*}) \tag{28}$$

This means that the accuracy in estimation of a flow with arbitrary duration satisfies the given bound with the proposed stratified random sampling, where n^* samples are collected at each time block.

Another important property to consider for flow is about aggregation. Flow statistics may further *aggregated into a bigger flow* later for different engineering purpose. It can be easily shown that the *accuracy* of the estimation is *conserved* for aggregated flows.

Stratified random sampling gives smaller variance of estimation (better accuracy) than simple random sampling when the variances within blocks are small compared to the variance for all the interval. Given a number of samples n, stratified random sampling increases the accuracy, when samples are selected *proportionally* to the population size in the stratum [1]. i.e.,

$$Var(\hat{X}_{str(prop),n}) \leq Var(\hat{X}_{sim,n}) \tag{29}$$

If we additionally update flow statistics within a block *without* changing sampling probability, it turns out to be a stratified random sampling. Then the sample sizes in strata (or subblock) become proportional to the population size. Therefore, one may update flow statistics more often than once per block to increase an accuracy of estimation.

E. Bounding Absolute Error

So far we described bounding *relative* error in estimating flow packet and byte count. We have derived the required sample size by linearly separating accuracy parameter from traffic parameter as

$$n \propto \text{ accuracy} \cdot \text{traffic parameter} \tag{30}$$

For the objective of *absolute* error bound

$$Pr\left\{\left|\hat{m}^f - m^f\right| > \varepsilon\right\} \leq \eta \text{ or } Pr\left\{\left|\hat{v}^f - v^f\right| > \varepsilon\right\} \leq \eta \tag{31}$$

the required sample size can be obtained by a similar analysis used in bounding relative error. Then, the required number of samples can be derived as

$$n \geq n^{*,p} = z_p \cdot p^f(1-p^f) \cdot m^2 \text{ or } n \geq n^{*,b} = z_p \cdot \left[p^f(1-p^f)\mu^f + \sigma^f\right] \cdot m^2 \tag{32}$$

for flow packet and byte count estimation respectively (see [**?**] for detail). It can be summarized in the following form:

$$n \propto \text{accuracy} \cdot \text{traffic parameter} \cdot m^2 \tag{33}$$

Assuming $p^f < 0.5$, the required sample size depends on the *maximum* flow proportion whose estimation error should be bounded. Suppose $n^{*,p}$ is computed using p^θ. Since the accuracy is not guaranteed for flows whose proportion are larger than p^θ, the largest flow proportion should be known ahead. Still sampling probability is much higher than the one for relative error bound in general since it is quadratically proportional to total packet count (m^2). For flow byte count estimation, two parameters of a flow - average packet size of a flow μ^f and its variance σ^f - should be known as opposed to one parameter of flow SCV for relative error. Furthermore, unlike SCV, mean packet size of a flow μ^f varies widely among all the flows as observed in Figure 6. Using maximum packet size (1500bytes for example) would give a *very* large number in the required number of samples (often as many as total packet count) resulting in oversampling for many elephant flows with smaller average packet size.

Therefore, bounding relative error is more practical, and it is suitable for the type of applications such as traffic profiling and engineering, where flows responsible for most of the traffic are of interest.

IV. Practical considerations

In this section, we discuss the issues involved in the implementation of the proposed sampling technique. We first discuss how to determine the flow timeout value and its impact on the performance of our method. Then we discuss how to reduce the overhead of random number generation.

A. Flow Timeout Value

The choice of timeout values may change the flow statistics for a given traffic even with full measurement (i.e., no sampling). While a large timeout value leads to maintaining unnecessarily large number of flow states, a small timeout may break otherwise large, long-lived flows into smaller flows. The impact of timeout values on flow statistics and a performance trade-off were studied in [13], [14].

With introduction of sampling, timeout value should be adjusted appropriately due to increased inter-packet arrival time of a flow. Suppose TO_{full} is used for a flow definition when no sampling is used. With random

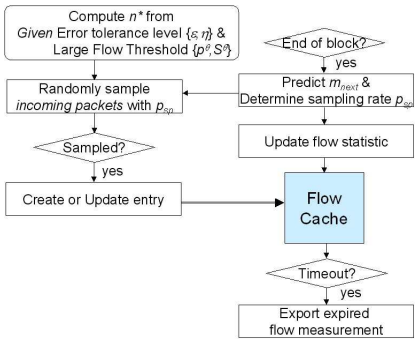


Fig. 9. Adaptive random sampling for flow volume measurement.

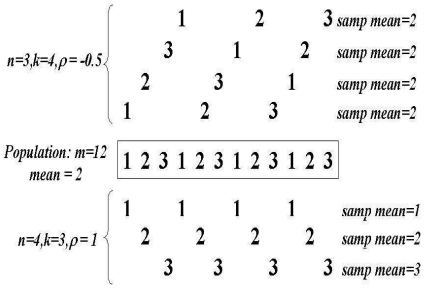


Fig. 10. Systematic sampling on periodic population.

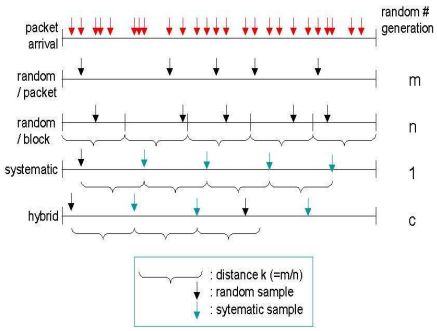


Fig. 11. Trade-off in random number generation.

sampling, intra-flow inter-packet arrival time of samples is increased by the inverse of the sampling rate on average (p_{avg}). Thus, one can use the timeout value under sampling as

$$TO_{samp} = \frac{TO_{full}}{p_{avg}} \tag{34}$$

However, in the proposed adaptive random sampling, the sampling rate changes adaptively over time. So we exponentially average the sampling rate as follows.

$$p_{avg,i} = \alpha p_i + (1-\alpha) p_{avg,i-1} \tag{35}$$

where p_i is the sampling probability of a block i and $p_{avg,i}$ is the averaged sampling probability in block i to be used for a timeout value. Given a timeout value for full measurement ($60sec$), we rarely observe truncation of elephant flows throughout the experiments. It is because from our definition, an elephant flow is expected to have a high packet rate on average over the flow's duration, thus making it *less sensitive* to a timeout value.

B. Utility of Systematic Sampling and Random Number Generation

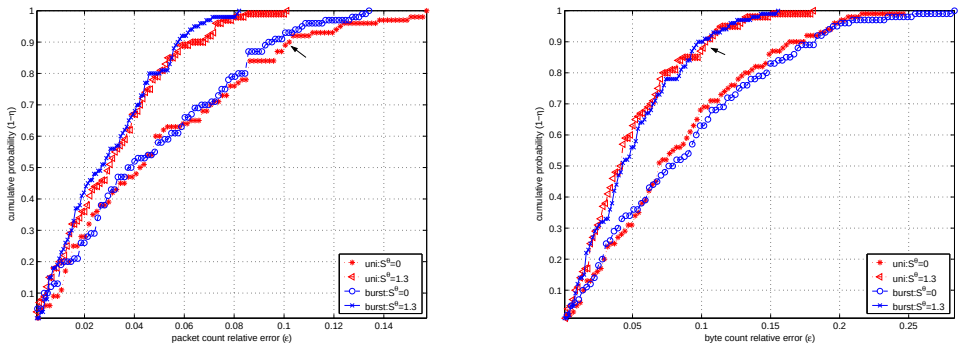
Systematic sampling is a popular sampling design employed in Cisco and Juniper routers ([33], [34]). In general, 1-out-of-k systematic sampling involves random selection of one element from the first k elements, and selection of every kth element thereafter requiring only one random number generation and a counter. Random sampling involves a random number generation per-packet. Even though modern routers already have the feature implemented for a mechanism such as RED (Random Early Detection), with a choice of sampling rate from our analysis one may want to consider using systematic sampling for a simplicity. However, understanding accuracy of systematic sampling has to precede its utility.

The performance of systematic sampling can be explained with the concept of correlation between samples of an experiment (sample set). The variance of sample mean using systematic sampling is given by [1], [2]

$$Var(\hat{X}_{sys,n}) = \frac{\sigma^2}{n}[1 + (n-1)\rho] \tag{36}$$

where ρ is a measure of the correlation between pairs of samples within the same systematic sample.

$$\rho = \frac{E(X_{ij} - \mu)(X_{ij'} - \mu)}{E(X_{ij} - \mu)} \tag{37}$$



(a) $S^\theta = 0$ (b) $S^\theta = 1.3$

Fig. 12. Synthetic data: Extremely bursty and uniformly distributed flows with variable packet sizes ($\{\eta, \varepsilon\} = \{0.1, 0.1\}$, $\{p^\theta, S^f\} = \{0.01, 1.3\}$, $p^f = p^\theta$).

where, $-\frac{1}{(n-1)} \leq \rho \leq 1$, $i = 1, \ldots, k$ and $j, j' = 1, \ldots, n$, $j \neq j'$.

Thus, a theoretical accuracy of systematic sampling is not practically assessable, as knowledge of all k systematic samples is necessary to calculate the variance of systematic samples. Eq. (36) also shows that when ρ is positive, the estimator is *not consistent*, since the accuracy is not increased with large n. If ρ is close to 1, then the variability of elements within the sample set is too small compared to variability among possible sample sets, and systematic sampling will yield a larger variance than using simple random sampling. If ρ is negative, then systematic sampling may be better than simple random sampling. The correlation may be negative if variability of elements *within* a systematic sample set tends to be larger than *among* systematic sample sets . For ρ close to 0, systematic sampling is roughly equivalent to simple random sampling. When the population is *randomly ordered*, systematic sampling will give us a precision approximately equivalent to that obtainable by simple random sampling. [5]

Figure 10 illustrates extreme performance of systematic sampling from the same population. Suppose one try to estimate a population mean with a systematic sampling. For the periodic population shown in Figure 10, when $k(= m/n)$ is the same as the period, the value of the sample is the same for all samples in any possible set of samples, thus, an increase of sample size would not increase the accuracy at all. Meanwhile, with $k = 4(> 3)$, it always gives the exact population mean with smaller number of samples which is better than random sampling. Therefore, randomness in samples is important to *assess* the accuracy, and to avoid extreme performance.

Scalability of measurement using adaptive random sampling may be further enhanced by infrequent random number generation. Suppose n packets out of m are collected (refer to illustration in Figure 11). Rather than generating a random number for each packet (first row), we maintain a counter initialized to $k(= m/n)$. The counter is decremented upon each packet arrival, and when it reaches 0 it is reset back to k. Whenever the counter is set to k, a random number i (from 0 to k) is generated and ith packet, counting from the time of counter reset, is sampled as illustrated in the second row of the figure. As a further enhancement, a hybrid approach could be used as shown in the third row of Figure 11. By changing starting point randomly several times ($c(< n)$), this process has the effect of shuffling the elements of the population.

[5] From our experiments, the performance of systematic sampling with adaptive sampling rate was close to the one of adaptive random sampling. However, in this paper, we limit our discussion on random sampling for the assessment of estimation error.

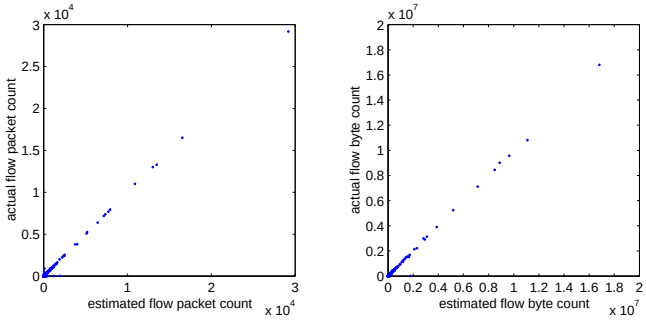


Fig. 13. Actual vs. estimated flow volume (trace Π_1, $\{\eta, \varepsilon\} = \{0.1, 0.1\}$).

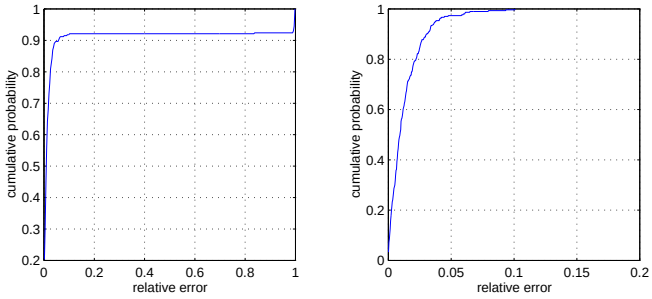


Fig. 14. Relative error of elephant flows (trace Π_1,$\{\eta, \varepsilon\} = \{0.1, 0.1\}$).

V. Experimental Results

In this section, we first validate our theoretical result with synthetic data. We then empirically evaluate the performance of our adaptive packet sampling technique for flow measurement using the real network traces.

In order to verify our theoretical results, first we conduct experiments with synthesized data where all flows are elephants whose proportions are the same as the threshold. In synthesized data, all flows have the same packet count and their durations fall within a block. They have different byte counts caused by various means and standard deviations of their packet sizes. The SCVs for all flows however, are the same ($SCV = 1.3$). Two types of traffic data are generated according to the flow's packet rate variability. Packets of flows in the first traffic data are uniformly distributed, while flows in the second traffic are bursty. We first use the sampling rate ignoring the packet size variability ($S^\theta = 0$), and repeat the measurements with a higher sampling rate using the actual value of packet size SCV.

In both cases we calculate the cumulative probability of relative errors for packet and byte count estimates. Figure 12(a) shows that packet count estimation indeed conforms to the pre-specified accuracy parameter, i.e., the probability of relative errors larger than $\varepsilon = 0.1$ is around $\eta = 0.1$ with $S^\theta = 0$. Meanwhile, as shown in Figure 12(b), the probability of relative errors larger than $\varepsilon = 0.1$ is a lot higher than $\eta = 0.1$, when the packet size variability is not considered for the sampling rate. When taking packet size variability into account ($S^\theta = 1.3$), byte count estimation error conforms to the pre-specified error bound. Due to increased sampling rate, the accuracy of the packet count estimates is increased, i.e., the proportion of all estimates whose relative error is smaller than ε is even higher than the predicted probability $1-\eta$. The above observations are true for both types of flows and therefore, independent of the flow's burstiness. Considering fluctuation of traffic, in particular a high day/night traffic ratio, adapting sampling rate appropriately is critical to achieve estimation accuracy as well as to avoid too much unnecessary oversampling.

We also validate the achieved accuracy of the proposed sampling technique, using real traces. In Figure 13, 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. Figure 14 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 $1 - \eta$. Recall that the sampling rate was aimed for an accuracy of minimum elephant whose proportion is equal to the threshold.

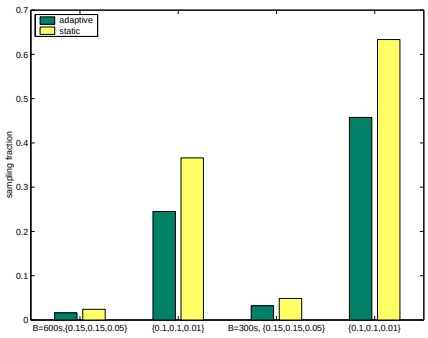


Fig. 15. Sampling fraction (trace Π_1).

TABLE III
FLOW CACHE SIZE REDUCTION.

Parameters $\{\eta, \varepsilon, p^\theta, S^\theta\}$	Trace Π_5 $(24hr)$		Trace Π_1 $(4hr)$		Trace Π_6 $(90sec)$	
	Adapt.	Static	Adapt.	Static	Adapt.	Static
$\{.1, .1, .001, .2\}$	.459	.711	.721	.772	.742	.746
$\{.15, .15, .005, .0\}$	.433	.697	.710	.735	.727	.730

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 14.

Next we examine the efficiency of the adaptive sampling in terms of reduction in packet measurement and flow cache size. Note that the amount of measurement in the adaptive scheme depends on accuracy parameters and elephant flow thresholds, and does not depend on the total traffic. In static sampling, however, the amount of samples is proportional to the total traffic. 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. Average sampling rate of adaptive sampling method is used for the sampling rate of static sampling method. In adaptive sampling, a higher accuracy requires larger number of samples while larger block size decreases sampling rate. As shown in Figure 15 the sampling fraction is higher for 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.

Reduction in flow cache size is another benefit from sampling, because some flows can be omitted in keeping flow statistics. Table III shows flow cache size reduction for both adaptive and static sampling with various traces and various sampling rate. Flow cache size reduction is computed in terms of average flow cache size in case of using sampling compared to the one without sampling for the trace. Packet sampling indeed reduces flow cache size for all cases of sampling rate and traces. The adaptive sampling performs better than static sampling, particularly when day/night traffic fluctuation is considered. However, its reduction is less relevant to the sampling parameters. This is because as sampling rate decreases, the timeout value in case of sampling increases accordingly. Then, flow statistics should be kept for longer time when sampling is used.

VI. CONCLUSIONS

In this paper, we have addressed the problem of flow volume measurement using packet sampling approach. 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. The proposed method based on stratified random sampling divides time into strata and within each stratum samples packets randomly at a rate determined according to the minimum number of samples needed to achieve the desired accuracy. 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. We have also discussed practical issues and argued that our method can be implemented efficiently to match the line speeds. We conclude that the ability to control the accuracy of estimation and thus tradeoff the utility and the overhead of measurement makes our adaptive sampling method a scalable and attractive solution for flow volume measurement.

REFERENCES

[1] T. Yamane, "Elementary Sampling Theory," Prentice-Hall, Inc., Engleweed Cliffs, N.J. 1967.
[2] R. Scheaffer, W. Mendenhall, R. Ott, "Elementary Survey Sampling," 5th ed., Duxbury Press, 1995.
[3] J. Gottman, "Time-series analysis," Cambridge University Press, 1981.
[4] P. Billingsley, "Probability Measures," Wiley-Interscience Publication, 1995.
[5] D. Berry and B. Lindgren, "Statistics theory and Methods," 2nd ed., Duxbury Press, ITP, 1996.
[6] "Internet Protocol Flow Informaion eXport (IPFIX)," IETF Working Group. see: http://ipfix.doit.wisc.edu
[7] Passive Packet Traces Archive http://moat.nlanr.net
[8] "Real time Flow Measurement," see: http://www.auckland.ac.nz/net/Internet/rtfm
[9] Packet Sampling Working Group https://ops.ietf.org/lists/psamp/
[10] J. Apisdorf, K. Claffy, K. Thompson and R. Wilder, "OC3MON: flexible, affordable, high performance statistics collection," see: http://www.nlanr.net/NA/Oc3mon
[11] C. Patridge. A proposed flow specification. RFC1363, 1992.
[12] R. Jain and S. A. Routhier. "Packet trains–measurements and a new model for computer network traffic," IEEE JSAC 4:986-995, 1986.
[13] B. Ryu, D. Cheney, H. Braun, "Internet flow characterization: adaptive timeout strategy and statistical modeling," Passive and Active Measurement Workshop, Amsterdam, April 2001.
[14] K. Claffy, H.-W. Braun, and G. C. Polyzos, "A parameterizable methodology for Internet traffic flow profiling," IEEE JSAC, 13:1481-1494, 1995.
[15] K. Claffy, "Internet traffic characterization," Ph.D thesis, University of California San Diego, 1994.
[16] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft, "Pop-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP," ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, November, 2001.
[17] L. Kleinrock and W Naylor, "On Measured Behavior of the Arpa Network," AFIPS Conference Proceedings, National Compuer Conference, vol 34, December 1999.
[18] W. Fang, L. Peterson, "Inter-AS Traffic Patterns and Their Implications," Proceedings of IEEE Globecom, Rio, Brazil, December. 1999.
[19] 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 Transactions on Networking, June 2001, pp. 265-279.
[20] R. Manajan, Steven M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. "Controlling High Bandwidth Flows at the Congested Router," Proc. of the 9th International Conference on Network Protocols (ICNP), Riverside, CA, November 2001.
[21] S. Kim and A. Reddy, "Identifying long term high rate flows at a router," in Proc. of High Performance Computing, Dec. 2001.
[22] J. Jedwab, P. Phaal, and B. Pinna, "Traffic estimation for the largest sources on a network, using packet sampling with limited storage," HP Labs Technical Report, HPL-92-35, 1992.
[23] InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, http://www.ietf.org/rfc/rfc3176.txt
[24] M. Borella, "Source Models of Network Game Traffic," Computer Communications, Vol. 23, No. 4, pp. 403-410, Feb. 2000.
[25] D. LaPointe, J. Winslow, and M. Claypool, "Analyzing and Simulating Network Game Traffic", Major Qualifying Project MQP-MLC-NG01, Computer Science Department,Worcester Polytechnic Institute, Fall 2001.
[26] Aimin Sang, S. Q. Li, "A Predictability Anal ysis of Network Traffic," in Proceedings of IEEE INFOCOM, 2000.
[27] A. Shaikh, J. Rexford, and K. Shin, "Load-sensitive routing of long-lived IP flows," In Proc. ACM SIGCOMM, 1999.
[28] I. Cozzani, S. Giordano, "A measurement based QoS evaluation through traffic sampling," IEEE SICON'98, 29th June - 3rd July, Singapore, 1998.
[29] N. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation," Proceedings of ACM SIGCOMM 2000 pp271-28.
[30] N. Duffield, C. Lund, and M. Thorup, "Charging from Sampled Network Usage," ACM SIGCOMM Internet Measurement Workshop 2001.
[31] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting," ACM SIGCOMM 2002.
[32] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the Characteristics and Origin of Internet Flow Rates," In Proc. ACM SIGCOMM, 2002.
[33] Sampled NetFlow. http://www.cisco.com
[34] Juniper packet sampling http://www.juniper.net

[35] A. Erramilli, O. Narayan, and W. Willinger, "Experimental Queuing Analysis with Long-Range Dependent Packet Traffic," IEEE/ACM Transactions on Networking, Vol. 4, No. 2, pp. 209-223, April 1996.
[36] RFC3234, "Middleboxes: Taxonomy and Issues," February 2002.
[37] B. Choi, J. Park, and Z.-L. Zhang, "Adaptive Random Sampling for Load Change Detection", *ACM Sigmetrics* 2002, *extended version* University of Minnesota, Technical Report TR-01-041, 2001.

APPENDIX

[Proof of Theorem 3]

$$Var\left(\frac{\hat{m}^f - m^f}{m^f}\right) = Var\left(\frac{\hat{p}^f - p^f}{p^f}\right) = \frac{Var\left(\hat{p}^f\right)}{p^{f2}} = \frac{p^f(1-p^f)}{np^{f2}} = \frac{(1-p^f)}{n} \tag{38}$$

where $n \geq n^{*,p} = \left\lceil z_p \cdot \left(\frac{1-p^f}{p^f}\right)\right\rceil$. Thus, $Var\left(\frac{\hat{m}^f - m^f}{m^f}\right) \leq \frac{1}{z_p}$.

Similarly,

$$Var\left(\frac{\hat{V}^f - V^f}{V^f}\right) = \frac{1 - p^f + S^f}{np^f} \tag{39}$$

where $n \geq n^{*,b} = \left\lceil z_p \cdot \left(\frac{1-p^f+S^f}{p^f}\right)\right\rceil$ Thus, $Var\left(\frac{\hat{V}^f - V^f}{V^f}\right) \leq \frac{1}{z_p}$

[Proof of Theorem 4]

$$Var(\hat{X}_{str,n\cdot L}) = \frac{L}{m^2}\frac{\sum_{i=1}^{L} m_i^2\sigma_i^2}{nL} = \frac{1}{mn}\sum_{i=1}^{L} m_i\sigma_i^2 \cdot \frac{m_i}{m} \leq \frac{1}{mn}\sum_{i=1}^{L} m_i\sigma_i^2 \leq Var(\hat{X}_{sim,n}) \tag{40}$$

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

[Proof of Eq. (14)]

$$\begin{aligned}
\hat{V}^f &= \frac{m}{n}n^f\hat{\mu}^f \approx \frac{m}{n}\left[n^f\mu^f + \sigma^f\sqrt{np^f}Y_b\right] = m\left[\frac{n^f}{n}\mu^f + \sigma^f\frac{\sqrt{np^f}}{n}Y_b\right] \\
&= m\left[\hat{p^f}\mu^f + \sigma^f\frac{\sqrt{p^f}}{\sqrt{n}}Y_b\right] = m\left[\left(p^f + \frac{\sqrt{p^f(1-p^f)}}{\sqrt{n}}Y_p\right)\mu^f + \frac{\sqrt{p^f}}{\sqrt{n}}\sigma^f Y_b\right] \\
&= m\left[p^f\mu^f + \frac{\sqrt{p^f}}{\sqrt{n}}\left(\mu^f\sqrt{1-p^f}Y_p + \sigma^f Y_b\right)\right] = mp^f\mu^f + m\left[\frac{\sqrt{p^f}}{\sqrt{n}}\left(\mu^f\sqrt{1-p^f}Y_p + \sigma^f Y_b\right)\right]
\end{aligned} \tag{41}$$

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

[Proof of Aggregation Property]

From the objective, with a probability of $1-\eta$, for each i in elephant flows of fine granularity,

$$\left|\frac{m_i^f - \hat{m}_i^f}{m_i^f}\right| \leq \varepsilon \tag{42}$$

Therefore, their aggregation accuracy is

$$\left|\frac{\sum_i m_i^f - \sum_i \hat{m}_i^f}{\sum_i m_i^f}\right| \leq \varepsilon \tag{43}$$

The aggregation property is similarly derived for aggregated flow byte count as well.